



KEMENTERIAN DIGITAL  
JABATAN DIGITAL NEGARA

**DevOps**  
*Sektor Awam*

**RANGKA KERJA  
PELAKSANAAN DEVOPS  
DALAM PEMBANGUNAN SISTEM APLIKASI  
SEKTOR AWAM**



**Kementerian Digital  
Jabatan Digital Negara**

**Rangka Kerja  
Pelaksanaan DevOps dalam  
Pembangunan Sistem Aplikasi  
Sektor Awam**

**Hak Cipta Terpelihara.**

**© Jabatan Digital Negara, Kementerian Digital, 2024.**

Hak cipta terpelihara, kecuali untuk tujuan pendidikan tanpa apa-apa kepentingan komersial. Tidak dibenarkan mengulang cetak mana-mana bahagian artikel, ilustrasi dan isi kandungan buku ini dalam apa jua bentuk dan dengan apa jua cara sama ada secara elektronik, rakaman atau cara lain sebelum mendapat kebenaran bertulis daripada Ketua Pengarah Jabatan Digital Negara.

**Diterbitkan oleh:**

Jabatan Digital Negara  
Kementerian Digital  
Bangunan MKN Embassy Techzone  
Blok B, No. 3200 Jalan Teknokrat 2  
63000 Cyberjaya, Sepang  
Selangor Darul Ehsan  
Malaysia  
Tel: 603 8000 8000  
Fax: 603 8872 3163  
Web: <http://www.jdn.gov.my>

## KANDUNGAN

<b>KANDUNGAN .....</b>	<b>ii</b>
<b>SENARAI JADUAL.....</b>	<b>iv</b>
<b>SENARAI RAJAH.....</b>	<b>vi</b>
<b>SENARAI LAMPIRAN .....</b>	<b>ix</b>
<b>PRAKATA .....</b>	<b>xi</b>
<b>AKRONIM .....</b>	<b>xiii</b>
<b>TAKRIFAN .....</b>	<b>xiv</b>
<b>RINGKASAN EKSEKUTIF .....</b>	<b>1</b>
<b>BAB 1 : PENDAHULUAN .....</b>	<b>5</b>
1.1. SKOP DOKUMEN .....	7
1.2. OBJEKTIF DOKUMEN .....	8
<b>BAB 2 : KONSEP DAN DEFINISI .....</b>	<b>11</b>
2.1. PERSEKITARAN SEMASA PEMBANGUNAN SISTEM APLIKASI.....	11
2.2. DEFINISI DEVOPS .....	12
2.3. EVOLUSI PEMBANGUNAN SISTEM APLIKASI.....	13
2.4. OBJEKTIF PELAKSANAAN DEVOPS .....	15
2.5. PERINGKAT-PERINGKAT DALAM PELAKSANAAN DEVOPS.....	16
2.6. MANFAAT PELAKSANAAN DEVOPS .....	19
<b>BAB 3 : KOMPONEN RANGKA KERJA .....</b>	<b>23</b>
3.1. KOMPONEN 1 (K1): PENERAPAN PRINSIP DAN BUDAYA .....	30
3.1.1. Prinsip DevOps .....	30
3.1.2. Budaya DevOps dan Penerapan dalam Pasukan .....	33
3.1.3. Pengenalan Metodologi <i>Agile</i> .....	38
3.2. KOMPONEN 2 (K2): PENGUKUHAN TADBIR URUS .....	44
3.2.1. Struktur Pelaporan Organisasi .....	44
3.2.2. Struktur Pasukan.....	45
3.2.3. Ahli Pasukan .....	49
3.3. KOMPONEN 3 (K3): PENGADAPTASIAN METODOLOGI .....	56
3.3.1. Aktiviti dalam Metodologi <i>Agile</i> .....	57
3.3.2. Peringkat Perancangan.....	101
3.3.3. Peringkat Pengekodan .....	117
3.3.4. Peringkat Pembangunan .....	120

3.3.5. Peringkat Pengujian .....	122
3.3.6. Peringkat Pelepasan .....	138
3.3.7. Peringkat Penempatan .....	147
3.3.8. Peringkat Pengoperasian .....	150
3.3.9. Peringkat Pemantauan .....	153
3.4. KOMPONEN 4 (K4): PENGUKURAN KEMATANGAN.....	167
3.4.1. Penilaian Persekitaran Semasa bagi Ketersediaan Pelaksanaan DevOps .....	167
3.4.2. Proses Peralihan .....	175
3.4.3. Pengukuran Tahap Pelaksanaan .....	179
3.5. KOMPONEN 5 (K5): PEMERKASAAN TEKNOLOGI.....	183
3.5.1. Infrastruktur <i>Pipeline</i> DevOps.....	183
3.5.2. Persekitaran <i>Pipeline</i> DevOps.....	188
3.5.3. <i>Tools</i> DevOps.....	192
3.5.4. Penggunaan <i>Tools</i> Mengikut Peringkat DevOps.....	195
3.6. KOMPONEN 6 (K6): PEMATUHAN KESELAMATAN .....	199
3.6.1. Aktiviti Keselamatan dalam DevSecOps .....	200
3.6.2. Reka Bentuk yang Selamat.....	203
3.6.3. <i>Tools</i> Keselamatan dalam DevSecOps .....	207
3.6.4. Analisis Kod yang Selamat.....	210
3.6.5. Penempatan yang Selamat .....	213
3.6.6. Penyenggaraan yang Selamat .....	215
3.6.7. Pengukuran Metrik Keselamatan .....	223
<b>BAB 4 : PENUTUP .....</b>	<b>226</b>
<b>LAMPIRAN .....</b>	<b>229</b>
<b>SUMBER RUJUKAN .....</b>	<b>286</b>
<b>PENGHARGAAN .....</b>	<b>292</b>

## SENARAI JADUAL

Jadual 3-1: Penerangan Berkaitan Aspek-Aspek Budaya dan Cadangan Aktiviti ....	34
Jadual 3-2: Cadangan Keanggotaan Jawatankuasa Pengurusan Produk .....	45
Jadual 3-3: Peranan, Tugasan dan Kemahiran Ahli Pasukan .....	51
Jadual 3-4: Komponen Artifak <i>Product Vision</i> .....	60
Jadual 3-5: Komponen Artifak <i>User Story</i> .....	63
Jadual 3-6: Contoh Artifak <i>User Story</i> .....	64
Jadual 3-7: Komponen Artifak <i>Product Backlog</i> .....	67
Jadual 3-8: Kategori Penetapan Keutamaan <i>User story</i> .....	68
Jadual 3-9: Contoh Artifak <i>Product Backlog</i> .....	71
Jadual 3-10: Komponen Artifak <i>Definition of Done</i> .....	72
Jadual 3-11: Contoh Senarai <i>Definition of Done</i> .....	73
Jadual 3-12: Komponen Artifak Perancangan Kapasiti .....	76
Jadual 3-13: Jadual Pengiraan Perancangan Kapasiti <i>Sprint 1</i> .....	78
Jadual 3-14: Komponen Artifak <i>Sprint Backlog</i> .....	79
Jadual 3-15: Contoh <i>Sprint Backlog</i> Hari 1 bagi <i>Sprint 1</i> .....	81
Jadual 3-16: Log <i>Daily Scrum</i> .....	83
Jadual 3-17: Log <i>Daily Scrum</i> bagi Tempoh Lima Hari .....	84
Jadual 3-18: Contoh Artifak <i>Sprint Backlog</i> sehingga Hari 10.....	85
Jadual 3-19: Komponen Carta <i>Burndown</i> .....	86
Jadual 3-20: <i>Product Backlog</i> Sebelum Aktiviti <i>Product Backlog Refinement</i> .....	91
Jadual 3-21: Pengemaskinian <i>Product Backlog</i> .....	92
Jadual 3-22: Artifak <i>Definition of Done</i> .....	95
Jadual 3-23: Komponen Artifak <i>Sprint Retrospective</i> .....	99
Jadual 3-24: Artifak <i>Sprint Retrospective</i> .....	100
Jadual 3-25: Aktiviti Pengurusan Konfigurasi .....	104
Jadual 3-26: Aktiviti Pengurusan Pindaan.....	106
Jadual 3-27: Aktiviti Pengurusan Insiden .....	108
Jadual 3-28: Aktiviti Pengurusan Pelepasan .....	110
Jadual 3-29: Jenis Pengujian berdasarkan Kuadran Pengujian <i>Agile</i> .....	113
Jadual 3-30: Keterangan Aliran Proses Pengendalian Ralat.....	124
Jadual 3-31: Komponen Templat <i>Traceability Matrix</i> .....	125
Jadual 3-32: Penetapan Konvensyen Nama dan Nombor <i>Traceability Matrix</i> .....	126

Jadual 3-33: Contoh Pengisian Templat <i>Traceability Matrix</i> .....	127
Jadual 3-34: Komponen Templat Kes Pengujian .....	130
Jadual 3-35: Contoh Pengisian Templat Kes Pengujian .....	132
Jadual 3-36: Konsep dan Teknik Pengujian Berterusan.....	136
Jadual 3-37: Contoh Pelan Pelepasan .....	140
Jadual 3-38: Jadual Penerangan <i>Semantic Versioning</i> .....	146
Jadual 3-39: Komponen Pelaporan Status Kemajuan Projek.....	149
Jadual 3-40 : Format Log .....	155
Jadual 3-41: Metrik Pemantauan Infrastruktur .....	159
Jadual 3-42: Metrik Pemantauan <i>Uptime</i> .....	159
Jadual 3-43: Kandungan Templat Perancangan Kapasiti .....	162
Jadual 3-44: Contoh Pengisian Templat Perancangan Kapasiti .....	163
Jadual 3-45: Keterangan Aliran Pemantauan Maklum Balas Pengguna .....	165
Jadual 3-46: Model Kematangan DevOps.....	171
Jadual 3-47: Metodologi Pelaksanaan Proses Peralihan kepada Amalan DevOps	176
Jadual 3-48: Jadual Pengukuran Prestasi Tahap Pelaksanaan DevOps .....	180
Jadual 3-49: Aliran Tugasan <i>Tools</i> Mengikut Peringkat DevOps .....	197
Jadual 3-50: Penerangan berkaitan <i>Tools</i> Keselamatan.....	209
Jadual 3-51: Penjelasan Berkenaan Automasi Pengujian Keselamatan.....	211
Jadual 3-52: Penerangan Aktiviti Pemantauan Keselamatan.....	216
Jadual 3-53: Penerangan Aktiviti Penyenggaraan Keselamatan.....	217
Jadual 3-54: Tindakan Terhadap Insiden .....	222
Jadual 4-1: Tahap Kritikal.....	253

## SENARAI RAJAH

Rajah 2-1: Evolusi Pembangunan Sistem Aplikasi.....	13
Rajah 2-2: Perbandingan antara <i>Waterfall</i> , <i>Agile</i> dan DevOps .....	14
Rajah 2-3: Peringkat-Peringkat dalam Pelaksanaan DevOps .....	16
Rajah 2-4: Manfaat Pelaksanaan DevOps .....	19
Rajah 3-1: Komponen Pelaksanaan DevOps.....	23
Rajah 3-2: Prinsip Asas Pelaksanaan DevOps .....	24
Rajah 3-3: Amalan Penambahbaikan Berterusan .....	25
Rajah 3-4: Pemboleh Daya Pelaksanaan DevOps.....	28
Rajah 3-5: Komponen Pelaksanaan DevOps.....	29
Rajah 3-6: Prinsip DevOps.....	31
Rajah 3-7: Budaya DevOps untuk Mencapai Prinsip DevOps.....	33
Rajah 3-8: Nilai-nilai Utama <i>Agile</i> .....	39
Rajah 3-9: Prinsip <i>Agile</i> .....	40
Rajah 3-10: Struktur Pelaporan dalam Pelaksanaan DevOps.....	44
Rajah 3-11: Topologi Struktur Pasukan DevOps.....	46
Rajah 3-12: Gambaran Pelaksanaan Topologi 3 .....	47
Rajah 3-13: Gambaran Pelaksanaan Topologi 4 .....	48
Rajah 3-14: Cadangan Keahlian Pasukan .....	49
Rajah 3-15: Aktiviti <i>Agile</i> dan Peringkat DevOps .....	56
Rajah 3-16: Aktiviti <i>Agile Scrum</i> dalam Satu <i>Sprint</i> .....	57
Rajah 3-17: Aktiviti dan Artifak <i>Agile Scrum</i> .....	58
Rajah 3-18: Aktiviti <i>Sprint Zero</i> .....	59
Rajah 3-19: Contoh Artifak <i>Product Vision</i> .....	61
Rajah 3-20: Hierarki Fungsi Bisnes Sistem Tempahan Bilik Mesyuarat.....	64
Rajah 3-21: <i>Sprint Planing Meeting</i> Pertama .....	66
Rajah 3-22: Kaedah <i>Playing Poker</i> .....	70
Rajah 3-23: <i>Sprint Planning Meeting</i> Kedua .....	75
Rajah 3-24: <i>Daily Scrum Meeting</i> .....	82
Rajah 3-25: Carta <i>Burndown</i> sehingga Hari ke 10 .....	87
Rajah 3-26: <i>Product Backlog Refinement</i> .....	89
Rajah 3-27: <i>Sprint Review</i> .....	94
Rajah 3-28: Aktiviti <i>Sprint</i> dan <i>Product Increment</i> .....	96

Rajah 3-29: <i>Sprint Retrospective</i> .....	97
Rajah 3-30: Peringkat Perancangan .....	101
Rajah 3-31: Perkembangan <i>Epic</i> ke <i>User Story</i> .....	102
Rajah 3-32: Langkah - Langkah bagi Penyediaan Pelan Pelepasan .....	109
Rajah 3-33: Peringkat Pengekodan .....	118
Rajah 3-34: Peringkat Pembangunan .....	120
Rajah 3-35: Aliran Proses Pengendalian Ralat .....	123
Rajah 3-36: Aktiviti Pengujian Berdasarkan Metodologi <i>Agile Scrum</i> .....	129
Rajah 3-37: Kuadran Pengujian <i>Agile</i> .....	134
Rajah 3-38: Aktiviti <i>Sprint</i> dan <i>Product Release</i> .....	144
Rajah 3-39: <i>Semantic Versioning</i> .....	145
Rajah 3-40: Contoh Carta Burndown dan Carta Gantt.....	148
Rajah 3-41: Contoh WBS .....	148
Rajah 3-42: Tiga Komponen yang Menyokong Konsep Kebolehperhatian .....	154
Rajah 3-43: Paparan Log .....	155
Rajah 3-44: Paparan Metrik .....	156
Rajah 3-45: Paparan Jejak <i>API Queries</i> .....	157
Rajah 3-46: Aliran Proses Pemantauan Maklum Balas Pengguna .....	164
Rajah 3-47: Pengukuran Kematangan .....	167
Rajah 3-48: Keluk Kematangan Agensi (Sumber: ACT-IAC) .....	168
Rajah 3-49: Proses Peralihan kepada Pelaksanaan DevOps .....	175
Rajah 3-50: Metrik Pengukuran Prestasi Pelaksanaan DevOps .....	179
Rajah 3-51: Infrastruktur <i>Pipeline DevOps</i> .....	184
Rajah 3-52: Pembinaan Persekutaran ke Arah Pelaksanaan <i>Pipeline DevOps</i> .....	188
Rajah 3-53: <i>Tools</i> untuk Persekutaran <i>Pipeline DevOps</i> .....	192
Rajah 3-54: Penggunaan <i>Tools</i> Mengikut Peringkat DevOps .....	196
Rajah 3-55: Penyepadan Ekosistem Keselamatan dalam DevOps yang membentuk DevSecOps.....	199
Rajah 3-56: Aktiviti Keselamatan DevSecOps. ....	200
Rajah 3-57: Aktiviti untuk Reka Bentuk yang Selamat .....	203
Rajah 3-58: Langkah-Langkah untuk Melaksanakan Penilaian Risiko ke Atas Sistem Aplikasi.....	205
Rajah 3-59: Gambaran Berkenaan Langkah Pemodelan Ancaman.....	207
Rajah 3-60: <i>Tools</i> Keselamatan dalam Membentuk DevSecOps.....	208
Rajah 3-61: Automasi Pengujian Keselamatan .....	210
Rajah 3-62: Aktiviti Berkaitan Penempatan yang Selamat .....	213

Rajah 3-63: Aktiviti Berkaitan Penyenggaraan yang Selamat .....	215
Rajah 3-64: Pengurusan Insiden Keselamatan .....	218
Rajah 3-65: Metrik Utama Pengukuran Pelaksanaan DevSecOps .....	223
Rajah 4-1: Templat Artifak <i>Product Vision</i> .....	238
Rajah 4-2: Artifak <i>User Story</i> .....	240
Rajah 4-3: Artifak <i>Product Backlog</i> .....	242
Rajah 4-4: Artifak Perancangan Kapasiti.....	243
Rajah 4-5: Artifak <i>Sprint Backlog</i> Hari 1 bagi <i>Sprint 1</i> .....	244
Rajah 4-6: Artifak Log <i>Daily Scrum</i> .....	246
Rajah 4-7: Artifak <i>Sprint Backlog</i> sehingga Hari 10 .....	247
Rajah 4-8: Carta <i>Burndown</i> sehingga Hari ke 10 .....	248
Rajah 4-9: Artifak <i>Sprint Retrospective</i> .....	249
Rajah 4-10: Artifak <i>Definition of Done</i> .....	251

## SENARAI LAMPIRAN

Lampiran 1: Templat Artifak <i>Product Vision</i> .....	230
Lampiran 2: Templat Artifak <i>User Story</i> .....	231
Lampiran 3: Templat Artifak <i>Product Backlog</i> .....	232
Lampiran 4: Templat Artifak <i>Definition of Done</i> .....	233
Lampiran 5: Templat Artifak Perancangan Kapasiti .....	234
Lampiran 6: Templat Artifak <i>Sprint Backlog</i> .....	235
Lampiran 7: Templat Artifak <i>Burndown Chart</i> .....	236
Lampiran 8: Templat Artifak <i>Sprint Retrospective</i> .....	237
Lampiran 9: Kajian Kes <i>Agile Scrum</i> .....	238
Lampiran 10: Templat Kes Pengujian .....	252
Lampiran 11: Tahap Kritikal .....	253
Lampiran 12: Templat <i>Traceability Matrix Ujian</i> .....	254
Lampiran 13: Pengukuran Tahap Pelaksanaan DevOps .....	255
Lampiran 14: Konsep MVP dalam Pembangunan Produk .....	261
Lampiran 15: Peranan dan Tanggungjawab dalam Pengurusan <i>Tools</i> dan Proses CI/CD .....	263
Lampiran 16: Konfigurasi dalam Pelaksanaan DevOps .....	265
Lampiran 17: Kawalan Versi dalam Pelaksanaan DevOps .....	266
Lampiran 18: Senarai <i>Exit Criteria</i> bagi Aktiviti Utama Pelaksanaan DevOps .....	267
Lampiran 19: Perbandingan Teknik Pengujian Automatik TDD dan BDD .....	268
Lampiran 20: Metrik Pemantauan DevOps .....	269
Lampiran 21: Senarai <i>Tools DevOps</i> .....	271
Lampiran 22: Pelaporan Insiden .....	272
Lampiran 23: Templat Pelaporan Status Kemajuan Pembangunan Produk .....	273

# **PRAKATA**



**PRAKATA  
KETUA PENGARAH  
JABATAN DIGITAL NEGARA**

**Assalamualaikum Warahmatullahi Wabarakatuh,  
Salam Sejahtera, Salam Malaysia MADANI.**

Dengan nama Allah yang Maha Pemurah lagi Maha Mengasihani.

Alhamdulillah, syukur ke hadrat Allah yang Maha Esa kerana dengan limpah kurnia dan izin-Nya juga, maka dokumen Rangka Kerja Pelaksanaan DevOps bagi Pembangunan Sistem Aplikasi Sektor Awam ini telah berjaya diterbitkan. Pembangunan Rangka Kerja Pelaksanaan DevOps Sektor Awam ini adalah amat signifikan dan bertepatan dengan masa, seiring dengan penjenamaan semula Jabatan Digital Negara (JDN) dalam memacu pendigitalan perkhidmatan kerajaan ke arah penyampaian perkhidmatan kerajaan berpaksikan rakyat dan berpacukan data melalui pendekatan *Whole of Government (WoG)*.

Saya ingin mengucapkan tahniah dan syabas kepada ahli pasukan projek yang telah bersungguh-sungguh memerlukan keringat dalam menjayakan penerbitan dokumen rangka kerja ini. Penghasilan dokumen ini amat penting bagi memastikan pembangunan aplikasi kerajaan dapat dilaksanakan dengan lebih efisien melalui pendekatan baharu yang lebih holistik dan secara tidak langsung memberi kelebihan kepada kerajaan untuk memberikan yang terbaik kepada rakyat, lebih-lebih lagi dalam situasi global yang semakin mencabar. Situasi ini memerlukan penjawat awam bertindak dengan lebih responsif dan inovatif dalam menyampaikan perkhidmatan kepada pelanggan.

Sebagai usaha merealisasikan inisiatif pendigitalan dalam penyampaian perkhidmatan, kerajaan telah melancarkan Rangka Tindakan Ekonomi Digital Malaysia bagi

mentransformasikan Malaysia menjadi negara berpendapatan tinggi menjelang 2025. Selain itu, Pelan Strategik Pendigitalan Sektor Awam (PSPSA) 2021-2025 juga telah menggariskan hala tuju strategik pendigitalan sektor awam, selaras dengan aspirasi Wawasan Kemakmuran Bersama 2030 dan Rancangan Malaysia ke-12 (RMKe-12). Hasrat ini tiada lain, selain agar segala inisiatif pendigitalan yang dilaksanakan dapat dimanfaatkan oleh rakyat sepenuhnya, seterusnya mentransformasikan Malaysia kepada negara mampan melalui ekonomi digital selaras dengan aspirasi Malaysia MADANI.

Dalam mengorak langkah ke hadapan, beberapa inisiatif pendigitalan baharu perlu diberi penekanan terutama dalam pembangunan sistem aplikasi yang dilihat perlu berubah kepada lebih berfokuskan pengalaman pengguna selain reka bentuk berteraskan keselamatan dan mengadaptasi konsep-konsep baharu yang lebih fleksibel seperti *Agile* dan *DevOps*. Pelaksanaan DevOps dapat membantu agensi dalam memperkasakan pembangunan sistem yang merupakan tonggak dalam merealisasikan pendigitalan penyampaian perkhidmatan kerajaan.

Justeru, Dokumen Rangka Kerja Pelaksanaan DevOps Dalam Pembangunan Sistem Aplikasi Sektor Awam telah dihasilkan sebagai rujukan dan panduan kepada semua agensi kerajaan dan penjawat awam khasnya, dalam pelaksanaan DevOps bagi pembangunan sistem aplikasi sektor awam dengan cerdas, tangkas, selamat dan efisien. Saya yakin dan percaya, melalui kaedah pembangunan baharu yang digariskan dalam dokumen ini, akan berupaya memacu prestasi dan produktiviti penjawat awam ke arah penghasilan sistem aplikasi yang lebih mantap dan efisien bagi memenuhi ekspektasi rakyat.

Sekian, terima kasih.

**Ts. Dr. Fazidah binti Abu Bakar**

Ketua Pengarah

Jabatan Digital Negara

Kementerian Digital

## AKRONIM

<b>ACT- IAC</b>	<i>American Council for Technology-Industry Advisory Council</i>
<b>BRS</b>	<i>Business Requirement Specification</i>
<b>DASA</b>	<i>DevOps Agile Skills Association</i>
<b>ICT</b>	<i>Information and Communication Technology</i>
<b>MVP</b>	<i>Minimum Viable Product</i>
<b>NC4</b>	Pusat Penyelaras dan Kawalan Siber Negara ( <i>National Cyber Coordination and Command Centre</i> ).
<b>PMO</b>	<i>Project Management Office</i>
<b>PPP</b>	Pelan Pengurusan Projek
<b>PPS</b>	Pelan Pembangunan Sistem
<b>WBS</b>	<i>Work Breakdown Structure</i>

## TAKRIFAN

<b>Agile</b>	Pendekatan pembangunan sistem yang menekankan kaedah iteratif melibatkan perancangan dan pelaksanaan berterusan serta berulang-ulang secara kolaboratif di antara pemilik sistem dengan pasukan pembangun bagi membolehkan perubahan dapat dilaksanakan dengan lebih cepat dan berkesan.
<b>Blameless post mortem</b>	Mesyuarat pasukan selepas pelaksanaan projek bagi menyenaraikan isu yang berlaku dalam pasukan. Ini adalah untuk mencari penyelesaian secara bersama dan menghalang isu daripada berulang. Tujuan utama <i>blameless post mortem</i> adalah untuk menggalakkan komunikasi secara objektif (mementingkan fakta).
<b>User Story</b>	Keperluan pengguna yang telah diperhalusi dan boleh dilaksanakan dalam jangka masa satu <i>iteration</i> iaitu antara dua hingga empat minggu. Cerita pengguna membantu dalam penyediaan skop bagi sistem aplikasi yang akan dibangunkan berdasarkan keperluan pengguna.
<b>Container</b>	Platform untuk menempatkan kod, fail konfigurasi, kod binari dan <i>libraries</i> aplikasi ke dalam satu pakej yang dipanggil imej <i>container</i> . Imej ini berfungsi sebagai sistem operasi maya yang ringkas dengan penggunaan sumber sistem yang minimum.
<b>Definition of done (DoD)</b>	Kriteria dan syarat yang telah dipersetujui oleh pasukan dan pengurus produk sebelum projek atau cerita pengguna, boleh dianggap selesai.
<b>Epic</b>	Keperluan-keperluan pengguna yang mempunyai objektif yang sama dan boleh dihasilkan melalui beberapa <i>iteration</i> . Epik terdiri daripada beberapa cerita pengguna yang boleh dibahagikan kepada tugas terperinci berdasarkan keperluan pengguna.
<b>Hutang teknikal</b>	<i>Trade-off</i> teknikal yang berlaku dalam pengekodan, disebabkan oleh kekangan sumber tenaga atau masa. Hutang teknikal akan menambah tugas pengekodan dalam kitaran pembangunan seterusnya.
<b>Infrastructure as a Code (IaC)</b>	Pengautomasian penyediaan infrastruktur melalui konfigurasi skrip atau kod yang boleh diubah dengan pantas bagi mempercepatkan penyediaan infrastruktur dalam pembangunan sistem aplikasi.

<b>Iteration</b>	Kitaran pembangunan projek, yang biasanya mengambil masa antara dua hingga empat minggu. Pasukan membangunkan ciri-ciri produk yang boleh digunakan dan dihantar ke persekitaran produksi dalam <i>iteration</i> yang berulang. Projek terdiri daripada satu siri <i>iteration</i> , bersama-sama dengan mesyuarat perancangan sebelum pembangunan dan mesyuarat retrospektif pada akhir setiap <i>iteration</i> .  <i>Sprint</i> adalah contoh <i>iteration</i> pembangunan berdasarkan rangka kerja Scrum.
<b>Konvensyen pengekodan</b>	Garis panduan untuk bahasa pengaturcaraan yang merangkumi gaya pengaturcaraan, sintaks, amalan dan kaedah bagi mengekalkan konsistensi pengekodan bahasa tersebut.
<b>Lean</b>	Amalan pengurusan untuk meningkatkan kecekapan dan keberkesanan dengan mengelakkan pembaziran.
<b>Mean Time to Acknowledge (MTTA)</b>	Metrik bagi mengukur purata masa yang diperlukan oleh pasukan untuk bertindak balas terhadap aduan, gangguan atau insiden.
<b>Mean Time to Resolve (MTTR)</b>	Metrik bagi mengukur purata masa yang diperlukan oleh sistem untuk mendapatkan semula prestasi operasi normal setelah berlaku kegagalan sistem.
<b>Microservices</b>	Arkitektur secara modular bagi memastikan setiap perkhidmatan aplikasi tidak bergantung kuat antara satu sama lain.
<b>Minimum Viable Product (MVP)</b>	Konsep pembangunan sistem aplikasi secara iteratif berdasarkan maklum balas berterusan. MVP menghasilkan sistem aplikasi dengan fungsi minimum serta memberikan nilai kepada pengguna.
<b>Orchestration</b>	Proses bagi mengkoordinasi proses automasi dan konfigurasi yang digunakan dalam <i>pipeline CI/CD</i> .
<b>Pair programming</b>	Amalan kerjasama antara dua pembangun, iaitu salah seorang pembangun menulis kod manakala seorang lagi akan memberi ulasan dan maklum balas. Peranan bagi kedua-dua pembangun akan bertukar secara kerap iaitu setiap jam atau beberapa jam.
<b>Peer review</b>	Memastikan kod yang ditulis menepati konvensyen pengekodan. <i>Peer review</i> mendidik pembangun untuk mempelajari cara penulisan kod yang teratur, berkongsi amalan terbaik serta mengurangkan ralat dan pepijat dari tatacara pengaturcaraan.
<b>Pelan Pengurusan Projek (PPP)</b>	Dokumen yang menjadi rujukan utama dalam mengurus dan mengawal projek sistem aplikasi ICT dalam sektor awam merujuk kepada PPrISA.

<b>Pipeline</b>	Satu set proses dan <i>tools</i> yang membolehkan pembangun dan pasukan operasi bekerjasama dan berkolaborasi dalam DevOps. <i>Pipeline CI/CD</i> berlaku dalam seluruh peringkat DevOps dan menitikberatkan pengautomasian proses pembangunan sistem aplikasi.
<b>Produk</b>	Sistem aplikasi yang dibangunkan untuk menyokong fungsi bisnes organisasi.
<b>Projek</b>	Satu usaha sementara yang terdiri daripada beberapa siri aktiviti dan tugas yang dibuat untuk membangunkan produk. Projek perlu mempunyai objektif, tempoh masa dan bajet tertentu, berdasarkan spesifikasi yang ditetapkan serta sumber manusia dan peralatan.
<b>Retrospektif</b>	Mesyuarat pasukan pada akhir <i>iteration</i> untuk membincangkan kaedah kerja pasukan. Retrospektif perlu berlaku semasa projek supaya penambahbaikan yang berhasil boleh digunakan dan relevan dengan kerja yang akan datang pada projek yang sama. Aspek retrospektif seperti masa, ahli pasukan dan format mesyuarat harus konsisten.
<b>Semakan analisis statik</b>	Analisis yang digunakan untuk mengenal pasti masalah keselamatan semasa peringkat pengekodan. Semakan analisis statik semasa proses integrasi mengenal pasti masalah keselamatan sebelum proses pelepasan dilaksanakan. Analisis statik yang kerap akan memastikan kod tiada kelemahan keselamatan yang serius dalam setiap peringkat.
<b>Telemetry</b>	Log data yang berhasil dari sumber yang sukar dicapai dan dihantar ke sistem yang berbeza untuk pemantauan dan analisis. <i>Telemetry</i> membolehkan masalah semasa dikenal pasti dengan pantas. Pasukan DevOps boleh mengenal pasti metrik corak pengguna melalui <i>telemetry</i> dan mencipta amaran jika anomalি berlaku.
<b>Tools DevOps</b>	<i>Tools</i> yang digunakan bagi menggalakkan prinsip berterusan DevOps. Setiap peringkat DevOps mempunyai <i>tools</i> dengan fungsi yang tersendiri serta terbahagi mengikut kesesuaian dalam kematangan pelaksanaan DevOps. <i>Tools DevOps</i> yang dipilih bergantung kepada model dan infrastruktur dalam pembangunan sistem aplikasi.
<b>Tuggakan tugasan</b>	Senarai keutamaan tugasan yang perlu diusahakan oleh pasukan dalam <i>iteration</i> untuk memenuhi keperluan pengguna.



# **RINGKASAN EKSEKUTIF**

## RINGKASAN EKSEKUTIF

Dokumen Rangka Kerja Pelaksanaan DevOps dalam Pembangunan Sistem Aplikasi Sektor Awam ini diterbitkan sebagai panduan asas kepada semua pegawai teknikal yang ingin menerapkan pendekatan DevOps dalam pembangunan sistem aplikasi kerajaan, dengan mengambil kira kaedah dan amalan terbaik pelaksanaan DevOps sebagai satu standard yang perlu diikuti oleh agensi sektor awam.

Rangka kerja ini menggariskan komponen yang perlu diambil kira oleh kementerian dan agensi kerajaan dalam melaksanakan DevOps bagi pembangunan sistem aplikasi secara dalaman. Jabatan Digital Negara, Kementerian Digital telah membangunkan rangka kerja ini bagi mengurus dan memastikan semua aktiviti pembangunan sistem yang dilaksanakan menggunakan pendekatan DevOps mematuhi panduan yang telah digariskan.

Rangka kerja pelaksanaan DevOps ini merangkumi enam komponen utama yang terdiri daripada penerapan prinsip dan budaya, pengukuhan tadbir urus, pengadaptasian metodologi, pengukuran kematangan, pemerkasaan teknologi dan pematuhan keselamatan. Manakala pemboleh daya yang menyokong kejayaan pelaksanaan DevOps adalah dasar, kompetensi, automasi, pengurusan perubahan dan kawalan. Kesemua pemboleh daya ini berpaksikan kepada tiga prinsip asas bagi menjamin keberkesanan pelaksanaan DevOps sektor awam iaitu melalui proses penambahbaikan yang berterusan, penerapan nilai-nilai dalam pembangunan produk yang bersifat komunikatif serta jalinan kerjasama antara ahli pasukan secara kolaboratif. Selain daripada itu, metodologi *Agile Scrum* juga diperkenalkan untuk melengkapi proses kerja DevOps. Aktiviti seperti *sprint zero*, *sprint planning meeting*, *daily scrum*, *product backlog refinement*, *sprint review* dan *sprint retrospective* akan diterangkan bersama artifik-artifik *agile* yang menyokong pelaksanaan pendekatan DevOps di sektor awam.

Secara keseluruhannya, pendekatan pembangunan produk melalui pelaksanaan DevOps berupaya mempertingkatkan kolaborasi dan komunikasi pasukan yang secara tidak langsung melonjak produktiviti agensi sektor awam menjadi lebih cekap dan efisien. Kelebihan utama dalam pelaksanaan DevOps adalah pengurangan

tempoh masa bagi proses penempatan dalam penghasilan sesebuah produk, mempercepatkan proses adaptasi metodologi *agile* serta memastikan produk yang terhasil adalah lebih berkualiti dalam memenuhi kehendak pengguna.

Rangka kerja ini menjadi rujukan bagi membantu agensi sektor awam dalam menentukan kejayaan pelaksanaan DevOps ke arah transformasi sistem penyampaian perkhidmatan awam yang terbaik.



## **BAB 1**

# **PENDAHULUAN**

## BAB 1 : PENDAHULUAN

Seiring dengan peredaran masa yang pantas berubah, perkembangan teknologi maklumat juga semakin hari mengalami rentak perubahan yang pesat. Dalam persekitaran semasa yang serba pantas ini, penggunaan teknologi dan pendekatan terkini yang relevan dalam pembangunan sistem aplikasi merupakan salah satu langkah transformasi sektor awam untuk meningkatkan kecekapan dalam penyampaian perkhidmatan kerajaan.

Pelbagai metodologi dan *tools* bagi pembangunan sistem aplikasi telah diperkenalkan dan berkembang mengikut keperluan dan peredaran masa. Pendekatan DevOps merupakan salah satu pendekatan terkini dalam meningkatkan keberkesanan penyampaian perkhidmatan sektor awam. Pendekatan DevOps menggabungkan pasukan pembangun dan pasukan operasi menjadi satu pasukan, yang dikenali sebagai pasukan DevOps. Melalui pendekatan ini, terdapat proses-proses yang terlibat dalam pembangunan sistem aplikasi yang dilaksanakan secara automatik dengan menggunakan *tools* atau perisian tertentu, seterusnya mempercepatkan tempoh masa pembangunan sistem aplikasi dan memudahkan aktiviti penyelenggaraan.

Jabatan Digital Negara selaku agensi penggerak dan peneraju ICT telah melaksanakan kajian dalam mengenal pasti komponen-komponen DevOps yang perlu ada bagi pelaksanaan DevOps dalam pembangunan sistem aplikasi sektor awam. Metodologi kajian merangkumi kajian literatur ke atas jurnal, penerbitan berkaitan pelaksanaan DevOps daripada negara luar, dan perbincangan kumpulan fokus dengan agensi sektor awam yang telah mula melaksanakan DevOps. Selain daripada itu, semakan kualiti dan penyelaras bersama *Subject Matter Expert* (SME) berkaitan dengan panduan pengurusan projek dan pembangunan sistem aplikasi turut dilaksanakan. Kajian ini lebih memfokuskan kepada masalah dan kekangan pembangunan sistem sedia ada, serta persekitaran agensi sektor awam dalam proses pembangunan sistem. Selain itu, kajian turut mengadaptasikan komponen-komponen utama pelaksanaan DevOps di sektor awam dan mengenal pasti penyelesaian yang sesuai sebagai asas Rangka Kerja Pelaksanaan DevOps dalam pembangunan sistem aplikasi sektor awam.

Dokumen Rangka Kerja Pelaksanaan DevOps diwujudkan sebagai panduan bagi menerangkan komponen utama yang perlu diambil kira ke arah kejayaan pelaksanaan pendekatan DevOps dalam pembangunan sistem aplikasi sektor awam. Dokumen ini juga merupakan sumber rujukan untuk meningkatkan pengetahuan pegawai teknikal sektor awam yang terlibat dalam pelaksanaan DevOps.

Struktur dokumen ini dibahagikan kepada empat bab, seperti berikut:

- a. **Bab 1** menerangkan tujuan, latar belakang, skop, kumpulan sasaran dan objektif dokumen rangka kerja pelaksanaan DevOps bagi pembangunan sistem aplikasi sektor awam,
- b. **Bab 2** memberikan penerangan terhadap kajian kebolehlaksanaan DevOps dalam pembangunan sistem aplikasi sektor awam, definisi, evolusi pembangunan sistem aplikasi, objektif pelaksanaan dan manfaat DevOps,
- c. **Bab 3** menjelaskan tentang rangka kerja pelaksanaan DevOps bagi pembangunan sistem aplikasi sektor awam yang merangkumi penerapan prinsip dan budaya, pengukuhan tadbir urus, pengadaptasian metodologi, pengukuran kematangan, pemerkasaan teknologi, serta pematuhan keselamatan dan
- d. **Bab 4** merumuskan kandungan dokumen rangka kerja ini, faktor-faktor yang menyumbang kepada kejayaan pelaksanaan DevOps dan harapan terhadap penggunaan dokumen ini.

Dokumen ini perlu dibaca bersama dengan dokumen dasar dan garis panduan berkaitan yang telah dikeluarkan sebelum ini oleh Unit Pemodenan Tadbiran dan Perancangan Pengurusan Malaysia (MAMPU), khususnya:

- a. Panduan Pengurusan Projek ICT Sektor Awam (PPrISA),
- b. Panduan Kejuruteraan Sistem Aplikasi Sektor Awam (KRISA), dan
- c. Panduan Pelan Induk *Enterprise Architecture* (EA) Sektor Awam (MyGovEA): Pelaksanaan Pendekatan Reka Bentuk Berstruktur Ekosistem Organisasi Perkhidmatan Awam.

## **1.1. SKOP DOKUMEN**

Skop pelaksanaan DevOps sektor awam seperti yang digariskan dalam dokumen ini adalah khusus untuk pembangunan sistem aplikasi secara dalaman oleh pegawai teknikal sektor awam. Dokumen ini juga boleh digunakan bagi aktiviti pembangunan sistem aplikasi baharu dan juga aktiviti penambahbaikan atau peningkatan kepada sistem aplikasi sedia ada.

Kumpulan sasaran utama dokumen rangka kerja ini adalah:

- a. Pegawai teknikal yang berperanan sebagai ahli pasukan pembangun dan ahli pasukan operasi yang terlibat dalam pembangunan dan penyelenggaraan sistem aplikasi,
- b. Bahagian/unit di agensi sebagai pemilik bisnes berperanan sebagai pemilik produk, dan
- c. Mana-mana pihak yang berkepentingan untuk menjayakan pelaksanaan DevOps dalam pembangunan sistem aplikasi.

## 1.2. OBJEKTIF DOKUMEN

Objektif penyediaan dokumen ini adalah seperti berikut:

- a. Memperkenalkan amalan pelaksanaan DevOps bagi meningkatkan keupayaan dan kualiti penyampaian perkhidmatan sektor awam.
- b. Memperkenalkan metodologi *agile* bagi menyokong pelaksanaan pendekatan DevOps dalam pembangunan sistem aplikasi sektor awam.
- c. Memberikan pemahaman dan panduan mengenai proses pembangunan sistem aplikasi dengan penerapan elemen keselamatan berterusan dalam pelaksanaan DevOps.



## **BAB 2**

# **KONSEP DAN DEFINISI**

## BAB 2 : KONSEP DAN DEFINISI

Pendekatan pembangunan DevOps secara asasnya dapat mempertingkatkan kolaborasi dan interaksi di antara pasukan pembangun dan pasukan operasi supaya lebih efisien dan dinamik. DevOps berupaya menggabungkan pembangunan perisian dan operasi ICT ke dalam satu pasukan dan platform.

### 2.1. PERSEKITARAN SEMASA PEMBANGUNAN SISTEM APLIKASI

Kajian kebolehlaksanaan DevOps dalam pembangunan sistem aplikasi sektor awam telah dilaksanakan pada Mac hingga April 2020 bagi meninjau landskap semasa dan tahap ketersediaan agensi kerajaan. Kajian ini juga bertujuan mendapatkan pandangan daripada personel ICT sektor awam terhadap kebolehlaksanaan DevOps dalam pembangunan sistem aplikasi di sektor awam. Berdasarkan dapatan kaji selidik yang telah dilaksanakan, didapati bahawa kesedaran dan pendedahan mengenai DevOps dalam kalangan responden masih rendah. Walau bagaimanapun, majoriti menyokong untuk melaksanakan DevOps memandangkan pendekatan ini dapat meningkatkan kualiti pembangunan sistem aplikasi.

Di samping itu, beberapa siri bengkel juga telah dianjurkan dari Januari 2022 hingga Mac 2024 melibatkan beberapa agensi sektor awam yang telah mula melaksanakan DevOps untuk mengenal pasti isu permasalahan dalam pelaksanaan DevOps. Hasil dapatan daripada bengkel-bengkel tersebut, masalah dan penyelesaian dalam pembangunan sistem sedia ada dan persekitaran di agensi sektor awam telah dikenal pasti. Selain itu, komponen-komponen utama rangka kerja serta penyelesaian yang sesuai bagi pelaksanaan DevOps dalam pembangunan sistem aplikasi sektor awam turut dihasilkan.

## 2.2. DEFINISI DEVOPS

DevOps merupakan cara pemikiran dan pendekatan yang baharu dengan menggabungkan falsafah budaya, amalan dan *tools* bagi meningkatkan keupayaan agensi dalam pembangunan sistem aplikasi dan penyampaian perkhidmatan. Pendekatan DevOps juga menggabungkan pasukan pembangun dan pasukan operasi untuk bekerjasama secara kolaboratif dan komunikatif sebagai satu pasukan bagi meningkatkan tahap produktiviti.

Menurut Gartner, DevOps ditafsirkan sebagai pendekatan berorientasikan sistem bertujuan merapatkan jurang antara pasukan pembangun dan pasukan operasi melalui metodologi *agile*. Penekanan terhadap peningkatan komunikasi dan kerjasama serta penggabungan amalan dan *tools* automasi dapat meningkatkan keupayaan agensi dalam menyampaikan perkhidmatan yang optimum.

*“DevOps represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach. DevOps emphasizes people (and culture), and it seeks to improve collaboration between operations and development teams. DevOps implementations utilize technology — especially automation tools that can leverage an increasingly programmable and dynamic infrastructure from a life cycle perspective.”*

- Gartner Glossary by  
Gartner, Inc. And/or its affiliates

(Sumber: Glossary dari Gartner, 2020<sup>1</sup>)

Menurut *American Council for Technology-Industry Advisory Council* (ACT-IAC), DevOps menerapkan perubahan budaya dan pemikiran secara kolaboratif dan komunikatif terhadap pasukan pembangun dan pasukan operasi dengan mengutamakan amalan kerja secara berpasukan.

*“DevOps is the move towards a collaborative and communicative mindset and culture with streamlined practices by bringing the development and operations groups together to work as one team”*

- ACT-IAC

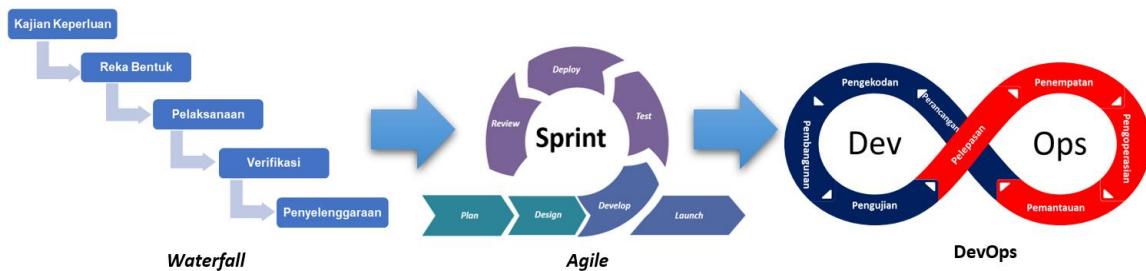
(Sumber: Glossary dari ACT-IAC, 2020<sup>2</sup>)

<sup>1</sup>Definition of DevOps - Gartner Information Technology Glossary. (2020). Gartner. <https://www.gartner.com/en/information-technology/glossary/devops>.

<sup>2</sup>American Council for Technology-Industry Advisory Council (ACT-IAC). (2020). *DevOps Primer: Case Studies and Best Practices from Across Government. What Is DevOps in Government.*

### 2.3. EVOLUSI PEMBANGUNAN SISTEM APLIKASI

Seiring perkembangan semasa, industri teknologi maklumat turut memperlihatkan perkembangan pembangunan sistem aplikasi melalui pendekatan, metodologi dan teknologi yang digunakan. Evolusi ini mencerminkan tindak balas terhadap perubahan dalam pengukuhan bisnes, perkembangan teknologi dan peningkatan pemahaman terhadap cara terbaik pembangunan sistem aplikasi. Organisasi sektor awam juga perlu beradaptasi untuk memanfaatkan kemajuan teknologi dan perkembangan metodologi bagi meningkatkan penyampaian yang lebih efisien dan efektif.



**Rajah 2-1: Evolusi Pembangunan Sistem Aplikasi**

Rajah 2-1 menunjukkan evolusi pembangunan sistem aplikasi yang melibatkan tiga metodologi utama iaitu *waterfall*, *agile* dan DevOps. Kesesuaian sesebuah metodologi pembangunan sistem aplikasi adalah bergantung kepada jenis organisasi, keupayaan dan keperluan teknikal, serta jenis dan pasukan projek. Penerangan untuk setiap metodologi adalah seperti berikut:

#### a. *Waterfall*

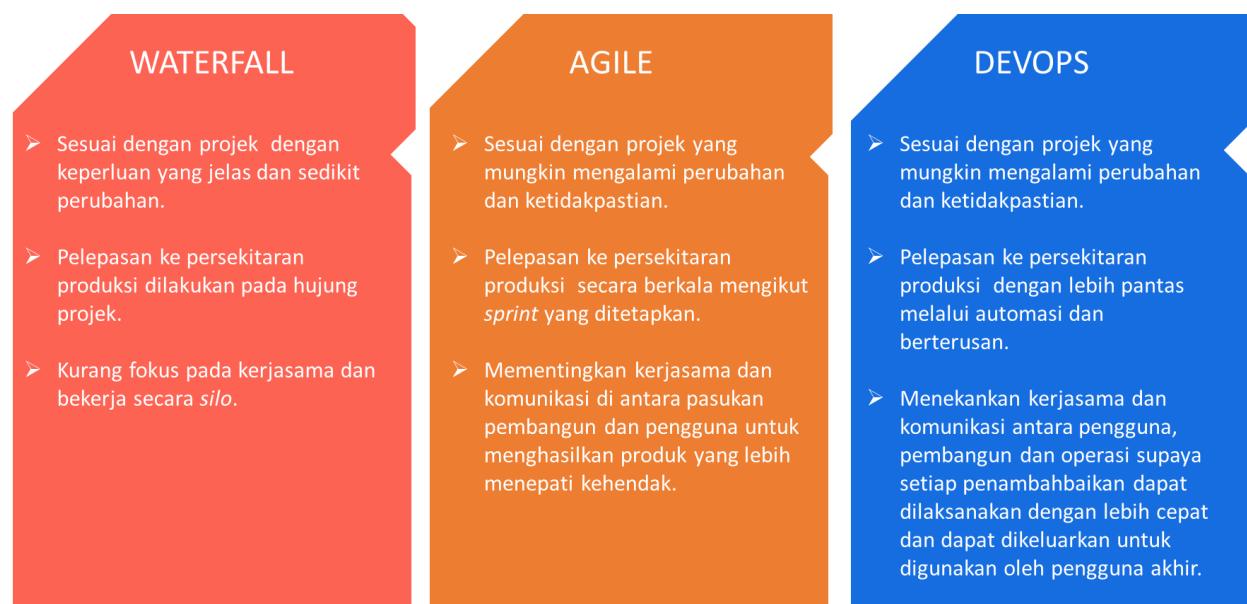
Metodologi *waterfall* juga dikenali metodologi jujukan linear. Metodologi ini menerapkan kawalan yang ketat terhadap setiap peringkat dalam pembangunan sistem aplikasi. Semakan dan pengesahan serahan dilakukan secara formal dengan pihak berkepentingan pada setiap penghujung peringkat. Sesuatu peringkat seterusnya tidak akan dimulakan sekiranya semakan dan pengesahan serahan bagi peringkat semasa tidak disempurnakan.

### b. Agile

Metodologi *agile* merujuk kepada pendekatan pembangunan secara peningkatan (incremental) dan pengulangan (iterative) yang menumpukan kepada penghasilan produk yang pantas kepada pengguna dalam tempoh yang singkat dan berulang. Metodologi *agile* mementingkan kerjasama erat antara pengguna dan pasukan pembangun untuk menghasilkan produk yang berkualiti.

### c. DevOps

DevOps merupakan pendekatan atau budaya dalam pembangunan sistem aplikasi yang membolehkan pasukan untuk membangun, menguji dan melepaskan sistem aplikasi dengan lebih pantas dan efisien dengan menggabungkan prinsip dan amalan *agile*. DevOps bertujuan merapatkan jurang antara pengguna, pasukan pembangun dan pasukan operasi dengan menggalakkan kerjasama yang erat. Penggabungan amalan dan *tools* antara pasukan secara automasi ini dapat meningkatkan keupayaan agensi dalam menyampaikan perkhidmatan yang lebih pantas dan konsisten.



**Rajah 2-2: Perbandingan antara Waterfall, Agile dan DevOps**

Rajah 2-2 memaparkan perbandingan secara ringkas antara metodologi pembangunan sistem aplikasi. Pemilihan metodologi bergantung kepada keperluan projek, saiz projek, keadaan dan matlamat organisasi. Kombinasi metodologi *agile* dan DevOps semakin menjadi trend untuk menyatukan pembangunan yang lebih pantas dengan pelepasan dan operasi yang efisien.

## **2.4. OBJEKTIF PELAKSANAAN DEVOPS**

Objektif pelaksanaan DevOps adalah seperti berikut:

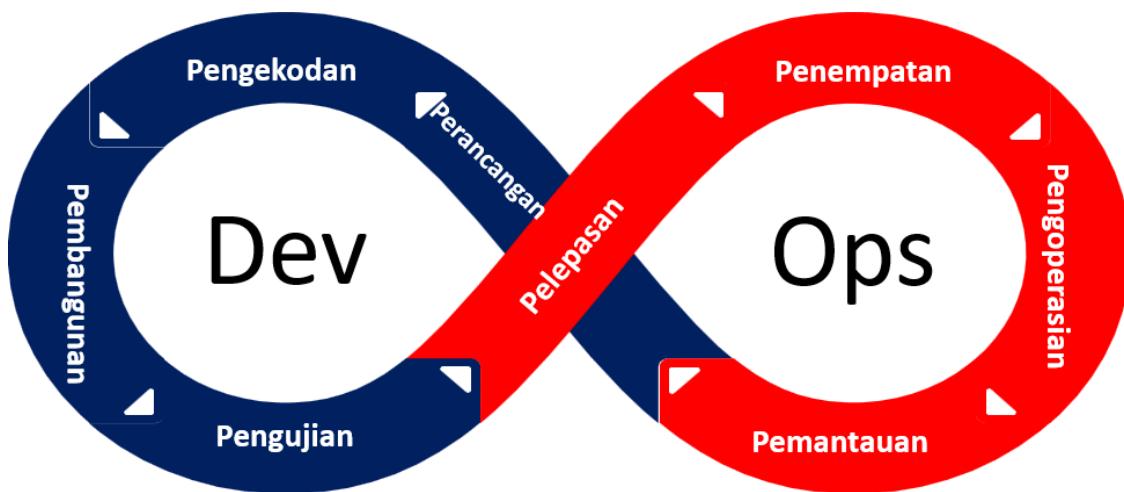
- a. Meningkatkan hubungan kerjasama antara pasukan pembangun dan pasukan operasi menerusi amalan kerja secara kolaboratif dan komunikatif,
- b. Mempercepatkan proses pembangunan sistem aplikasi bagi meningkatkan tahap penyampaian perkhidmatan digital di agensi, dan
- c. Meningkatkan tahap kualiti dan ketersediaan sistem aplikasi serta mengoptimumkan kos dan penggunaan tenaga sumber manusia.

Objektif tersebut adalah bagi menyokong visi pelaksanaan DevOps sektor awam iaitu 'Memacu kecekapan pembangunan sistem aplikasi sektor awam ke arah penyampaian perkhidmatan digital yang berkualiti'.

## 2.5. PERINGKAT-PERINGKAT DALAM PELAKSANAAN DEVOPS

Pelaksanaan DevOps membentuk kitaran dan digunakan untuk mengenal pasti peringkat-peringkat dalam kitaran hayat pembangunan sistem aplikasi. Terdapat lapan peringkat iteratif dalam kitaran hayat DevOps yang merangkumi keupayaan, proses dan teknik yang penting untuk pembangunan sistem aplikasi. Peringkat-peringkat DevOps adalah seperti di Rajah 2-3.<sup>3</sup>

Peringkat iteratif yang signifikan dalam kitar hayat DevOps ialah integrasi berterusan (Continuous Integration-CI), penempatan berterusan (Continuous Deployment-CD) dan penyampaian berterusan (Continuous Delivery). Penggunaan *tools* yang sesuai pada setiap peringkat kitar hayat DevOps adalah sangat penting untuk memastikan proses pembangunan sistem aplikasi dapat dilaksanakan dengan lancar dan berkesan.



**Rajah 2-3: Peringkat-Peringkat dalam Pelaksanaan DevOps**

<sup>3</sup> MAMPU (2019). Section 2 - DevOps Approach. Dalam *Software Quality Management Using Devops Approach*. Diakses pada 9 Jun 2022 dari <https://sqa.mampu.gov.my/index.php/ms/section-1-introduction>.

**a. Peringkat Perancangan (Plan)**

Peringkat ini melibatkan pembangunan/penyediaan kes bisnes, penetapan tujuan dan objektif, penyediaan pelan projek serta pembentukan struktur organisasi.

**b. Peringkat Pengekodan (Code)**

Peringkat ini melibatkan reka bentuk arkitektur sistem aplikasi, pemilihan metodologi pembangunan sistem aplikasi, penentuan teknologi untuk digunakan dalam projek dan pembangunan kod sumber.

**c. Peringkat Pembangunan (Build)**

Peringkat ini melibatkan kompilasi dan integrasi kod, serta *packaging* sistem aplikasi. Kod dikompilasi dan dibina menjadi binari yang boleh digunakan atau *packages code*.

**d. Peringkat Pengujian (Test)**

Peringkat ini melibatkan validasi dan verifikasi sistem aplikasi. Ujian merangkumi pengujian fungsian dan bukan fungsian.

**e. Peringkat Pelepasan (Release)**

Peringkat ini melibatkan proses pengurusan pelepasan yang merangkumi proses kawalan perubahan dan jadual pelepasan sistem aplikasi.

**f. Peringkat Penempatan (Deploy)**

Peringkat ini melibatkan penempatan sistem aplikasi ke persekitaran produksi.

#### **g. Peringkat Pengoperasian (Operate)**

Peringkat ini melibatkan pengoperasian sistem aplikasi dalam persekitaran produksi. Pasukan akan memantau prestasi, ketersediaan dan keselamatan sistem aplikasi serta bertindak balas terhadap sebarang isu yang timbul

#### **h. Peringkat Pemantauan (Monitor)**

Peringkat ini melibatkan aktiviti pemantauan berterusan bagi memastikan sistem berada dalam keadaan baik dari segi kebolehgunaan, prestasi dan ketersediaan sistem. Peringkat ini penting bagi pasukan DevOps untuk mendapatkan maklum balas berterusan berkenaan status dan prestasi sistem.

## 2.6. MANFAAT PELAKSANAAN DEVOPS

Manfaat yang diperolehi daripada pelaksanaan DevOps adalah seperti Rajah 2-4.<sup>4</sup>



**Rajah 2-4: Manfaat Pelaksanaan DevOps**

### a. Penghasilan produk yang lebih pantas dan berkualiti

Mempercepatkan pembangunan sistem aplikasi melalui maklum balas pengguna sepanjang tempoh *iteration* dan meningkatkan kualiti sistem aplikasi melalui pendekatan pembangunan sistem aplikasi secara *incremental*.

### b. Peningkatan komunikasi

Meningkatkan tahap komunikasi di antara pasukan pembangun dan pasukan operasi untuk mengelakkan pasukan bekerja secara *silo*.

<sup>4</sup> Devops Agile Skills Association LLC. (2020, September). DASA DevOps Fundamentals (2.0.1 ed.) [E-book]. Devops Agile Skills Association LLC. .

**c. Peningkatan nilai penyampaian**

Mewujudkan persekitaran supaya proses integrasi dan penyampaian (delivery) dilaksanakan secara berterusan bagi memastikan tumpuan diberikan kepada perubahan dan nilai tambah terhadap sistem aplikasi.

**d. Masa penyampaian yang optimum**

Integrasi dan penempatan berterusan menjadi teras dalam pendekatan pelaksanaan DevOps dapat memastikan masa yang diperlukan untuk penghasilan produk adalah optimum.

**d. Penjimatkan sumber**

Automasi merupakan asas dalam pelaksanaan DevOps. Aktiviti seperti pengujian, pengurusan infrastruktur, pemantauan serta pelaporan yang dilaksanakan secara automasi dapat membantu proses pembangunan dan pengoperasian menjadi lebih efisien dan menjimatkan penggunaan sumber (manusia, masa dan kos).

**e. Keselamatan yang Dipertingkatkan**

Mempertingkatkan ekosistem keselamatan sistem aplikasi secara ketara dengan melaksanakan langkah pengujian keselamatan sepanjang kitaran pembangunan sistem aplikasi. Pendekatan proaktif ini membantu pasukan DevOps mengenal pasti kelemahan keselamatan lebih awal seterusnya dapat mengurangkan risiko keselamatan pada produk akhir.

**f. Pengurusan Risiko yang Berkesan**

Mengenal pasti, menilai dan mengurangkan potensi risiko keselamatan secara berkesan sepanjang kitaran hayat pembangunan sistem aplikasi. Pendekatan ini melibatkan penyepaduan langkah keselamatan di peringkat awal pembangunan sistem aplikasi, memantau kelemahan secara berterusan, dan menangani sebarang isu dengan pantas untuk memastikan sistem aplikasi yang lebih selamat dan berdaya tahan.



## **BAB 3**

# **KOMPONEN RANGKA KERJA**

## BAB 3 : KOMPONEN RANGKA KERJA

Terdapat enam komponen (K1 hingga K6) dalam rangka kerja ini iaitu Penerapan Prinsip dan Budaya, Pengukuran Tadbir Urus, Pengadaptasian Metodologi, Pengukuran Kematangan, Pemerksaan Teknologi dan Pematuhan Keselamatan. Rajah 3-1 memaparkan komposisi komponen dalam melaksanakan amalan DevOps bagi pembangunan sistem aplikasi di sektor awam.

# RANGKA KERJA PELAKSANAAN DEVOPS SEKTOR AWAM



Rajah 3-1: Komponen Pelaksanaan DevOps

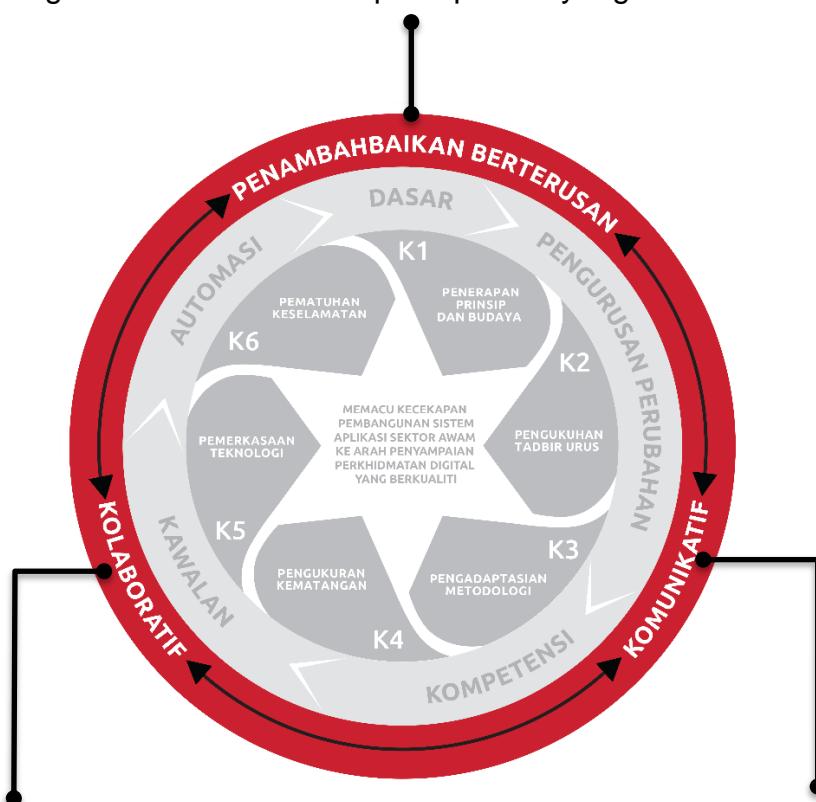
### VISI

Memacu kecekapan pembangunan sistem aplikasi sektor awam ke arah penyampaian perkhidmatan digital yang berkualiti.

## PRINSIP ASAS

Tiga prinsip asas yang menyokong kejayaan pelaksanaan DevOps seperti di Rajah 3-2.

Fokus kepada amalan integrasi, pengujian, penempatan, penyampaian, pemantauan dan keselamatan yang berterusan di sepanjang kitar hayat pembangunan sistem aplikasi. Amalan ini dapat mengoptimumkan penggunaan sumber seperti masa, tenaga dan kos serta meningkatkan kecekapan pembangunan sekali gus menambah nilai kepada produk yang dihasilkan



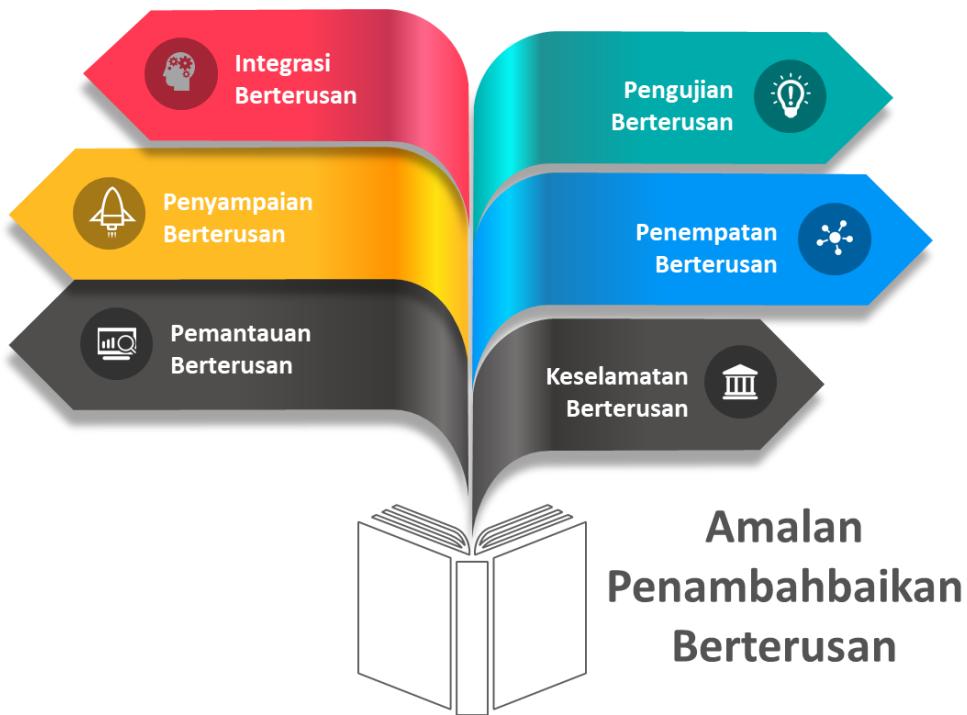
Jalinan usaha sama strategik di antara pasukan bagi tujuan perkongsian sumber, pengalaman, dan kepakaran. Budaya kerja secara kolaboratif akan mengerakkan semua ahli pasukan seperti pengurus produk, pasukan pembangun dan pasukan operasi untuk bertanggungjawab sepenuhnya dalam pembangunan sistem

Keupayaan berkomunikasi dengan jelas dan berkesan penting bagi meningkatkan persefahaman ahli pasukan. Penerapan nilai komunikasi yang efektif secara tidak langsung dapat menghasilkan pasukan yang berwibawa dan mendorong kepada peningkatan produktiviti

**Rajah 3-2: Prinsip Asas Pelaksanaan DevOps**

Amalan penambahbaikan berterusan merupakan pendekatan dalam pengurusan kualiti yang bertujuan meningkatkan kecekapan, keberkesan dan kebolehpercayaan sistem aplikasi melalui penambahbaikan berterusan. Dengan amalan penambahbaikan berterusan, sistem aplikasi yang dibangunkan akan ditambah baik, diuji dan dilepaskan ke persekitaran produksi dalam kitaran yang berulang (iterative). Kaedah pengulangan (iteration) ini dapat mengurangkan risiko kegagalan sistem melalui perubahan kecil dan lepasan secara kerap (frequent release). Konsep integrasi dan penyampaian berterusan (CI/CD) melibatkan pelaksanaan automasi merentas peringkat DevOps bagi meminimumkan proses manual dari pengekodan sehingga penempatan kod ke persekitaran yang bersesuaian.

Enam pendekatan utama dalam amalan penambahbaikan berterusan seperti di Rajah 3-3:



**Rajah 3-3: Amalan Penambahbaikan Berterusan**

### a. Integrasi berterusan

Integrasi berterusan atau *continuous integration* (CI) adalah pendekatan dalam DevOps yang melibatkan pengintegrasian perubahan kod sumber secara automatik dan berterusan. Proses integrasi berterusan ini berlaku apabila pembangun membuat perubahan pada kod sumber dan mengintegrasikan kod secara berkala ke repositori pengurusan kod sumber berpusat. Proses ini membolehkan pembangun mengesan masalah dengan lebih awal dalam proses pembangunan, mempercepat pengesahan perubahan dan meminimumkan risiko yang terlibat apabila dilepaskan ke persekitaran produksi. Integrasi berterusan merupakan elemen penting dalam penyampaian dan penempatan berterusan supaya sistem aplikasi dapat dilepaskan dengan selamat, berulang dan cepat.

### b. Pengujian berterusan

Pengujian berterusan atau *continuous testing* merupakan satu pendekatan DevOps yang melibatkan pengujian secara automatik dan berterusan daripada sistem aplikasi yang sedang dibangunkan. Tujuan pengujian berterusan ini adalah untuk memastikan bahawa setiap perubahan pada kod sumber diuji secara automatik dan diintegrasikan dengan kod sumber yang sedia ada serta memastikan bahawa sistem aplikasi yang dihasilkan memenuhi keperluan bisnes dan mempunyai kualiti yang baik.

### c. Penyampaian berterusan

Penyampaian berterusan atau *continuous delivery* (CD) merupakan pendekatan dalam DevOps yang melibatkan penghantaran sistem aplikasi ke persekitaran yang berkaitan secara automatik dan berterusan. Tujuan penyampaian berterusan ini adalah untuk memastikan setiap perubahan pada kod sumber diintegrasikan, diuji secara automatik dan sedia untuk dihantar ke persekitaran yang berkaitan.

#### **d. Penempatan berterusan**

Penempatan berterusan atau *continuous deployment* (CD) merupakan pendekatan dalam DevOps yang melibatkan penempatan sistem aplikasi di persekitaran yang berkaitan secara automatik dan berterusan. Tujuan penempatan berterusan ini adalah untuk mempercepatkan penghantaran dan meminimumkan waktu pelepasan perubahan kod sumber yang telah diuji ke persekitaran produksi.

#### **e. Pemantauan berterusan**

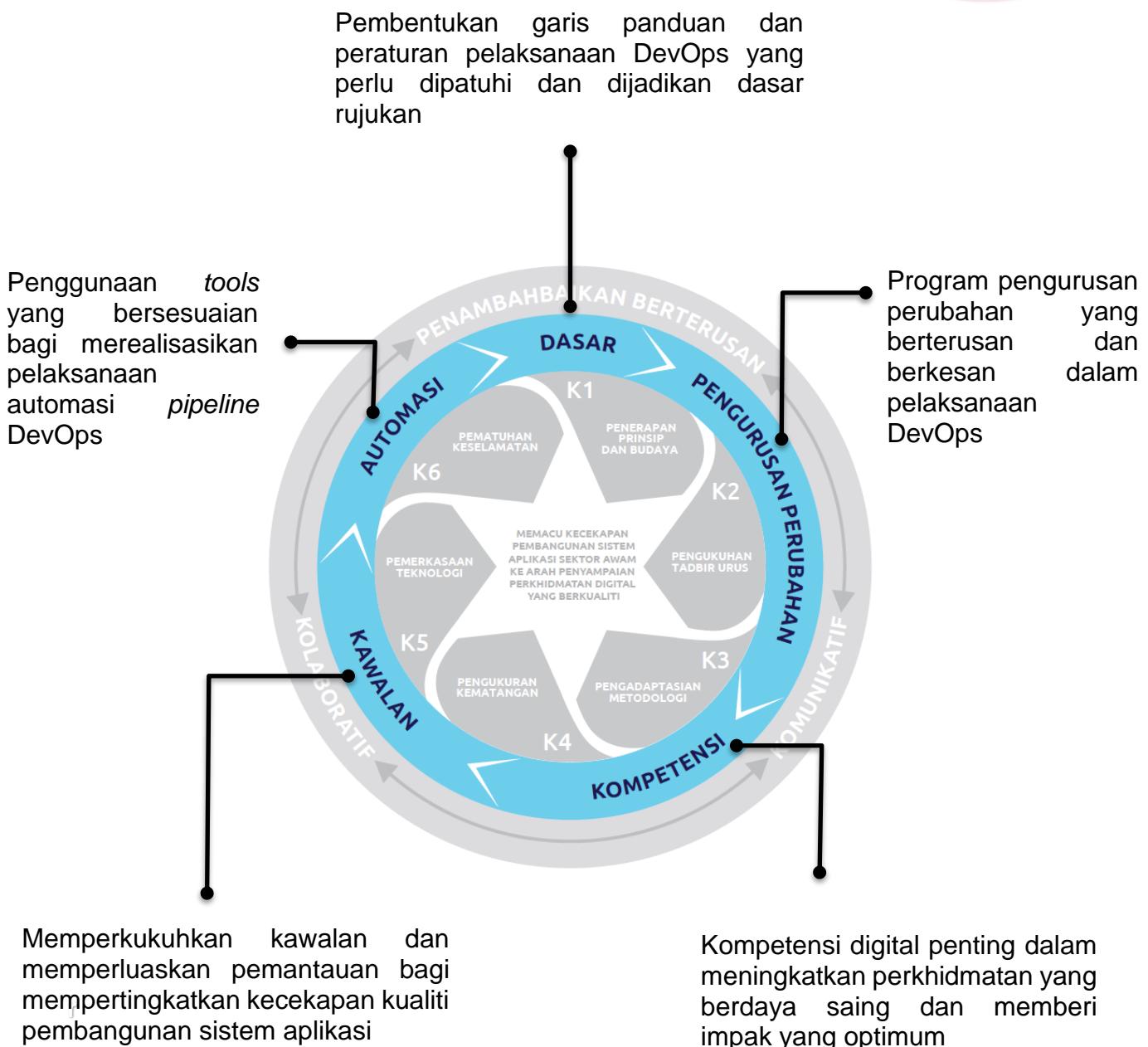
Pemantauan berterusan atau *continuous monitoring* adalah pendekatan dalam DevOps yang melibatkan pemantauan dan analisis secara berterusan daripada *tools* pemantauan. Tujuan proses pemantauan berterusan ini adalah supaya sistem aplikasi berjalan lancar dan pasukan DevOps dapat memberikan maklum balas dengan pantas terhadap masalah atau ralat. Pemantauan berterusan juga dapat mengurangkan atau meminimumkan gangguan fungsi produk kepada pengguna akhir dan memastikan kebolehcapaian produk berada pada tahap optimum.

#### **f. Keselamatan berterusan**

Keselamatan berterusan atau *continuous security* merupakan penginterasian amalan keselamatan dalam pembangunan sistem aplikasi secara berterusan sepanjang *pipeline* DevOps. Keselamatan berterusan berupaya untuk mengenal pasti, mencegah dan mengawal isu keselamatan secara automatik dan berterusan. Sejajar dengan prinsip kolaborasi dan komunikasi antara pasukan pembangun dan pasukan operasi, produk dapat dihasilkan dengan lebih pantas dan selamat.

## PEMBOLEH DAYA

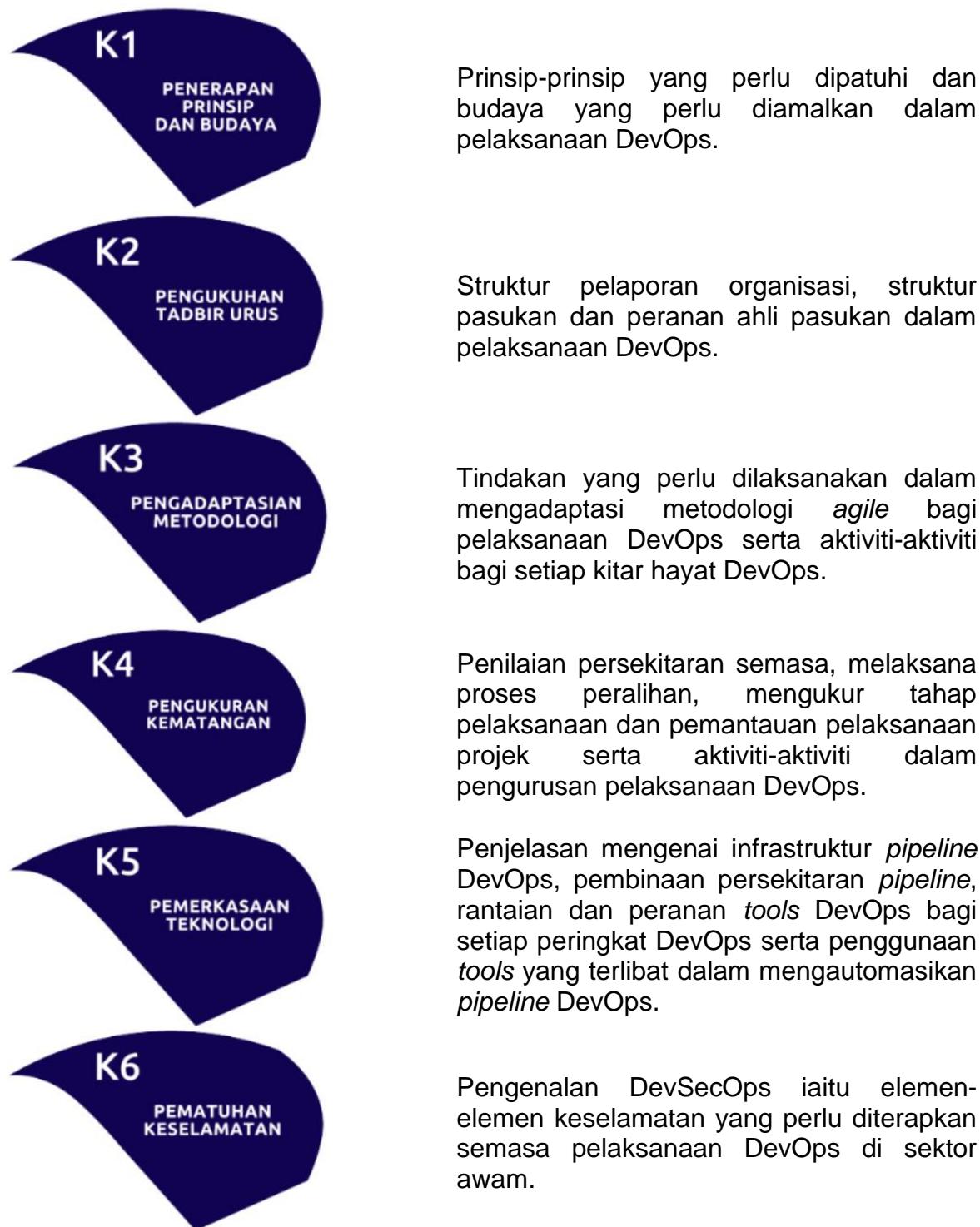
Rajah 3-4 menunjukkan pemboleh daya yang menyokong kejayaan pelaksanaan Komponen DevOps:



**Rajah 3-4: Pemboleh Daya Pelaksanaan DevOps**

## KOMPONEN

Enam komponen utama bagi pelaksanaan DevOps sektor awam adalah seperti di Rajah 3-5:



Rajah 3-5: Komponen Pelaksanaan DevOps

### **3.1. KOMPONEN 1 (K1): PENERAPAN PRINSIP DAN BUDAYA**

DevOps merupakan satu pendekatan yang berkait rapat dengan amalan kerja sesuatu pasukan. Pelaksanaan DevOps akan mencapai kejayaan dengan mengamalkan prinsip dan budaya DevOps. Bahagian ini akan menerangkan secara terperinci terhadap prinsip dan budaya yang perlu dipatuhi dan diamalkan dalam pelaksanaan DevOps.

#### **3.1.1. Prinsip DevOps**

Prinsip DevOps yang diadaptasi daripada *DevOps Agile Skills Association* (DASA) serta penerapan keselamatan melalui pendekatan *shifting security left*<sup>5,6</sup> daripada *DevOps Research and Assessment* (DORA) boleh diamalkan bagi membantu memudahkan pelaksanaan DevOps adalah seperti di Rajah 3-6.<sup>7, 8, 9</sup>.

---

<sup>5</sup> DORA | *DevOps Capabilities: Shifting left on Security*. (n.d.-b). <https://dora.dev/devops-capabilities/process/shifting-left-on-security/>

<sup>6</sup> Myrbakken, H., & Colomo-Palacios, R. (2017). DevSecOps: a multivocal literature review. In *Software Process Improvement and Capability Determination: 17th International Conference, SPICE 2017, Palma de Mallorca, Spain, October 4–5, 2017, Proceedings* (pp. 17-29). Springer International Publishing.

<sup>7</sup> Coupland, M. (2021). *DevOps Adoption Strategies: Principles, Processes, Tools, and Trends: Embracing DevOps through effective culture, people, and processes*. Packt Publishing.

<sup>8</sup> Devops Agile Skills Association LLC. (2020, September). DASA DevOps Fundamentals (2.0.1 ed.) [E-book]. Devops Agile Skills Association LLC.

<sup>9</sup> Deloitte. (2020). *DevOps Point of View. An Enterprise Architecture Perspective*, 17.



**Rajah 3-6: Prinsip DevOps**

**a. Mengutamakan Kehendak Pengguna**

Prinsip ini memerlukan ahli pasukan untuk sentiasa mengambil kira semua maklum balas daripada pengguna untuk melaksanakan penambahbaikan terhadap produk.

**b. Menetapkan Matlamat Berdasarkan *Milestone***

Penetapan matlamat berdasarkan hasil yang ingin dicapai sama ada untuk jangka masa pendek, sederhana atau panjang.

**c. Memberi Komitmen Menyeluruh dalam Penghasilan Produk**

Pendekatan DevOps memastikan setiap ahli pasukan perlu melaksanakan peranan dan tanggungjawab bersama pada setiap peringkat kitar hayat pembangunan sistem aplikasi. Setiap ahli pasukan akan terlibat dari peringkat awal hingga peringkat akhir bagi pembangunan produk. Dengan cara ini, tanggungjawab yang perlu digalas dapat dikurangkan serta meningkatkan tahap akauntabiliti dalam kalangan pasukan.

#### **d. Mewujudkan Pasukan Pelbagai Disiplin**

Setiap ahli pasukan perlu terdiri daripada individu yang mempunyai set kemahiran yang seimbang supaya mampu melaksanakan setiap tugas yang memerlukan kemahiran yang berbeza. Selain itu, setiap ahli pasukan juga perlu mempunyai kemampuan untuk menguruskan tugas dan membuat keputusan yang tepat.

#### **e. Melaksanakan Penambahbaikan Produk Secara Berterusan**

Penggunaan metodologi *Agile* dalam pembangunan produk adalah digalakkan dalam pelaksanaan DevOps bagi memastikan penambahbaikan produk dapat dilaksanakan secara berterusan. Melalui kaedah ini, produk dapat diserahkan penggunaannya secara berperingkat bagi memberi pengalaman awal terhadap penggunaan produk tersebut. Sebarang kelemahan atau kekurangan produk dapat dibaiki pada peringkat awal.

#### **f. Mengautomasikan Proses Pembangunan Produk**

Mengautomasikan proses membolehkan segala aktiviti iteratif dilaksanakan dengan mudah, cepat dan berkesan serta mengurangkan risiko berlakunya ralat dan meningkatkan produktiviti ahli pasukan.

#### **g. *Shifting Security Left***

Mengintegrasikan elemen keselamatan seawal mungkin dalam kitar hayat pembangunan sistem aplikasi yang melibatkan pelaksanaan kawalan dan semakan keselamatan pada setiap peringkat DevOps. Prinsip ini bertujuan untuk membina amalan terbaik berkaitan keselamatan dalam pembangunan sistem aplikasi untuk mengesan dan menyelesaikan potensi kelemahan keselamatan seawal mungkin dalam proses pembangunan dengan bantuan *tools* automasi.

Dengan cara ini, pengguna dapat melihat penambahbaikan yang dilaksanakan terhadap produk secara nyata tanpa sebarang gangguan terhadap penggunaan dan secara tidak langsung akan meningkatkan tahap ketersediaan produk.

### 3.1.2. Budaya DevOps dan Penerapan dalam Pasukan

Pendekatan DevOps dapat dilaksanakan dengan lebih berkesan, melalui penerapan budaya seperti di Rajah 3-7, seiring dengan prinsip-prinsip DevOps yang digariskan.

Tujuan pelaksanaan budaya DevOps adalah untuk meningkatkan kualiti dan produktiviti dalam pelaksanaan tugas atau aktiviti, membentuk komunikasi yang lebih baik dan menambah baik kerjasama merentasi ahli pasukan.



**Rajah 3-7: Budaya DevOps untuk Mencapai Prinsip DevOps**

Jadual 3-1 menyenaraikan penerangan berkaitan aspek budaya dan cadangan aktiviti yang boleh dilaksanakan oleh agensi berdasarkan prinsip DevOps.

**Jadual 3-1: Penerangan Berkaitan Aspek-Aspek Budaya dan Cadangan Aktiviti**

Aspek Budaya	Penerangan	Cadangan Aktiviti
<b>Menggalakkan Pembelajaran dan Peningkatan Kemahiran yang Berterusan</b>	Membangunkan tahap kemahiran dan daya kompeten ahli pasukan dengan menekankan budaya pembelajaran secara berterusan dan perkongsian pengetahuan berkaitan idea dan teknologi baharu	<ul style="list-style-type: none"> <li>i. Melaksanakan sesi perkongsian pengetahuan terhadap teknologi atau penyelesaian baharu dan menggalakkan setiap ahli pasukan untuk terlibat dalam sesi ini, dan</li> <li>ii. menyediakan program latihan dan coaching untuk ahli pasukan bagi menambahkan dan meningkatkan tahap kemahiran.</li> </ul>
<b>Menggalakkan Eksperimentasi dan Keberanian Mengambil Risiko</b>	Mewujudkan pengembangan idea-idea baharu dengan kaedah menggalakkan ahli pasukan meneroka dan bereksperimentasi dengan idea mereka	<ul style="list-style-type: none"> <li>i. Menyediakan platform bagi melaksanakan sesi perkongsian pengetahuan untuk ahli pasukan membentangkan idea-idea mereka,</li> <li>ii. memberi kepercayaan kepada ahli pasukan dan menghargai setiap idea-idea baharu yang dicadangkan oleh setiap pasukan, dan</li> <li>iii. menyediakan kemudahan yang diperlukan oleh setiap ahli pasukan bagi mencuba dan melaksanakan idea-idea baharu mereka.</li> </ul>
<b>Mengutamakan Kualiti Produk</b>	Menerapkan tanggungjawab bersama dalam memastikan kualiti	<ul style="list-style-type: none"> <li>i. Menggalakkan setiap ahli pasukan untuk bertanggungjawab dan terlibat dalam setiap peringkat kitar</li> </ul>

Aspek Budaya	Penerangan	Cadangan Aktiviti
	produk yang dihasilkan berada pada tahap tertinggi	<p>hayat pembangunan sistem aplikasi,</p> <ul style="list-style-type: none"> <li>ii. mengamalkan ketelusan dalam penerimaan maklum balas atau pengujian dengan berkongsi dan memaparkan secara terperinci hasil pengujian atau maklum balas yang diterima kepada setiap ahli pasukan,</li> <li>iii. mengautomasi proses-proses yang dilaksanakan secara manual bagi meminimumkan ralat yang terhasil daripada kelalaian ahli pasukan serta meningkatkan kualiti produk yang dihasilkan, dan</li> <li>iv. melaksanakan penambahbaikan secara berterusan terhadap produk berdasarkan setiap maklum balas yang diterima daripada pengguna.</li> </ul>
<b>Menyokong Pemikiran Kreatif dan Inovatif</b>	Menerapkan pemikiran secara kreatif dan inovatif dalam menyelesaikan masalah yang dihadapi bagi mengoptimumkan proses dan masa pembangunan produk	<ul style="list-style-type: none"> <li>i. Melaksanakan program-program inovasi dan menggalakkan penyertaan oleh setiap ahli pasukan bagi membolehkan mereka berfikir secara kreatif dan inovatif,</li> <li>ii. menyediakan platform dan kemudahan yang dapat membantu setiap ahli pasukan untuk berinovasi bagi menghasilkan penyelesaian baharu, dan</li> <li>iii. membentuk komuniti pengamal (communities of practices - COP) untuk peningkatan dan perkongsian pengetahuan dan pengalaman. Ini akan memberi keyakinan kepada setiap ahli pasukan untuk</li> </ul>

Aspek Budaya	Penerangan	Cadangan Aktiviti
		berinovasi dalam menyelesaikan sesuatu tugasan berdasarkan kreativiti mereka.
<b>Menerapkan Budaya Pemikiran Berasaskan Produk</b>	Menggalakkan penyelesaian isu dengan melihat produk sebagai satu penyelesaian lengkap tanpa menumpukan kepada fungsi-fungsi khusus sahaja	<ul style="list-style-type: none"> <li>i. Menggalakkan pengguna untuk memberikan maklum balas tentang produk dan memastikan ahli pasukan memahami serta bertindak balas secara cepat dan efisien, dan</li> <li>ii. menggalakkan ahli pasukan untuk menyampaikan taklimat berkaitan produk yang dibangunkan di seminar dan penerbitan di laman sesawang.</li> </ul>
<b>Mewujudkan Persekitaran Kerja yang Ideal dan Kondusif</b>	Mewujudkan persekitaran kerja yang selesa dari segi fizikal dalam memperkuuhkan budaya organisasi berprestasi tinggi dan inovatif	<ul style="list-style-type: none"> <li>i. Melaksanakan bengkel bagi membangun dan menetapkan peranan dan tanggungjawab ahli pasukan,</li> <li>ii. mewujudkan matlamat yang ingin dicapai oleh ahli pasukan, dan</li> <li>iii. meningkatkan komunikasi secara berkesan bagi mewujudkan kepercayaan dan ketelusan antara ahli pasukan.</li> </ul>

Aspek Budaya	Penerangan	Cadangan Aktiviti
<b>Mengutamakan Nilai dan Kepuasan Pengguna</b>	<p>Memberi tumpuan bagi memastikan produk yang dibangunkan memenuhi kehendak dan ekspektasi pengguna secara keseluruhan</p>	<ul style="list-style-type: none"> <li>i. Menggalakkan setiap ahli pasukan untuk berinteraksi dengan pengguna bagi memahami dengan lebih mendalam terhadap keperluan dan kehendak pengguna,</li> <li>ii. melaksanakan sesi demonstrasi kepada pengguna di setiap peringkat penyampaian produk untuk mendapatkan maklum balas terhadap produk yang dibangunkan, dan</li> <li>iii. melaksanakan penambahbaikan terhadap produk secara berterusan berdasarkan maklum balas yang diterima daripada pengguna bagi memastikan produk yang dibangunkan mencapai matlamat dan objektif yang telah ditetapkan.</li> </ul>

### **3.1.3. Pengenalan Metodologi *Agile***

DevOps melengkapi *agile* dengan memperluaskan prinsip dan amalan *agile* untuk merangkumi seluruh kitaran hidup pembangunan dan penghantaran perisian. Sementara *agile* memberi tumpuan terutamanya kepada peringkat pembangunan, DevOps mengambil pendekatan yang lebih luas, menangani perbezaan antara pembangunan dan operasi, dan memastikan peralihan yang lancar dari pembangunan kod ke penempatan sistem aplikasi. DevOps berperanan sebagai penghubung yang memperluaskan prinsip dan amalan *agile* melebihi peringkat pembangunan. DevOps memperkayakan metodologi *agile* dengan menyediakan pendekatan yang komprehensif yang merangkumi seluruh kitaran hidup pembangunan dan penyampaian sistem aplikasi.

Pembangunan sistem aplikasi dengan metodologi *agile* merujuk kepada pendekatan pembangunan secara peningkatan (incremental) dan pengulangan (iterative) iaitu penambahbaikan keperluan pengguna adalah selari dengan aktiviti pembangunan melalui kerjasama di antara pasukan pembangun dan pengurus produk serta semua pihak berkepentingan yang berkaitan. Matlamat utama pembangunan *agile* adalah untuk membolehkan pasukan pembangun menghasilkan produk yang lebih pantas, berkualiti dan bertindak balas secara cepat terhadap perubahan.

#### **3.1.3.1. Nilai dan Prinsip *Agile***

Falsafah *agile* ini disokong oleh empat nilai dan 12 prinsip yang didokumenkan dalam manifesto *agile*<sup>10</sup>. Empat nilai *agile* menggalakkan proses pembangunan produk yang memberi tumpuan kepada kualiti dan penghasilan produk yang memenuhi keperluan dan jangkaan pengguna. Manakala 12 prinsip *agile* yang digariskan bertujuan menyediakan persekitaran kerja yang tertumpu kepada keperluan pengguna dan sejajar dengan objektif penghasilan produk.

---

<sup>10</sup> Manifesto for Agile Software Development. (n.d.). Diakses pada April 4, 2023, dari <https://agilemanifesto.org/>

Pelaksanaan manifesto *agile* memberikan nilai yang lebih baik untuk pengguna dan meningkatkan kecekapan serta keberkesanannya dalam proses pembangunan sistem aplikasi. Berikut merupakan penerangan terhadap empat nilai utama *manifesto agile* seperti di Rajah 3-8:



**Rajah 3-8: Nilai-nilai Utama Agile**

**a. Individu dan interaksi melebihi proses dan alatan**

Menekankan kepentingan komunikasi, kerjasama dan kerja berpasukan di kalangan ahli pasukan. Interaksi antara ahli pasukan membantu meningkatkan kerjasama dan faktor terpenting dalam pembangunan sistem aplikasi yang berjaya.

**b. Produk yang berfungsi melebihi dokumentasi komprehensif**

Menekankan kepentingan menyampaikan produk yang berfungsi kepada pengguna seawal mungkin dan meminimumkan dokumentasi yang tidak diperlukan.

### c. Kerjasama pelanggan melebihi rundingan kontrak

Menekankan kepentingan melibatkan pengguna akhir dalam proses pembangunan dan bekerjasama rapat untuk memahami keperluan dan kehendak pengguna. Perubahan keperluan pengguna dari semasa ke semasa dapat di peroleh secara segera seterusnya membawa kepada hasil produk yang lebih baik.

### d. Bertindak balas terhadap perubahan melebihi perancangan

Menyesuaikan perancangan dan pembangunan sistem aplikasi terhadap perubahan keadaan, keutamaan dan keperluan pengguna secara fleksibel.



**Rajah 3-9: Prinsip Agile**

Terdapat 12 prinsip *agile* yang terkandung dalam manifesto *agile* seperti Rajah 3-9 iaitu:

- i. Penyampaian produk yang memenuhi keperluan dan jangkaan pengguna melalui hasil kerja yang lebih cepat serta berterusan.
- ii. Perubahan keperluan pengguna diutamakan dan disesuaikan dengan cepat sepanjang tempoh pembangunan.
- iii. Penyampaian produk yang berfungsi pada setiap lelaran secara kerap.
- iv. Pemilik bisnes dan pembangun produk bekerjasama sepanjang tempoh pembangunan.
- v. Produk dibangunkan dalam kalangan individu yang bermotivasi, saling memberi sokongan dan kepercayaan agar tugas boleh diselesaikan.
- vi. Perbincangan secara bersemuka.
- vii. Produk yang berfungsi adalah ukuran utama kemajuan.
- viii. Persekitaran pembangunan yang mampan membantu pasukan bekerja dengan cekap dan produktif.
- ix. Perhatian yang berterusan kepada kecemerlangan teknikal.
- x. Penumpuan kepada perkara yang penting dan perlu.
- xi. Pasukan yang berdikari.
- xii. Penilaian prestasi pasukan secara berkala ke arah penambahbaikan.

### **3.1.3.2. Agile Scrum**

Terdapat beberapa pendekatan pembangunan sistem aplikasi berdasarkan metodologi *agile* seperti *Scrum*, *Kanban*, *Lean Software Development* dan *Extreme Programming*. Setiap pendekatan tersebut mengandungi konsep dan prinsipnya yang tersendiri. Namun begitu, metodologi *scrum* merupakan metodologi yang paling terkenal dan berkesan turut digunakan oleh organisasi besar seperti Google, Apple, Facebook, Adobe, AirBnB, Spotify dan Yahoo dalam mengurus operasi harian mereka.

Pendekatan *scrum* merupakan salah satu pendekatan *agile* yang menggalakkan kolaborasi yang efektif dan berterusan dalam kalangan ahli pasukan DevOps. Pendekatan ini dilengkapi dengan aktiviti, peranan dan artifikat yang boleh diperlakukan oleh pasukan DevOps bagi membolehkan mereka bekerja secara adaptif dan responsif terhadap perubahan keperluan pengguna. Gabungan falsafah *agile* dan pendekatan *scrum* membentuk metodologi *agile scrum* yang memfokuskan kepada pembangunan secara meningkat dan berulang (incremental and iterative development) melalui kaedah pelaksanaan yang berstruktur.

Berikut merupakan empat manfaat utama pelaksanaan DevOps menggunakan metodologi *agile scrum*:

**a. Berupaya mengurus perubahan keperluan pengguna dengan cepat**

Bertepatan dengan prinsip *agile* yang menggalakkan perubahan di sepanjang tempoh pembangunan, sebarang perubahan pada keperluan pengguna dapat diurus dengan lebih efektif dan berkesan.

**b. Ketelusan yang lebih tinggi**

Pasukan pembangun melaporkan dan mengemas kini status aktiviti *sprint* setiap hari melalui artifik *agile scrum*. Ini menjadikan kandungan dan status kemajuan pembangunan semasa termasuk keputusan ujian, dapat dilihat oleh pasukan, pengurusan dan semua pihak yang berkepentingan.

**c. Kolaborasi yang berkesan**

Menggalakkan kolaborasi yang berterusan dengan menghapuskan *silo* melalui komunikasi terbuka dalam kalangan ahli pasukan DevOps.

**d. Pelepasan produk yang pantas**

*Minimum viable product* (MVP) boleh dilepas dan digunakan oleh pengurus produk dengan pantas pada setiap akhir *sprint*. Melalui maklum balas yang diterima, penambahbaikan berterusan akan dibangun dan dilepaskan pada *sprint* seterusnya.

Pelaksanaan DevOps dalam pembangunan sistem aplikasi sektor awam adalah menggunakan metodologi *agile* dengan mengadaptasikan amalan dan proses *agile scrum*.

### 3.2. KOMPONEN 2 (K2): PENGUKUHAN TADBIR URUS

Tadbir urus dalam pelaksanaan DevOps penting bagi menyelaras pembangunan produk. Pelaporan mengenai prestasi pembangunan produk secara DevOps boleh dilaksanakan melalui jawatankuasa sedia ada atau perlu menubuhkan Jawatankuasa Pengurusan Produk. Jawatankuasa sedia ada merujuk kepada Jawatankuasa Pemandu ICT (JPICT) atau Mesyuarat Pengurusan agensi dengan menambah terma rujukan berkaitan dengan pelaksanaan DevOps. Terma rujukan ini diwujudkan bagi memastikan pengurusan produk yang sistematik dan berkesan.

Sekiranya Jawatankuasa Pengurusan Produk perlu diwujudkan, jawatankuasa ini dicadangkan berada di bawah JPICT atau Mesyuarat Pengurusan agensi.

#### 3.2.1. Struktur Pelaporan Organisasi

Cadangan struktur pelaporan dalam pelaksanaan DevOps dengan Jawatankuasa Pengurusan Produk adalah seperti di Rajah 3-10.



**Rajah 3-10: Struktur Pelaporan dalam Pelaksanaan DevOps**

Cadangan bagi keanggotaan Jawatankuasa Pengurusan Produk adalah seperti di Jadual 3-2.

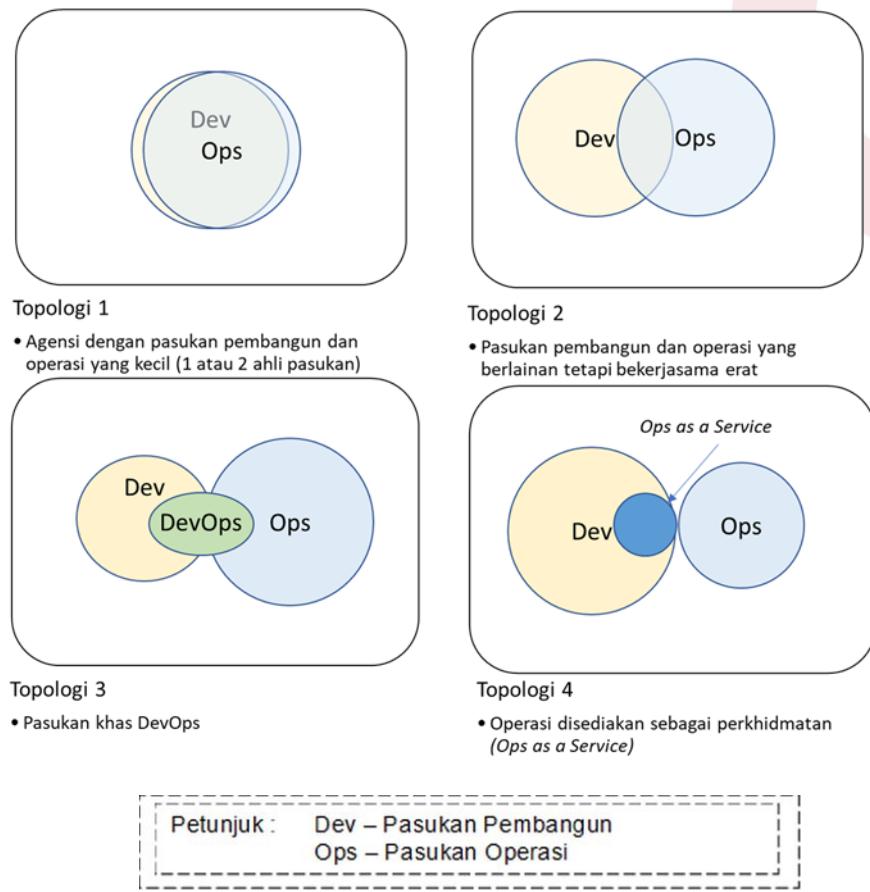
**Jadual 3-2: Cadangan Keanggotaan Jawatankuasa Pengurusan Produk**

No.	Keahlian	Cadangan
1.	Pengerusi	Ketua Pegawai Digital atau Ketua Agensi/Pengarah ICT Agensi
2.	Ahli-ahli	Pengurus-pengurus produk di agensi yang terlibat dalam pelaksanaan DevOps
3.	Kekerapan Mesyuarat	Sebulan sekali atau mengikut keperluan
4.	Terma Rujukan	<ul style="list-style-type: none"> <li>i. Menentukan hala tuju dan strategi dalam hal ehwal produk di agensi,</li> <li>ii. Membincang dan mengesyorkan penyelesaian isu-isu berkaitan produk, dan</li> <li>iii. Memantau pembangunan dan pelaksanaan produk.</li> </ul>

### 3.2.2. Struktur Pasukan

Struktur pasukan harus dirancang dengan teliti sebelum pembangunan produk. Manakala lokasi persekitaran pasukan juga perlu dipertimbangkan bagi menggalakkan komunikasi pasukan secara efisien. Topologi struktur pasukan<sup>11</sup> yang menyokong prinsip DevOps adalah seperti di Rajah 3-11.

<sup>11</sup> Skelton, M., & Pais, M. (2019). Team topologies: organizing business and technology teams for fast flow. It Revolution.



**Rajah 3-11: Topologi Struktur Pasukan DevOps**

### a. Topologi 1 - Perkongsian Tanggungjawab dan Peranan

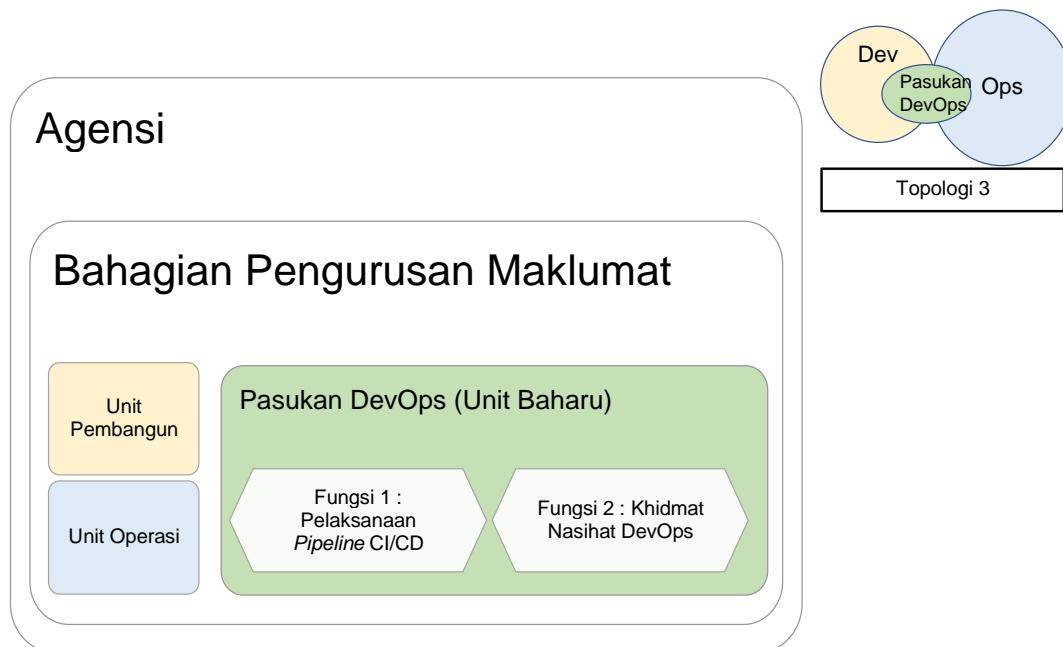
Topologi ini berlaku kepada agensi yang mempunyai perjawatan yang kecil dimana peranan pembangunan dan operasi dilaksanakan oleh pegawai yang sama. Struktur pasukan DevOps ini menggalakkan kerjasama yang erat dan cekap antara pasukan pembangun dan pasukan operasi (termasuk pasukan sokongan seperti pasukan kualiti dan keselamatan).

### b. Topologi 2 - Kolaborasi Pembangun dan Operasi

Topologi ini membentuk struktur pasukan pembangun dan pasukan operasi yang bekerjasama erat walaupun pasukan ini berada dalam unit yang berbeza. Struktur ini memerlukan sokongan daripada pengurusan atasan dalam memastikan kolaborasi pasukan berjalan lancar.

### c. Topologi 3 - DevOps sebagai Perkhidmatan (DevOps as a Service)

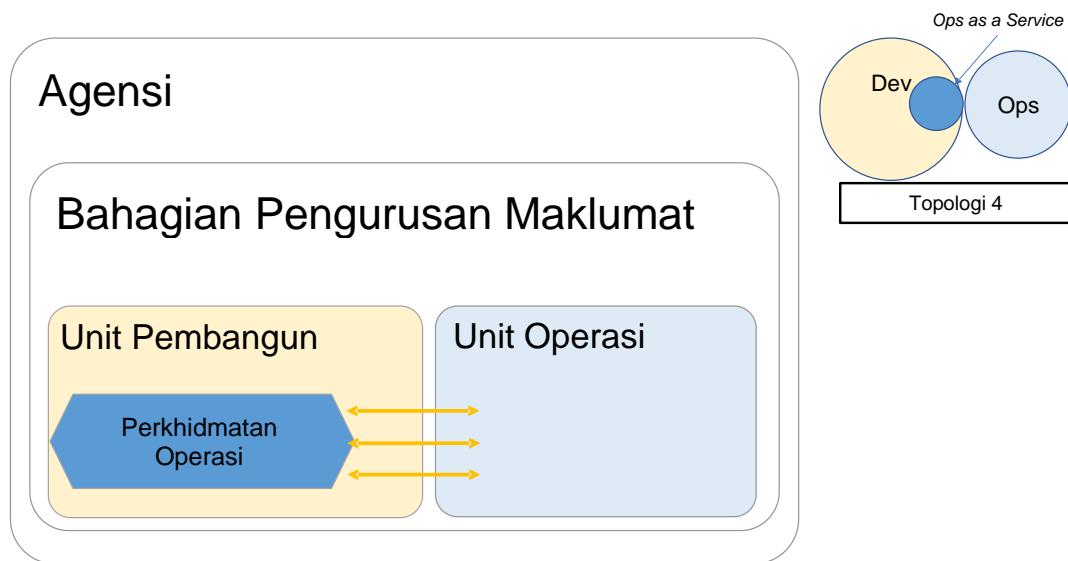
Struktur ini menyediakan pasukan DevOps khas yang memudahkan pelaksanaan DevOps dan membantu menyebarkan kesedaran tentang amalan DevOps. Pasukan DevOps khas ini mempunyai dua fungsi iaitu melaksanakan pengautomasian proses dan membangunkan *pipeline CI/CD* bagi kegunaan dalaman, dan memberi khidmat nasihat serta bimbingan untuk amalan dan penggunaan *tools* DevOps. Topologi ini boleh menjadi bertentangan dengan amalan DevOps jika pasukan ini terus bekerja secara *silo*. Gambaran pelaksanaan topologi ini adalah seperti di Rajah 3-12.



Rajah 3-12: Gambaran Pelaksanaan Topologi 3

#### d. Topologi 4 - Operasi sebagai Perkhidmatan (Ops as a Service)

Struktur ini memastikan pasukan operasi mengautomasikan penyediaan infrastruktur termasuk rangkaian, sistem pemantauan, dan penyediaan persekitaran bagi kegunaan pasukan pembangun. Pasukan pembangun akan menjalankan tanggungjawab dalam pengoperasian. Kolaborasi antara pasukan operasi dan pasukan pembangun adalah terhad dan hanya tertumpu kepada perkhidmatan operasi yang disediakan. Pasukan pembangun akan beransur-ansur menjadi pasukan DevOps dengan memahirkan diri dalam proses pengoperasian. Gambaran pelaksanaan topologi ini adalah seperti di Rajah 3-13.

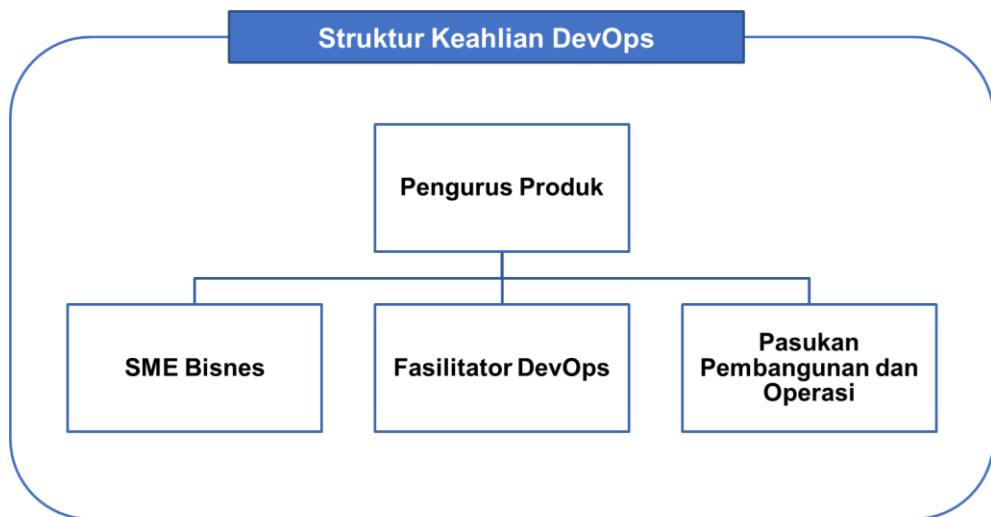


Rajah 3-13: Gambaran Pelaksanaan Topologi 4

Berdasar empat topologi yang dibincangkan, topologi terbaik yang disarankan adalah topologi 3 atau 4 dengan mengambil kira sumber tenaga di agensi.

### 3.2.3. Ahli Pasukan

Bilangan ahli pasukan DevOps yang dicadangkan adalah antara lima hingga 10 orang dalam satu pasukan dan bilangan ahli pasukan yang minimum adalah digalakkan.<sup>12</sup> Beberapa pasukan DevOps boleh diwujudkan jika terdapat ramai ahli pasukan dalam pembangunan produk. Cadangan keahlian pasukan DevOps adalah seperti Rajah 3-14.



**Rajah 3-14: Cadangan Keahlian Pasukan**

#### 3.2.3.1. Pengurus Produk

Pengurus produk merupakan individu yang mengetuaui dan bertanggungjawab untuk membuat keputusan berkenaan hala tuju produk yang ingin dibangunkan. Pengurus produk boleh terdiri daripada pemilik bisnes atau pegawai teknikal.

#### 3.2.3.2. SME Bisnes

SME ialah seseorang yang mempunyai kepakaran dan pengetahuan dalam proses bisnes produk yang dibangunkan. SME bisnes tidak terlibat secara

<sup>12</sup> Cagle, R., Rice, T., & Kristan, M. (2018). *DevOps for federal acquisition*. MITRE CORP BEDFORD MA.

langsung dalam aktiviti *scrum*, tetapi perlu bekerja rapat dengan pasukan pembangunan dan pengurus produk untuk memberikan panduan dan pandangan berkenaan dengan bidang kepakarannya.

SME bisnes bertanggungjawab untuk memberikan input, spesifikasi dan keperluan sebenar produk. Penglibatan SME yang berkemahiran dan kompeten dalam proses bisnes diperlukan dalam melaksanakan aktiviti pembangunan produk.

### **3.2.3.3. Fasilitator DevOps**

Fasilitator DevOps merupakan individu yang bertanggungjawab membantu pasukan memudahkan, mempraktikkan dan mematuhi pelaksanaan prinsip serta amalan *scrum*. Fasilitator DevOps memainkan peranan sebagai *scrum master* dalam mengurus dan melancarkan setiap aktiviti *scrum* seperti *scrum planning meeting*, *daily scrum*, *sprint review* dan *sprint retrospective*. Fasilitator DevOps juga bertanggungjawab bagi melancarkan aktiviti yang berkaitan teknikal.

### **3.2.3.4. Pasukan Pembangunan dan Operasi**

Pasukan pembangun dan operasi merupakan sebuah pasukan yang bertanggungjawab membangunkan produk yang dikehendaki oleh pengurus produk. Secara idealnya, pasukan ini terdiri daripada lima hingga 10 orang yang mempunyai pelbagai kemahiran dalam menghasilkan produk untuk memenuhi *definition of done* (DoD) pada setiap *sprint*. Cadangan peranan ahli pasukan ini adalah terdiri daripada juruanalisis bisnes, pembangun, penguji dan pereka antara muka. Ahli pasukan boleh memainkan beberapa peranan bergantung kepada kemahiran, saiz pasukan dan kompleksiti produk.

### 3.2.3.5. Peranan Ahli Pasukan

Peranan, tugasan dan kemahiran ahli pasukan DevOps adalah seperti di Jadual 3-3.

**Jadual 3-3: Peranan, Tugasan dan Kemahiran Ahli Pasukan**

Peranan	Tugasan	Kemahiran
<b>Pengurus Produk</b>	<ul style="list-style-type: none"> <li>i. Menentukan visi dan hala tuju produk dengan berkolaborasi bersama SME Bisnes untuk memahami keperluan pengguna, mengumpulkan maklum balas dan memastikan produk memenuhi keperluan pengguna,</li> <li>ii. menguruskan <i>product backlog</i> dan menyusun keutamaan ciri-ciri produk mengikut keperluan pengguna. Pengurus produk secara berkala perlu meninjau dan menyesuaikan <i>product backlog</i> berdasarkan maklum balas dan keutamaan yang berubah,</li> <li>iii. memastikan produk yang dihasilkan menepati keperluan pengguna seperti keperluan <i>User Interface (UI)</i> dan fungsi sistem,</li> <li>iv. menjadi penyelaras bagi pihak berkepentingan dalam memahami nilai produk yang bakal dibangunkan,</li> <li>v. mengutamakan penghasilan produk jangka panjang berdasarkan keperluan pengguna,</li> <li>vi. merancang strategi pelaksanaan DevOps,</li> <li>vii. mengenal pasti set <i>tools</i> yang boleh mengoptimumkan produktiviti pasukan,</li> </ul>	<ul style="list-style-type: none"> <li>i. Berpengetahuan tentang sistem pengurusan yang digunakan oleh pasukan,</li> <li>ii. pantas dan dinamik dalam menangani perubahan fungsi produk,</li> <li>iii. perlu mewujudkan hubungan yang baik dalam pasukan dan menyokong prinsip dan amalan DevOps secara proaktif, dan</li> <li>iv. boleh menerangkan keperluan teknikal dan keperluan pengguna dengan mudah kepada pihak berkepentingan.</li> </ul>

Peranan	Tugasan	Kemahiran
	<ul style="list-style-type: none"> <li>viii. memastikan pasukan mempunyai sumber yang diperlukan bagi <i>pipeline CI/CD</i> yang berterusan,</li> <li>ix. memastikan penghasilan produk berdasarkan keperluan pengguna dalam tempoh masa dan kos yang ditetapkan, dan</li> <li>x. menyediakan pelaporan kepada pihak berkepentingan.</li> </ul>	
<b>SME Bisnes</b>	<ul style="list-style-type: none"> <li>i. Menjadi pakar rujuk berkaitan bisnes,</li> <li>ii. mengesahkan produk yang dihasilkan menepati keperluan bisnes,</li> <li>iii. menyediakan panduan mengenai keperluan dan skop produk,</li> <li>iv. membantu pasukan memahami dan melaksanakan amalan terbaik yang berkaitan dengan bidang kepakarannya,</li> <li>v. membantu dalam pembangunan <i>user story</i> dan <i>acceptance criteria</i>,</li> <li>vi. menyertai <i>sprint planning meeting</i> untuk membantu mengutamakan dan menganggarkan tugasan <i>product backlog</i>, dan</li> <li>vii. menyediakan maklum balas mengenai tugasan yang selesai dan mencadangkan penambahbaikan untuk <i>sprint</i> berikutnya.</li> </ul>	Mahir dalam proses bisnes.

Peranan	Tugasan	Kemahiran
<b>Fasilitator DevOps</b>	<p>i. <i>Scrum Master</i>:</p> <ul style="list-style-type: none"> <li>a. Mengautomasikan proses pembangunan Menjadi pemudah cara kepada permasalahan yang dihadapi oleh ahli pasukan yang memberi kesan kepada kemajuan aktiviti scrum semasa pembangunan sistem aplikasi,</li> <li>b. memudahkan serta menggalakkan komunikasi dan kerjasama dalam pasukan juga dengan pihak berkepentingan luar,</li> <li>c. membantu menambah baik peranan dan tugas ahli pasukan dalam melaksanakan aktiviti pembangunan, dan</li> <li>d. membantu menyediakan pelaporan kepada pengurus produk.</li> </ul> <p>ii. <i>Jurutera DevOps</i>:</p> <ul style="list-style-type: none"> <li>a. Menyelia integrasi, pembangunan, ujian dan penggunaan <i>tools</i> untuk menyokong pelepasan berterusan,</li> <li>b. merancang dan melaksanakan proses pelepasan dengan efektif dan bertanggungjawab bagi setiap pelepasan produk, dan</li> <li>c. memantau penggunaan <i>tools</i> DevOps serta pelepasan produk secara menyeluruh.</li> <li>d. Mengautomasikan proses pembangunan dan operasi serta berkongsi maklumat dengan ahli pasukan, dan</li> <li>e. memastikan proses konfigurasi dilakukan secara teratur dan selamat.</li> </ul>	<ul style="list-style-type: none"> <li>i. Berpengetahuan dan mahir dalam penggunaan <i>tools</i> pelepasan,</li> <li>ii. mampu mengkoordinasi aktiviti-aktiviti pelepasan dengan pihak berkepentingan, dan</li> <li>iii. mempunyai kemahiran sebagai <i>scrum master</i> dan memiliki sijil pengiktirafan <i>scrum master</i>.</li> </ul>

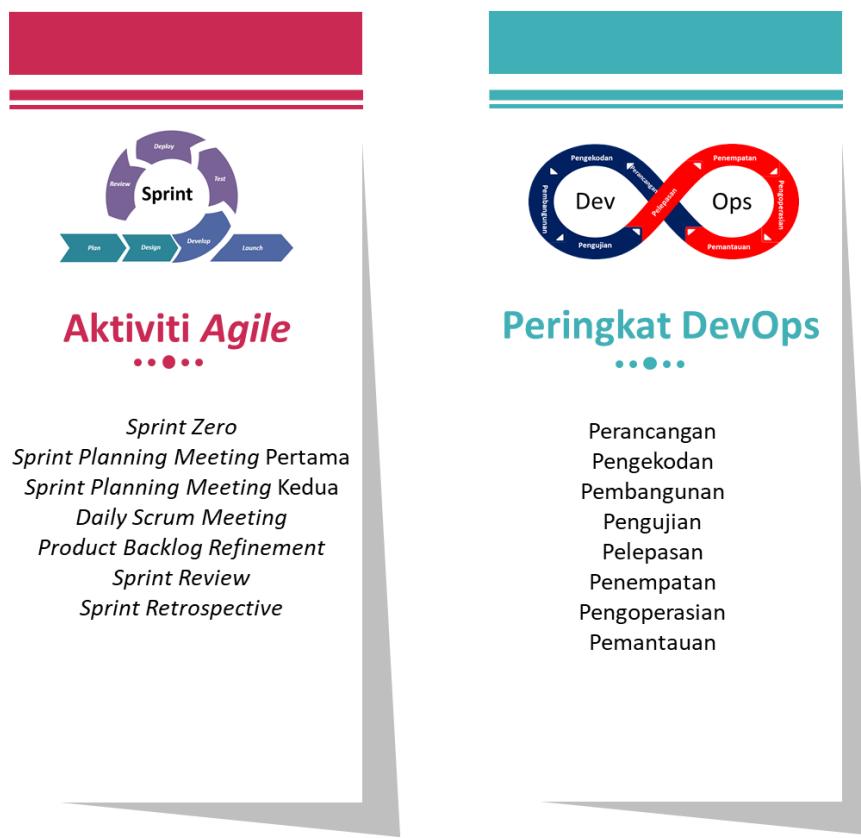
Peranan	Tugasan	Kemahiran
<b>Pasukan Pembangunan dan Operasi</b>	<p><b>Pembangun</b></p> <ul style="list-style-type: none"> <li>i. Membangunkan kod serta melaksanakan pengujian <i>back-end</i> dan <i>front-end</i>,</li> <li>ii. memastikan kualiti produk dikekalkan dari peringkat awal</li> <li>iii. memberi perkhidmatan sokongan produk.</li> <li>iv. Kualiti: <ul style="list-style-type: none"> <li>a. Menentukan prosedur ujian dan bekerjasama rapat dengan ahli pasukan untuk mengautomasikan ujian yang diperlukan,</li> <li>b. mereka bentuk dan melaksanakan ujian yang memerlukan pengujian secara manual,</li> <li>c. mengesan ralat atau pepijat dan membetulkan produk seawal mungkin, dan</li> <li>d. meningkatkan kualiti produk yang dibangunkan.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>i. Mahir dalam bahasa pengaturcaraan yang digunakan,</li> <li>ii. berpengetahuan mengenai <i>tools</i> DevOps dan <i>pipeline CI/CD</i>, dan</li> <li>iii. berpengalaman dalam proses pembangunan produk.</li> </ul>
	<p><b>Operasi</b></p> <ul style="list-style-type: none"> <li>i. Memastikan penggunaan, penyelenggaraan dan pemantauan <i>pipeline CI/CD</i> sentiasa tersedia.</li> <li>ii. Pelepasan: <ul style="list-style-type: none"> <li>a. Menyelia integrasi, pembangunan, ujian dan penggunaan <i>tools</i> untuk menyokong pelepasan berterusan,</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>i. Berpengetahuan mengenai <i>tools</i> DevOps dan <i>pipeline CI/CD</i>,</li> <li>ii. berpengetahuan dan mahir dalam penggunaan <i>tools</i> pelepasan,</li> <li>iii. mampu mengkoordinasi aktiviti-aktiviti pelepasan dengan pihak berkepentingan,</li> </ul>

Peranan	Tugasan	Kemahiran
	<p>b. merancang dan melaksanakan proses pelepasan dengan efektif dan bertanggungjawab bagi setiap pelepasan produk, dan</p> <p>c. memantau penggunaan <i>tools</i> DevOps serta pelepasan produk secara menyeluruh</p> <p>iii. Keselamatan:</p> <ul style="list-style-type: none"> <li>a. Menyelenggara alat dan teknik pengesanan keselamatan,</li> <li>b. memastikan <i>pipeline</i> CI/CD di integrasi dan diurus secara selamat dalam semua persekitaran,</li> <li>c. memantau insiden keselamatan dan bekerjasama dengan pasukan untuk memastikan proses pembangunan dan operasi berjalan lancar, dan</li> <li>d. menyediakan laporan terperinci insiden kepada pihak berkepentingan.</li> </ul>	<p>iv. berpengetahuan dan mahir dalam penggunaan <i>tools</i> konfigurasi,</p> <p>v. pantas mengenal pasti punca masalah dalam isu konfigurasi,</p> <p>vi. berpengetahuan luas mengenai konfigurasi <i>tools</i> DevOps,</p> <p>vii. mahir dalam pengurusan kualiti,</p> <p>viii. berpengetahuan dan mahir dalam pengujian sistem,</p> <p>ix. teliti dalam proses semakan kualiti,</p> <p>x. berpengetahuan dan mahir dalam penggunaan <i>tools</i> keselamatan DevOps, dan</p> <p>xi. arif dalam perkembangan terkini teknologi keselamatan.</p>

Semua peranan boleh diambil oleh pasukan pembangun/operasi kecuali pengurus produk. Ahli pasukan bertanggungjawab dalam pengagihan keseluruhan tugas pembangunan serta pengoperasian produk.

### 3.3. KOMPONEN 3 (K3): PENGADAPTAISAN METODOLOGI

Pelaksanaan DevOps dalam pembangunan sistem aplikasi sektor awam adalah menggunakan metodologi *agile* dengan mengadaptasikan amalan dan proses *agile scrum*. Komponen ini dibahagikan kepada dua bahagian iaitu bahagian yang melibatkan aktiviti *agile* dan bahagian peringkat-peringkat dalam DevOps. Rajah 3-15 memaparkan senarai tujuh aktiviti *agile* dan lapan peringkat DevOps yang akan diterangkan dalam komponen ini.



**Rajah 3-15: Aktiviti Agile dan Peringkat DevOps**

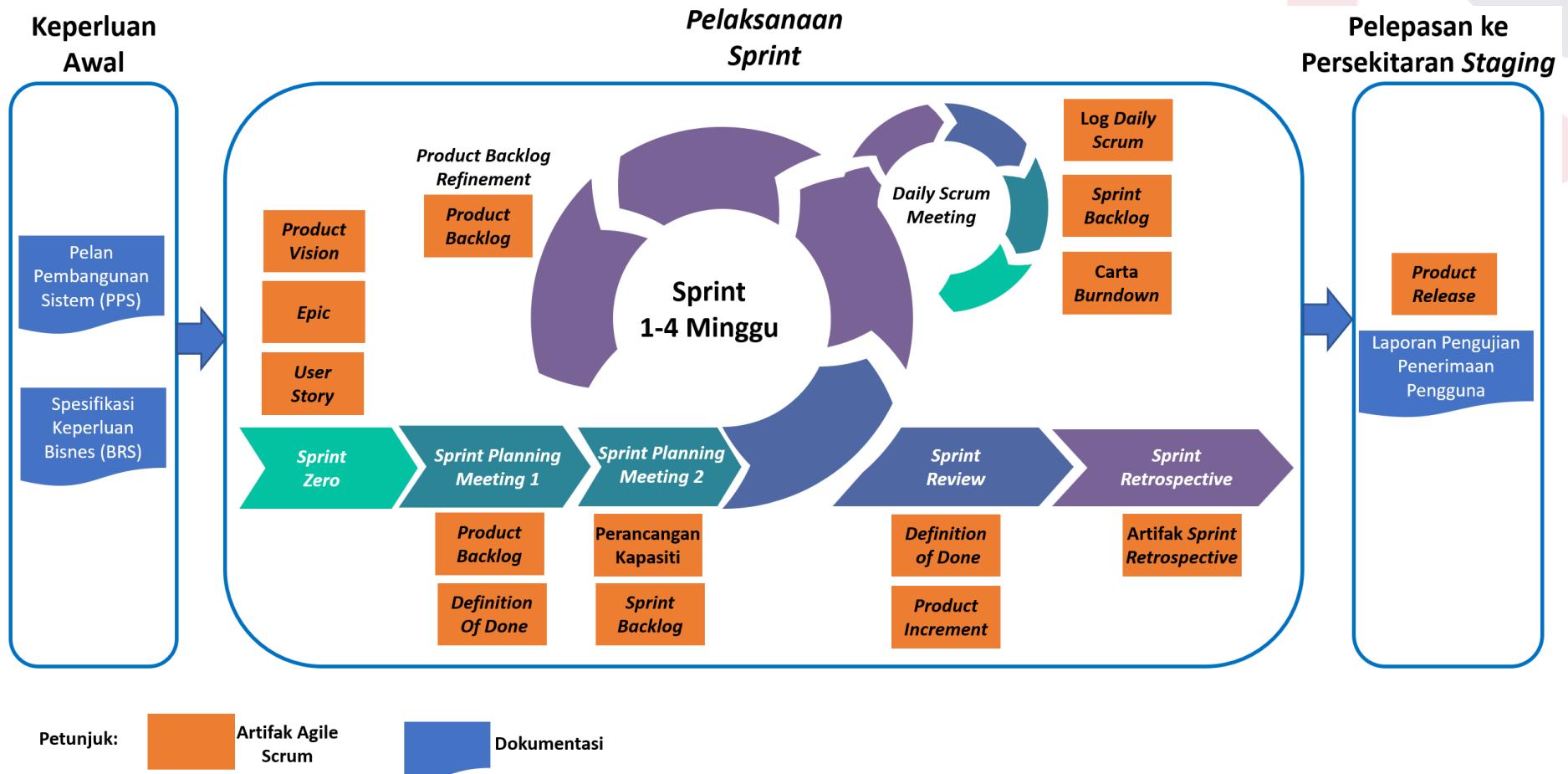
### 3.3.1. Aktiviti dalam Metodologi *Agile*

Metodologi *agile scrum* terdiri daripada aktiviti berikut iaitu *Sprint Zero*, *Sprint Planning Meeting*, *Daily Scrum Meeting*, *Product Backlog Refinement*, *Sprint Review* dan *Sprint Retrospective*. Rajah 3-16 memaparkan aktiviti *scrum* yang perlu dilaksanakan pada satu *sprint*. Setiap aktiviti ini melibatkan semua ahli pasukan iaitu pengurus produk, pembangun/operasi dan fasilitator DevOps.



Rajah 3-16: Aktiviti *Agile Scrum* dalam Satu *Sprint*

Rajah 3-17 memaparkan aktiviti *agile scrum* yang disokong oleh artifak yang dihasilkan daripada setiap aktiviti tersebut. Artifak merujuk kepada sumber informasi dan komunikasi antara pasukan DevOps. Artifak juga membantu pasukan dalam mengurus tugas bagi memastikan produk dapat dihasilkan mengikut keperluan pengguna dan perlu dikemas kini di sepanjang aktiviti *scrum*.



Rajah 3-17: Aktiviti dan Artifikat Agile Scrum

### 3.3.1.1. Aktiviti *Sprint Zero*

*Sprint zero* adalah aktiviti yang dilaksanakan sebagai persiapan sebelum *sprint* pertama bermula. Tempoh masa pelaksanaan aktiviti ini adalah dari satu hari hingga satu minggu mengikut kesesuaian pasukan. Tujuan *sprint zero* adalah untuk memastikan bahawa pasukan bersedia untuk mulakan pembangunan, mempunyai pemahaman yang jelas tentang tugas yang perlu dilakukan, jadual pelaksanaan dan peranan serta tanggungjawab untuk setiap ahli pasukan. Rajah 3-18 menunjukkan senarai dokumen rujukan sebelum aktiviti *sprint zero* bermula dan artifak hasil daripada aktiviti *sprint zero*.



**Rajah 3-18: Aktiviti *Sprint Zero***

**Aktiviti *sprint zero*** melibatkan perkara seperti berikut:

- Menyiapkan persekitaran pembangunan.
- Menentukan skop dan matlamat pembangunan produk.
- Mengenal pasti keperluan pembangunan.
- Menetapkan peranan dan tanggungjawab kepada ahli pasukan.

**Dokumen Rujukan** kepada aktiviti *sprint zero* adalah seperti berikut:

- Dokumen Spesifikasi Keperluan Bisnes (BRS) dan keperluan pengguna adalah daripada Hierarki Fungsi Bisnes.
- Dokumen Pelan Pembangunan Sistem (PPS) digunakan untuk penetapan visi pembangunan produk.

**Artifak hasil** daripada aktiviti *sprint zero* adalah seperti berikut:

- a. *Product vision.*
- b. *Epic.*
- c. *User story.*

### a. Artifak Product Vision

*Product vision* merupakan artifak yang dihasilkan oleh pengurus produk. *Product vision* memberi gambaran jelas tentang tujuan dan matlamat utama produk dibangunkan. Artifak ini bertujuan membantu pasukan DevOps mencapai persefahaman dan jelas dengan hala tuju produk.

#### i. Komponen Artifak *Product Vision*

Jadual 3-4 menerangkan komponen yang membentuk artifak *product vision*.

**Jadual 3-4: Komponen Artifak *Product Vision***

No.	Komponen	Penerangan	Dokumen Rujukan
1.	Visi	Menerangkan hala tuju produk.	BRS
2.	Kumpulan Sasaran	Menerangkan sasaran pengguna yang akan menggunakan produk.	PPS
3.	Justifikasi	Memberi justifikasi kepada permasalahan yang dihadapi oleh pengguna sebelum produk dibangunkan.	PPS
4.	Produk	Menerangkan produk yang akan dibangunkan berserta modul-modulnya.	BRS
5.	Tujuan	Menerangkan berkenaan matlamat pembangunan produk dan manfaat produk kepada agensi.	BRS

Templat artifak *product vision* boleh dirujuk dalam **Lampiran 1: Templat Artifak Product Vision**.

## ii. Contoh Pengisian Artifak *Product Vision*

Rajah 3-19 memaparkan contoh pengisian artifak *product vision*.

### **PRODUCT VISION BOARD UNTUK SISTEM TEMPAHAN BILIK MESYUARAT**

VISI			
KUMPULAN SASARAN	JUSTIFIKASI	PRODUK	TUJUAN
Memberi penerangan kepada sasaran pengguna yang akan menggunakan produk yang akan dibangunkan. (**PPS)	Memberi justifikasi kepada permasalahan yang dihadapi oleh pengguna buat masa sekarang. (**PPS)	Penerangan kepada produk yang akan dibangunkan beserta modul-modulnya. (*BRS)	Penerangan berkenaan matlamat pembangunan produk. (*BRS)

**Rajah 3-19: Contoh Artifak *Product Vision***

### b. Artifak *Epic* dan *User Story*

*Epic* merupakan fungsi utama produk atau elemen besar keperluan pengguna yang perlu dibangunkan bagi mencapai matlamat keseluruhan. *Epic* seterusnya dipecahkan kepada beberapa keperluan pengguna yang lebih kecil dikenali sebagai *user story*. *Epic* dan *user story* yang dikumpul akan digunakan pada aktiviti seterusnya bagi tujuan perancangan dan pelaksanaan pembangunan produk. Oleh itu, adalah digalakkan agar kesemua *user story* disenaraikan sebelum bermulanya *sprint*.

Berikut merupakan format penulisan *user story*:

**<peranan> <keperluan> <impak>**

**Contoh:**

**Peranan – Pengguna (Warga JDN)**

**Keperluan – BOLEH** mendaftar profil pengguna baharu

**Impak – SUPAYA** boleh mendaftar masuk sistem tempahan bilik mesyuarat

Sebagai pengguna (Warga JDN), saya **BOLEH** mendaftar profil pengguna baharu, **SUPAYA** boleh mendaftar masuk sistem tempahan bilik mesyuarat

### i. Komponen Artifak *User Story*

Jadual 3-5 menerangkan komponen artifak *user story*.

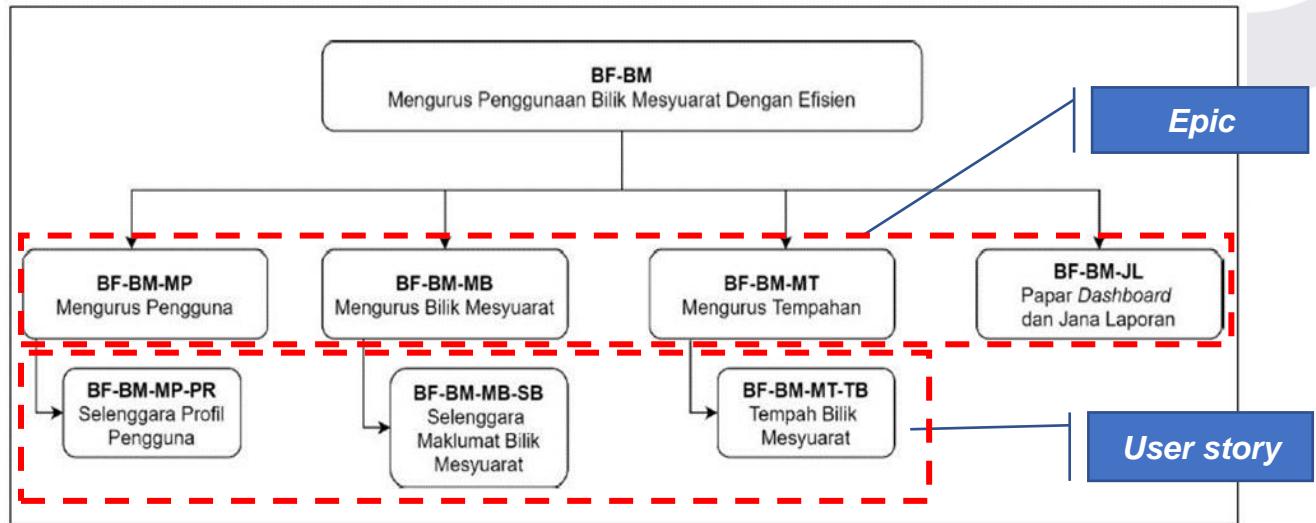
**Jadual 3-5: Komponen Artifak *User Story***

No.	Komponen	Penerangan
1.	<i>User story</i> ID	Merujuk kepada perwakilan ID unik (naming convention) untuk setiap <i>user story</i> .
2.	Peranan	Merujuk kepada jenis-jenis pengguna yang akan menggunakan produk tersebut.
3.	Keperluan	Keperluan produk yang ingin dibangunkan.
4.	Impak	Kesan ketara atau manfaat yang dapat digunakan hasil daripada pembangunan ciri-ciri tersebut.
5.	<i>Acceptance criteria</i>	Ciri-ciri keperluan fungsian yang perlu dipenuhi untuk melengkapkan <i>user story</i> .

Templat artifak *user story* boleh dirujuk dalam **Lampiran 2: Templat Artifak *User Story***.

### ii. Contoh Pengisian Artifak *Epic* dan *User Story*

Hierarki Fungsi Bisnes yang terdapat pada dokumen BRS Sistem Tempahan Bilik Mesyuarat boleh dijadikan sebagai panduan untuk menterjemahkan *epic* dan *user story*. Rajah 3-20 memaparkan *epic* dan *user story* yang diekstrak daripada Hierarki Fungsi Bisnes Sistem Tempahan Bilik Mesyuarat.



**Rajah 3-20: Hierarki Fungsi Bisnes Sistem Tempahan Bilik Mesyuarat**

Epic yang telah dikenal pasti adalah **Mengurus Pengguna**, **Mengurus Bilik Mesyuarat**, **Mengurus Tempahan** dan **Papar Dashboard** serta **Jana Laporan**. Pengurus produk menterjemahkan keperluan pengguna ke dalam format *user story* untuk memberi kefahaman kepada pasukan scrum. Jadual 3-6 merupakan contoh pengisian artifak *user story*.

**Jadual 3-6: Contoh Artifak User Story**

User Story Id	Peranan	Keperluan	Impak	Acceptance Criteria
<b>Modul Mengurus Pengguna</b>				
BF-BM-EP01-US01	Pengguna (warga JDN)	<b>BOLEH</b> mendaftar profil pengguna baharu	<b>SUPAYA</b> boleh mendaftar masuk sistem tempahan bilik mesyuarat	Semasa melengkapkan pendaftaran perlu pastikan: i. Format nombor kad pengenalan 12 digit dimasukan dan tidak termasuk simbol '-'.

User Story Id	Peranan	Keperluan	Impak	Acceptance Criteria
				<ul style="list-style-type: none"> <li>ii. Hanya alamat e-mel rasmi agensi sahaja yang boleh diterima.</li> <li>iii. Semak dan pastikan tiada pertindihan untuk kad pengenalan dan e-emel.</li> </ul>
<b>BF-BM-EP01-US02</b>	Pengguna (warga JDN)	<b>BOLEH</b> mengemas kini maklumat profil pengguna yang telah didaftarkan	<b>SUPAYA</b> akaun pengguna dapat disahkan dan selamat.	<ul style="list-style-type: none"> <li>i. Saiz gambar dicadangkan tidak boleh melebih dari 800 x 800 piksel.</li> <li>ii. Format gambar yang dicadangkan adalah .jpeg dan .png sahaja.</li> </ul>

### 3.3.1.2. Aktiviti *Sprint Planning Meeting* Pertama

*Sprint planning meeting* pertama merupakan aktiviti setelah bermulanya *sprint*. Aktiviti ini diadakan pada hari pertama pada setiap *sprint*. Tujuan *sprint planning meeting* pertama adalah seperti berikut:

- a. Membincangkan dan menganalisis artifak yang telah dihasilkan daripada *sprint zero*.
- b. Menetapkan keutamaan dan saiz *user story* menggunakan kaedah MoSCoW dan turutan Fibonacci.
- c. Merancang *sprint* dan pelepasan produk.
- d. Membincangkan DoD.

Rajah 3-21 menunjukkan senarai artifak rujukan sebelum aktiviti *sprint planning meeting* pertama bermula dan artifak hasil daripada aktiviti *sprint planning meeting* pertama.



**Rajah 3-21: Sprint Planning Meeting Pertama**

**Artifak rujukan** kepada aktiviti *sprint planning meeting* pertama adalah *product vision*, *epic* dan *user story* yang telah dihasilkan pada *Sprint Zero*.

**Artifak hasil** daripada aktiviti *sprint planning meeting* pertama adalah *product backlog* dan DoD.

**Penglibatan ahli pasukan** adalah seperti berikut:

- a. Peranan pengurus produk adalah menjelaskan hala tuju produk dan memberi gambaran yang lebih jelas terhadap keperluan pengguna melalui lukisan, model atau mockup. Selain daripada itu, pengurus produk boleh menambahbaik, mengeluarkan dan memberi keutamaan semula *product backlog* di sepanjang pembangunan.
- b. SME Bisnes digalakkan untuk menyertai aktiviti ini bagi membantu pasukan menganggar dan mengutamakan *user story* yang telah disenaraikan.
- c. Fasilitator DevOps bertindak sebagai *scrum master* bagi mengurus dan melancarkan sepanjang aktiviti ini berlangsung.
- d. Pasukan pembangun memainkan peranan dalam menentukan *story point* bagi setiap *user story* yang disenaraikan. *Story point* merupakan skor yang menggambarkan saiz dan kerumitan *user story* tersebut.

### a. Artifak Product Backlog

*Product backlog* merupakan senarai keperluan yang terdiri daripada ciri-ciri baharu, penambahbaikan, eksperimentasi, keperluan prestasi, sekuriti atau ralat. Artifak ini boleh ditambah baik, dikeluarkan dan diberi keutamaan semula di sepanjang pembangunan sekiranya perlu.

#### i. Komponen Artifak *Product Backlog*

Jadual 3-7 menerangkan komponen artifak *product backlog*.

**Jadual 3-7: Komponen Artifak *Product Backlog***

No.	Komponen	Penerangan
1.	<b>User story ID</b>	Pengenalan unik ID untuk setiap <i>user story</i> .
2.	<b>Keutamaan</b>	Penetapan keutamaan untuk setiap <i>user story</i> berdasarkan kepentingan dan nilai kepada pengguna. Penetapan keutamaan dilakukan oleh pengurus produk dan pasukan pembangun dengan menggunakan kaedah MoSCoW.
3.	<b>Turutan Keutamaan</b>	Turutan keutamaan yang ditetapkan oleh pengurus produk bergantung kepada nilai keutamaan dan <i>story point</i> .
4.	<b>User story</b>	Penerangan ringkas tentang ciri atau fungsi dari perspektif pengguna yang merangkumi peranan, fungsi dan impak.
5.	<b>Story point</b>	Anggaran bagi masa, usaha, atau kompleksiti yang diperlukan untuk menyelesaikan <i>user story</i> . Kaedah turutan Fibonacci digunakan oleh pengurus produk dan pasukan pembangun pasukan untuk menetapkan saiz <i>user story</i> .
6.	<b>Status</b>	Status semasa <i>user story</i> tersebut sama ada Baharu, Dalam Tindakan atau Selesai.

Templat artifik *product backlog* boleh dirujuk dalam **Lampiran 3: Templat Artifak Product Backlog**.

**Keutamaan setiap user story** boleh ditentukan dengan menggunakan kaedah MoSCow untuk mengurus keperluan pengguna.

Dalam pembangunan sistem aplikasi, kaedah MoSCoW merujuk kepada pendekatan menetapkan keutamaan keperluan pengguna berdasarkan pembahagian keperluan kepada empat kategori iaitu *must have, should have, could have* dan *will not have*.

Jadual 3-8 memberi penerangan setiap kategori berdasarkan kaedah MosCow.

**Jadual 3-8: Kategori Penetapan Keutamaan User story**

No.	Kategori	Penerangan
1.	<b>Perlu Ada (Must Have)</b>	Keperluan yang wajib ada dan jika tiada produk tidak berfungsi
2.	<b>Patut Ada (Should Have)</b>	Keperluan yang kurang penting namun akan memberi impak yang tinggi
3.	<b>Mungkin Ada (Could Have)</b>	Keperluan yang kurang penting dan akan memberi impak yang rendah
4.	<b>Tidak Sepatuinya Ada (Will Not Have)</b>	Keperluan yang tidak menjadi keutamaan dalam tempoh pembangunan

**Penentuan story point** boleh ditentukan menggunakan kaedah *playing poker*. *Playing poker* adalah teknik yang digunakan oleh pasukan untuk menganggarkan nilai kompleksiti bagi setiap *user story*.

Turutan Fibonacci digunakan sebagai nilai kompleksiti untuk menganggarkan usaha yang diperlukan dalam menyelesaikan setiap *user story*. Turutan Fibonacci adalah siri nombor di mana setiap nombor adalah jumlah dua nombor sebelumnya, bermula dengan 0 dan 1. Turutan tersebut adalah 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 dan seterusnya dengan hasilnya membawa kepada kenaikan konsisten 60%. Adalah digalakkan skor tertinggi *story point* dihadkan sehingga 89 kerana nilai yang lebih besar akan menyebabkan anggaran saiz dan kompleksiti *user story* menjadi kurang tepat.

Proses ini membantu pasukan pembangun menganggar tempoh masa pembangunan dengan lebih tepat, mencapai persetujuan dan merancang tugas dengan lebih strategik. Jika terdapat perbezaan pendapat antara ahli pasukan berkenaan *story point*, pengurus produk selaku ketua pasukan perlu meminta setiap ahli pasukan untuk mengusulkan pendapat supaya persefahaman dapat dicapai dalam penentuan *story point*. Rajah 3-22 menerangkan langkah-langkah kaedah *playing poker*.



**Rajah 3-22: Kaedah *Playing Poker***

## ii. Contoh Pengisian Artifak *Product Backlog*

Jadual 3-9 memaparkan contoh pengisian artifak *product backlog*.

**Jadual 3-9: Contoh Artifak *Product Backlog***

PRODUCT BACKLOG ID	KEUTAMAAN	TURUTAN KEUTAMAA N	USER STORY			STORY POINT	SPRINT NO	STATUS
<b>Epic 1 : Pengurusan Pengguna</b>								
BF-BM-EP01-PB01	Mesti Ada	1	Pengguna (warga MAMPU)	<b>BOLEH</b> mendaftar profil pengguna baru	<b>SUPAYA</b> boleh mendaftar masuk sistem tempahan bilik mesyuarat	3	Sprint 1	Dalam Tindakan
BF-BM-EP01-PB02	Mesti Ada	2	Pengguna (warga MAMPU)	<b>BOLEH</b> mengemaskini maklumat profil pengguna yang telah didaftarkan	<b>SUPAYA</b> akaun dapat disahkan dan selamat.	5	Sprint 1	Baharu
<b>Epic 2 : Pengurusan Bilik Mesyuarat</b>								
BF-BM-EP02-PB01	Mesti Ada	4	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menyelenggara maklumat bilik mesyuarat dengan mewujudkan rekod bilik yang baharu atau mengemaskini rekod sedia ada.	<b>SUPAYA</b> mendapatkan maklumat bilik mesyuarat yang tepat.	13		
BF-BM-EP02-PB02	Mesti Ada	3	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menyemak maklum balas penggunaan bilik mesyuarat yang diterima daripada pengguna (warga Agensi)	<b>SUPAYA</b> mendapatkan sumber rujukan untuk melaporkan kerosakan dan perlukan pembaikan yang berkaitan.	8	Sprint 1	

### b. Artifak Definition of Done (DoD)

DoD merupakan senarai kriteria yang mesti dipenuhi untuk item *product backlog* dianggap lengkap dan produk sedia untuk pelepasan pada akhir *sprint*. DoD harus ditakrifkan dengan jelas, difahami dan dipersetujui oleh semua ahli pasukan *scrum* untuk memastikan produk yang dibangunkan adalah berkualiti tinggi dan memenuhi jangkaan pengurus produk.

#### i. Komponen Artifak *Definition of Done*

Artifak DoD terdiri daripada komponen seperti keperluan fungsian, keperluan bukan fungsian, penempatan dan dokumentasi. Jadual 3-10 menerangkan setiap komponen yang membentuk artifak DoD.

**Jadual 3-10: Komponen Artifak *Definition of Done***

No.	Komponen	Penerangan
1.	Keperluan Fungsian	Senarai semak yang berkaitan pengujian fungsian.
2.	Keperluan Bukan Fungsian	Senarai semak yang berkaitan pengujian bukan fungsian.
3.	Penempatan	Persekutaran penempatan (deployment) produk sama ada pembangunan, <i>staging</i> atau produksi.
4.	Dokumentasi	Jenis dokumentasi dan dokumen serahan yang perlu disediakan semasa <i>sprint</i> .
5.	Status	Senarai semak telah diselesaikan atau tidak.
6.	Catatan	Penerangan tambahan mengenai komponen senarai semak seperti pelaksanaan pada <i>sprint</i> ke berapa.

Templat artifak DoD boleh dirujuk dalam **Lampiran 4: Templat Artifak *Definition of Done***.

## ii. Contoh Pengisian Artifak *Definition of Done*

Jadual 3-11 merupakan DoD yang telah dikenal pasti dan dipersetujui oleh pasukan *scrum*.

**Jadual 3-11: Contoh Senarai *Definition of Done***

Komponen	Senarai Semak	Status	Catatan
<b>Pengujian Keperluan Fungsian</b>	1. Kod sumber selesai dibangunkan		
	2. Kod sumber disemak		
	3. Kod sumber digabungkan (merged)		
	4. Melepas pengujian unit		
	5. Melepas pengujian integrasi		
	6. Melepas pengujian sistem dan memenuhi <i>acceptance criteria</i> bagi setiap <i>user story</i>		
	7. Melepas pengujian penerimaan pengguna		
<b>Pengujian Keperluan Bukan Fungsian</b>	1. Melepas pengujian kualiti kod		
	2. Melepas pengujian SAST		
	3. Melepas pengujian prestasi		
<b>Penempatan</b>	1. Penempatan ke persekitaran <i>staging</i> pada <i>sprint</i> yang melibatkan <i>product release</i> sahaja		
<b>Dokumentasi</b>	1. Artifak <i>product vision</i>		
	2. Dokumen Laporan UAT		
	3. Artifak <i>sprint backlog</i>		
	4. Artifak Carta <i>burndown</i>		

Komponen	Senarai Semak	Status	Catatan
	5. Artifak Perancangan Kapasiti		
	6. Artifak log <i>daily scrum</i>		
	7. Artifak <i>sprint retrospective</i>		

### 3.3.1.3. Aktiviti *Sprint Planning Meeting* Kedua

*Sprint planning meeting* kedua merupakan aktiviti seterusnya selepas *sprint planning meeting* pertama. Aktiviti ini dilaksanakan pada hari yang sama dengan *sprint planning meeting* pertama. Tempoh masa pelaksanaan aktiviti ini adalah dari dua hingga empat jam mengikut kesesuaian pasukan.

Tujuan aktiviti *sprint planning meeting* kedua adalah:

- i. Melakukan perancangan kapasiti.
- ii. Menyediakan artifak *sprint backlog*.
- iii. Aktiviti *sprint planning meeting* kedua meliputi perkara seperti berikut:
- iv. Pasukan pembangun akan berkongsi ketersediaan mereka sepanjang *sprint*.
- v. Pasukan pembangun akan menyemak *user story* yang telah dikenal pasti semasa *sprint planning meeting* pertama dan memutuskan cara pelaksanaannya.
- vi. Membahagikan *user story* tersebut kepada tugas kecil dan membuat anggaran kompleksiti, masa dan usaha yang diperlukan untuk menyelesaikan setiap *user story*.
- vii. Mengenal pasti sebarang pergantungan antara tugas.
- viii. Menentukan susunan tugas berdasarkan keutamaannya.

Rajah 3-23 menunjukkan senarai artifak rujukan sebelum aktiviti *sprint planning meeting* kedua bermula dan artifak hasil daripada aktiviti *sprint planning meeting* kedua.



**Rajah 3-23: Sprint Planning Meeting Kedua**

**Artifikat Rujukan** kepada aktiviti *Sprint Planning Meeting* Kedua adalah *product backlog* dan DoD yang telah dihasilkan pada *Sprint Planning Meeting* Pertama.

**Artifikat Hasil** daripada aktiviti *Sprint Planning Meeting* Kedua adalah perancangan kapasiti dan artifikat *sprint backlog*.

**Penglibatan ahli pasukan** adalah seperti berikut:

- i. Pengurus produk bertanggungjawab untuk memastikan pasukan pembangun jelas tentang item *product backlog* yang perlu dilengkapkan dan merekodkan perkara-perkara yang dibincangkan ke dalam *sprint backlog*.
- ii. Fasilitator DevOps bertindak sebagai *scrum master* bagi mengurus dan melancarkan sepanjang aktiviti ini berlangsung.
- iii. Pasukan pembangun berperanan untuk mengenal pasti tugas-tugasan kecil yang perlu dilaksanakan serta ketersediaan mereka semasa *sprint*.

### a. Artifak Perancangan Kapasiti

Kapasiti pasukan dikira mengikut ketersediaan ahli pasukan dalam setiap *sprint*. Aktiviti ini melibatkan menganalisis kapasiti pasukan, bilangan ahli pasukan, ketersediaan dan kemahiran ahli pasukan. Maklumat ini digunakan untuk merancang jumlah tugas yang boleh dilaksanakan semasa *sprint*. Namun, perancangan kapasiti ini perlu mengambil kira hari ahli pasukan itu bercuti, jumlah hari menghadiri latihan, komitmen pada tugas lain, hal peribadi dan lain-lain.

#### i. Komponen Artifak Perancangan Kapasiti

Jadual 3-12 menerangkan artifak perancangan kapasiti.

**Jadual 3-12: Komponen Artifak Perancangan Kapasiti**

No.	Komponen	Penerangan
1.	Nama	Nama ahli pasukan yang telah ditugaskan untuk menyelesaikan keperluan pengguna.
2.	Jumlah Hari Bekerja <i>sprint</i> semasa (Hari)	Bilangan hari bekerja setiap ahli pasukan untuk <i>sprint</i> semasa. Perlu mengambil kira hari pegawai itu bercuti, menghadiri latihan, hal peribadi dan cuti umum yang dirancang.
3.	Jumlah Masa Bekerja Sehari (Jam)	Bilangan jam bekerja yang boleh diperuntukkan oleh ahli pasukan tersebut dalam sehari untuk menyiapkan tugasan.
4.	Jumlah Masa yang Diperuntukkan (Jam)	Jumlah jam bekerja keseluruhan yang ada untuk setiap ahli pasukan.  <b>Jumlah Masa yang Diperuntukkan = Jumlah Hari Bekerja setiap <i>sprint</i> x Jumlah Waktu Bekerja Sehari</b>
5.	Jumlah Masa Keseluruhan (Jam)	Merujuk kepada penambahan keseluruhan jumlah masa yang diperuntukkan dalam <i>sprint</i> semasa.

No.	Komponen	Penerangan
6.	5-10% <i>interrupt buffer</i> (Jam)	<i>Interrupt buffer</i> 5% hingga 10% adalah masa kerja untuk isu luar jangka yang telah diambil kira daripada Jumlah Masa Keseluruhan yang diperuntukkan bagi setiap <i>sprint</i> dan perlu ditolak semasa kiraan Jumlah Masa Bersih.
7.	5% -10% <i>Product Backlog Refinement</i> (Jam)	5% hingga 10% masa yang telah diambil kira bagi memperhalusi <i>product backlog</i> .
8.	Jumlah Masa Bersih (Jam)	Jumlah Masa Keseluruhan ditolak dengan masa <i>interrupt buffer</i> dan <i>product backlog refinement</i> .

Templat artifak perancangan kapasiti boleh dirujuk dalam **Lampiran 5: Templat Artifikat Perancangan Kapasiti**.

## ii. Contoh Pengisian Artifak Perancangan Kapasiti

Contoh artifak perancangan kapasiti yang telah diisi berdasarkan lima orang ahli pasukan pembangun bagi 10 hari bekerja adalah seperti Jadual 3-13.

**Jadual 3-13: Jadual Pengiraan Perancangan Kapasiti *Sprint 1***

Perancangan Kapasiti <i>Sprint 1</i>			
Tempoh <i>Sprint</i> , 2 minggu = 10 hari bekerja			
Nama	Jumlah Hari Bekerja Semasa <i>Sprint</i> (Hari) (a)	Jumlah Masa Bekerja Sehari (Jam) (b)	Jumlah Masa yang Diperuntukkan (Jam) (a*b)
Ali	8	5	40
Azhim	9	5	45
Ahmad	7	5	35
Raziman	6	2	12
Fauzi	4	4.5	18
Jumlah Masa Keseluruhan (Jam) (c)			150
(-)	5-10% <i>Interrupt Buffer</i> (Jam) (d)		15
(-)	5-10% <i>PB Refinement</i> (Jam) (e)		15
Jumlah Masa Bersih (Jam) = (c - (d + e))			120

Berdasarkan perancangan kapasiti yang dihasilkan, jumlah masa bersih yang diperuntukkan oleh ahli pasukan pembangun bagi *sprint 1* adalah sebanyak 120 jam.

## b. Artifak Sprint Backlog

*Sprint backlog* adalah artifak yang mengandungi senarai tugas yang perlu diselesaikan oleh pasukan pembangun semasa *sprint*. *Sprint backlog* bersifat dinamik yang boleh berkembang di sepanjang *sprint* apabila tugas ditambah, dialih keluar atau diubah suai.

*Sprint backlog* berfungsi sebagai panduan yang boleh digunakan pasukan pembangun untuk membantu mereka kekal fokus kepada matlamat *sprint*. Artifak ini juga membantu pasukan pembangun mengurus tugas yang sedang dilaksana dan memantau kemajuan tugas mereka sepanjang *sprint*.

### i. Komponen Artifak *Sprint Backlog*

Jadual 3-14 menerangkan artifak *sprint backlog*.

**Jadual 3-14: Komponen Artifak *Sprint Backlog***

No.	Komponen	Penerangan
1.	User Story ID	Mewakili ID unik untuk setiap <i>product backlog</i> .
2.	Product Backlog Item	Senarai <i>user story</i> yang telah dikenal pasti dalam setiap <i>sprint</i> .
3.	ID Tugasan	Mewakili ID unik untuk setiap tugas <i>sprint</i> .
4.	Tugasan	Tugas bagi setiap <i>sprint backlog</i> yang telah dikenal pasti oleh pasukan <i>scrum</i> .
5.	Ahli Pasukan	Nama ahli pasukan yang ditugaskan untuk menyelesaikan tugas.
6.	Status	Status semasa tugas tersebut sama ada Baharu, Dalam Tindakan atau Selesai.
7.	Story Point	Anggaran saiz yang ditetapkan oleh pasukan <i>scrum</i> menerusi kaedah turutan fibonacci.
8.	Anggaran Masa (Jam)	Masa yang diperlukan untuk menyelesaikan tugas tersebut.

No.	Komponen	Penerangan
9.	Masa Sebenar (jam)	Jumlah jam yang telah digunakan untuk menyelesaikan tugas setiap hari.
10.	<i>Remaining Effort</i>	Jumlah masa yang berbaki untuk menyelesaikan tugas setiap hari.  <b><i>Remaining effort = Jumlah Anggaran Masa (Hari Sebelumnya) – Masa Sebenar</i></b>
11.	<i>Ideal Trend</i>	Merujuk kepada trend penurunan ideal untuk mencapai zero effort remaining pada penghujung sprint.  <b><i>Ideal Trend = Jumlah Anggaran Masa – (Jumlah Anggaran Masa / Jumlah Hari Sprint * Hari)</i></b>

Templat artifikat *sprint backlog* boleh dirujuk dalam **Lampiran 6: Templat Artifak Sprint Backlog**.

## ii. Contoh Pengisian Artifikat *Sprint Backlog*

Jadual 3-15 memaparkan contoh pengisian artifikat *sprint backlog*. Terdapat dua *epic* dan tiga *product backlog* yang disenaraikan dalam *sprint* 1. Setiap *product backlog* dipecahkan kepada beberapa tugas. Setiap pembangun bertanggungjawab untuk menyelesaikan tugas tersebut berdasarkan anggaran masa yang ditetapkan. Hari 1 melibatkan aktiviti *sprint planning meeting* pertama dan kedua manakala Hari 10 melibatkan aktiviti *sprint review* dan *sprint retrospective*. Pembangun akan mula membangunkan produk bermula dari Hari 2 sehingga Hari 9.

### Jadual 3-15: Contoh Sprint Backlog Hari 1 bagi Sprint 1

PRODUCT BACKLOG ID	PRODUCT BACKLOG ITEM	ID TUGASAN	TUGASAN	AHLI PASUKAN	STATUS	STORY POINT	ANGGARAN MASA (JAM)	HARI 1	HARI 2	HARI 3	HARI 4	HARI 5	HARI 6	HARI 7	HARI 8	HARI 9	HARI 10	
<b>Epic 1: Pengurusan Pengguna</b>																		
BF-BM-EP01-PB01	Pengguna (warga MAMPU)BOLEH mendaftar profil pengguna baru SUPAYA boleh mendaftar masuk sistem tempahan bilik mesyuarat	US-001-ST01	<i>new table format</i>	Azhim	Selesai	3	8											
		US-001-ST02	<i>new table UI</i>	Ahmad	Selesai		4											
		US-001-ST03	<i>implement new db format</i>	Fauzi	Selesai		6											
		US-001-ST04	<i>template setup</i>	Raziman	Selesai		2											
		US-001-ST05	<i>creation of auto approve</i>	Ali	Selesai		4											
BF-BM-EP01-PB02	Pengguna (warga MAMPU)BOLEH mengemaskini maklumat profil pengguna yang telah didaftarkan SUPAYA akaun dapat disahkan dan selamat.	US-002-ST01	<i>Install contact page.</i>	Ali	Selesai	5	5											
		US-002-ST02	<i>new table UI</i>	Raziman	Selesai		6											
		US-002-ST03	<i>implement new db format</i>	Ahmad	Selesai		8											
		US-002-ST04	<i>template setup</i>	Azhim	Selesai		8											
		US-002-ST05	<i>export configuration</i>	Ahmad	Selesai		7											
		US-002-ST06	<i>creation of auto approve</i>	Ali	Selesai		8											
		US-002-ST07	<i>audit trail</i>	Azhim	Selesai		5											
		US-002-ST08	<i>creation of new table</i>	Raziman	Selesai		5											
<b>Epic 2: Pengurusan Bilik Mesyuarat</b>																		
BF-BM-EP02-PB02	Pentadbir Bilik Mesyuarat BOLEH menyemak maklum balas penggunaan bilik mesyuarat yang diterima daripada pengguna (warga Agensi)SUPAYA mendapatkan sumber rujukan untuk melaporkan kerrosakan dan perlukan pembaikan yang berkaitan.	US-004-ST01	<i>Install contact page</i>	Azhim	Selesai	8	8											
		US-004-ST02	<i>new table UI</i>	Raziman	Selesai		6											
		US-004-ST03	<i>implement new db format</i>	Ali	Selesai		7											
		US-004-ST04	<i>template setup</i>	Ahmad	Selesai		7											
		US-004-ST05	<i>table implementation</i>	Ali	Selesai		6											
		US-004-ST06	<i>export configuration</i>	Fauzi	Selesai		5											
		US-004-ST07	<i>creation of auto approve</i>	Raziman	Selesai		6											
<b>JUMLAH STORY POINT</b>						16												
<b>MASA SEBENAR (Jam) (a)</b>						0												
<b>REMAINING EFFORT (Jam) (b)</b> (b=b hari sebelum – a hari semasa)						121												
<b>IDEAL TREND (jam) (c)</b> (c = c rancang – (c rancang/jumlah hari x hari semasa))						121												

### 3.3.1.4. Aktiviti *Daily Scrum Meeting*

Aktiviti *daily scrum meeting* diadakan selepas *sprint planning meeting* kedua.

Aktiviti ini merupakan aktiviti harian yang diadakan maksimum 15 minit setiap hari. Menerusi aktiviti ini, setiap ahli pasukan *scrum* berpeluang untuk berkongsi kemajuan masing-masing, mengkoordinasi kerja dan melaporkan perkara berikut:

- i. Tugasan yang telah diselesaikan hari sebelum.
- ii. Tugasan yang bakal diselesaikan hari ini.
- iii. Isu yang dihadapi dalam menyelesaikan tugas (sekiranya ada).

Rajah 3-24 memaparkan senarai artifak rujukan sebelum aktiviti *daily scrum meeting* bermula dan artifak hasil daripada aktiviti *daily scrum meeting*.



**Rajah 3-24: Daily Scrum Meeting**

**Artifak Rujukan** kepada aktiviti *daily scrum meeting* adalah *sprint backlog* dan carta *burndown* hari sebelumnya.

**Artifak Hasil** daripada aktiviti *daily scrum meeting* adalah log *daily scrum*, *sprint backlog* dan carta *burndown* yang dikemas kini setiap hari.

**Penglibatan ahli pasukan** adalah seperti berikut:

- i. Pengurus produk bertanggungjawab untuk merekod perkara-perkara yang dikemas kini pasukan pembangun.
- ii. Fasilitator DevOps bertindak sebagai *scrum master* bagi mengurus dan melancarkan sepanjang aktiviti ini berlangsung di samping membantu mengkoordinasikan isu yang dihadapi oleh pasukan pembangun. Sekiranya isu tidak dapat diselesaikan dan berlanjutan, mesyuarat lanjutan bersama SME Bisnes dan pihak berkepentingan boleh diadakan.
- iii. Pasukan pembangun berperanan untuk berkongsi status kemajuan kerja masing-masing.

#### a. Artifak Log *Daily Scrum*

Log *daily scrum* merekodkan kemajuan pembangunan dan isu yang menghalang perancangan untuk hari seterusnya. Log ini digunakan sebagai rujukan kepada pasukan pembangun untuk menjelaki kemajuan mereka dan mengenal pasti mana-mana tugas yang memerlukan perhatian.

##### i. Komponen Artifak Log *Daily Scrum*

Jadual 3-16 menerangkan komponen artifak log *daily scrum*.

**Jadual 3-16: Log *Daily Scrum***

No.	Komponen	Penerangan
1.	Hari	Mewakili hari dalam <i>sprint</i> .
2.	Ahli Pasukan	Nama ahli pasukan yang ditugaskan untuk menyelesaikan tugas.
3.	Kemas kini Kemajuan	<ol style="list-style-type: none"><li>i. Tugas yang telah diselesaikan hari sebelum.</li><li>ii. Tugas yang bakal diselesaikan hari ini.</li><li>iii. Isu yang dihadapi dalam menyelesaikan tugas (sekiranya ada).</li></ol>

Templat artifak log *daily scrum* boleh dirujuk dalam **Lampiran 6: Templat Artifak Sprint Backlog**.

### **ii. Contoh Pengisian Log Daily Scrum**

Jadual 3-17 menunjukkan contoh artifak log *daily scrum* yang telah diisi bagi seorang pembangun untuk tempoh lima hari bekerja.

**Jadual 3-17: Log Daily Scrum bagi Tempoh Lima Hari**

<b>LOG DAILY SCRUM</b>						
Ahli Pasukan	Kemajuan	Hari 1 Isnin	Hari 2 Selasa	Hari 3 Rabu	Hari 4 Khamis	Hari 5 Jumaat
Ali	Tugasan yang telah diselesaikan pada hari sebelum	<i>Sprint Planning Meeting 1 &amp; 2</i>	sambungan penyediaan fungsi <i>auto approve</i> (30% siap)	bercuti	bercuti	penyediaan fungsian <i>auto approve</i> (80% siap)
	Tugasan yang bakal diselesaikan pada hari ini	penyediaan fungsian <i>auto approve</i>	sambungan penyediaan fungsian <i>auto approve</i>	bercuti	sambungan penyediaan fungsian <i>auto approve</i> (50% siap)	sambungan penyediaan fungsian <i>auto approve</i>
	Isu yang dihadapi semasa menyelesaikan tugasan (sekiranya ada)	memerlukan maklumat lanjut berkenaan db <i>table auto approve</i>	tiada	bercuti	tiada	Tiada

### **iii. Pengemaskinian Artifak Sprint Backlog**

Artifak *sprint backlog* perlu dikemas kini bagi mencatatkan masa yang diperuntukkan oleh pasukan pembangun dalam menyelesaikan tugasan. Masa yang diperuntukkan bagi setiap tugas akan dicatatkan pada artifak ini. Contoh pengemaskinian artifak *sprint backlog* bagi tempoh 10 hari bekerja adalah seperti Jadual 3-18.

### Jadual 3-18: Contoh Artifak Sprint Backlog sehingga Hari 10

PRODUCT BACKLOG ID	PRODUCT BACKLOG ITEM	ID TUGASAN	TUGASAN	AHLI PASUKAN	STATUS	STORY POINT	ANGGARAN MASA (JAM)	HARI 1	HARI 2	HARI 3	HARI 4	HARI 5	HARI 6	HARI 7	HARI 8	HARI 9	HARI 10	
<b>Epic 1: Pengurusan Pengguna</b>																		
BF-BM-EP01-PB01	Pengguna (warga MAMPU) BOLEH mendaftar profil pengguna baru SUPAYA boleh mendaftar masuk sistem tempahan bilik mesyuarat	US-001-ST01	<i>new table format</i>	Azhim	Selesai	3	8		3	0	2	2	1	0	0	0	0	
		US-001-ST02	<i>new table UI</i>	Ahmad	Selesai		4		3	1	0	0	0	0	0	0	0	
		US-001-ST03	<i>implement new db format</i>	Fauzi	Selesai		6		2	0	0	4	0	0	0	0	0	
		US-001-ST04	<i>template setup</i>	Raziman	Selesai		2		0	0	2	1	0	0	0	0	0	
		US-001-ST05	<i>creation of auto approve</i>	Ali	Selesai		4		2	3	0	0	0	0	0	0	0	
BF-BM-EP01-PB02	Pengguna (warga MAMPU) BOLEH mengemaskini maklumat profil pengguna yang telah didaftarkan SUPAYA akaun dapat disahkan dan selamat.	US-002-ST01	<i>Install contact page.</i>	Ali	Selesai	5	5		0	3	0	3	0	0	0	0	0	
		US-002-ST02	<i>new table UI</i>	Raziman	Selesai		6		0	0	0	2	2	0	0	2		
		US-002-ST03	<i>implement new db format</i>	Ahmad	Selesai		8		0	2	3	0	3	1	0	0		
		US-002-ST04	<i>template setup</i>	Azhim	Selesai		8		0	0	4	2	3	0	0	0		
		US-002-ST05	<i>export configuration</i>	Ahmad	Selesai		7		0	0	0	0	4	2	0	0		
		US-002-ST06	<i>creation of auto approve</i>	Ali	Selesai		8		0	0	0	0	4	4	1	0		
		US-002-ST07	<i>audit trail</i>	Azhim	Selesai		5		0	0	0	1	2	3	1	0		
		US-002-ST08	<i>creation of new table</i>	Raziman	Selesai		5		0	0	0	0	2	0	0	2		
<b>Epic 2: Pengurusan Bilik Mesyuarat</b>																		
BF-BM-EP02-PB02	Pentadbir Bilik Mesyuarat BOLEH menyemak maklum balas penggunaan bilik mesyuarat yang diterima daripada pengguna (warga Agensi) SUPAYA mendapatkan sumber rujukan untuk melaporkan kerosakan dan perlukan pembaikan yang berkaitan.	US-004-ST01	<i>Install contact page</i>	Azhim	Selesai	8	8		0	0	0	0	0	0	2	3		
		US-004-ST02	<i>new table UI</i>	Raziman	Selesai		6		0	0	0	0	0	0	0	3		
		US-004-ST03	<i>implement new db format</i>	Ali	Selesai		7		1	0	0	0	0	2	3	1		
		US-004-ST04	<i>template setup</i>	Ahmad	Selesai		7		0	2	0	0	0	2	0	0		
		US-004-ST05	<i>table implementation</i>	Ali	Selesai		6		3	0	0	0	0	0	3	1		
		US-004-ST06	<i>export configuration</i>	Fauzi	Selesai		5		0	0	0	2	0	3	0	3		
		US-004-ST07	<i>creation of auto approve</i>	Raziman	Selesai		6		0	0	0	3	2	0	0	0		
<b>JUMLAH STORY POINT</b>						<b>16</b>												
<b>MASA SEBENAR (Jam) (a)</b>							121		14	11	11	20	23	17	10	15		
<b>REMAINING EFFORT (Jam) (b) (b=b hari sebelum – a hari semasa)</b>							121		107	96	85	65	42	25	15	0		
<b>IDEAL TREND (jam) (c) (c = c rancang – (c rancang/jumlah hari x hari semasa))</b>							121		105.875	90.75	75.625	60.5	45.375	30.25	15.125	0		

### b. Artifak Carta *Burndown*

Carta *burndown* merupakan *tool* visual yang dijana hasil daripada pengemaskinian *sprint backlog* yang dilaksanakan setiap hari untuk menjelak kemajuan *sprint*. Carta ini berbentuk menurun dan menuju kepada *zero effort remaining* pada akhir *sprint*.

#### i. Komponen Artifak Carta *Burndown*

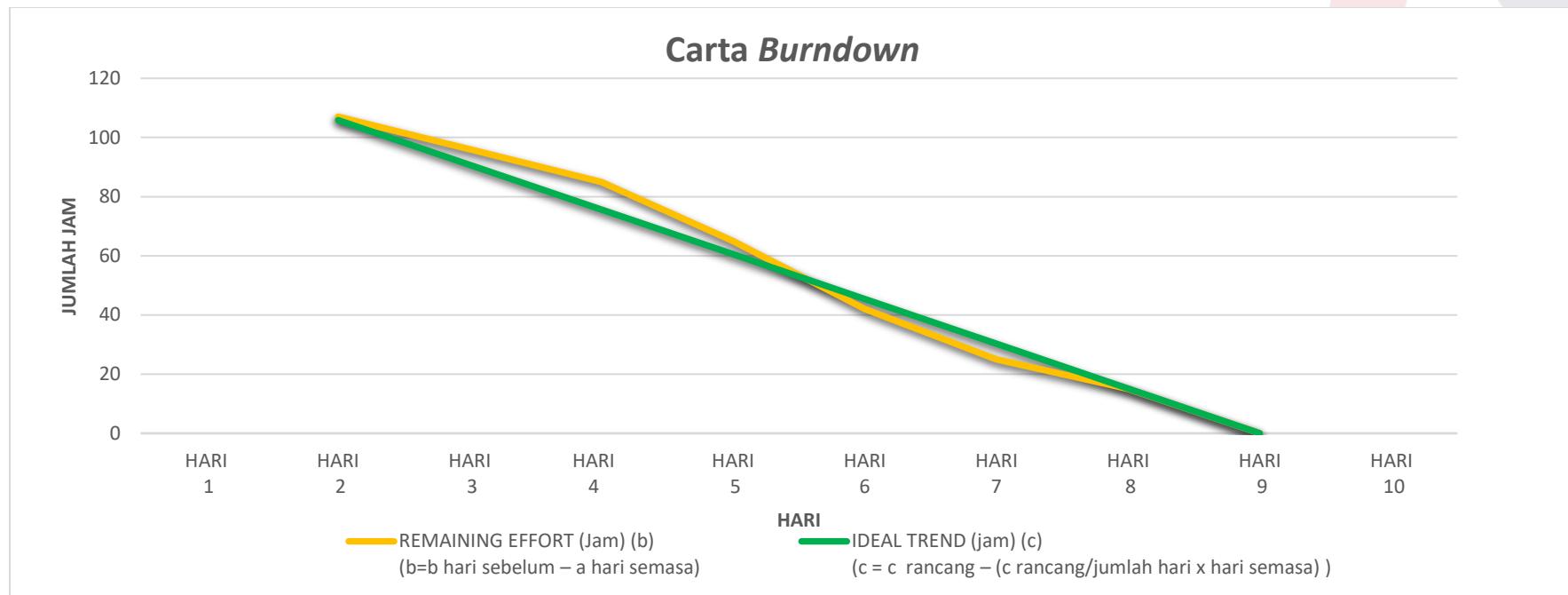
Jadual 3-19 menerangkan komponen yang terdapat berdasarkan artifak carta *burndown*.

**Jadual 3-19: Komponen Carta *Burndown***

No.	Elemen	Penerangan
1.	Paksi Menegak (Paksi-Y)	Jumlah jam dalam <i>sprint</i> .
2.	Paksi Mendatar (Paksi-X)	Hari dalam <i>sprint</i> .
3.	<i>Ideal Trend</i>	Trend penurunan ideal untuk mencapai <i>zero effort remaining</i> pada penghujung <i>sprint</i> .
4.	<i>Remaining Effort</i>	Jumlah masa (jam) yang berbaki untuk menyelesaikan tugas setiap hari.

#### ii. Contoh Pengisian Artifak Carta *Burndown*

Rajah 3-25 menunjukkan contoh artifak carta *burndown* yang dihasilkan daripada artifak *sprint backlog* yang telah diisi bagi tempoh 10 hari bekerja.



	Jumlah Jam	HARI 1	HARI 2	HARI 3	HARI 4	HARI 5	HARI 6	HARI 7	HARI 8	HARI 9	HARI 10
REMAINING EFFORT (Jam) (b) (b=b hari sebelum – a hari semasa)	121		107	96	85	65	42	25	15	0	0
IDEAL TREND (jam) (c) (c = c rancang – (c rancang/jumlah hari x hari semasa))	121		105.875	90.75	75.625	60.5	45.375	30.25	15.125	0	0

Rajah 3-25: Carta Burndown sehingga Hari ke 10

### 3.3.1.5. Aktiviti *Product Backlog Refinement*

*Product backlog refinement* adalah aktiviti menyemak tugas dalam senarai *product backlog*, menilai keutamaan *user story* berdasarkan nilai kepada pengguna dan kompleksiti, serta mengemas kini *product backlog*. Ini bertujuan bagi memastikan item tersebut ditakrifkan dengan baik, difahami, relevan dan memenuhi keperluan pengguna. Panduan *Scrum* mencadangkan bahawa tempoh aktiviti ini dilaksanakan tidak melebihi 10% daripada kapasiti pasukan pembangun<sup>13</sup>.

Aktiviti yang dilaksanakan semasa sesi *product backlog refinement* adalah seperti berikut:

- i. Menyemak produk backlog untuk memastikan item adalah terkini dan relevan.
- ii. Menambah *user story* baharu ke dalam produk backlog berdasarkan maklum balas pengguna atau keperluan baru.
- iii. Menganggarkan masa, usaha, atau kompleksiti yang diperlukan untuk menyelesaikan setiap *user story* dalam produk backlog. Ini membantu pasukan merancang dan mengutamakan setiap *user story*.
- iv. Membahagikan *user story* yang lebih besar menjadi tugas yang lebih kecil dan mudah diuruskan. Ini membantu memastikan setiap *user story* adalah difahami dengan baik dan boleh diselesaikan dalam satu *sprint*.
- v. Mengemas kini keperluan, butiran item dan menambah kriteria penerimaan setiap *user story*.
- vi. Mengutamakan *user story* dalam produk backlog berdasarkan nilai kepada pengguna, maklum balas dan faktor lain.

---

<sup>13</sup> Product Backlog Refinement Time. (n.d.). Scrum.org. Diakses pada April 11, 2023, from <https://www.scrum.org/>

vii. Mengemas kini status setiap *user story* dalam produk backlog berdasarkan kemajuan, perubahan, atau maklum balas.

Rajah 3-26 memaparkan senarai artifak rujukan sebelum aktiviti *product backlog refinement* bermula dan artifak hasil daripada aktiviti *product backlog refinement*.



**Rajah 3-26: *Product Backlog Refinement***

**Artifikat Rujukan** kepada aktiviti *product backlog refinement* adalah *product backlog*, *sprint backlog* dan DoD.

**Artifikat Hasil** daripada aktiviti *product backlog refinement* adalah pengemaskinian artifak *product backlog* yang mengandungi perkara berikut:

- i. Item baharu yang ditambah ke dalam produk backlog.
- ii. Pengemaskinian yang dibuat pada item sedia ada.
- iii. Keutamaan yang ditetapkan pada setiap item.
- iv. Anggaran *story point* yang diberikan pada setiap item.

**Penglibatan ahli pasukan** adalah seperti berikut:

- i. Pengurus produk bertanggungjawab untuk merekod, menyemak dan menetapkan keutamaan keatas senarai item *product backlog*.
- ii. Pasukan pembangun perlu bekerja sama dengan pengurus produk dengan memberikan input dan maklum balas kepada perkara berikut:
  - a. Memahami keutamaan item yang ditetapkan.
  - b. Memberikan anggaran *story point* untuk setiap item.
  - c. Memberikan maklum balas untuk memastikan *product backlog* sentiasa dikemas kini.
- iii. Fasilitator DevOps bertindak sebagai *scrum master* bagi mengurus dan melancarkan sepanjang aktiviti ini berlangsung.

*Product backlog refinement* melibatkan pasukan *scrum* sahaja bagi memastikan ahli pasukan fokus dan mengelakkan perbincangan yang tidak berkaitan. Walau bagaimanapun, SME Bisnes boleh dijemput jika boleh memberikan input dan penyelesaian yang diperlukan oleh pasukan.

#### i. Pengemaskinian Artifak *Product Backlog*

Senarai *product backlog* pada Jadual 3-20 digunakan semasa sesi *product backlog refinement* sebagai artifak perbincangan. Lajur Catatan *Product Backlog Refinement* digunakan sebagai penerangan mengenai kemaskini yang dilaksanakan pada *product backlog*.

**Jadual 3-20: Product Backlog Sebelum Aktiviti Product Backlog Refinement**

PRODUCT BACKLOG								
USER STORY ID	KEUTAMAAN	TURUTAN KEUTAMAAN	USER STORY			STORY POINT	SPRINT NO	CATATAN PRODUCT BACKLOG REFINEMENT
<b>Modul Mengurus Pengguna</b>								
PB-BF-BM-MP-PR-01	Mesti Ada	1	Pengguna (warga JDN)	<b>BOLEH</b> mendaftar profil pengguna baharu	<b>SUPAYA</b> boleh mendaftar masuk sistem tempahan bilik mesyuarat	3	Sprint 1	
PB-BF-BM-MP-PR-02	Mesti Ada	2	Pengguna (warga JDN)	<b>BOLEH</b> mengemas kini maklumat profil pengguna yang telah didaftarkan	<b>SUPAYA</b> akaun dapat disahkan dan selamat.	5	Sprint 1	

Contoh *product backlog* yang dikemas kini hasil daripada aktiviti *Product Backlog Refinement* adalah seperti Jadual 3-21.

- Product backlog* PB-BF-BM-MP-PR-01 memerlukan fungsi tambahan, iaitu notifikasi melalui email perlu dihantar kepada pengguna setelah pendaftaran pengguna berjaya. *Story point* dikemaskini untuk pertambahan aktiviti pembangunan API.
- Product backlog* PB-BF-BM-MP-PR-03 ditambah dalam *product backlog* sebagai item baru untuk dilaksanakan pada *sprint* kedua.

### Jadual 3-21: Pengemaskinian *Product Backlog*

PRODUCT BACKLOG								
USER STORY ID	KEUTAMAAN	TURUTAN KEUTAMAAN	USER STORY			STORY POINT	SPRINT NO	CATATAN PRODUCT BACKLOG REFINEMENT
<b>Modul Mengurus Pengguna</b>								
PB-BF-BM-MP-PR-01	Mesti Ada	1	Pengguna (warga JDN)	<b>BOLEH</b> mendaftar profil pengguna baharu	<b>SUPAYA</b> boleh mendaftar masuk sistem tempahan bilik mesyuarat	5	Sprint 1	Dipecahkan kepada <i>product backlog</i> baharu, PB-BF-BM-MP-PR-03
PB-BF-BM-MP-PR-02	Mesti Ada	2	Pengguna (warga JDN)	<b>BOLEH</b> mengemas kini maklumat profil pengguna yang telah didaftarkan	<b>SUPAYA</b> akaun dapat disahkan dan selamat.	5	Sprint 1	
PB-BF-BM-MP-PR-03	Mesti Ada	3	Pengguna (warga JDN)	<b>BOLEH</b> menghantar notifikasi melalui email apabila profil pengguna baharu didaftarkan	<b>SUPAYA</b> pengguna menerima notifikasi pendaftaran selesai	5	Sprint 2	Pecahan fungsi daripada PB-BF-BM-MP-PR-01

### 3.3.1.6. Aktiviti *Sprint Review*

*Sprint review* adalah aktiviti yang berlangsung pada akhir setiap *sprint* semasa. Tujuan *sprint review* adalah untuk menilai kerja yang telah dilakukan dalam *sprint* semasa, mendapatkan maklum balas dari SME Bisnes dan memastikan bahawa kerja yang dilakukan selaras dengan *product vison* dan keperluan SME Bisnes. Penglibatan *scrum master* adalah amat penting bertindak sebagai fasilitator aktiviti.

Aktiviti yang dilaksanakan semasa sesi *sprint review* adalah seperti berikut:

- i. Pasukan pembangun mempamerkan kerja yang telah dilaksanakan kepada SME Bisnes dan menerima maklum balas daripada mereka.
- ii. Pengurus produk membentangkan kemajuan pembangunan dalam mencapai matlamat projek secara keseluruhan.
- iii. Pasukan pembangun memberikan maklum balas mengenai kerja yang telah dilakukan, yang boleh digunakan untuk meningkatkan produk dalam *sprint* seterusnya.
- iv. Pengurus produk mengemas kini *product backlog* berdasarkan maklum balas yang diterima dan merancang untuk *sprint* seterusnya.
- v. Pasukan boleh menggunakan peluang untuk meraikan kemajuan yang telah dicapai dan menghargai usaha ahli pasukan dalam pelaksanaan pembangunan produk.

Rajah 3-27 memaparkan senarai artifak rujukan sebelum aktiviti *sprint review* bermula dan artifak hasil daripada aktiviti *sprint review*.



**Rajah 3-27: Sprint Review**

**Artifak Rujukan** kepada aktiviti *sprint review* adalah *product backlog*, *sprint backlog* dan DoD.

**Artifak Hasil** daripada aktiviti *sprint review* adalah artifak DoD dan *product increment*.

#### i. Contoh Pengisian Artifak *Definition of Done*

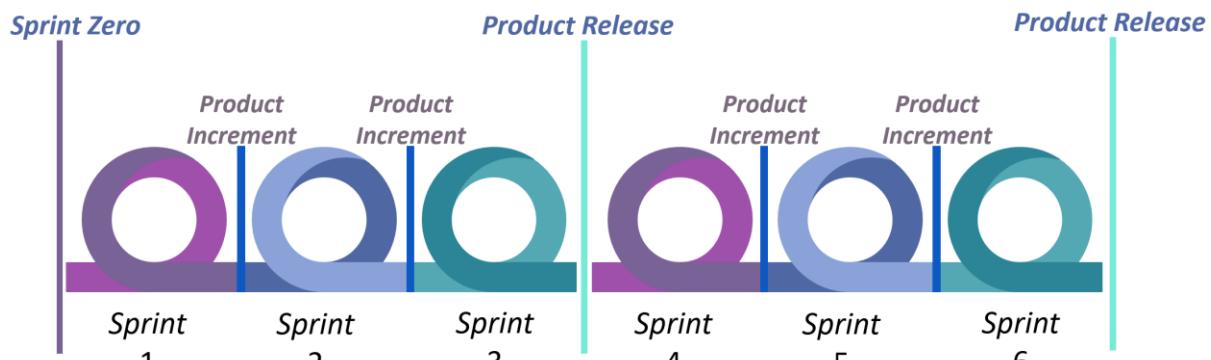
Contoh artifak DoD yang telah lengkap diisi adalah seperti Jadual 3-22.

### Jadual 3-22: Artifak *Definition of Done*

Komponen	Senarai Semak	Status	Catatan
<b>Pengujian Keperluan Fungsian</b>	1. Kod sumber selesai dibangunkan	✓	<i>Sprint 1</i>
	2. Kod sumber disemak	✓	<i>Sprint 1</i>
	3. Kod sumber digabungkan (merged)	✓	<i>Sprint 1</i>
	4. Melepas pengujian unit	✓	<i>Sprint 1</i>
	5. Melepas pengujian integrasi	✓	<i>Sprint 1</i>
	6. Melepas pengujian sistem dan memenuhi <i>acceptance criteria</i> bagi setiap <i>user story</i>	✓	<i>Sprint 1</i>
	7. Melepas pengujian penerimaan pengguna		<i>Sprint 3</i>
<b>Pengujian Keperluan Bukan Fungsian</b>	1. Melepas pengujian kualiti kod	✓	<i>Sprint 1</i>
	2. Melepas pengujian SAST	✓	<i>Sprint 1</i>
	3. Melepas pengujian prestasi	✓	<i>Sprint 1</i>
<b>Penempatan</b>	1. Penempatan ke persekitaran <i>staging</i> pada <i>sprint</i> yang melibatkan <i>product release</i> sahaja		<i>Sprint 3</i>
<b>Dokumentasi</b>	1. Artifak <i>product vision</i>	✓	<i>Sprint 1</i>
	2. Dokumen laporan UAT		<i>Sprint 3</i>
	3. Artifak <i>sprint backlog</i>	✓	<i>Sprint 1</i>
	4. Artifak carta <i>burndown</i>	✓	<i>Sprint 1</i>
	5. Artifak perancangan kapasiti	✓	<i>Sprint 1</i>
	6. Artifak log <i>daily scrum</i>	✓	<i>Sprint 1</i>
	7. Artifak <i>sprint retrospective</i>	✓	<i>Sprint 1</i>

### a. Product Increment

*Product increment* adalah sejumlah *product backlog* yang telah diselesaikan pada *sprint* semasa. *Product increment* merupakan produk yang telah dibangunkan oleh pasukan *scrum* dan memenuhi DoD yang telah dipersetujui. Rajah 3-28 menunjukkan gambaran bahawa *product increment* akan dihasilkan pada setiap *sprint* manakala *product release* akan dilepaskan ke persekitaran produksi setelah berakhirnya *sprint* 3. Penerangan berkaitan *product release* akan diterangkan pada para 3.3.6.2.



Rajah 3-28: Aktiviti *Sprint* dan *Product Increment*

#### 3.3.1.7. Aktiviti *Sprint Retrospective*

*Sprint retrospective* merupakan aktiviti yang diadakan pada akhir *sprint* semasa. Semasa aktiviti ini, pasukan *scrum* menyemak perkara yang berjalan dengan lancar semasa *sprint* semasa, perkara yang tidak berjalan seperti dijangka dan perkara yang perlu ditambah baik dalam *sprint* berikutnya.

Tujuan *sprint retrospective* adalah untuk mengenal pasti perkara untuk penambahbaikan dan membuat perubahan untuk meningkatkan prestasi pasukan.

Aktiviti yang dilaksanakan semasa *sprint retrospective* adalah untuk menggalakkan komunikasi dan maklum balas terbuka adalah seperti berikut:

- a. Menjelaskan tujuan retrospektif dan menetapkan peraturan asas seperti hanya seseorang dibenarkan bercakap pada satu-satu masa, tidak menyalahkan sesama ahli pasukan dan memberikan maklum balas secara membina.
- b. Membincang dan mengumpul maklumat berkaitan *sprint* semasa seperti perkara yang berjalan lancar dan perkara yang tidak berjalan seperti dijangka.
- c. Menganalisis maklumat untuk mengenal pasti perkara yang boleh ditambah baik.
- d. Menentukan tindakan yang boleh diambil untuk memperbaiki proses pelaksanaan pembangunan.
- e. Menyimpulkan perbincangan, bersetuju tentang item tindakan dan menetapkan rancangan untuk *sprint* seterusnya.
- f. Memastikan tindak lanjut pada item tindakan dilaksanakan dan memberi impak positif.

Rajah 3-29 memaparkan senarai artifik rujukan sebelum aktiviti *sprint review* bermula dan artifik hasil daripada aktiviti *sprint review*.



**Rajah 3-29: Sprint Retrospective**

**Artifak Rujukan** kepada aktiviti *sprint retrospective* adalah artifikat DoD dan *product increment*.

**Artifak Hasil** daripada aktiviti *sprint retrospective* adalah artifikat *sprint retrospective* yang mengandungi senarai penambahbaikan untuk *sprint* seterusnya.

**Penglibatan ahli pasukan** adalah seperti berikut:

- i. Pengurus produk boleh menyertai retrospektif, memberikan maklum balas tentang *product backlog* dan membantu mengenal pasti perkara yang perlu untuk peningkatan produk.
- ii. Fasilitator DevOps bertindak sebagai *scrum master*, memastikan ahli pasukan berpeluang memberikan idea dan maklum balas.
- iii. Pasukan pembangun perlu terlibat secara aktif dengan memberikan input dan maklum balas.

*Sprint retrospective* melibatkan pasukan *scrum* sahaja untuk memastikan ahli pasukan bebas mengutarakan idea dan maklum balas. Walau bagaimanapun, SME Bisnes boleh dijemput jika boleh memberikan input dan penyelesaian yang diperlukan oleh pasukan.

### a. Artifak Sprint Retrospective

Artifak *sprint retrospective* merujuk kepada templat yang mengandungi soalan-soalan *post mortem* yang perlu dibincang dan dipersetujui oleh ahli pasukan scrum pada penghujung *sprint*. Artifak ini akan menjadi input penambahbaikan yang boleh dilaksanakan pada *sprint* seterusnya.

#### i. Komponen Artifak Sprint Retrospective

Jadual 3-23 menerangkan komponen dalam artifak *sprint retrospective*.

**Jadual 3-23: Komponen Artifak Sprint Retrospective**

No.	Komponen	Penerangan
1.	Perkara yang Berjalan dengan Lancar	Merujuk kepada perkara positif yang perlu dilakukan lebih lagi.
2.	Perkara yang Tidak Berjalan seperti Dijangka	Merujuk kepada perkara negatif yang sedang dilakukan dan perlu diberhentikan.
3.	Perkara yang Perlu Ditambah baik	Merujuk kepada penambahbaikan yang perlu dilaksanakan.

Templat artifak *sprint retrospective* boleh dirujuk dalam **Lampiran 8: Templat Artifak Sprint Retrospective**.

#### ii. Contoh Pengisian Artifak Sprint Retrospective

Contoh pengisian artifak *sprint retrospective* adalah seperti Jadual 3-24.

### Jadual 3-24: Artifak Sprint Retrospective

ID	Perkara Yang Berjalan Dengan Lancar	Perkara Yang Tidak Berjalan Seperti Dijangka	Perkara Yang Perlu Ditambah Baik
SP001	Nota <i>daily scrum meeting</i> telah diedarkan sebaik selesai aktiviti <i>daily scrum meeting</i>	Tempoh masa <i>daily scrum meeting</i> kerap melebihi 15 minit	Menyediakan pelan penempatan di akhir setiap <i>sprint</i>
SP001	Komunikasi dan kolaborasi yang konsisten dengan pengurus produk	Sesetengah ralat mengambil masa yang agak lama untuk diselesaikan	

Terdapat dua dokumen asas yang boleh digunakan sebagai rujukan seperti yang diperincikan dalam Buku KRISA.

Dokumen yang relevan dan sesuai digunakan untuk pelaksanaan *agile* dan DevOps dalam pembangunan sistem aplikasi adalah seperti berikut:

- a. Pelan Pembangunan Sistem (PPS).
- b. Spesifikasi Keperluan Bisnes (BRS).

Satu kajian kes pembangunan sistem aplikasi iaitu Sistem Pengurusan Bilik Mesyuarat (SPBM) digunakan sebagai contoh pelaksanaan metodologi *agile*. Contoh pelaksanaan metodologi *agile* berdasarkan kajian kes ini adalah seperti di **Lampiran 9: Kajian Kes Agile Scrum**.

### 3.3.2. Peringkat Perancangan

Peringkat perancangan DevOps melibatkan aktiviti-aktiviti sebelum pengekodan bermula seperti memperhalusi keperluan dan maklum balas pengguna dalam skop produk serta merancang pengurusan bagi *tools* dan proses CI/CD.<sup>14</sup> Pelaksanaan DevOps perlu menepati keperluan yang terdapat dalam PPrISA dan KRISA.

Aktiviti *agile* seperti *sprint zero*, *sprint planning meeting* pertama dan *sprint planning meeting* kedua akan dilaksanakan serta artifak yang terhasil daripada aktiviti tersebut akan menjadi input bagi aktiviti pada peringkat ini. Rajah 3-30 memaparkan aktiviti utama dalam peringkat perancangan.



**Rajah 3-30: Peringkat Perancangan**

Peringkat perancangan terdiri daripada empat aktiviti utama iaitu:

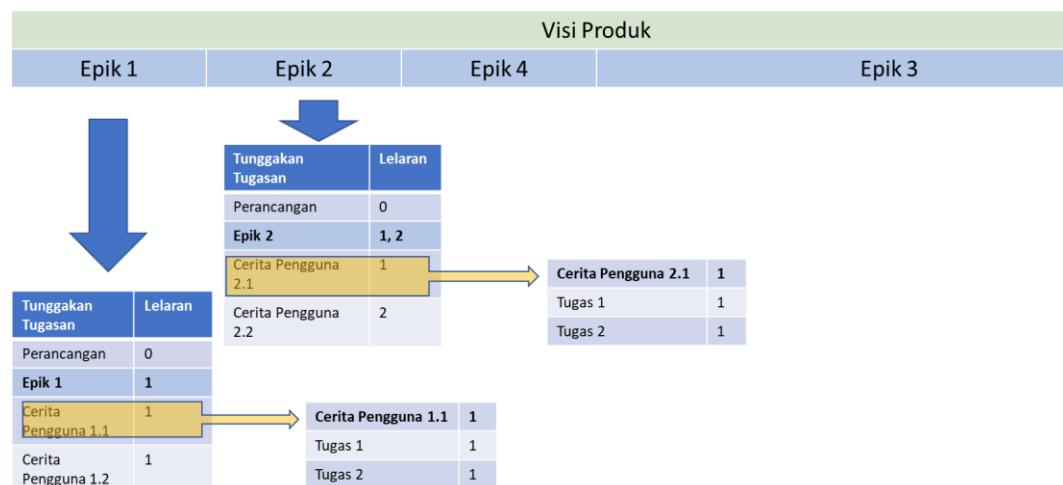
- a. Perancangan Produk,
- b. Pengurusan Komunikasi,
- c. Pengurusan *Tools* dan Proses CI/CD, serta
- d. Perancangan Pengujian.

<sup>14</sup> Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016, May). What is DevOps? A systematic mapping study on definitions and practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016* (pp. 1-11);

### 3.3.2.1. Perancangan Produk

Perancangan produk menjadi asas kepada langkah kerja yang akan dilaksanakan bagi semua ahli pasukan. Perancangan produk menumpukan kepada pelan hala tuju produk bagi menyelaraskan objektif dan memfokuskan kepada ciri-ciri produk yang akan dibangunkan. Aktiviti perancangan produk juga merangkumi penyediaan *tools* yang akan digunakan sepanjang pembangunan sistem aplikasi. Pelan hala tuju produk perlu direkodkan bagi membantu pemantauan kemajuan sepanjang aktiviti pembangunan. Pelan ini terdiri daripada artifik *product vision*, *epic* dan *user story*. Persediaan *tools* boleh dibuat setelah *sprint zero* dilaksanakan iaitu setelah pelan hala tuju produk telah ditetapkan.

Pasukan akan mengenal pasti aktiviti-aktiviti dan tempoh masa yang diperlukan serta menyenaraikan ciri-ciri *user story* yang akan dihasilkan dalam pelan pelepasan. Perkembangan *epic* kepada *user story* adalah seperti di Rajah 3-31.



Rajah 3-31: Perkembangan *Epic* ke *User Story*

Pasukan akan menakrif cerita pengguna menggunakan konsep *Minimum Viable Product* (MVP) dan menghasilkan tunggakan tugas bagi satu atau dua *iteration* awal. Konsep MVP dalam pembangunan produk adalah seperti di **Lampiran 14: Konsep MVP dalam Pembangunan Produk**.

Konsep MVP memastikan produk dihasilkan dengan ciri-ciri dan fungsi asas bagi pengesahan awal produk. MVP membantu pengurus produk memperolehi maklum balas pengguna secepat mungkin bagi penambahbaikan produk. Pengesahan dan penambahbaikan berterusan dalam MVP menjadikan MVP penting dalam metodologi *agile*.

Ciri-ciri produk akan menjadi lebih jelas dengan iteratif dan ciri-ciri baharu ditambah ke dalam pelan pelepasan bagi *iteration* seterusnya. Pasukan dan pengurus produk akan mengesahkan DoD bagi setiap pelepasan berdasarkan *iteration* bagi mengesahkan ciri-ciri produk (berdasarkan *user story*) yang telah selesai. Ciri-ciri produk yang tidak diperlukan akan dikeluarkan daripada pelan pelepasan.

### **3.3.2.2. Pengurusan Komunikasi**

Pengurusan komunikasi merangkumi komunikasi, perbincangan dan pengumuman di antara ahli pasukan sepanjang pembangunan produk. Penglibatan semua ahli pasukan boleh menyumbang kepada keberkesanan komunikasi untuk menyokong aspek budaya DevOps. Antara contoh aktiviti dalam pengurusan komunikasi termasuklah penggunaan *ChatOps*, pengumuman berkaitan pelepasan versi produk dan juga perkongsian fail. Penggunaan *tools* komunikasi yang diintegrasikan dengan *pipeline CI/CD* memudahkan perkongsian maklumat antara ahli pasukan.

### **3.3.2.3. Pengurusan *Tools* dan Proses CI/CD**

Pengurusan *tools* dan proses CI/CD dibahagikan kepada empat pengurusan utama iaitu:

- a. Pengurusan Konfigurasi,
- b. Pengurusan Pindaan,
- c. Pengurusan Insiden, dan
- d. Pengurusan Pelepasan.

Setiap ahli pasukan mempunyai peranan dalam pengurusan *tools* dan proses CI/CD bagi memastikan pembangunan sistem berjalan lancar.<sup>15</sup> Peranan dan tanggungjawab ahli pasukan dalam pengurusan-pengurusan ini adalah seperti di **Lampiran 15: Peranan dan Tanggungjawab dalam Pengurusan Tools dan Proses CI/CD.**

### a. Pengurusan Konfigurasi

Pengurusan konfigurasi merujuk kepada maklumat dan *tools* yang perlu dikonfigurasikan untuk mewujudkan dan mengekalkan keharmonian persekitaran pembangunan.

Tiga aktiviti yang terlibat di bawah pengurusan konfigurasi adalah seperti di Jadual 3-25.

**Jadual 3-25: Aktiviti Pengurusan Konfigurasi**

Aktiviti Pengurusan Konfigurasi	Huraian Aktiviti
<b>Mengenal pasti Konfigurasi</b>	Mengenal pasti maklumat konfigurasi, merekod dan menyenaraikan maklumat dalam repositori perlu dilakukan pada permulaan transformasi ke arah DevOps. <i>Tools</i> pengurusan konfigurasi digunakan untuk mengenal pasti maklumat konfigurasi bagi setiap peringkat DevOps.
<b>Mengawal Versi dan Konfigurasi</b>	Perubahan dalam maklumat konfigurasi dan versi perlu dikawal selia dan aktiviti ini perlu mengambil kira perkara berikut: <ol style="list-style-type: none"><li>Penguatkuasaan,</li><li>Keupayaan Integrasi, dan</li><li>Kawalan Versi.</li></ol>
<b>Perakuan Pematuhan Konfigurasi dan Semakan Audit</b>	Konfigurasi akan sentiasa berubah dan aktiviti ini memastikan pematuhan konfigurasi dilaksanakan dengan teliti melalui prosedur yang betul.

---

<sup>15</sup> Cagle, R., Rice, T., & Kristan, M. (2018). *DevOps for federal acquisition*. MITRE CORP BEDFORD MA;

Maklumat konfigurasi hendaklah diuruskan sepanjang kitaran hayat pembangunan produk. Konfigurasi dalam pelaksanaan DevOps adalah seperti di **Lampiran 16: Konfigurasi dalam Pelaksanaan DevOps**.

**Tiga komponen utama** dalam pengurusan konfigurasi adalah seperti berikut:

- i. **Repositori** - Repositori digunakan untuk menguruskan versi dan menyimpan item konfigurasi dengan selamat. Tiga jenis repositori pengurusan konfigurasi adalah:
  - **Repositori Artifak** - bertujuan menyimpan fail-fail artifak, termasuk binari, API dan *libraries*. Repositori artifak adalah arkib pangkalan data bagi fail-fail artifak berdasarkan sejarah versi (version history),
  - **Repositori Kod Sumber** - merupakan pangkalan data kod sumber yang digunakan oleh pasukan pembangun. Pangkalan data ini berfungsi sebagai *container* untuk semua kod sumber, skrip automasi dan fail konfigurasi, dan
  - **Repositori Pangkalan Data Konfigurasi** - pangkalan data hubungan pelbagai sistem dan aplikasi yang berkaitan dengan pengurusan konfigurasi, termasuk perkhidmatan aplikasi (application services), pelayan, peranti infrastruktur dan pangkalan data.
- ii. **Infrastruktur sebagai Kod** - kod atau skrip yang menyediakan persekitaran infrastruktur yang diperlukan secara automatik supaya bersedia untuk pembangunan dan aktiviti pengujian. Penyediaan semula persekitaran infrastruktur yang sama dan selamat pada bila-bila masa adalah penting dalam *pipeline CI/CD*,<sup>16</sup> dan

---

<sup>16</sup> Artac, M., Borovssak, T., Di Nitto, E., Guerriero, M., & Tamburri, D. A. (2017, May). DevOps: introducing infrastructure-as-code. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)* (pp. 497-498). IEEE.

**iii. Konfigurasi sebagai Kod** - Kod atau skrip yang menyeragamkan konfigurasi dalam sumber tertentu, seperti pelayan atau rangkaian. Kod atau skrip ini sentiasa digunakan sepanjang peringkat pembangunan DevOps.

### b. Pengurusan Pindaan

Pengurusan pindaan membantu meminimumkan risiko berkaitan pindaan dalam *pipeline* CI/CD. Pengurusan pindaan perlu diautomasi secara berperingkat menggunakan *tools* DevOps bagi mengurangkan tempoh masa proses pindaan. Empat aktiviti dalam pengurusan pindaan adalah seperti di Jadual 3-26.

**Jadual 3-26: Aktiviti Pengurusan Pindaan**

Aktiviti-aktiviti Pengurusan Pindaan	Huraian Aktiviti
<b>Permohonan Pindaan</b>	Pindaan <i>standard</i> boleh diautomasi menggunakan <i>tools</i> pengurusan pindaan. Pindaan yang tidak dirancang dan mempunyai impak tinggi kepada pelaksanaan projek memerlukan semakan lanjut.
<b>Menganalisis Kesan Pindaan</b>	Analisis diautomasi berdasarkan kesan, risiko dan faedah yang telah dikenal pasti dan direkodkan ke dalam sistem pengurusan pindaan.
<b>Meluluskan/Menolak Permohonan Pindaan</b>	Maklumat yang perlu direkodkan adalah permohonan pindaan manakala proses membuat keputusan diautomasi berdasarkan risiko dan kesan kepada pengguna. Aktiviti ini memerlukan kelulusan mengikut tadbir urus agensi.
<b>Melaksanakan Pindaan</b>	Merekodkan permohonan pindaan yang telah lulus bagi penyelarasan proses kerja pasukan.

Pengurusan pindaan dalam pelaksanaan DevOps memerlukan pengautomasian secara berperingkat bagi elemen-elemen pindaan iaitu Borang Permohonan Pindaan, Penyata Pindaan, Log Penyelesaian Isu dan Aliran Proses Kawalan Pindaan. Berikut adalah contoh-contoh pindaan berimpak tinggi dalam skop projek menggunakan metodologi *Agile*:

- i. Penambahan dan penghapusan *epic* dari MVP atau produk yang akan datang,
- ii. penambahan *epic* baru bagi ciri-ciri produk yang tidak disenaraikan dalam *epic* sedia ada,
- iii. perubahan terhadap DoD bagi *iteration* semasa yang memerlukan penambahan masa projek,
- iv. penyusunan semula keutamaan serahan *epic* dalam *roadmap* produk yang telah dipersetujui, dan
- v. perubahan pada garis masa kerana berlaku penambahan *iteration* yang perlu dilaksanakan bagi pembangunan MVP produk yang dipersetujui.

### c. Pengurusan Insiden

Pengurusan insiden melibatkan proses menganalisis insiden, memulihkan sistem yang terlibat serta mengelakkan pengulangan insiden. Matlamat pengurusan insiden adalah untuk menangani insiden dengan cekap supaya mengurangkan impak kepada masa dan kualiti penghasilan produk. *Tools* pengurusan insiden memantau dan menganalisis sebarang gangguan terhadap perkhidmatan produk dan rantaian *tools* CI/CD secara proaktif dan berterusan.

**Lima aktiviti** yang terlibat dalam pengurusan insiden adalah seperti di Jadual 3-27.

**Jadual 3-27: Aktiviti Pengurusan Insiden**

Aktiviti-aktiviti Pengurusan Insiden	Huraian Aktiviti
<b>Merekod Insiden</b>	Pengesanan insiden berlaku lebih awal melalui penggunaan <i>tools</i> DevOps dan akan direkodkan secara automatik.
<b>Berkomunikasi dengan Pihak Berkepentingan</b>	<i>Tools</i> komunikasi digunakan untuk memaklumkan insiden dengan segera serta melaporkan situasi terkini insiden dari semasa ke semasa kepada pihak berkepentingan. <i>Tools</i> komunikasi adalah penting dan perlu diintegrasikan dengan <i>tools</i> pengurusan insiden.
<b>Mengenal pasti Resolusi Insiden</b>	Diagnosis dibuat bagi menyelesaikan insiden yang berlaku dengan cepat. Pasukan DevOps perlu mempunyai capaian kepada log dan berupaya untuk memahami implikasi insiden. Ahli pasukan bekerjasama secara komited bagi menyelesaikan isu dengan segera.
<b>Melaksanakan Kajian Penutupan Insiden</b>	<i>Tools</i> pengurusan insiden mengeluarkan laporan insiden secara terperinci untuk makluman dan perhatian pihak berkepentingan. <i>Blameless post-mortem</i> dilakukan sebaik sahaja insiden selesai oleh pasukan.
<b>Menyediakan Dokumentasi Insiden</b>	Maklumat mengenai penambahbaikan dan tindakan yang perlu direkodkan untuk kegunaan pasukan dan mengurangkan insiden berlaku pada masa hadapan.

*Tools* pengurusan insiden yang menyokong amalan DevOps mempunyai ciri-ciri seperti papan pemuka, mempunyai *timeline* kejadian, menjana laporan penutupan insiden, menunjukkan masa untuk mengesan isu MTTA dan masa untuk menyelesaikan isu MTTR.

#### d. Pengurusan Pelepasan

Pengurusan pelepasan melibatkan aktiviti menyelaras dan mengurus pelepasan sistem dalam persekitaran yang berbeza tanpa gangguan perkhidmatan. Pelepasan automatik yang berterusan adalah satu pencapaian utama dalam pelaksanaan DevOps.

Pelan pelepasan adalah dokumen yang menyenaraikan ciri-ciri yang dirancang berserta jangkaan masa pelepasan. Pelan ini menggunakan maklum balas daripada *iteration* sebelumnya dan menetapkan skop, garis masa dan sumber untuk setiap pelepasan. Tujuh langkah dalam penyediaan pelan pelepasan adalah seperti di Rajah 3-32.



**Rajah 3-32: Langkah - Langkah bagi Penyediaan Pelan Pelepasan**

Tools yang digunakan bagi pengurusan pelepasan memberi maklumat dan gambaran yang jelas tentang ciri, tarikh dan masa pelepasan, persekitaran serta proses pelepasan. Tools kawalan versi dan pengurusan konfigurasi bertindak sebagai pengurus pelepasan iaitu mengautomasikan pengesahan pindaan serta memperakui pelepasan aplikasi. Pelepasan berlaku apabila pengesahan berjaya dilakukan bagi produk yang dibangunkan. Lima aktiviti utama dalam pengurusan pelepasan adalah seperti di Jadual 3-28.

### **Jadual 3-28: Aktiviti Pengurusan Pelepasan**

Aktiviti-aktiviti Pengurusan Pelepasan	Huraian Aktiviti
<b>Perancangan</b>	<ul style="list-style-type: none"> <li>i. Menentukan keperluan dan kriteria khusus untuk pelepasan.</li> <li>ii. Memastikan kriteria penerimaan produk ditakrifkan dengan jelas dalam pelan pelepasan.</li> </ul>
<b>Pembinaan</b>	Menentukan kriteria penerimaan menepati keperluan yang ditetapkan bagi membina pelepasan.
<b>Pengujian</b>	Menyemak persekitaran pengujian untuk memastikan semua kriteria penerimaan pelepasan dipenuhi.
<b>Penyediaan</b>	Membolehkan pindaan dan kemas kini produk dilakukan tanpa memberi kesan langsung kepada versi semasa pengguna sedia ada.
<b>Penempatan</b>	Penempatan pelepasan adalah serahan produk kepada pengguna dengan fungsi terkini.

Metrik prestasi pelepasan digunakan untuk menambah baik proses pelepasan yang berikutnya. Aktiviti penilaian dilakukan sebaik sahaja proses pelepasan selesai, dengan menggunakan metrik-metrik seperti masa henti pelepasan, jenis dan keutamaan pelepasan serta bilangan pelepasan yang menepati pelan pelepasan.

### 3.3.2.4. Perancangan Pengujian

Perancangan pengujian dilaksanakan bagi membolehkan pengurus produk, pasukan pembangun dan penguji merancang, menjadual dan melaksana aktiviti pengujian termasuklah serahan yang akan digunakan untuk mengesahkan bahawa aplikasi itu memenuhi keperluan pengguna yang ditetapkan.

Bermula pada *sprint zero*, pengurus produk dan pasukan pembangun akan membincangkan perkara yang perlu dicapai dalam aktiviti pengujian bagi setiap *sprint*. Pelan Pengujian disediakan dengan tujuan untuk:

- a. Merancang dan mengurus aktiviti pengujian sistem aplikasi secara menyeluruh.
- b. Menentukan maklumat berkaitan keperluan *tools* pengujian, data pengujian dan persekitaran pengujian yang digunakan untuk menyediakan keperluan pengujian.
- c. Memahami peranan dan tanggungjawab pihak yang terlibat yang akan melaksanakan proses dan prosedur pengujian.

#### a. Pelan Pengujian

Pelan Pengujian mengandungi perkara berikut:

- i. Penetapan jenis dan skop aktiviti pengujian yang akan dilaksanakan.
- ii. Pendekatan pelaksanaan pengujian.
- iii. Penetapan tahap kritikal untuk pengendalian ralat.
- iv. Penetapan *traceability matrix* untuk liputan kes pengujian.

## b. Jenis dan Skop Aktiviti Pengujian

Pendekatan bagi aktiviti pengujian ini dikenali sebagai *incremental testing* iaitu pengujian dilakukan secara berperingkat bermula daripada pengujian unit/komponen terkecil sistem aplikasi seperti fungsi, kelas, prosedur dan antara muka (Pengujian Unit); seterusnya menguji dua atau lebih modul/sistem/element yang disepadukan (Pengujian Integrasi); dan akhirnya semua modul yang terlibat diuji secara menyeluruh (Pengujian Sistem).

Skop pengujian dalam panduan ini memfokuskan kepada jenis dan tahap pengujian sistem aplikasi seperti berikut:

### a. Pengujian keperluan fungsian (functional test)

- i. Pengujian Unit
- ii. Pengujian Integrasi
- iii. Pengujian Sistem
- iv. Pengujian Penerimaan Pengguna (UAT)

### b. Pengujian keperluan bukan fungsian (non-functional test)

- i. Kualiti Kod
- ii. Pengujian Keselamatan
  - *Static Application Security Testing* (SAST)
  - *Software Composition Analysis* (SCA)
  - *Dynamic Application Security Testing* (DAST)
  - *Interactive Application Security Testing* (IAST)
  - *Secret Detection*
- iii. Pengujian Prestasi

Terdapat 11 jenis pengujian sistem aplikasi yang akan digunakan dalam pelaksanaan panduan ini seperti Jadual 3-29.

**Jadual 3-29: Jenis Pengujian berdasarkan Kuadran Pengujian Agile**

<b>Pengujian Keperluan Fungsian</b>			
No.	Jenis Pengujian	Keterangan	Skop Pengujian
1.	Pengujian Unit	Pengujian bagi menilai unit atau komponen terkecil (lowest level) sistem aplikasi seperti fungsi, kelas, prosedur dan antara muka.	<ul style="list-style-type: none"> <li>i. Dilaksanakan secara automasi menggunakan <i>pipeline</i> CI/CD dengan bantuan <i>tools</i> pengujian unit atau secara manual.</li> <li>ii. Skrip automasi pengujian unit disediakan oleh pembangun dengan bantuan <i>tools</i> pengujian unit.</li> </ul>
2.	Pengujian Integrasi	Pengujian dua atau lebih modul/sistem/element yang disepadukan dan telah berjaya melepas pengujian unit.	<ul style="list-style-type: none"> <li>i. Dilaksanakan secara automasi menggunakan <i>pipeline</i> CI/CD atau secara manual.</li> <li>ii. Menguji keupayaan di antara modul/sistem/element untuk menerima, memproses dan mengeluarkan output dalam sistem atau persekitaran yang sama.</li> <li>iii. Menguji keupayaan komunikasi API.</li> <li>iv. Menguji semula modul/sistem/element dan API yang telah selesai pada <i>sprint</i> sebelumnya bersama fungsi baharu pada <i>sprint</i> semasa.</li> </ul>

<b>Pengujian Keperluan Fungsian</b>			
No.	Jenis Pengujian	Keterangan	Skop Pengujian
3.	Pengujian Sistem	Pengujian semua modul yang terlibat pada <i>sprint</i> semasa secara menyeluruh.	<ul style="list-style-type: none"> <li>i. Dilaksanakan secara automasi menggunakan <i>pipeline CI/CD</i> dengan bantuan <i>tools pengujian</i> atau secara manual.</li> <li>ii. Menguji keperluan fungsian (functional requirements) semua modul yang terlibat melalui antara muka pengguna mengikut kes pengujian yang telah ditetapkan.</li> <li>iii. Menguji semula keperluan fungsian yang telah selesai pada <i>sprint</i> sebelumnya bersama fungsi baharu pada <i>sprint</i> semasa.</li> </ul>
4.	Pengujian Penerimaan Pengguna	Pengujian keperluan fungsian sistem aplikasi secara menyeluruh untuk memastikan sistem aplikasi memenuhi spesifikasi keperluan yang ditetapkan oleh pengurus produk.	<ul style="list-style-type: none"> <li>i. Dilaksanakan secara manual oleh pengurus produk berdasarkan kes pengujian yang telah dipersetujui.</li> <li>ii. Menguji dan mengesahkan keperluan fungsian yang dibangunkan memenuhi spesifikasi keperluan yang telah ditetapkan oleh pengurus produk.</li> </ul>

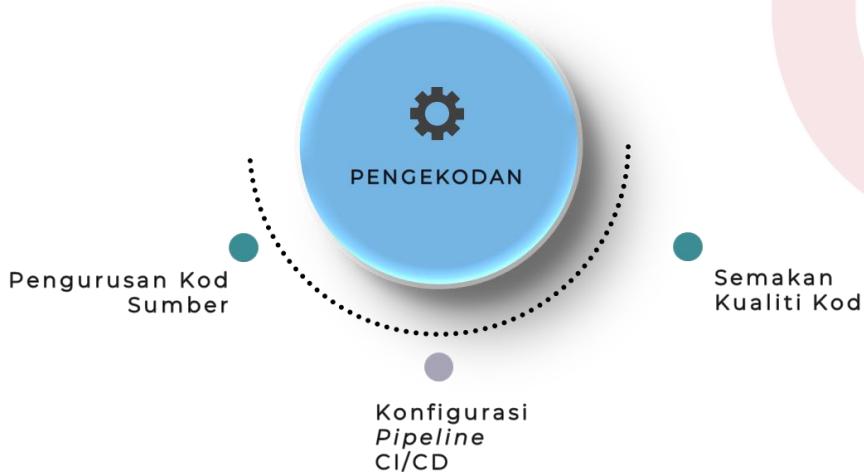
<b>Pengujian Keperluan Bukan Fungsian</b>			
No.	Jenis Pengujian	Keterangan	Skop Pengujian
5.	Kualiti Kod	Pengawalan kualiti kod sumber dengan memastikan pengekodan kod sumber mengikut standard yang ditetapkan.	<ul style="list-style-type: none"> <li>i. Dilaksanakan secara automasi menggunakan <i>pipeline CI/CD</i>.</li> <li>ii. Dilaksanakan dengan menganalisis kod sumber untuk mengesan ralat dan kualiti berkaitan pengekodan kod sumber.</li> </ul>
6.	Pengujian SAST	Pengujian kod sumber untuk mengenal pasti kelemahan atau ralat dari segi keselamatan ke atas sistem aplikasi.	<ul style="list-style-type: none"> <li>i. Dilaksanakan secara automasi menggunakan <i>pipeline CI/CD</i>.</li> <li>ii. Menganalisis kod sumber sistem aplikasi dari segi reka bentuk dan pengekodan untuk mengenal pasti kelemahan atau ralat dari segi keselamatan yang boleh menjadi ancaman ke atas sistem aplikasi.</li> </ul>
7.	Pengujian SCA	Pengimbasan dan analisis keselamatan berkaitan komponen pihak ketiga dan perpustakaan sumber terbuka dalam kod sumber.	<ul style="list-style-type: none"> <li>i. Dilaksanakan secara automasi menggunakan <i>pipeline CI/CD</i>.</li> <li>ii. Mengenal pasti semua komponen sumber terbuka atau pihak ketiga yang digunakan dalam pembangunan sistem aplikasi dengan menganalisis <i>dependencies manifest file</i> projek untuk mencari sebarang isu keselamatan.</li> </ul>
8.	Pengujian DAST	Pengujian keselamatan dengan mensimulasikan serangan terhadap sistem aplikasi dan menganalisis respons untuk mengenal pasti ancaman keselamatan.	<ul style="list-style-type: none"> <li>i. Dilaksanakan secara automasi menggunakan <i>pipeline CI/CD</i>.</li> <li>ii. Menguji sistem aplikasi yang sedang berjalan untuk mengesan kelemahan keselamatan seperti suntikan</li> </ul>

<b>Pengujian Keperluan Bukan Fungsian</b>			
No.	Jenis Pengujian	Keterangan	Skop Pengujian
			SQL, <i>cross-site scripting</i> , menilai mekanisme pengesahan, menyemak salah konfigurasi dan memastikan keselamatan API.
9.	Pengujian IAST	Pengimbasan keselamatan yang mengumpul, memantau dan mengenal pasti maklumat dari sistem aplikasi semasa simulasi DAST dilaksanakan.	<ul style="list-style-type: none"> <li>i. Dilaksanakan secara automasi menggunakan <i>pipeline CI/CD</i>.</li> <li>ii. Dilaksanakan dengan menyepadukan IAST dalam sistem aplikasi yang beroperasi untuk mendapatkan maklum balas terperinci berkaitan operasi sistem aplikasi, butiran aliran data, laluan kawalan dan juga maklumat konfigurasi.</li> </ul>
10.	Pengujian Secret Detection	Pengimbasan dan pengenalpastian maklumat atau rahsia sensitif dalam kod sumber dan fail konfigurasi aplikasi.	<ul style="list-style-type: none"> <li>i. Dilaksanakan secara automasi menggunakan <i>pipeline CI/CD</i>.</li> <li>ii. Mengimbas dan menganalisis pangkalan kod dan fail teks dalam repositori untuk mengenal pasti corak yang sepadan dengan format maklumat sensitif seperti kunci API, token akses, kata laluan dan kelayakan(<i>credential</i>).</li> </ul>
11.	Pengujian Prestasi	Pengujian terhadap keupayaan sistem dari segi kelajuan tindak balas dan kestabilan berdasarkan tahap bebanan yang telah ditetapkan mengikut sasaran pengguna sistem aplikasi.	<ul style="list-style-type: none"> <li>i. Dilaksanakan untuk mengenal pasti bagaimana sistem bertindak di bawah beban kerja dan kekangan tertentu untuk melaksanakan sesuatu fungsi.</li> <li>ii. Dilaksanakan menggunakan <i>tools</i> secara automasi menggunakan <i>pipeline CI/CD</i> atau secara manual dengan bantuan <i>tools</i> pengujian prestasi.</li> </ul>

### 3.3.3. Peringkat Pengekodan

Pengekodan merupakan peringkat pembangunan kod aplikasi berdasarkan *user story* yang telah ditetapkan pada *sprint* semasa. Ahli pasukan DevOps menggunakan sistem kawalan versi untuk mendaftar masuk kod dan membangunkan produk. Integrasi berterusan diamalkan dengan penggunaan *tools* automasi dalam proses pembangunan, pengujian dan pelepasan. Pelaksanaan *pair programming* dan *peer review* dapat meningkatkan kerjasama dan kemahiran ahli pasukan. Pengujian secara berpasangan silang (cross pairing) dilaksanakan bagi mengawal mutu kod aplikasi. Dalam kaedah ini, penguji pertama akan melaksanakan pengujian manakala penguji kedua akan membuat pemerhatian, mengambil nota dan memberi maklum balas terhadap ujian tersebut. Pengemaskinian kod sumber turut berlaku pada peringkat ini sekiranya berlaku perubahan. Aktiviti *daily scrum meeting* akan diadakan pada peringkat ini bagi tujuan pemantauan dan pelaporan tahap kemajuan pembangunan produk yang dibangunkan. Pembangun akan mendaftar masuk dan mengintegrasikan kod sumber ke repositori sistem pengurusan kod sumber berpusat (repositori berpusat).

Setelah pasukan pembangun menyelesaikan *sprint task*, mereka akan *commit* dan *push* kod sumber mereka ke repositori berpusat. Pembangun akan menghantar perubahan kod sumber dengan *merge request* iaitu permintaan untuk menggabungkan kod baharu mereka ke repositori berpusat. Penilai iaitu pembangun yang mempunyai kepakaran dalam semakan kod sumber akan menyemak perubahan yang telah pasukan pembangun laksanakan. Setelah analisis dibuat dan tidak menemui sebarang masalah, mereka akan meluluskan *merge request*. Semakan secara manual ini sepatutnya pantas, tetapi berkesan untuk mengenal pasti masalah lebih awal. Rajah 3-33 memaparkan aktiviti utama dalam peringkat pengekodan.



**Rajah 3-33: Peringkat Pengekodan**

Terdapat tiga aktiviti utama pada peringkat ini:

- a. Pengurusan Kod Sumber,
- b. Konfigurasi *Pipeline* CI/CD dan
- c. Semakan Kualiti Kod.

### 3.3.3.1. Pengurusan Kod Sumber

Pengurusan kod sumber melibatkan aktiviti pewujudan *feature branch* bagi setiap ahli pasukan pembangun berdasarkan *product backlog* atau *sprint backlog* yang telah ditetapkan bergantung kepada kesesuaian dan penggunaan pasukan pembangun. Selain daripada itu, aktiviti lain dalam pengurusan kod sumber ini adalah *merge request* daripada *feature branch* ke repositori berpusat bagi tujuan pengujian.

### 3.3.3.2. Konfigurasi *Pipeline* CI/CD

*Pipeline* CI/CD akan dikonfigurasikan pada peringkat ini berdasarkan kesesuaian sistem aplikasi. Konfigurasi *pipeline* CI/CD adalah dibuat berdasarkan perancangan pada para 3.3.2.3.

### **3.3.3.3. Semakan Kualiti Kod**

Semakan kualiti kod bertujuan memastikan kod sumber dibina berdasarkan standard pembangunan sistem aplikasi yang telah ditetapkan berdasarkan Jaminan Kualiti Perisian(SQA). Penerangan berkaitan SQA dan atribut kualiti perisian boleh didapati dalam buku panduan KRISA. Bagi memastikan SQA terjamin di bawah pelaksanaan DevOps, proses semakan awal pada *pipeline* CI/CD perlu dilaksanakan. Antara pengujian awal yang boleh dibuat adalah seperti berikut:

#### **a. Pengujian Unit**

Aktiviti pengujian bagi menilai unit terkecil (lowest level) pada sistem aplikasi seperti kod sumber, fungsi dan sebagainya.

#### **b. Pengujian SAST**

Aktiviti pengujian ini bagi menyemak *vulnerabilities* yang terdapat pada kod sumber tersebut.

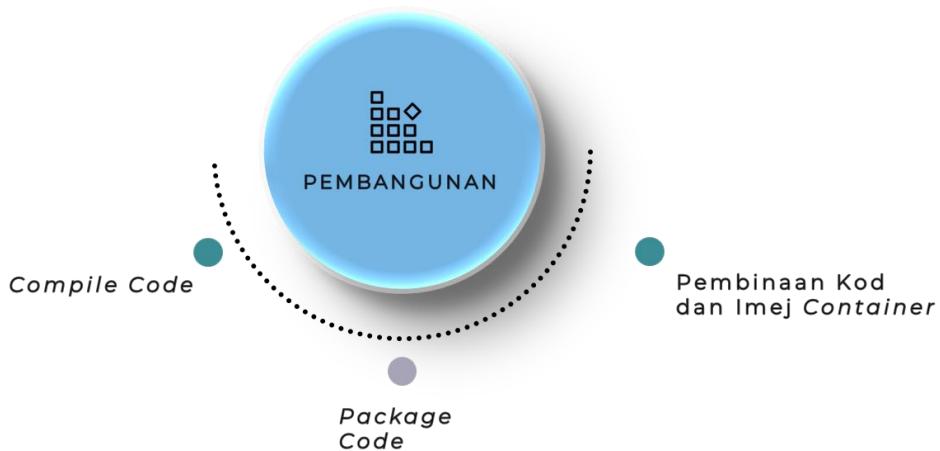
#### **c. Kualiti kod**

Aktiviti pengujian atau analisis ini dibuat pada *pipeline* CI/CD bagi memastikan perubahan kod tersebut dapat meningkatkan prestasi sistem aplikasi tersebut.

Penerangan lebih lanjut pengujian awal ini akan diterangkan pada peringkat pengujian.

### 3.3.4. Peringkat Pembangunan

Peringkat pembangunan merupakan proses pengubahsuaian fail dan aset lain dibawah tanggungjawab pasukan pembangun menjadi produk dalam bentuk akhir atau boleh digunakan. Pembinaan ini termasuk menyusun (compiling) fail kod sumber. Pada peringkat ini, setelah *merge request* diluluskan, kod sumber yang dibangunkan akan dibina. Peringkat pembangunan ini dikonfigurasikan dalam *pipeline CI/CD*. Rajah 3-34 memaparkan aktiviti utama dalam peringkat pembangunan.



**Rajah 3-34: Peringkat Pembangunan**

Peringkat Pembangunan terdapat tiga aktiviti utama iaitu:

- a. *Compile Code*,
- b. *Package Code* dan
- c. *Pembinaan Kod dan Imej Container*

### **3.3.4.1. *Compile Code***

*Compile code* merupakan proses penyusunan dan pengubahsuaian kod sumber daripada *human-readable code* kepada *machine-executable code*. Proses ini melibatkan penggunaan *code compiler* bagi menganalisis dan menterjemahkan kod sumber kepada bentuk fail *executable* serta mengandungi *libraries* dan fail yang berkaitan. Fail ini boleh dipasang pada mesin atau komputer untuk menjalankan program tersebut.

### **3.3.4.2. *Package Code***

*Package code* merujuk kepada satu atau lebih fail kod yang telah disusun dan dikumpulkan bersama bagi membentuk satu pakej fail (distributable format) yang sedia untuk diimport atau digunakan semula ke aplikasi pelayan di peringkat penempatan. *Package code* memudahkan pengurusan kod sumber kerana membolehkan pembangun memisahkan kod sumber menjadi modul yang berbeza dan memfokuskan pada setiap modul pada masa yang berbeza.

### **3.3.4.3. *Pembinaan Kod Sumber dan Imej Container***

Pembinaan kod sumber dan imej *container* produk melibatkan proses pembangunan atau pembinaan produk daripada kod sumber kepada sistem yang boleh *execute* dan diuji. Pada peringkat ini, aktiviti pengintegrasian berterusan (CI) seperti menyusun kod sumber, memuat turun *dependencies*, membina imej *container* dan menghasilkan pakej perisian yang sedia untuk digunakan. Penggunaan *pipeline CI/CD* akan dikonfigurasi pada peringkat ini berdasarkan kesesuaian produk.

### 3.3.5. Peringkat Pengujian

Peringkat pengujian merupakan aktiviti verifikasi yang dilakukan terhadap komponen atau sistem aplikasi untuk memastikan sistem dibangunkan berdasarkan kepada spesifikasi keperluan dan reka bentuk sistem. Jenis-jenis pengujian yang dijalankan adalah pengujian keperluan fungsian, pengujian keperluan bukan fungsian serta verifikasi terhadap ralat yang telah dibaiki.

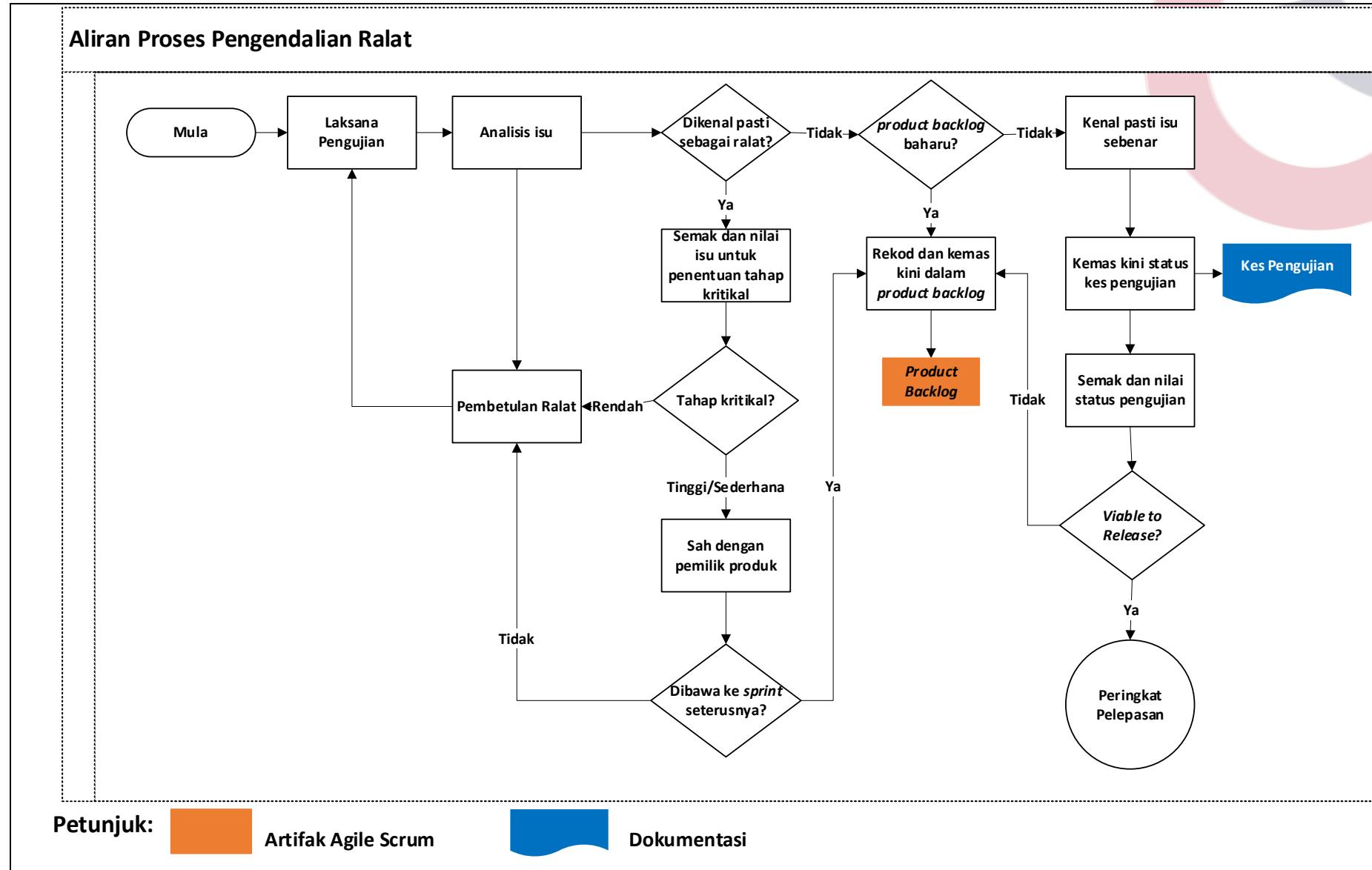
#### 3.3.5.1. Penetapan Tahap Kritikal (Severity Level)

*Severity level* bermaksud tahap kritikal terhadap kegagalan fungsian atau penemuan ralat semasa pengujian. Faktor dalam menentukan keutamaan (priority) boleh dirujuk pada **Lampiran 11: Tahap Kritikal** sebagai panduan.

Masalah yang diperolehi semasa proses pengujian akan dianalisis dan dikategorikan kepada berdasarkan Tahap Kritikal dan kesannya, rujuk **Lampiran 11: Tahap Kritikal**. Pasukan DevOps boleh menggunakan Tahap Kritikal untuk menentukan pelan tindakan selanjutnya seperti berikut:

- a. Penentuan keutamaan kerja terhadap item daripada senarai *product backlog*.
- b. Penentuan impak ralat ke atas produk serta penetapan keutamaan kerja untuk pembetulan setiap ralat dalam pengujian.
- c. Penentuan kriteria penerimaan dan kelulusan pengujian (*viable to release*).
- d. Aliran Proses Pengendalian Ralat.

Proses pengendalian ralat diterangkan dalam aliran proses pada Rajah 3-35.



Rajah 3-35: Aliran Proses Pengendalian Ralat

Berikut adalah penerangan proses dalam aliran pengendalian ralat.

**Jadual 3-30: Keterangan Aliran Proses Pengendalian Ralat**

Proses	Keterangan
<b>Pelaksanaan Pengujian</b>	Penguji melaksanakan pengujian dan melaporkan ralat yang ditemui semasa proses pengujian.
<b>Menganalisis Isu</b>	Pasukan pembangunan dan pengurus produk akan menganalisis isu yang dilaporkan.
<b>Kenal pasti isu sebenar</b>	Jika isu yang dilaporkan dikenal pasti sebagai bukan ralat, tetapi keperluan baharu, isu akan direkodkan semasa aktiviti <i>product backlog refinement</i> .
<b>Kemas kini status kes pengujian</b>	Wakil pasukan pembangun akan mengemas kini log isu berdasarkan ralat yang telah dikenal pasti.
<b>Semakan dan penilaian isu bagi menentukan keutamaan dan menentukan tahap kritikal</b>	Jika isu yang dilaporkan dikenal pasti sebagai ralat, pasukan pembangun dan penguji akan melakukan semakan dan penilaian isu bagi penentuan keutamaan dan menentukan tahap kritikal ralat.
<b>Pembetulan ralat</b>	Jika tahap kritikal ralat adalah rendah, pasukan pembangunan akan terus membetulkan ralat dan melaksanakan pengujian semula.
<b>Sahkan dengan pengurus produk</b>	Jika tahap kritikal ralat adalah tinggi atau sederhana, perbincangan di antara pasukan pembangun dan pengurus produk akan dilakukan untuk menentukan sama ada ralat harus diselesaikan pada <i>sprint</i> semasa atau <i>sprint</i> berikutnya.
<b>Semakan dan penilaian status pengujian</b>	Pasukan pembangun dan pengurus produk seterusnya akan memutuskan sama ada DoD dipenuhi berdasarkan status kes pengujian.

### 3.3.5.2. Penetapan *Traceability Matrix*

*Traceability Matrix* disediakan untuk menjelaki hubungan diantara keperluan (requirement) dengan kes pengujian sepanjang kitar hayat pembangunan produk. Ia berupaya memastikan:

- a. Semua keperluan sistem telah dibangunkan.
- b. Keperluan atau fungsi yang dibangunkan telah dilaksanakan pengujian.
- c. Semua pindaan ke atas aktiviti yang berkaitan dilaksanakan.
- d. Kelebihan penggunaan *Traceability Matrix* adalah seperti berikut:
- e. Memastikan keseluruhan keperluan diuji dan dipenuhi.
- f. Mengenal pasti keperluan yang tidak dinyatakan atau tidak konsisten.

Berikut merupakan komponen dan contoh *Traceability matrix*.

#### a. Penyediaan Templat *Traceability Matrix*

*Traceability matrix* disediakan berdasarkan templat seperti di **Lampiran 12: Templat *Traceability Matrix Ujian*** dengan komponen seperti berikut:

**Jadual 3-31: Komponen Templat *Traceability Matrix***

No.	Komponen	Keterangan
1.	ID Fungsi Bisnes	ID unik bagi fungsi bisnes yang dirujuk dari dokumen BRS.
2.	ID <i>User Story</i>	ID unik bagi <i>user story</i> yang merujuk kepada ringkasan fungsi bisnes.
3.	ID <i>Product Backlog</i>	ID unik bagi <i>product backlog</i> yang merujuk kepada senario pengujian fungsi bisnes.

No.	Komponen	Keterangan
4.	ID Kes Ujian	ID unik bagi kes ujian yang merujuk kepada kes senario pengujian fungsi bisnes.
5.	Keterangan Kes Ujian	Keterangan ringkas bagi kes ujian.

### b. Penetapan Konvensyen Nama dan Nombor *Traceability Matrix*

Contoh penetapan Konvensyen Nama dan Nombor *Traceability Matrix* adalah seperti Jadual 3-32.

**Jadual 3-32: Penetapan Konvensyen Nama dan Nombor *Traceability Matrix***

ID Fungsi Bisnes	ID User Story	ID Product Backlog	ID Kes Ujian	Keterangan Kes Ujian
BF-BM-EP01	BF-BM-EP01-US01	BF-BM-EP01-PB01	BF-BM-EP01-PB01-TC01	Selenggara Profil Pengguna.
Keterangan Kandungan ID: BF : Ringkasan teknik fungsi bisnes BM : Ringkasan Fungsi Utama Nama Bilik Mesyuarat EP : Ringkasan Epic 01 : Bilangan Epic	Keterangan Kandungan ID: US : Ringkasan User Story 01 : Bilangan User Story	Keterangan Kandungan ID: PB : Ringkasan Product Backlog 01 : Bilangan Product Backlog	Keterangan Kandungan ID: TC : Ringkasan Kes Ujian 01 : Bilangan Kes Ujian	

### c. Pengisian Templat Traceability Matrix

Contoh Templat Kes Pengujian yang telah diisi adalah seperti Jadual 3-33.

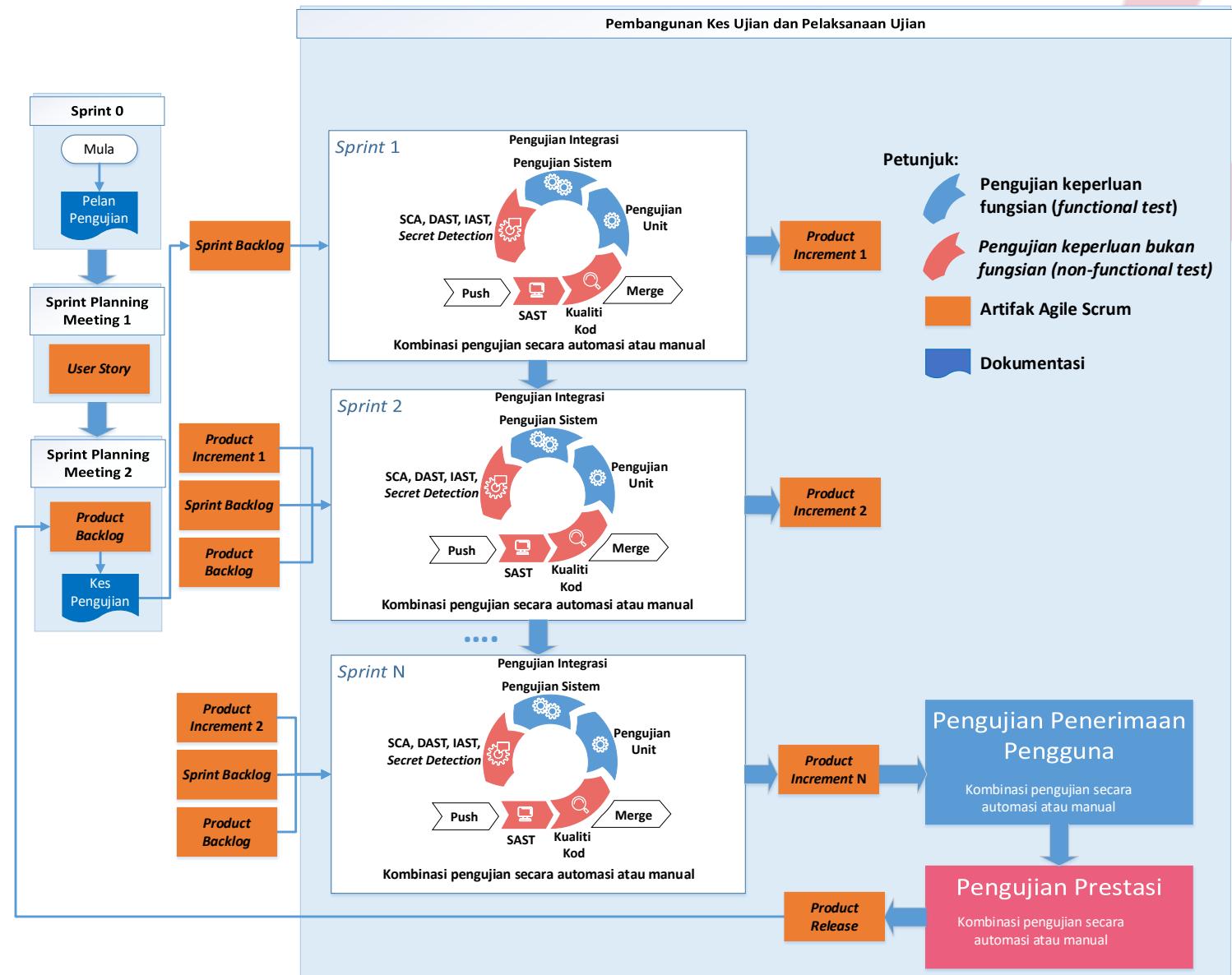
**Jadual 3-33: Contoh Pengisian Templat *Traceability Matrix***

ID Fungsi Bisnes	ID User Story	ID Product Backlog	ID Kes Ujian	Keterangan Kes Ujian
BF-BM-EP01	BF-BM-EP01-US01	BF-BM-EP01-PB01	BF-BM-EP01-PB01-TC01	Mendaftar profil pengguna baharu.
			BF-BM-EP01-PB01-TC02	Memuat naik gambar profil pengguna baharu.
	BF-BM-EP01-US02	BF-BM-EP01-PB02	BF-BM-EP01-PB02-TC01	Pentadbir sistem boleh mengemas kini profil pengguna.
			BF-BM-EP01-PB02-TC02	Pengguna boleh mengemas kini profil sendiri.
BF-BM-EP02	BF-BM-EP02-US01	BF-BM-EP02-PB01	BF-BM-EP02-PB01-TC01	Pentadbir sistem boleh mewujudkan rekod bilik mesyuarat.
			BF-BM-EP02-PB01-TC02	Pentadbir sistem boleh mengemas kini rekod bilik mesyuarat sedia ada.

### 3.3.5.3. Pelaksanaan Aktiviti Pengujian

Aktiviti pengekodan serta pengujian dilakukan secara *incremental* dan *iterative* semasa pembangunan dapat memastikan penghasilan produk yang berkualiti.

Setiap *sprint* menghasilkan *product increment* dan maklum balas yang diperolehi akan menjadi input kepada *sprint* seterusnya. Pelaksanaan aktiviti pengujian secara *iterative* mengikut *sprint* melalui pembangunan produk berdasarkan metodologi *Agile Scrum* adalah seperti Rajah 3-36.



Rajah 3-36: Aktiviti Pengujian Berdasarkan Metodologi Agile Scrum

Aktiviti pengujian berdasarkan *Agile Scrum* diterangkan seperti berikut:

### a. Pembangunan Kes Pengujian

Pembangunan kes pengujian dilaksanakan bersama semasa pembangunan produk pada setiap *sprint*. Reka bentuk kes pengujian memastikan bahawa pengujian kualiti merangkumi keseluruhan keperluan pengguna seperti yang terdapat pada *user story*, *product backlog* dan DoD. Kes pengujian di dokumentasikan pada templat kes pengujian seperti **Lampiran 10: Templat Kes Pengujian**.

Kes pengujian yang didokumenkan kemudiannya diserahkan kepada pengurus produk untuk semakan. Seterusnya pasukan pembangun akan memutuskan kes pengujian yang boleh diautomatiskan.

Berikut merupakan komponen dan contoh pembangunan kes pengujian.

#### i. Penyediaan Templat Kes Pengujian

Kes pengujian dibangunkan berpandu pada Templat Kes Pengujian seperti di **Lampiran 10: Templat Kes Pengujian** dengan komponen seperti berikut:

**Jadual 3-34: Komponen Templat Kes Pengujian**

Komponen	Keterangan
ID Kes Ujian	ID unik bagi kes ujian yang merujuk kepada senario pengujian fungsi bisnes.
Nama Kes Ujian	Nama bagi kes ujian.
Keterangan Kes Ujian	Keterangan ringkas bagi kes ujian.
ID Product Backlog	ID unik bagi <i>product backlog</i> yang merujuk kepada senario pengujian fungsi bisnes.

Komponen	Keterangan
<b>Prasyarat</b>	Prasyarat bagi membolehkan kes ujian dilaksanakan.
<b>Input/ Langkah- Langkah Ujian</b>	Turutan langkah-langkah dalam melaksanakan kes ujian.
<b>Jangkaan Hasil</b>	Jangkaan hasil bagi kes ujian.
<b>Hasil Sebenar</b>	Hasil sebenar yang diperolehi setelah kes ujian dilaksanakan.
<b>Status</b>	Keputusan/status kes ujian sama ada lulus atau gagal.

### iii. Contoh Pengisian Templat Kes Pengujian

Contoh Templat Kes Pengujian yang telah diisi adalah seperti Jadual 3-35.

**Jadual 3-35: Contoh Pengisian Templat Kes Pengujian**

<b>ID Kes Ujian</b>	BF-BM-EP01-PB01-TC01		
<b>Nama Kes Ujian</b>	Mendaftar profil pengguna baharu.		
<b>Keterangan Kes Ujian</b>	Sebagai Pengguna (warga JDN), saya BOLEH mendaftar profil pengguna baharu SUPAYA pengguna boleh mendaftar masuk sistem tempahan bilik mesyuarat.		
<b>ID Product Backlog</b>	BF-BM-EP01-PB01		
<b>Prasyarat</b>	Nama pengguna dan alamat email yang sah untuk didaftarkan.		
<b>Input/ Langkah-Langkah Ujian</b>	<b>Jangkaan Hasil</b>	<b>Hasil Sebenar</b>	<b>Status (Lulus/Gagal)</b>
1. Daftar nama dan email pengguna.	<ul style="list-style-type: none"> <li>a. Antara muka daftar pengguna baharu dipaparkan.</li> <li>b. Sistem akan memaparkan mesej “Pendaftaran pengguna baharu berjaya”.</li> </ul>		

### b. Pelaksanaan Aktiviti Pengujian

Semasa pembangunan produk berdasarkan *product backlog* pada setiap *sprint*, pengujian secara berulang dilakukan untuk mengenal pasti ralat pada peringkat awal supaya pembetulan dapat dilakukan dengan segera.

Aktiviti pengujian boleh dipecahkan kepada pelaksanaan dalam atau luar *sprint*. Walau bagaimanapun, aktiviti pada luar *sprint* boleh dilaksanakan dalam *sprint* menggunakan *tools* automasi dan disahkan pada *sprint review*.

#### a. Pengujian dalam *sprint* – pengujian yang dilakukan secara berulang dalam *sprint* adalah seperti berikut:

- i. Pengujian SAST.
- ii. Pengujian SCA.
- iii. Pengujian DAST.
- iv. Pengujian IAST.
- v. Pengujian *Secret Detection*.
- vi. Kawalan Kualiti Kod.
- vii. Pengujian Unit.
- viii. Pengujian Integrasi.
- ix. Pengujian Sistem.

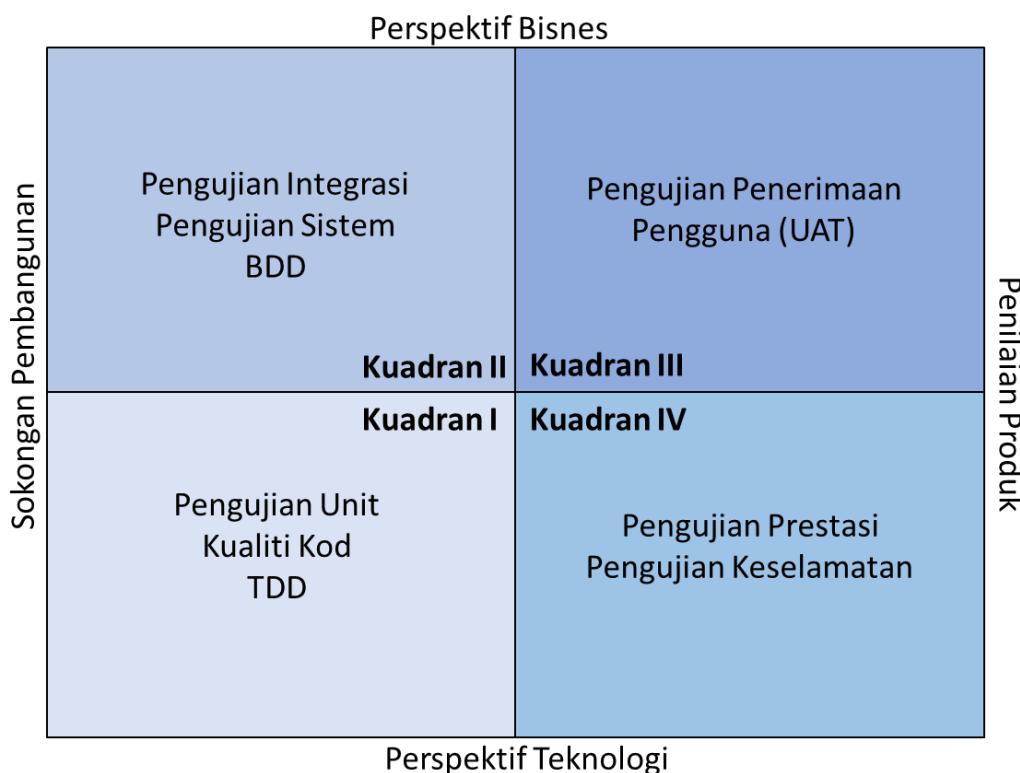
#### b. Pengujian luar *sprint* – pengujian yang dilakukan luar *sprint* adalah seperti berikut:

- i. Pengujian Penerimaan Pengguna.
- ii. Pengujian Prestasi.

### 3.3.5.4. Pengujian Berterusan

Konsep pengujian berterusan adalah untuk memastikan pengujian dilaksanakan di setiap peringkat pembangunan produk bagi mengurangkan risiko kegagalan dalam menepati keperluan pengguna. Pengujian automatik boleh dipraktikkan bagi pelepasan produk yang berkualiti tinggi secara berterusan. Strategi pengujian berterusan perlu mempertimbangkan perkara-perkara seperti kekerapan pelepasan, ketersediaan *tools* ujian automatik, kematangan produk, data ujian dan persekitaran ujian.

Terdapat dua teknik bagi melaksanakan pengujian automatik dengan teknik *Test-Driven Development* (TDD) dan *Behaviour-Driven Development* (BDD) menurut DASA. Penggunaan salah satu atau gabungan kedua-dua teknik ini digunakan dalam pelaksanaan DevOps bagi pengujian automatik. Perbandingan kedua-dua teknik ujian ini adalah seperti di **Lampiran 19: Perbandingan Teknik Pengujian Automatik TDD dan BDD**.



**Rajah 3-37: Kuadran Pengujian Agile**

Kuadran Pengujian *Agile*<sup>17</sup> pada Rajah 3-37 menyediakan perwakilan visual pelbagai aktiviti pengujian yang diperlukan dalam projek *agile* untuk membantu pasukan merancang pengujian secara menyeluruh serta memastikan pendekatan seimbang yang merangkumi perspektif perniagaan dan teknologi. Kuadran dibahagikan kepada empat kategori yang mewakili aspek pengujian berbeza. Kuadran ini juga berfungsi sebagai panduan untuk memahami dan merancang jenis pengujian serta memastikan liputan pengujian komprehensif. Penerangan berkaitan setiap kuadran adalah seperti berikut:

**a. Kuadran I – Pengujian dari segi perspektif teknologi sebagai sokongan pembangunan**

Pengujian automasi yang dilakukan untuk menguji kod unit, komponen dan API untuk memastikan keupayaan komunikasi antara komponen yang berbeza. Matlamat pengujian ini adalah untuk memberikan maklum balas dengan cepat kepada pembangun berkaitan kualiti kod secara berterusan melalui bantuan *tools* automasi.

**b. Kuadran II – Pengujian dari perspektif bisnes sebagai sokongan dan panduan pembangunan**

Pengujian yang berorientasikan pengesahan terhadap keperluan sistem aplikasi yang dibangunkan bagi memenuhi spesifikasi keperluan bisnes dan pengguna seperti yang telah ditetapkan oleh pengurus produk.

**c. Kuadran III – Pengujian dari perspektif bisnes untuk menganalisis dan menilai produk**

Pengujian yang dijalankan untuk menganalisis dan menilai produk serta meneroka tingkah laku produk dalam pelbagai senario. Matlamat pengujian ini adalah untuk mendedahkan isu-isu yang berpotensi

---

<sup>17</sup> Black, R., Coleman, G., Walsh, M. K., Cornanger, B., Forgács, I., Kakkonen, K., & Sabak, J. (2017). *Agile Testing Foundations: An ISTQB Foundation Level Agile Tester guide*.

wujud dan keperluan fungsian yang mungkin tidak memenuhi spesifikasi keperluan bisnes dan pengguna.

**d. Kuadran IV – Pengujian dari perspektif teknologi untuk menganalisis dan menilai produk**

Pengujian yang menumpukan kepada menilai aspek bukan fungsian sistem aplikasi seperti prestasi dan keselamatan. Matlamat pengujian ini bagi mengenal pasti dan mengurangkan risiko berkaitan dengan aspek teknikal sistem aplikasi yang boleh mempengaruhi kualiti dan kebolehpercayaan secara keseluruhan.

Jadual 3-36 menerangkan gabungan TDD, BDD dan Kuadran Pengujian *Agile* yang memainkan peranan penting dalam mencapai pengujian berterusan yang berkesan.

**Jadual 3-36: Konsep dan Teknik Pengujian Berterusan**

Test-Driven Development (TDD)	Behaviour-Driven Development (BDD)	Kuadran Pengujian Agile
<ul style="list-style-type: none"> <li>i. TDD membentuk asas pengujian berterusan dengan memastikan pembangun menulis kod pengujian automatik sebagai sebahagian dari pengekodan.</li> <li>ii. Pengujian ini mengesahkan ketepatan kod dengan setiap perubahan.</li> </ul>	<ul style="list-style-type: none"> <li>i. BDD ialah lanjutan daripada TDD yang memfokuskan pada kerjasama antara pembangun, penguji dan pihak berkepentingan bukan teknikal.</li> <li>ii. Menggunakan spesifikasi bahasa semula jadi (Gherkin syntax) untuk menerangkan sifat sistem aplikasi yang diperlukan daripada</li> </ul>	<ul style="list-style-type: none"> <li>i. Kuadran Pengujian <i>Agile</i> membantu pasukan mengenal pasti dan mengatur pelbagai jenis pengujian untuk memastikan liputan komprehensif.</li> <li>ii. Pengujian Kuadran I dan II menyokong pembangunan dan penyepaduan.</li> <li>iii. Kuadran III dan IV meliputi keperluan bisnes dan</li> </ul>

<b>Test-Driven Development (TDD)</b>	<b>Behaviour-Driven Development (BDD)</b>	<b>Kuadran Pengujian Agile</b>
	<p>perspektif pengguna akhir.</p> <p>iii. Pengujian ini disepadukan dalam <i>pipeline CI/CD</i> sebagai pengujian berterusan.</p>	<p>pengalaman pengguna.</p> <p>iv. Pengujian ini disepadukan dalam proses pengujian berterusan.</p>

Dengan menggabungkan TDD, BDD dan Kuadran Pengujian *Agile*, pasukan boleh mewujudkan strategi pengujian berterusan yang mantap serta merangkumi aspek teknikal, sejajar dengan matlamat bisnes dan memudahkan kerjasama merentas peranan yang berbeza dalam persekitaran DevOps. Pendekatan ini meningkatkan kualiti perisian, mempercepatkan pusingan dapatan maklum balas, dan menyokong penghantaran berterusan produk sistem aplikasi yang boleh berkualiti.

### 3.3.6. Peringkat Pelepasan

Peringkat pelepasan adalah proses untuk mengeluarkan versi baru produk kepada persekitaran penempatan. Pelepasan perubahan kod ke persekitaran penempatan boleh dilaksanakan secara automatik dalam *pipeline* CI/CD atau manual. Pelepasan produk ke persekitaran produksi telah ditetapkan berdasarkan hasil persetujuan bersama pengurus produk dan ahli pasukan DevOps. Pada peringkat pelepasan, penetapan versi perlu dilaksanakan untuk digunakan pada peringkat penempatan.

Peringkat pelepasan melibatkan tiga persekitaran utama iaitu:

- a. Pelepasan ke persekitaran pembangunan,
- b. Pelepasan ke persekitaran *staging* dan
- c. Pelepasan ke persekitaran produksi.

Jadual 3-37 adalah contoh pelan pelepasan di mana tempoh pelepasan merujuk kepada tempoh masa manakala versi produk merujuk kepada versi pelepasan produk. Objektif produk pula merujuk kepada objektif produk yang akan dihasilkan dan ciri-ciri produk merujuk kepada ciri-ciri produk untuk mencapai objektif produk yang ditetapkan. Metrik produk merujuk kepada pengukuran kejayaan produk, contohnya bilangan pengguna yang melayari portal agensi (*traffic*), bilangan pengguna yang menggunakan *chatbot* dan masa penggunaan *chatbot*.

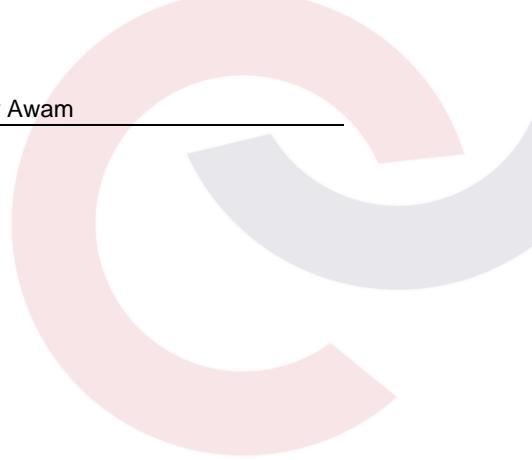
### Jadual 3-37: Contoh Pelan Pelepasan

Tempoh Pelepasan	Versi Produk	Objektif Produk	Ciri-Ciri Produk	Metrik Produk
<b>Q1 2022</b>	Pelepasan 1 (versi 1.0)	Objektif 1	Ciri-Ciri 1	Metrik 1
		Objektif 2	Ciri-Ciri 2	Metrik 2 Metrik 3
<b>Q2 2022</b>	Pelepasan 2 (versi 1.1)	Objektif 1	Ciri-Ciri 1	Metrik 1 Metrik 2
<b>Q3 2022</b>	Pelepasan 3 (versi 1.2)	Objektif 3	Ciri-Ciri 3	Metrik 1
			Ciri-Ciri 4	Metrik 2
			Ciri-Ciri 5	Metrik 3
<b>Q4 2022</b>	Pelepasan 4 (versi 1.3)	Objektif 4	Ciri-Ciri 4	Metrik 1
			Ciri-Ciri 6	Metrik 2
<b>Q1 2023</b>	Pelepasan 5 (versi 2.0)	Objektif n Objektif 2	Ciri-Ciri n	Metrik 1 Metrik n

Scenario pelan pelepasan yang merujuk kepada

Jadual 3-37 adalah contoh pelan pelepasan di mana tempoh pelepasan merujuk kepada tempoh masa manakala versi produk merujuk kepada versi pelepasan produk. Objektif produk pula merujuk kepada objektif produk yang akan dihasilkan dan ciri-ciri produk merujuk kepada ciri-ciri produk untuk mencapai objektif produk yang ditetapkan. Metrik produk merujuk kepada pengukuran kejayaan produk, contohnya bilangan pengguna yang melayari portal agensi (traffic), bilangan pengguna yang menggunakan chatbot dan masa penggunaan chatbot.

Penerangan pada



Jadual 3-37 adalah seperti berikut:

- a. Suku pertama tahun 2022 (Q1 2022),
- b. pelepasan 1 bagi produk versi 1.0 akan dihasilkan dengan dua objektif iaitu objektif 1 dan objektif 2,
- c. ciri-ciri produk yang perlu dihasilkan adalah ciri-ciri 1 bagi objektif 1 dan ciri-ciri 2 bagi objektif 2,
- d. metrik produk yang ditetapkan adalah metrik 1, metrik 2 dan metrik 3,
- e. pada suku kedua tahun 2022 (Q2 2022),
- f. pelepasan 2 bagi produk versi 1.1 akan dihasilkan bagi memenuhi objektif 1 dan ciri-ciri 1, dan
- g. metrik produk yang ditetapkan iaitu metrik 1 dan metrik 2.

### **3.3.6.1. Pelepasan Sistem Aplikasi**

Sebelum proses pelepasan dilaksanakan, pengurus produk perlu membuat penerimaan berdasarkan *acceptance criteria*. *Acceptance criteria* juga dikenali sebagai DoD kerana produk yang dihasilkan telah memenuhi skop dan keperluan yang telah ditetapkan.

Pelepasan secara automatik adalah untuk memastikan produk yang dibangunkan sentiasa tersedia bagi peringkat penempatan. Penempatan ke persekitaran produksi adalah kesinambungan daripada proses pelepasan berterusan.

Proses kelulusan pelepasan secara berperingkat, sama ada automatik atau manual, perlu dirancang oleh pasukan. Setiap proses dan tugas mempunyai *exit criteria* yang jelas. Ini menjadikan *pipeline CI/CD* terkawal dan membolehkan pelepasan secara berterusan yang lebih berkualiti.<sup>18</sup>

---

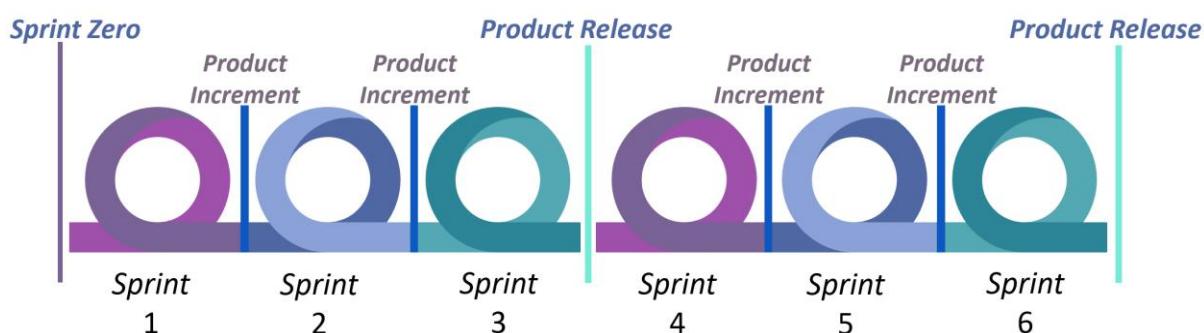
<sup>18</sup> Government Accountability Office. (2020). Agile Assessment Guide : Best Practices for Agile Adoption and Implementation, (GAO Publication No. 20-590G). Washington, D.C.: U.S. Government Printing Office.

Persekuturan-persekuturan yang diguna pakai seperti pembangunan, pengujian dan produksi memerlukan konfigurasi infrastruktur yang ditadbir sepenuhnya menggunakan *tools* konfigurasi CI/CD. Teknologi yang diguna pakai adalah *Infrastructure as a Code* (IaC), *containers* dan *microservices*. *Tools* orkestrasi DevOps mengurus dan menyelaras teknologi-teknologi<sup>19</sup> ini, antaranya seperti Terraform, Ansible dan Puppet.

### 3.3.6.2. *Product Release*

*Product release* adalah *product increment* yang terdiri dari beberapa *sprint* di mana produk direka, dibangunkan, diuji dan ditempatkan seperti Rajah 3-38.

*Product increment* yang terdiri dari setiap *sprint* 1 hingga *sprint* 3 dan dari setiap *sprint* 4 hingga *sprint* 6 akan dilepaskan kepada pengguna sebagai *product release*. Penetapan aktiviti ini boleh berubah mengikut kesesuaian produk dan hasil perbincangan bersama ahli pasukan scrum.



**Rajah 3-38: Aktiviti Sprint dan Product Release**

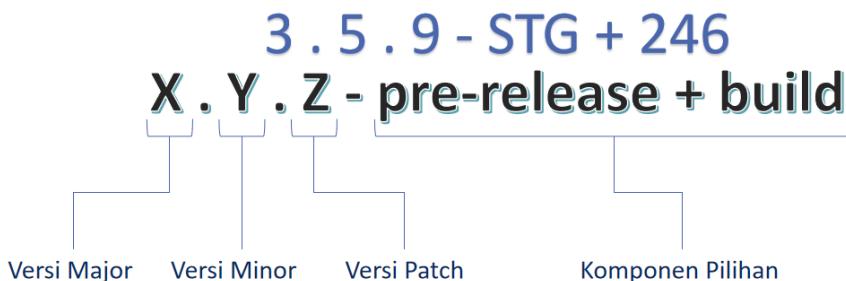
Pelepasan membolehkan *snapshot* sistem aplikasi disediakan untuk pengguna. Antara perkara yang dijana bagi aktiviti ini adalah seperti berikut:

<sup>19</sup> Bahadori, K., & Vardanega, T. (2018, March). DevOps meets dynamic orchestration. In *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment* (pp. 142-154). Springer, Cham.

- a. *Snapshot* kod sumber.
- b. *Generic packages*.
- c. Fail metadata yang berkaitan dengan versi pelepasan.
- d. Nota pelepasan (release note).

### 3.3.6.3. Penetapan Versi *Release*

Pelepasan sistem aplikasi ke setiap persekitaran memerlukan penetapan versi untuk menyelaras penempatan, penambahbaikan dan integrasi antara sistem dengan lebih efisien. Kaedah yang *Semantic Versioning* atau lebih dikenali sebagai SemVer merupakan antara kaedah yang biasa digunakan dalam penetapan versi pada sistem aplikasi.



**Rajah 3-39: Semantic Versioning**

Penggunaan SemVer merangkumi tiga komponen teras, ditulis dalam bentuk  $x.y.z$  dimana  $x,y$  dan  $z$  adalah nombor bulat. Manakala komponen pilihan terdiri daripada *pre-release* dan *build*.

Jadual 3-38 akan menerangkan penggunaan SemVer berdasarkan Rajah 3-39.

### Jadual 3-38: Jadual Penerangan *Semantic Versioning*

Komponen	Contoh	Penerangan
Versi Major (X)	3.0.0	<ul style="list-style-type: none"> <li>i. X merujuk kepada versi major.</li> <li>ii. Penambahan nombor berlaku apabila terdapat perubahan pada sistem aplikasi yang melibatkan <i>API</i> terjejas.</li> <li>iii. Apabila nombor versi major ditambah, nombor versi minor dan versi <i>patch</i> akan ditetapkan semula kepada sifar.</li> <li>iv. Contoh versi semasa 3.5.9, maka versi seterusnya adalah 4.0.0.</li> </ul>
Versi Minor (Y)	3.5.0	<ul style="list-style-type: none"> <li>i. Y merujuk kepada versi minor.</li> <li>ii. Penambahan nombor berlaku apabila fungsi baru dilepaskan ke persekitaran produksi tanpa melibatkan perubahan pada API.</li> <li>iii. Apabila nombor versi minor ditambah, nombor versi <i>patch</i> akan ditetapkan semula kepada sifar.</li> <li>iv. Contoh versi semasa 3.5.9, maka versi seterusnya adalah 3.6.0.</li> </ul>
Versi Patch (Z)	3.5.9	<ul style="list-style-type: none"> <li>i. Z merujuk kepada versi <i>patch</i>.</li> <li>ii. Penambahan nombor berlaku apabila terdapat pembetulan ralat dan tiada perubahan fungsi pada sistem aplikasi tersebut.</li> <li>iii. Tiada limit nombor bagi penambahan versi <i>patch</i>.</li> <li>iv. Contoh versi semasa 3.5.9, maka versi seterusnya adalah 3.5.10.</li> </ul>
Pre-release	STG	<ul style="list-style-type: none"> <li>i. <i>Pre-release</i> merupakan komponen pilihan dan merujuk kepada persekitaran penempatan bagi tujuan pengujian.</li> <li>ii. Sekiranya pelepasan untuk persekitaran produksi, komponen <i>pre-release</i> tidak perlu dinyatakan.</li> <li>iii. Contoh <b>DEV</b> untuk persekitaran pembangunan dan <b>STG</b> untuk persekitaran <i>staging</i>.</li> </ul>
Build	246	<ul style="list-style-type: none"> <li>i. <i>Build</i> merupakan komponen pilihan dan merujuk kepada <i>Pipeline ID</i> daripada GitLab CI/CD.</li> </ul>

### 3.3.7. Peringkat Penempatan

Penempatan sistem aplikasi secara automasi oleh *pipeline* CI/CD memudahkan pasukan menempatkan versi baharu sistem aplikasi dengan pantas, mengurangkan masa henti dan mengurangkan ralat semasa proses pemasangan imej sistem aplikasi ke persekitaran penempatan.

Perkara yang perlu dikawal bagi kejayaan sesuatu projek DevOps adalah pencapaian kemajuan projek dan kualiti.<sup>20</sup> Kawalan dalam DevOps diintegrasikan dalam *pipeline* CI/CD secara automatik menggunakan *tools* DevOps.

#### 3.3.7.1. Penempatan Sistem Aplikasi

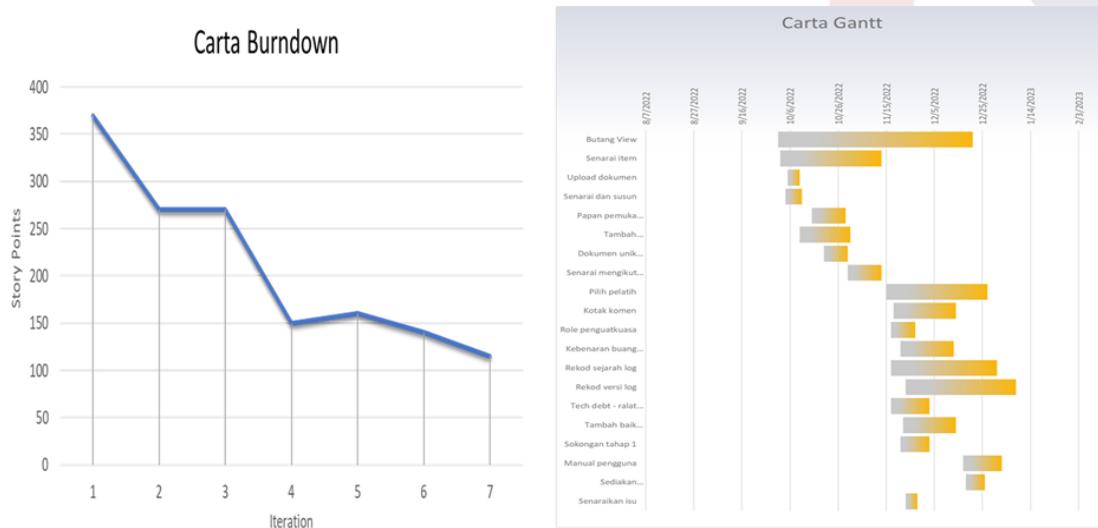
Pengurusan penempatan sistem aplikasi ke pelbagai persekitaran secara automasi dapat dilaksanakan pada *pipeline* CI/CD dengan konfigurasi yang berasingan untuk setiap persekitaran.

#### 3.3.7.2. Kawalan Kemajuan Projek

Carta Burndown digunakan bagi menjelak pencapaian kemajuan projek dalam pelaksanaan *Agile* namun mekanisme seperti *Work Breakdown Structure* (WBS) dan carta Gantt masih boleh digunakan. WBS dalam *Agile* merujuk kepada tuggakan tugas *iteration* yang dijadualkan berdasarkan pelepasan, serta diperhalusi secara iteratif sepanjang pembangunan produk. Contoh carta Burndown dan carta Gantt serta WBS bagi memantau prestasi tuggakan tugas *iteration* adalah seperti di Rajah 3-40 dan Rajah 3-41.

---

<sup>20</sup> Bolhuis, W. T. C. (2021). How Can (Large Scale) Agile be Effectively Adopted and Scaled Up in Dutch Public Sector Organisations (Master's thesis, University of Twente).



Rajah 3-40: Contoh Carta Burndown dan Carta Gantt

Projek ABC								
Nama Projek	Jumlah Tugas		Tempoh Masa		Peratus Selesai			
	Selesai	Dalam Proses	Tarikh Mula	Tarikh Akhir	55%	45%		
mlah Tugas	20							
Selesai	10							
Dalam Proses	2							
Belum Bermula	8							
ITERATION	TUGAS	AHU	STORY POINTS	TARIKH MULA	TARIKH AKHIR	TEMPOH MASA	STATUS	NOTA
Iteration 1	Butang View	Endik M	10	10/1/2022	12/20/2022	81	Selesai	
	Senarai item	Puan F	10	10/2/2022	11/12/2022	42	Selesai	
	Upload dokumen	Endik S	5	10/5/2022	10/9/2022	5	Selesai	
	Senarai dan susun	Endik M	5	10/4/2022	10/10/2022	7	Selesai	
	Papan permuka dinamik	Puan T	40	10/15/2022	10/28/2022	14	Selesai	
	Tambah dokumen baru	Endik S	5	10/10/2022	10/30/2022	21	Selesai	
Iteration 2	Dokumen unik pengguna	Puan T	10	10/20/2022	10/29/2022	10	Dalam Proses	
	Senarai mengikut preference	Endik M	5	10/30/2022	11/12/2022	14	Selesai	
	Pilih pelatih	Endik M	10	11/15/2022	12/26/2022	42	Selesai	
	Kotak komen	Puan F	40	11/18/2022	12/13/2022	26	Dalam Proses	
	Role penguatkuasa	Mizz T	10	11/17/2022	11/26/2022	10	Selesai	
	Kebenaran buang dokumen	Endik S	40	11/21/2022	12/12/2022	22	Belum Bermula	

Rajah 3-41: Contoh WBS

### a. Pelaporan Status Kemajuan Projek

Pengurus produk bertanggungjawab untuk melaporkan status kemajuan projek kepada semua yang terlibat dalam projek berkenaan. Rujuk pada para 3.2.3.1 yang menjelaskan peranan pengurus produk.

Perkara yang perlu dilaporkan ialah seperti berikut:

- i. Status kemajuan aktiviti utama projek berdasarkan kepada skop,
- ii. perincian status kemajuan setiap aktiviti utama, status kemajuan kewangan, senarai permohonan pindaan dengan impak melaksanakan pindaan, laporan mengenai isu dan risiko projek serta cadangan untuk menangani isu dan risiko.

### b. Format Pelaporan Status Kemajuan Projek

Perkara yang perlu dilaporkan pada Format Pelaporan Status Kemajuan Projek seperti di **Lampiran 23: Templat Pelaporan Status Kemajuan Pembangunan Produk** dengan komponen seperti berikut:

**Jadual 3-39: Komponen Pelaporan Status Kemajuan Projek**

Komponen	Keterangan
<b>Ringkasan Status Keseluruhan</b>	Menerangkan senarai status kemajuan aktiviti utama projek berdasarkan kepada skop pembangunan produk.
<b>Release Burn-up Chart</b>	Menerangkan secara carta untuk menjelak status kemajuan pasukan dengan membandingkan kerja yang dirancang berbanding kerja yang telah siap untuk mencapai <i>product release</i> .
<b>Ringkasan Status Kemajuan <i>Product Backlog</i> (User Story)</b>	Menyenaraikan status kemajuan <i>product backlog</i> .
<b>Ringkasan Status Kemajuan <i>Product Backlog</i> (Ralat)</b>	Menyenaraikan status kemajuan <i>product backlog</i> untuk kategori ralat.

### 3.3.8. Peringkat Pengoperasian

Peringkat pengoperasian DevOps menekankan pengautomasi proses penyelenggaraan dalam persekitaran produksi melalui aktiviti berikut:

- a. Penyandaran pangkalan data (database backup).
- b. Pengemasan *container registry* (housekeeping).
- c. Pemulihan sistem aplikasi (system recovery)

Matlamat pengoperasian adalah untuk memastikan infrastruktur sistem aplikasi beroperasi pada tahap optimum semasa pelepasan, *update*, *patch*, peningkatan versi dan penyelenggaraan berjadual seperti pembaharuan lesen dan sijil (certificate).

#### 3.3.8.1. Pengenalan Konsep *Infrastructure as a Code*

IaC adalah pendekatan pengurusan infrastruktur sistem aplikasi secara automasi dengan penyepaduan penggunaan *pipeline* CI/CD dan *tools* yang bersesuaian. Penerapan IaC memberi penekanan kepada aktiviti pengurusan infrastruktur sistem aplikasi yang dilaksanakan secara peningkatan (incremental) dan pengulangan (iterative) seperti pengurusan dan penyelenggaraan sistem pengoperasian, pelayan, pangkalan data dan penempatan sistem aplikasi.

#### 3.3.8.2. Penggunaan IaC dalam DevOps

Penyelenggaraan *tools* DevOps adalah bagi menjamin keselamatan dan kestabilan *tools* yang digunakan dalam pelaksanaan *pipeline* CI/CD. Konsep IaC memudahkan penyelenggaraan *tools* DevOps manakala kawalan versi pula memudahkan penyelenggaraan ke atas kod aplikasi yang dibangunkan. Pengautomasi penyelenggaraan meningkatkan kebolehpercayaan (reliability) dan ketersediaan sistem, mengurangkan masa henti serta

mengukuhkan keselamatan dan tadbir urus. Contoh penggunaan IaC dalam DevOps adalah seperti berikut:

- a. Pengautomasian aktiviti pengurusan dan penyelenggaraan berkala terhadap infrastruktur sistem aplikasi.
- b. Pembangunan persekitaran penempatan dan konfigurasi dilaksanakan dengan lebih konsisten melalui pengautomasian.
- c. Penyeragaman kemas kini versi dan sistem aplikasi merentasi pelbagai persekitaran penempatan dengan lebih efisien.

Aktiviti-aktiviti pengoperasian seperti *restart services*, memindahkan log antara data repositori, dan *scaling-up* atau *scaling-down* infrastruktur boleh di kod atau diskrip dan dilaksanakan mengikut permintaan (on-demand). Penyelenggaraan infrastruktur DevOps dengan seni bina *micro service* hanya melibatkan komponen yang berubah sahaja dan tidak mengganggu komponen lain.

### **3.3.8.3. Penyandaran Pangkalan Data**

Penyandaran merujuk kepada proses membuat salinan data atau fail data untuk digunakan sekiranya data asal atau fail data mengalami kerosakan. Penyandaran bagi panduan ini merujuk kepada aktiviti proses membuat sandaran arkitektur dan data tersimpan bagi pangkalan data. Penyandaran boleh dibuat sama ada secara berkala atau mengikut keperluan agensi.

### **3.3.8.4. Pengemasan *Container Registry***

Pengemasan adalah proses pengoptimuman dan pengemasan sistem yang melibatkan aktiviti seperti pengalihan dan pengemasan ruang storan, fail dan program. Hasil daripada pengemasan memastikan sistem dan storan dapat berfungsi pada keadaan prestasi yang optimum.

Pengemasan imej *container* melalui aktiviti pemadaman imej yang tidak digunakan atau dari versi terdahulu daripada *container registry* bertujuan memastikan penggunaan ruang storan yang optimum. Polisi pemadaman imej yang disarankan adalah mengikut *tools* pengurusan imej yang digunakan atau mengikut keperluan agensi. Pengemasan boleh dibuat sama ada secara berkala atau secara mengikut keperluan agensi.

### **3.3.8.5. Pemulihan Sistem Aplikasi**

Pemulihan sistem aplikasi adalah proses memulihkan sistem aplikasi kepada keadaan operasi asal apabila terjadi kegagalan atau kerosakan. Proses ini memainkan peranan penting untuk memastikan sistem aplikasi memiliki tahap ketersediaan yang tinggi, meskipun terjadi masalah teknikal atau kegagalan sistem.

Proses pemulihan sistem aplikasi adalah seperti berikut:

- a. Pasukan mendapat notifikasi kegagalan sistem melalui *tools* pemantauan.
- b. Proses penempatan semula imej sistem aplikasi dari *container registry* ke persekitaran yang terkesan yang dijalankan secara automasi oleh *pipeline CI/CD*.

### **3.3.9. Peringkat Pemantauan**

Pemantauan adalah proses menjelak, menganalisis dan memaparkan laporan prestasi sistem aplikasi untuk membantu pasukan operasi mengenal pasti dan menyelesaikan isu dengan lebih efisien. Aktiviti pemantauan secara berterusan melalui pengumpulan dan analisis data pemantauan adalah seperti berikut:

- a. Pemantauan sistem
  - i. Pemantauan log.
  - ii. Pemantauan infrastruktur.
  - iii. Pemantauan *uptime*.
- b. Pemantauan prestasi sistem aplikasi
  - i. Pemantauan Prestasi Aplikasi (APM).
  - ii. Pemantauan Pengguna Sebenar (RUM).
  - iii. Pemantauan sintetik.
- c. Perancangan kapasiti (capacity planning).
- d. Pemantauan maklum balas pengguna.

#### **3.3.9.1. Pengenalan Konsep Kebolehperhatian (Observability)**

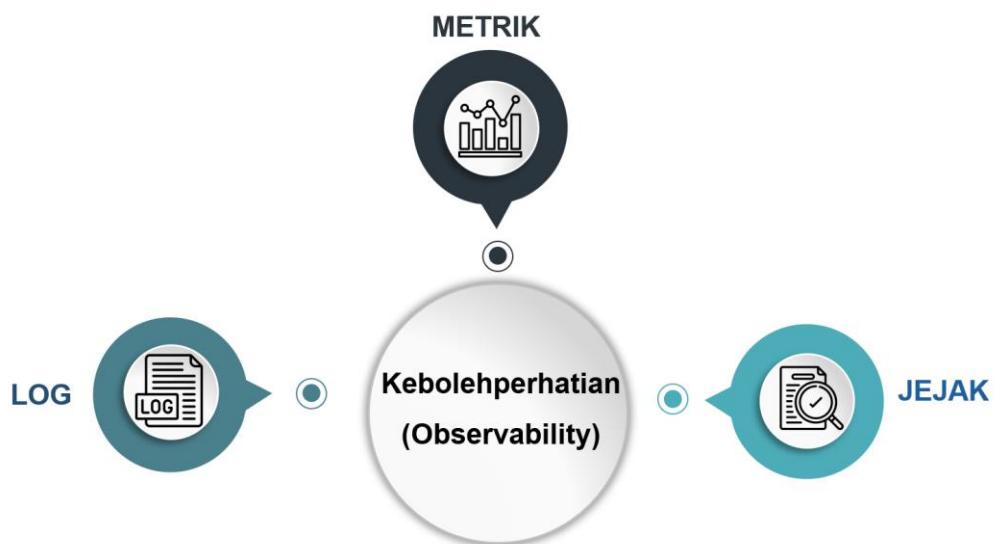
Kebolehperhatian ditakrifkan sebagai bagaimana pasukan membuat tafsiran terhadap keadaan sistem berdasarkan output yang diperoleh daripada aktiviti pemantauan.

Dengan penggunaan arkitektur dan infrastruktur *cloud-native* seperti *microservices* dan *container*, konsep kebolehperhatian membolehkan pasukan melaksanakan aktiviti berikut:

- a. Pemantauan sistem dan infrastruktur yang kompleks dengan lebih berkesan,

- b. Pemantauan dari pelbagai sumber data dapat di satukan untuk memberikan analisis yang lebih lengkap dan
- c. Pemantauan secara automasi dapat ditingkatkan dengan penggunaan *tools* yang bersesuaian.

Terdapat tiga komponen utama yang menyokong konsep kebolehperhatian seperti Rajah 3-42 iaitu log, metrik dan jejak (trace). Tiga komponen ini memberikan data dan analisis yang berbeza berkaitan prestasi sistem. Apabila data dari tiga komponen ini dianalisis bersama, gambaran lengkap berkaitan keadaan sistem, sistem aplikasi dan infrastruktur dapat diperoleh.



**Rajah 3-42: Tiga Komponen yang Menyokong Konsep Kebolehperhatian**

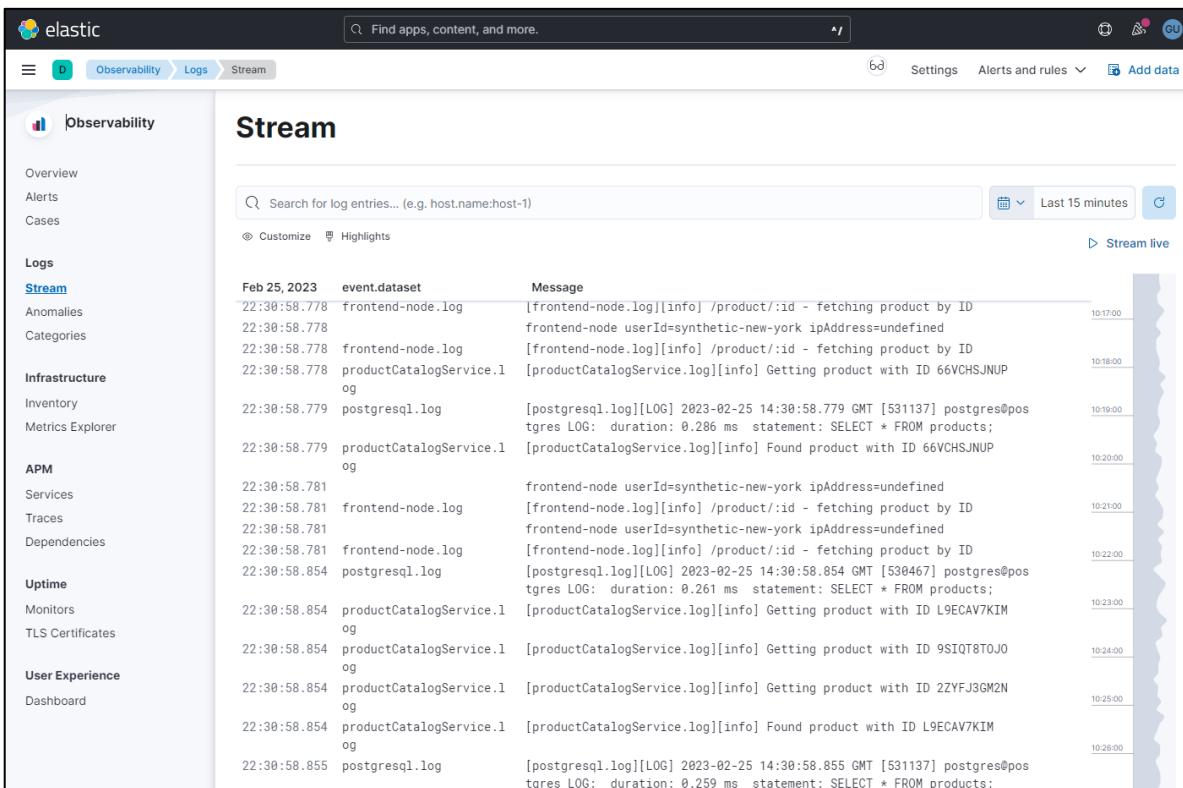
**a. Log**

Log adalah rekod peristiwa yang berlaku pada masa tertentu dan mengandungi maklumat seperti waktu dan peristiwa yang berlaku. Log terdiri daripada format berikut.

### Jadual 3-40 : Format Log

Format Log	Penerangan
<b>Log teks kosong</b>	Mengandungi data berbentuk teks.
<b>Log berstruktur</b>	Mengandungi data tambahan seperti metadata.
<b>Log binari</b>	Mengandungi data berbentuk jujukan bait.

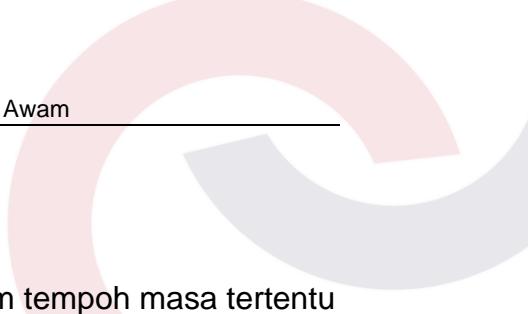
Log yang diperoleh dari pelbagai sumber akan dipaparkan seperti Rajah 3-43 menggunakan aplikasi pemantauan log iaitu Elastic Observability.



The screenshot shows the Elastic Observability Stream interface. The left sidebar has sections for Overview, Alerts, Cases, Logs (Stream selected), Anomalies, Categories, Infrastructure (Inventory, Metrics Explorer), APM (Services, Traces, Dependencies), Uptime (Monitors, TLS Certificates), and User Experience (Dashboard). The main area is titled 'Stream' and shows log entries from February 25, 2023. It includes a search bar, a time range selector set to 'Last 15 minutes', and a 'Stream live' button. The log entries table has columns for Date, event.dataset, and Message. The messages show interactions between frontend-node.log and productCatalogService.log, including product fetches and database queries. The right side of the interface features a vertical timeline with time markers from 10:17:00 to 10:26:00.

Date	event.dataset	Message
Feb 25, 2023	frontend-node.log	[frontend-node.log][info] /product/:id - fetching product by ID
	frontend-node.log	frontend-node userId=synthetic-new-york ipAddress=undefined
	productCatalogService.log	[frontend-node.log][info] /product/:id - fetching product by ID
	productCatalogService.log	[productCatalogService.log][info] Getting product with ID 66VCHSJNUP
	postgresql.log	[postgresql.log][LOG] 2023-02-25 14:30:58.779 GMT [531137] postgres@postgres LOG: duration: 0.286 ms statement: SELECT * FROM products;
	productCatalogService.log	[productCatalogService.log][info] Found product with ID 66VCHSJNUP
	frontend-node.log	frontend-node userId=synthetic-new-york ipAddress=undefined
	frontend-node.log	[frontend-node.log][info] /product/:id - fetching product by ID
	frontend-node.log	frontend-node userId=synthetic-new-york ipAddress=undefined
	frontend-node.log	[frontend-node.log][info] /product/:id - fetching product by ID
	postgresql.log	[postgresql.log][LOG] 2023-02-25 14:30:58.854 GMT [530467] postgres@postgres LOG: duration: 0.261 ms statement: SELECT * FROM products;
	productCatalogService.log	[productCatalogService.log][info] Getting product with ID L9ECAV7KIM
	productCatalogService.log	[productCatalogService.log][info] Getting product with ID 9SIQT8TOJO
	productCatalogService.log	[productCatalogService.log][info] Getting product with ID 2ZYFJ3GM2N
	productCatalogService.log	[productCatalogService.log][info] Found product with ID L9ECAV7KIM
	postgresql.log	[postgresql.log][LOG] 2023-02-25 14:30:58.855 GMT [531137] postgres@postgres LOG: duration: 0.259 ms statement: SELECT * FROM products;

**Rajah 3-43: Paparan Log**

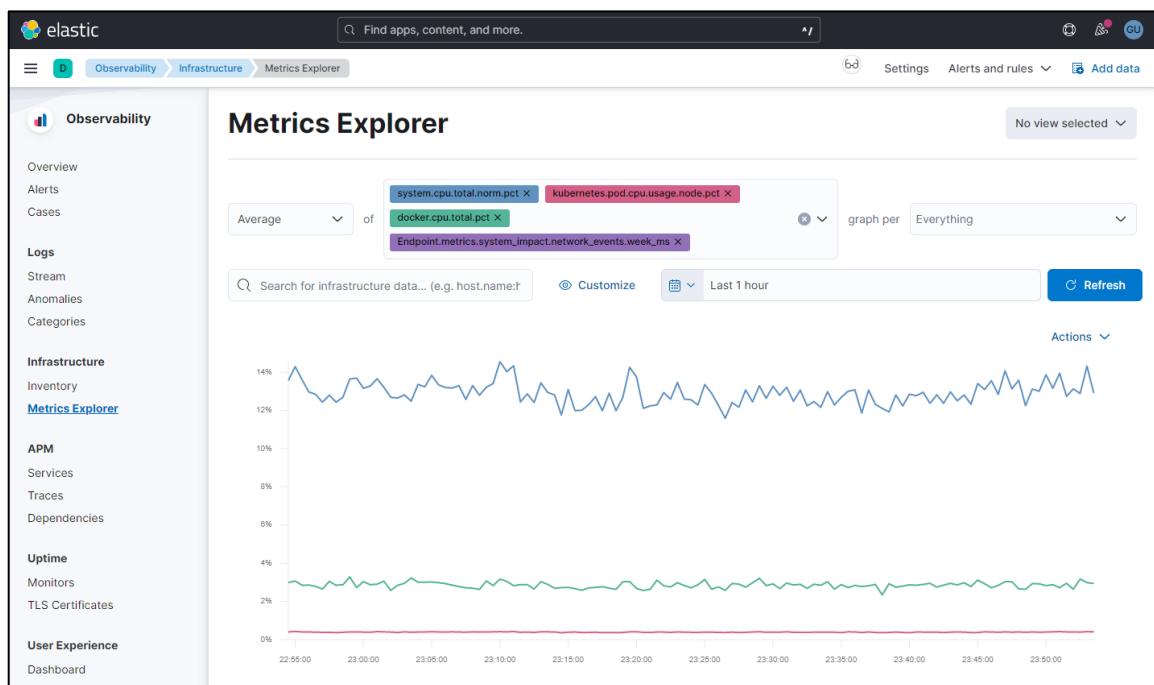


## b. Metrik

Metrik adalah nilai angka berstruktur yang diukur dalam tempoh masa tertentu dan disimpan secara data siri masa. Metrik diperoleh dari prestasi sistem adalah seperti masa operasi sistem, masa tindak balas, bilangan permintaan sesaat dan jumlah kuasa pemprosesan atau memori yang digunakan sistem aplikasi. Pasukan boleh menggunakan pangkalan data siri masa sebagai pelan tindakan masa hadapan dan unjuran untuk memahami perkara yang berlaku pada sistem.

Sebagai contoh, pemantauan terhadap bilangan permintaan sesaat dalam perkhidmatan HTTP menunjukkan peningkatan trafik yang mendadak. Metrik memberikan data secara siri masa dan memaparkan perubahan trend bagi sebarang tindakan susulan untuk mengenal pasti punca lonjakan.

Rajah 3-44 merupakan metrik yang diperoleh daripada aplikasi pemantauan metrik iaitu Elastic Observability.



**Rajah 3-44: Paparan Metrik**

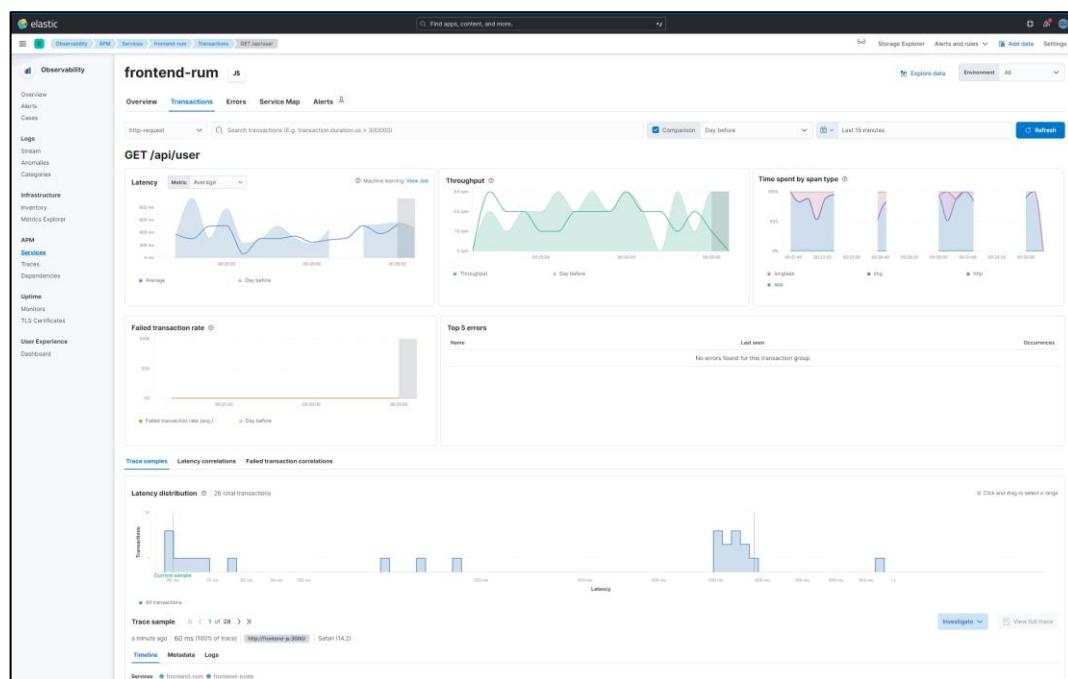
### c. Jejak

Jejak mewakili kitaran perjalanan untuk tindakan atau permintaan semasa merentasi beberapa sistem seperti aplikasi *container* atau arkitektur *microservices*. Analisis terhadap data jejak membolehkan pasukan mengukur tahap prestasi sistem, menentukan punca perubahan prestasi sistem, mengenal pasti dan menyelesaikan isu dengan kadar segera.

Di antara jejak yang boleh digunakan untuk mengenal pasti punca perubahan prestasi sistem adalah seperti berikut:

- i. *API Queries*
- ii. *Server-to-Server Workload*
- iii. *Internal API Calls*
- iv. *Frontend API Traffic*

Rajah 3-45 merupakan data jejak yang diperoleh daripada aplikasi pemantauan jejak iaitu Elastic Observability.



**Rajah 3-45: Paparan Jejak *API Queries***

### 3.3.9.2. Pemantauan Sistem

Pemantauan sistem adalah proses untuk memantau log, penggunaan sumber dan kadar ketersediaan sistem. Aktiviti dalam pemantau sistem adalah seperti berikut:

- a. Pemantauan log,
- b. Pemantauan infrastruktur dan
- c. Pemantauan *uptime*.

#### a. Pemantauan Log

Pemantauan log adalah proses memantau log output sistem untuk menentukan peristiwa yang berlaku pada sistem. Dengan perisian pemantauan log, pasukan boleh mengumpul maklumat, mendapat notifikasi dan melaksanakan tindakan penyelesaian berdasarkan log yang diperolehi.

Contoh log yang boleh dipantau adalah seperti berikut:

- i. Log *container* seperti Docker dan Kubernetes,
- ii. Log pangkalan data seperti MySQL dan PostgreSQL,
- iii. Log web server seperti Nginx dan Apache Tomcat,
- iv. Log sistem pengoperasian seperti Linux dan Windows dan
- v. Log dari pelbagai sumber sistem lain.

#### b. Pemantauan Infrastruktur

Pemantauan infrastruktur adalah proses untuk mengumpul data keadaan dan prestasi dari server, mesin maya, *container*, pangkalan data dan komponen *backend* yang lain.

### i. Metrik Pemantauan Infrastruktur

**Jadual 3-41: Metrik Pemantauan Infrastruktur**

Perkara	Penerangan
Penggunaan CPU	Menunjukkan peratusan kadar penggunaan CPU yang memberi kesan kepada prestasi.
Penggunaan Memori	Menunjukkan peratusan saiz memori yang digunakan semasa program dijalankan.
Penggunaan Storan	Menunjukkan peratusan saiz storan yang digunakan untuk menyimpan fail, gambar dan kandungan yang lain.

### c. Pemantauan *Uptime*

Pemantauan *uptime* adalah pemeriksaan berkala untuk melihat ketersediaan perkhidmatan seperti laman web atau sistem aplikasi. Apabila perkhidmatan mengalami gangguan (downtime), pemantauan *uptime* mengesan isu dan memberi notifikasi untuk tindakan susulan.

### i. Metrik Pemantauan *Uptime*

Aktiviti pemantauan *uptime* adalah berdasarkan metrik berikut.

**Jadual 3-42: Metrik Pemantauan *Uptime***

Perkara	Penerangan
Memantau HTTP	Pemantauan ketersediaan dan tempoh masa tindak balas sistem aplikasi.
Memantau ICMP	Pemantauan kebolehcapaian dan ketersediaan server pada rangkaian.
Memantau TCP	Pemantauan kebolehcapaian dan ketersediaan port perkhidmatan sistem aplikasi.

### **3.3.9.3. Pemantauan Prestasi Sistem Aplikasi**

Pemantauan prestasi sistem aplikasi (APM) adalah proses mengumpul data untuk membantu pasukan mengesan ralat, memantau penggunaan sumber dan mengesan perubahan prestasi yang berlaku dalam sistem aplikasi seterusnya memberi kesan terhadap pengalaman pengguna.

#### **a. Pemantauan Prestasi Aplikasi**

Pemantauan prestasi aplikasi melibatkan pemerhatian perkhidmatan perisian dan sistem aplikasi pada masa nyata dengan mengumpulkan maklumat terperinci berkaitan prestasi mengenai masa tindak balas untuk permintaan masuk, transaksi permintaan (query) pangkalan data dan banyak lagi.

Pemantauan prestasi aplikasi memfokuskan pada pemantauan lima komponen utama berikut terhadap prestasi aplikasi.

- i. *Runtime* seni bina aplikasi.
- ii. Pemantauan pengguna sebenar.
- iii. Transaksi aplikasi.
- iv. Pemantauan komponen.
- v. Analisis dan pelaporan.

### b. Pemantauan Pengguna Sebenar (RUM)

Pemantauan pengguna sebenar (RUM) adalah pemantauan pasif yang merekodkan semua interaksi pengguna terhadap sistem aplikasi berasaskan laman web. *Tools* pemantauan akan mengumpul data berkaitan maklumat berikut:

- i. Peranti pengguna.
- ii. Versi pelayar.
- iii. Tindakan yang diambil oleh pengguna.
- iv. Maklum balas daripada sistem aplikasi.
- v. Pengenalan unik pengguna.

### c. Pemantauan Sintetik

Pemantauan sintetik adalah kaedah mensimulasikan interaksi pengguna terhadap penggunaan sistem aplikasi untuk memberikan maklumat tambahan berkaitan prestasi sistem aplikasi berdasarkan senario yang ditetapkan.

Semasa RUM, sebarang kemerosotan sistem aplikasi hanya dapat diperoleh apabila digunakan oleh pengguna sebenar. Masalah kekurangan data transaksi juga akan menjadi kekangan dalam mengenal pasti masalah sistem aplikasi pada peringkat awal.

Mewujudkan pemantauan sintetik sistem aplikasi dalam persekitaran yang ditetapkan dan mensimulasikan tindakan pengguna boleh membantu mengatasi kelemahan RUM seterusnya mengenal pasti sebarang permasalahan pada peringkat awal sebelum sampai kepada pengguna.

Contoh metrik pemantauan prestasi aplikasi, infrastruktur dan rangkaian adalah seperti di **Lampiran 20: Metrik Pemantauan DevOps**.

### 3.3.9.4. Perancangan Kapasiti Sistem (Capacity Planning)

Perancangan kapasiti bertujuan memastikan perkhidmatan sistem aplikasi dan kapasiti infrastruktur dapat menyokong penyampaian perkhidmatan berdasarkan sasaran tahap perkhidmatan yang telah ditetapkan. Selain itu, perancangan kapasiti boleh digunakan sebagai asas perancangan untuk permohonan sumber tambahan yang diperlukan untuk menyokong unjuran pertumbuhan dan pengoptimuman sistem aplikasi dan infrastruktur pada masa hadapan.

Templat berikut menyediakan panduan dan maklumat yang digunakan untuk merancang keperluan kapasiti.

#### a. Penyediaan Templat Perancangan Kapasiti

Membangunkan perancangan kapasiti dengan berpandukan maklumat-maklumat seperti Jadual 3-43.

**Jadual 3-43: Kandungan Templat Perancangan Kapasiti**

No.	Item	Keterangan
1.	Perkhidmatan/Item Dipantau	Penerangan berkenaan perkhidmatan atau item yang dianalisis.
2.	Keperluan Kapasiti	Keperluan dan pengukuran kapasiti.
3.	% Peningkatan Diperlukan Setiap Tahun	Unjuran peningkatan untuk setiap tahun.
4.	Nilai Ambang	Nilai tahap ambang di mana perkhidmatan atau item melepassi had yang ditetapkan.
5.	Pelan Tindakan	Strategi tindak balas kepada had kapasiti.

### b. Analisis Perancangan Kapasiti

Analisis perancangan kapasiti adalah senario yang dianalisis dari segi impak proses bisnes supaya pasukan boleh menentukan keperluan kapasiti dengan optimum.

Berikut digariskan perkara yang menjadi pertimbangan untuk membangunkan analisis perancangan kapasiti terhadap penggunaan sistem aplikasi.

- i. Keperluan kapasiti CPU.
- ii. Keperluan memori.
- iii. Kapasiti rangkaian.
- iv. Keperluan kapasiti storan.

### c. Pengisian Templat Perancangan Kapasiti

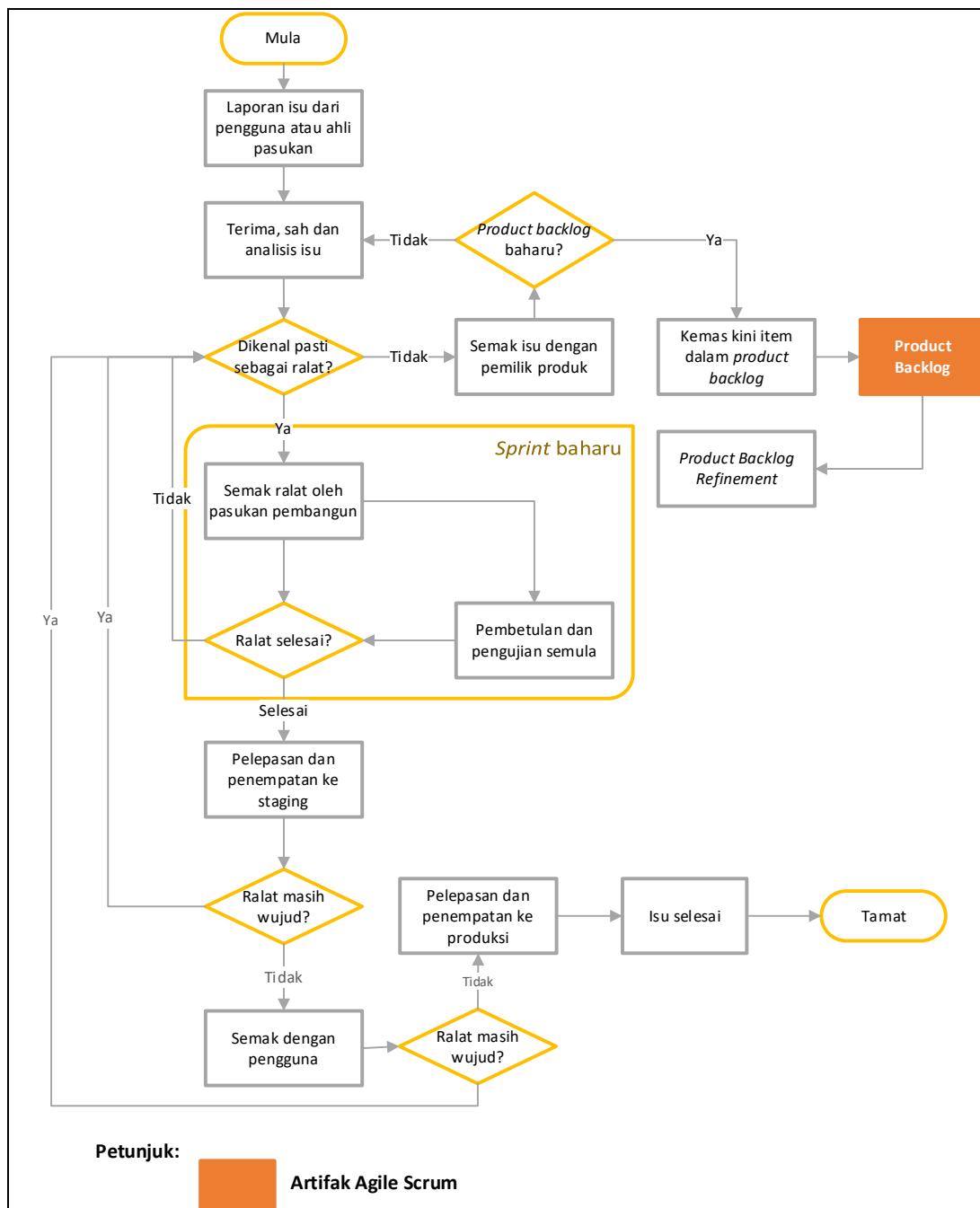
Contoh Templat Perancangan Kapasiti yang telah diisi adalah seperti Jadual 3-44.

**Jadual 3-44: Contoh Pengisian Templat Perancangan Kapasiti**

Perkhidmatan/Item Dipantau	Keperluan Kapasiti	% Peningkatan Diperlukan Setiap Tahun	Nilai Ambang	Pelan Tindakan
Storan Penyimpanan Data Sistem Aplikasi	10 MB per profil pengguna	5%	80%	Memantau peningkatan storan dari Elastic Observability dan juga pelan peluasan sistem aplikasi.

### 3.3.9.5. Pemantauan Maklum Balas Pengguna

Maklum balas pengguna adalah penting dalam membantu meningkatkan kualiti sistem aplikasi. Saluran maklum balas seperti meja bantuan boleh digunakan sebagai medium pengantara untuk pengguna memberikan maklum balas berkaitan sistem aplikasi.



Rajah 3-46: Aliran Proses Pemantauan Maklum Balas Pengguna

Jadual 3-45 adalah penerangan proses dalam aliran proses pemantauan maklum balas pengguna berdasarkan Rajah 3-46.

**Jadual 3-45: Keterangan Aliran Pemantauan Maklum Balas Pengguna**

Proses	Keterangan
<b>Laporan isu dari pengguna atau ahli pasukan.</b>	Pengguna atau ahli pasukan akan melaporkan isu yang ditemui semasa proses penggunaan sistem aplikasi di persekitaran produksi.
<b>Terima, sah dan analisis isu.</b>	Meja bantuan akan mengesahkan penerimaan isu dan mendapatkan maklumat lanjut berkenaan isu yang dilaporkan dari pelapor.
<b>Kenal pasti isu.</b>	Jika isu yang dilaporkan dikenal pasti sebagai permintaan baharu, isu akan dihantar ke pemilik produk untuk pengesahan seterusnya direkodkan sebagai item baharu dalam <i>product backlog</i> .
<b>Semak isu dengan pemilik produk.</b>	Isu yang disahkan sebagai permintaan baru akan dibincangkan secara lanjut semasa sebagai <i>product backlog refinement</i> pada <i>sprint planning meeting</i> . Jika isu tersebut memerlukan penjelasan dan maklumat lanjut dari pelapor, isu akan di ajukan semula ke meja bantuan untuk semakan.
<b>Semak ralat oleh pasukan pembangun.</b>	Jika isu dikenal pasti sebagai ralat, ralat tersebut akan di ajukan ke pasukan pembangun untuk semakan. Pasukan pembangun memasukan ralat yang diterima ke dalam <i>product backlog</i> untuk pembetulan yang akan dilaksanakan dalam <i>sprint</i> ditetapkan.
<b>Pembetulan dan pengujian semula.</b>	Ralat yang telah dikemas kini dan diuji akan dilepaskan serta ditempatkan pada persekitaran <i>staging</i> untuk pengesahan dengan pengguna.
<b>Semak dengan pengguna.</b>	Semakan dengan pengguna dilaksanakan untuk pengesahan penerimaan. Jika ralat masih wujud, semakan semula akan dilakukan untuk mengenal pasti isu sebenar.
<b>Penempatan ke produksi.</b>	<i>Product release</i> akan ditempatkan ke persekitaran produksi setelah pengesahan penerimaan pengguna lulus.

Pasukan DevOps bertanggungjawab mengurus *tools* serta menambah baik proses kerja. Kepercayaan antara sesama sendiri, autonomi, maklum balas secara berterusan, kolaborasi dengan pelbagai pasukan dan berkomunikasi secara terbuka menjadi teras dalam amalan DevOps.

Antara amalan-amalan dalam pelaksanaan DevOps adalah:

- a. Menjadikan pelepasan satu rutin harian,
- b. mengamalkan strategi yang sesuai dalam *branching* kod,
- c. memperuntukkan masa bagi ujian unit dan *code refactoring*,
- d. mengurangkan hutang teknikal secara konsisten,
- e. menggunakan *plug-ins* bagi menjamin konsistensi dan mengelakkan kelemahan kod,
- f. membincangkan maklum balas yang diterima secara bersama, dan
- g. amalan dalam pelaksanaan DevOps yang disenaraikan di atas tidak terhad dan boleh ditambah baik berdasarkan kepentingan dan keperluan agensi.

### 3.4. KOMPONEN 4 (K4): PENGUKURAN KEMATANGAN

Pelaksanaan DevOps boleh diukur berdasarkan kematangan agensi melalui pelbagai metrik dan petunjuk prestasi yang memberi gambaran tentang sejauh mana organisasi telah berjaya mengintegrasikan prinsip-prinsip DevOps ke dalam proses dan budaya mereka. Rajah 3-47 memaparkan aktiviti yang dilaksanakan pada komponen keempat ini.



Rajah 3-47: Pengukuran Kematangan

#### 3.4.1. Penilaian Persekutaran Semasa bagi Ketersediaan Pelaksanaan DevOps

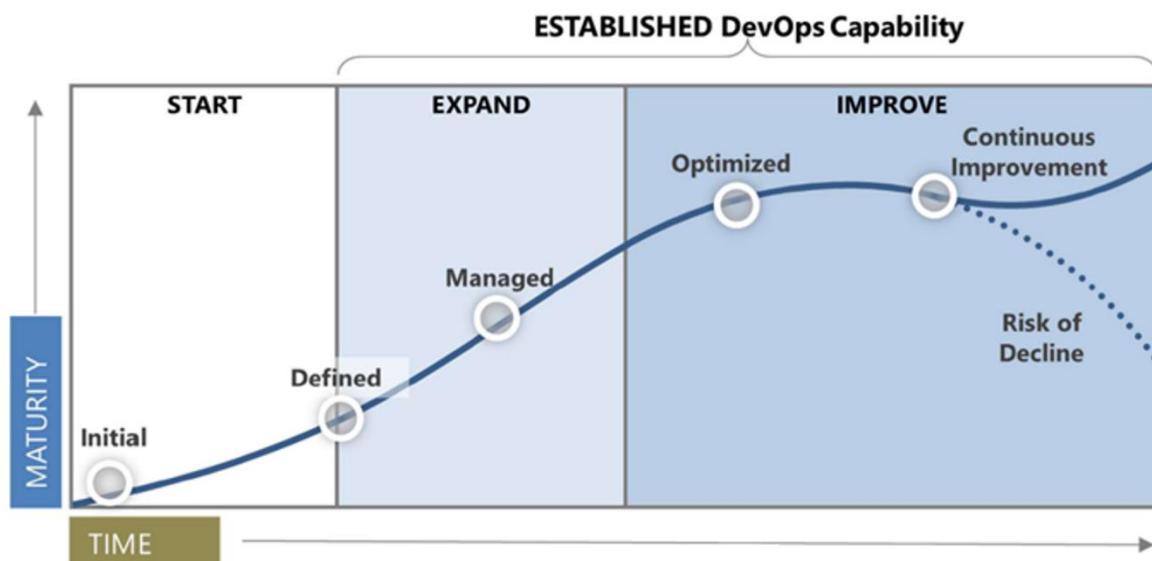
Setelah matlamat dan objektif pelaksanaan DevOps ditetapkan, penilaian terhadap persekitaran semasa perlu dilaksanakan bagi mengenal pasti ketersediaan agensi. Penilaian tahap ketersediaan seperti sumber, *tools* dan infrastruktur menyumbang kepada kejayaan pelaksanaan DevOps.

### 3.4.1.1. Tujuan

Menilai dan mengenal pasti tahap ketersediaan agensi terhadap pelaksanaan DevOps.

### 3.4.1.2. Metodologi Pelaksanaan

Penilaian terhadap persekitaran semasa DevOps boleh dilaksanakan berpandukan kepada Model Keluk Kematangan Keupayaan Agensi yang dikeluarkan oleh American Council for Technology-Industry Advisory Council (ACT-IAC).<sup>21</sup> Model ini boleh digunakan oleh agensi untuk menilai tahap ketersediaan untuk pelaksanaan DevOps. Lima peringkat dalam model ini menunjukkan evolusi ke arah kematangan keupayaan DevOps bermula dari peringkat permulaan, penetapan, pelaksanaan, pengoptimuman sehingga peringkat penambahbaikan berterusan.



Rajah 3-48: Keluk Kematangan Agensi (Sumber: ACT-IAC)

<sup>21</sup> American Council for Technology-Industry Advisory Council (ACT-IAC). (2020). DevOps Primer. Case Studies and Best Practices from Across Government, 11–12.

Rajah 3-48 di atas memaparkan lima peringkat dalam keluk kematangan agensi bagi mengenal pasti tahap ketersediaan pelaksanaan DevOps iaitu:

**a. Peringkat Permulaan (Initial)**

Proses atau aktiviti pembangunan produk yang dilaksanakan dalam peringkat ini tidak mempunyai perancangan dan kawalan yang berkesan. Kaedah atau pendekatan yang digunakan dalam melaksanakan sesuatu tugas tidak efektif dan terdedah kepada risiko serta ketidakcekapan. Dalam peringkat ini juga, tugas berjaya diselesaikan tetapi cenderung ke arah kelewatan dan pembaziran kos.

**b. Peringkat Penetapan (Defined)**

Dalam peringkat ini, segala proses dan aktiviti pembangunan produk yang akan dilaksanakan dirancang dan dikawal dengan terperinci dan berkesan. Penetapan terhadap piawaian, polisi dan skop kerja dilaksanakan sebagai rujukan dan pematuhan dalam melaksanakan sesuatu aktiviti. Dalam peringkat ini juga, keperluan-keperluan bagi melaksanakan sesuatu aktiviti dikenal pasti dan disediakan.

**c. Peringkat Pelaksanaan (Managed)**

Proses dan aktiviti pembangunan produk dalam peringkat ini dilaksanakan dengan menggunakan pendekatan yang berkesan berdasarkan perancangan yang telah dibuat. Dalam peringkat ini, proses atau aktiviti diselesaikan dengan mudah, cepat dan berkesan melalui penggunaan *tools* yang telah dikenal pasti dan disediakan.

**d. Peringkat Pengoptimuman (Optimised)**

Proses dan aktiviti pembangunan produk dilaksanakan sepenuhnya dengan menggunakan pendekatan yang berkesan berdasarkan perancangan yang telah dibuat. Setiap proses atau aktiviti pembangunan sistem aplikasi dilaksanakan dengan mematuhi piawaian dan polisi yang telah ditetapkan. Proses dan aktiviti yang

dilaksanakan dalam peringkat ini, dikawal dan diukur bagi mengenal pasti tahap keberkesanan terhadap pendekatan atau kaedah yang digunakan.

#### e. Peringkat Penambahbaikan Berterusan (Continuous Improvement)

Peringkat ini memberi tumpuan kepada pemantauan berterusan terhadap pelaksanaan sesuatu proses atau aktiviti dalam pembangunan sistem aplikasi. Segala kekurangan atau kelemahan dalam pelaksanaan dikenal pasti, dipantau dan ditambah baik secara berterusan.

Jadual 3-46 menunjukkan model kematangan DevOps yang telah diadaptasikan dari ACT-IAC serta boleh dirujuk bagi mengenal pasti tahap semasa atau tahap yang ingin dicapai dalam pelaksanaan DevOps. Model kematangan DevOps ini dibentuk berdasarkan keluk kematangan agensi seperti di Rajah 3-48, merangkumi lima aspek pelaksanaan iaitu Proses Pembinaan, Pelepasan dan Penempatan, Proses Pengujian, Keselamatan, Infrastruktur, Budaya serta Tadbir Urus.<sup>22</sup>

---

<sup>22</sup> American Council for Technology-Industry Advisory Council (ACT-IAC). (2020). DevOps Primer. Case Studies and Best Practices from Across Government, 11–12.

**Jadual 3-46: Model Kematangan DevOps**

Aspek	Tahap Kematangan DevOps				
	<i>Initial</i>	<i>Defined</i>	<i>Managed</i>	<i>Optimised</i>	<i>Continuous Improvement</i>
<b>1. Pembinaan, Pelepasan dan Penempatan</b>	Melaksanakan aktiviti pembinaan, pelepasan dan penempatan secara manual	<ul style="list-style-type: none"> <li>i. Mengenal pasti dan merancang proses yang boleh diautomasikan, dan</li> <li>ii. mengenal pasti <i>tools</i> integrasi bagi proses automasi</li> </ul>	Melaksanakan sebahagian aktiviti pembinaan, pelepasan dan penempatan secara automasi	Melaksanakan semua aktiviti pembinaan, pelepasan dan penempatan secara automasi	<ul style="list-style-type: none"> <li>i. Melaksanakan semua aktiviti pembinaan, pelepasan dan penempatan secara automasi</li> <li>ii. Melaksanakan strategi pemantauan berterusan terhadap aktiviti pembinaan, pelepasan dan penempatan</li> </ul>

Aspek	Tahap Kematangan DevOps				
	Initial	Defined	Managed	Optimised	Continuous Improvement
<b>2. Pengujian</b>	Melaksanakan pengujian secara manual	<ul style="list-style-type: none"> <li>i. Mengenal pasti dan merancang strategi pengujian,</li> <li>ii. mengenal pasti <i>tools</i> pengujian bagi proses automasi, dan</li> <li>iii. memulakan proses automasi dengan pengujian unit</li> </ul>	<ul style="list-style-type: none"> <li>i. Melaksanakan proses pengujian unit dan integrasi secara automasi, dan</li> <li>ii. melaksanakan analisis kod statik secara automasi</li> </ul>	<ul style="list-style-type: none"> <li>i. Melaksanakan pengujian unit, integrasi, penerimaan, prestasi dan sistem integrasi secara automasi, dan</li> <li>ii. melaksanakan analisis kod statik secara automasi</li> </ul>	Melaksanakan proses pengujian berterusan sebagai sebahagian daripada proses pemantauan berterusan
<b>3. Keselamatan</b>	<ul style="list-style-type: none"> <li>i. Melaksanakan aspek keselamatan di akhir kitar hayat pembangunan sistem aplikasi,</li> <li>ii. melaksanakan pengujian kawalan keselamatan secara manual</li> </ul>	Merancang dan membangunkan polisi dan piawaian keselamatan	Menekankan dan melaksanakan aspek keselamatan di awal kitar hayat pembangunan sistem aplikasi untuk mengenal pasti kelemahan dan tindakan mitigasi dengan lebih cepat	Melaksanakan pengujian kawalan keselamatan secara automasi	Melaksanakan pemantauan berterusan terhadap aspek keselamatan sistem aplikasi

Aspek	Tahap Kematangan DevOps				
	Initial	Defined	Managed	Optimised	Continuous Improvement
<b>4. Infrastruktur</b>	<ul style="list-style-type: none"> <li>i. Membangunkan piawaian penyediaan persekitaran infrastruktur, dan</li> <li>ii. mewujudkan <i>pipeline</i> serahan sistem aplikasi berdasarkan konsep <i>Minimum Viable Product</i> (MVP) yang melibatkan repositori kod sumber, <i>tools continuous integration/continuous delivery</i> (CI/CD) dan <i>tools</i> kolaborasi</li> </ul>	<ul style="list-style-type: none"> <li>i. Melaksanakan integrasi berterusan dan pengautomasi n <i>pipeline</i> serahan sistem aplikasi, dan</li> <li>ii. meningkatkan penggunaan repositori kod sumber berpusat berdasarkan amalan terbaik</li> </ul>	<ul style="list-style-type: none"> <li>i. Mengintegras i <i>tools</i> pembangunan dan penempatan ke persekitaran sedia ada berdasarkan <i>containerization</i>, dan</li> <li>ii. migrasi sistem legasi kepada persekitaran berasaskan seni bina modular (<i>microservices</i>)</li> </ul>	<p>Menyediakan infrastruktur yang boleh melaksanakan proses penskalaan secara automasi berdasarkan keperluan melalui pelaksanaan IaC</p>	<ul style="list-style-type: none"> <li>i. Memperluaskan pelaksanaan infrastruktur di premis (on-premises) ke pengkomputeran awan (cloud computing), dan</li> <li>ii. melaksanakan strategi pemantauan berterusan terhadap persekitaran infrastruktur</li> </ul>

Aspek	Tahap Kematangan DevOps				
	<i>Initial</i>	<i>Defined</i>	<i>Managed</i>	<i>Optimised</i>	<i>Continuous Improvement</i>
<b>5. Budaya dan Tadbir Urus</b>	<ul style="list-style-type: none"> <li>i. Proses membuat keputusan melalui pelbagai peringkat,</li> <li>ii. pasukan pembangun dan pasukan operasi bekerja secara berasingan dalam pembangunan sistem aplikasi, dan</li> <li>iii. terdapat jurang komunikasi di antara pasukan pembangun dan pasukan operasi</li> </ul>	<ul style="list-style-type: none"> <li>i. Merancang dan mengenal pasti strategi peralihan budaya ke arah DevOps,</li> <li>ii. mewujudkan pasukan pengamal DevOps, dan</li> <li>iii. mengenal pasti tools komunikasi dan kolaborasi</li> </ul>	Menggabungkan pasukan pembangun dan pasukan operasi menjadi satu pasukan yang bekerjasama secara komunikatif dan kolaboratif dalam melaksanakan pembangunan sistem aplikasi	Komunikasi yang cekap antara ahli pasukan dengan menggunakan tools komunikasi dan kolaborasi	<ul style="list-style-type: none"> <li>i. Proses membuat keputusan diagihkan kepada ahli pasukan, dan</li> <li>ii. melaksanakan pemantauan berterusan dan mengukur tahap keberkesanan pelaksanaan DevOps</li> </ul>

### 3.4.2. Proses Peralihan

Amalan terbaik peralihan kepada pelaksanaan DevOps adalah secara pendekatan berperingkat seperti Rajah 3-49.<sup>23</sup> Proses peralihan bermula dengan pasukan yang kecil di agensi atau projek tertentu sebelum dilaksanakan secara meluas. Penilaian terhadap pandangan dan maklum balas harus dilakukan pada setiap langkah untuk memastikan peralihan berjalan dengan lancar.



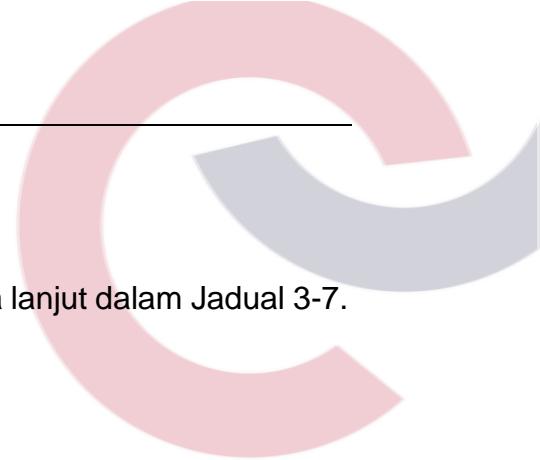
**Rajah 3-49: Proses Peralihan kepada Pelaksanaan DevOps**

#### 3.4.2.1. Tujuan

Memastikan proses peralihan dapat direalisasikan seperti dirancang yang merangkumi kaedah pembangunan sistem aplikasi, *tools* dan teknologi, tadbir urus, budaya, serta pengurusan dalam perlaksanaan DevOps.

---

<sup>23</sup> Muñoz, M., & Rodríguez, M. N. (2021). A guidance to implement or reinforce a DevOps approach in organizations: A case study. Journal of Software: Evolution and Process, e2342.



### 3.4.2.2. Metodologi Pelaksanaan

Metodologi pelaksanaan proses peralihan secara berperingkat seperti Rajah 3-49 dijelaskan secara lanjut dalam Jadual 3-7.

**Jadual 3-47: Metodologi Pelaksanaan Proses Peralihan kepada Amalan DevOps**

Peringkat	Penjelasan	Aktiviti
1. Mengadaptasi Metodologi dalam Pembangunan Produk <b>Agile</b>	Mengadaptasi metodologi <i>Agile</i> ke dalam pembangunan sistem. Kesedaran terhadap manifesto <i>Agile</i> merangkumi budaya, proses kerja dan <i>tools</i> membantu memberikan hala tuju yang jelas terhadap proses peralihan yang dilaksanakan	<ul style="list-style-type: none"> <li>i. Membina budaya dan minda pasukan ke arah metodologi pembangunan berdasarkan <i>Agile</i>,</li> <li>ii. mempelajari dan mengadaptasi budaya DevOps secara berperingkat ke dalam pembangunan sistem aplikasi, dan</li> <li>iii. mewujudkan <i>pipeline</i> bagi serahan sistem aplikasi berdasarkan konsep <i>Minimum Viable Product</i> (MVP) yang melibatkan repositori kod sumber dan <i>tools CI/CD</i>.</li> </ul>
2. Menambah baik dan Menyelaras Proses Pembangunan Produk	Menggabungkan, menambah baik dan menyelaras amalan DevOps, sumber manusia, tadbir urus, <i>tools</i> serta dokumentasi berkaitan dalam penyampaian produk dan perkhidmatan dengan lebih pantas dan efisien	<ul style="list-style-type: none"> <li>i. Memperkenalkan dan menggunakan pakai konsep <i>Agile</i> dalam pasukan,</li> <li>ii. menyelaraskan proses pembangunan produk, dan</li> <li>iii. menambah baik amalan DevOps pembangunan sedia ada seperti:</li> </ul>

Peringkat	Penjelasan	Aktiviti
		<ul style="list-style-type: none"> <li>a. Integrasi berterusan dan pengautomasian <i>pipeline</i> serahan sistem aplikasi,</li> <li>b. Meningkatkan penggunaan repositori kod sumber berpusat berdasarkan amalan terbaik,</li> <li>c. Mengenal pasti proses dokumentasi yang boleh diminimumkan supaya pasukan lebih fokus kepada pembangunan produk, dan</li> <li>d. Merancang dan melaksana proses pelepasan ke persekitaran produksi dan menambah baik proses kerja yang terlibat.</li> </ul>
<b>3. Meningkatkan Pelaksanaan Infrastruktur dan Penggunaan Tools</b>	Memberi tumpuan kepada integrasi infrastruktur dan <i>tools</i> dari segi kebolehfungsian serta mengoptimumkan pelaksanaan infrastruktur dan penggunaan <i>tools</i> bagi meningkatkan kecekapan pembangunan sistem aplikasi	<ul style="list-style-type: none"> <li>i. Mengintegrasikan <i>tools</i> pembangunan dan penempatan melalui pengaplikasian pendekatan <i>containerization</i>,</li> <li>ii. migrasi persekitaran sistem legasi kepada persekitaran infrastruktur berdasarkan DevOps, dan</li> <li>iii. mengoptimumkan penggunaan <i>tools</i> dan infrastruktur DevOps dalam proses pembangunan dan operasi ke persekitaran produksi.</li> </ul>
<b>4. Mengoperasikan Pipeline DevOps</b>	Mengoperasikan <i>pipeline</i> DevOps pada tahap yang optimum bagi menyokong kadar penempatan yang lebih kerap dan berterusan. Dengan pendekatan ini, ralat dalam setiap peringkat kitar hayat	<ul style="list-style-type: none"> <li>i. Mengintegrasikan <i>tools</i> CI/CD dengan <i>tools</i> perisian pembangunan sedia ada atau <i>tools</i> perisian pihak ketiga,</li> </ul>

Peringkat	Penjelasan	Aktiviti
	pembangunan produk dapat dikenal pasti dengan lebih awal	<ul style="list-style-type: none"> <li>ii. mengautomasi persekitaran infrastruktur melalui pelaksanaan IaC, dan</li> <li>iii. menyesuaikan proses kerja berdasarkan prinsip DevOps.</li> </ul>
<b>5. Mengoptimum dan Memperluaskan Pelaksanaan DevOps</b>	Memberi tumpuan kepada mengoptimumkan dan memperluaskan pelaksanaan DevOps serta menggalakkan ahli pasukan untuk mempertingkatkan tahap pelaksanaan DevOps	<ul style="list-style-type: none"> <li>i. Melaksanakan seminar, bengkel dan program berkaitan DevOps bagi memberi kesedaran tentang faedah yang diperolehi dari pelaksanaan DevOps di agensi serta meningkatkan pengetahuan kompetensi pasukan,</li> <li>ii. melaksanakan sesi perkongsian teknologi terkini, program latihan dan <i>coaching</i> mengenai <i>tools</i> CI/CD dalam proses pembangunan aplikasi di agensi,</li> <li>iii. memperkenalkan insentif untuk ahli pasukan bagi meningkatkan pelaksanaan DevOps seperti memberi penghargaan dan pengiktirafan atau insentif yang dipilih secara bersama oleh ahli pasukan,</li> <li>iv. meningkatkan keupayaan infrastruktur DevOps di premis (on-premises) atau menggalakkan penggunaan perkhidmatan pengkomputeran awan (cloud services), dan</li> <li>v. merancang strategi bagi mengoptimumkan pelaksanaan DevOps melalui penempatan dan pemantauan secara berterusan.</li> </ul>

### 3.4.3. Pengukuran Tahap Pelaksanaan

Pelaksanaan DevOps perlu diukur bagi memastikan proses peralihan mencapai matlamat dan objektif yang telah ditetapkan.



**Rajah 3-50: Metrik Pengukuran Prestasi Pelaksanaan DevOps**

Metrik Pengukuran Prestasi Pelaksanaan DevOps seperti di Rajah 3-50 digunakan sebagai penanda aras pengukuran tahap perlaksanaan DevOps. Metrik tersebut adalah bagi mengukur kekerapan penempatan (deployment frequency), masa pusingan perubahan (lead time for changes), purata masa untuk pemulihan (mean time to restore service (mttr)), dan kadar kegagalan perubahan (change failure rate).<sup>24</sup>

Empat profil tahap pengukuran prestasi seperti di Jadual 3-1 iaitu elit, tinggi, sederhana dan rendah digunakan bagi menentukan tahap pencapaian agensi dalam perlaksanaan DevOps.<sup>25</sup>

<sup>24</sup> Smith, D., Villalba, D., Irvine, M., Stanke, D., & Harvey, N. (2021). Accelerate State of DevOps 2021. Available: <https://services.google.com/fh/files/misc/state-of-devops-2021.pdf>.

<sup>25</sup> Sallin, M., Kropp, M., Anslow, C., Quilty, J. W., & Meier, A. (2021, June). Measuring software delivery performance using the four key metrics of devops. In International Conference on Agile Software Development (pp. 103-119). Springer, Cham.

### 3.4.3.1. Tujuan

Memastikan keberkesanan pelaksanaan proses peralihan melalui pengukuran prestasi tahap pelaksanaan DevOps secara berkala.

### 3.4.3.2. Metodologi

Tahap pengukuran prestasi pelaksanaan DevOps dari segi penyampaian dan pengoperasian sistem aplikasi menggunakan empat metrik adalah seperti di Jadual 3-48.

**Jadual 3-48: Jadual Pengukuran Prestasi Tahap Pelaksanaan DevOps**

Metrik/Tahap Prestasi	Elit	Tinggi	Sederhana	Rendah
<b>Kekerapan Penempatan (Deployment Frequency)</b>	Atas permintaan (beberapa penempatan dalam sehari)	Minimum sehari sekali Maksimum seminggu sekali	Minimum seminggu sekali Maksimum sebulan sekali	Minimum seminggu sekali Maksimum tiga bulan sekali
<b>Masa Pusingan Perubahan (Lead Time for Changes)</b>	Kurang dari satu hari	Satu hingga minggu hari satu	Satu minggu hingga satu bulan	Satu minggu hingga satu bulan
<b>Purata Masa Untuk Pemulihian (Mean Time to Restore Service (MTTR))</b>	Kurang dari satu jam	Kurang satu jam dari	Kurang satu hari dari	Lebih dari Satu hari
<b>Kadar Kegagalan Perubahan (Change Failure Rate)</b>	0-15%	0-15%	0-15%	46-60%

Penerangan berkaitan empat metrik pengukuran prestasi tahap pelaksanaan DevOps dari segi penyampaian dan operasi sistem aplikasi dijelaskan seperti berikut:

**a. Kekerapan Penempatan (Deployment Frequency)**

Kekerapan penempatan merujuk kepada kekerapan penempatan kod sumber ke produksi apabila terdapat ciri-ciri baharu perlu dioperasikan/go-live. Kekerapan penempatan diukur secara mingguan atau harian.

**b. Masa Pusingan Perubahan (Lead Time for Change)**

Masa Pusingan Perubahan mengukur tempoh masa yang diambil untuk kod sumber didaftar masuk (commit) sehingga ke peringkat penempatan sistem aplikasi. Semakin singkat masa pusingan perubahan, semakin cepat maklum balas diterima, hasilnya penambahbaikan dapat dilaksanakan dan versi baharu dapat dikeluarkan.

**c. Purata Masa untuk Pemulihan (Mean Time to Restore Service - MTTR)**

MTTR adalah metrik pengukuran tempoh masa yang diambil untuk memulihkan perkhidmatan setelah insiden berlaku. Insiden yang diselesaikan dengan kadar segera menjadi keutamaan bagi ketersediaan produk serta mengekalkan prestasi produk pada tahap optimum. Pengukuran metrik ini dapat membantu pasukan untuk mengukuhkan garis panduan dan prosedur standard operasi bagi meningkatkan tahap keberkesanan pengurusan insiden.

**d. Kadar Kegagalan Perubahan (Change Failure Rate)**

Kadar Kegagalan Perubahan adalah kekerapan penempatan yang memerlukan pemulihan atau pembaikan segera (hotfix). Mengukur kadar kegagalan membolehkan pasukan menjelak dan menambah

baik proses aliran kerja untuk memastikan kebarangkalian kegagalan dapat diminimumkan. Kegagalan perubahan merujuk kepada:

- i. Penempatan ke produksi yang menyebabkan gangguan sistem/produk tidak berfungsi,
- ii. penempatan ke produksi yang memerlukan pemberian segera, atau
- iii. penempatan ke produksi yang memerlukan pemulihian semula ke versi sebelumnya.

Pengukuran prestasi tahap perlaksanaan DevOps berdasarkan empat metrik adalah seperti **Lampiran 13: Pengukuran Tahap Pelaksanaan DevOps**.

### **3.5. KOMPONEN 5 (K5): PEMERKASAAN TEKNOLOGI**

Infrastruktur DevOps diperlukan bagi mewujudkan persekitaran yang menyokong pembangunan dan penyampaian produk. Dua elemen penting dalam infrastruktur DevOps bagi meningkatkan pembangunan produk adalah pengintegrasian dan pengautomasian tugas manual melalui penggunaan *tools*. Seni bina DevOps yang menggabungkan pelbagai *tools* seperti *tools* CI/CD dan *tools* pengujian membolehkan automasi sistem aplikasi dan infrastruktur yang terletak di persekitaran pembangunan dan persekitaran produksi berfungsi secara efisien.

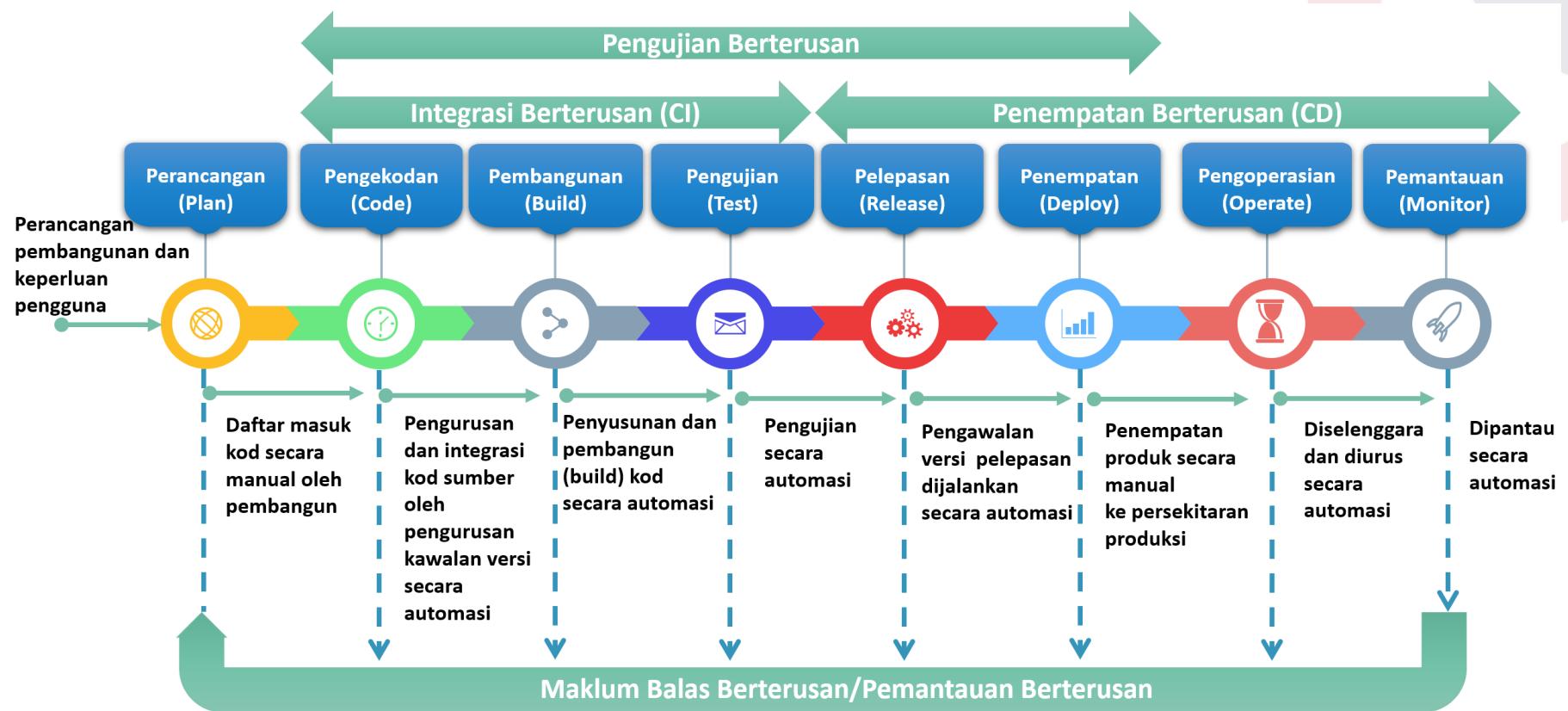
#### **3.5.1. Infrastruktur *Pipeline* DevOps**

*Pipeline* DevOps digunakan untuk mengautomasikan penghantaran perisian, yang merangkumi pembangunan kod sumber sehingga penempatan persekitaran produksi dan pemantauan. Infrastruktur *pipeline* DevOps adalah seperti di Rajah 3-51.<sup>26, 27</sup>

---

<sup>26</sup> Davis, J., & Daniels, R. (2016). *Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale* (1st ed.). O'Reilly Media.

<sup>27</sup> Coupland, M. (2021). *DevOps Adoption Strategies: Principles, Processes, Tools, and Trends: Embracing DevOps through effective culture, people, and processes*. Packt Publishing.



Rajah 3-51: Infrastruktur *Pipeline DevOps*

Berikut adalah penerangan proses yang berlaku dalam infrastruktur *pipeline* DevOps berdasarkan Rajah 3-51.

**a. Perancangan (Plan)**

Maklumat dan maklum balas dikumpul daripada pengguna dan perancangan perubahan perlu dilakukan untuk penambahbaikan produk.

**b. Pengekodan (Code)**

Pembangun membangunkan produk mengikut keperluan pengguna. Pembangun akan mendaftar masuk dan mengintegrasikan kod sumber ke dalam pengurusan kawalan sumber.

**c. Pembangunan (Build)**

Perubahan kod sumber yang berlaku di pengurusan kawalan sumber akan membolehkan proses automasi bermula bagi tujuan untuk penyusunan dan pembangunan (build) kod sebelum memasuki peringkat pengujian.

**d. Pengujian (Test)**

Pengujian secara automasi terhadap kod yang telah dibina (build) dilaksanakan menggunakan tools automasi.

**e. Pelepasan (Release)**

Langkah ini melibatkan perancangan, penjadualan dan kawalan versi proses pembinaan kod ke dalam persekitaran yang berbeza.

**f. Penempatan (Deploy)**

Pada peringkat ini, semua produk dan fail penempatan dipasang pada pelayan.

### **g. Pengoperasian (Operate)**

Selepas penempatan produk, pasukan operasi akan menyelenggara dan mengurus produk dalam persekitaran pengeluaran menggunakan tools automasi.

### **h. Pemantauan (Monitor)**

Dalam langkah ini, produk atau aplikasi yang dihantar kepada pengguna akan dipantau untuk memastikan produk berfungsi dengan baik. Maklumat ralat atau kegagalan fungsi akan dicatat dan diambil tindakan pembaikan sekiranya berlaku sebarang kegagalan fungsi dan ralat.

#### **3.5.1.1. Pembentukan Infrastruktur *Pipeline DevOps***

Berdasarkan Rajah 3-51, komponen berikut membentuk kitaran hayat pembangunan dan aliran kerja infrastruktur *pipeline DevOps*.

##### **a. Integrasi Berterusan dan Penempatan Berterusan (CI/CD)**

CI/CD merupakan salah satu asas kepada *pipeline DevOps*.

Antara ciri-ciri integrasi berterusan (**CI**) adalah:

- i. Membolehkan penyepaduan kod baharu daripada beberapa pembangun ke dalam repositori dengan lebih pantas dan
- ii. membolehkan pengujian kod secara automatik untuk mengenal pasti ralat lebih awal.

Antara ciri-ciri penempatan berterusan (**CD**) adalah:

- i. Membolehkan pembangun melakukan ujian tambahan seperti ujian UI, yang membantu memastikan penggunaan bebas pepijat dan

- ii. membantu dalam pengurusan keluaran kod, meningkatkan kekerapan keluaran ciri baharu dan mengautomasikan keluaran sistem aplikasi.

**b. Pengujian Berterusan**

Pengujian automatik boleh dilakukan pada setiap peringkat proses pembangunan. Pengujian mula dijalankan secara automatik setelah kod sumber disepadukan.

**c. Pemantauan Berterusan**

Pemantauan berterusan akan dapat mengurangkan atau meminimumkan gangguan fungsi produk kepada pengguna akhir dan memastikan kebolehcapaian produk berada pada tahap optimum.

**d. Maklum Balas Berterusan**

Setelah kod berada di persekitaran produksi, maklumat balas dari pengguna terhadap penggunaan produk dapat diperolehi untuk penambahbaikan berterusan.

### 3.5.2. Persekutaran *Pipeline* DevOps

Persekutaran yang menyokong pelaksanaan *pipeline* DevOps perlu diwujudkan seperti yang diterangkan dalam Rajah 3-52.<sup>28, 29</sup>



Rajah 3-52: Pembinaan Persekutaran ke Arah Pelaksanaan *Pipeline* DevOps

Langkah-langkah terperinci dalam pembinaan persekitaran *pipeline* DevOps adalah seperti berikut:

#### a. Perancangan dan Keperluan Pengguna

Agensi perlu menilai ketidakcekapan semasa perancangan dan keperluan pengguna sebelum mulakan penambahbaikan.

Metrik berikut digunakan untuk mendapatkan maklumat, memfokuskan kepada isu dan mulakan proses penambahbaikan:

<sup>28</sup> Gruver, G. (2016). Starting and Scaling DevOps in the Enterprise. In G. Gruver, Starting and Scaling DevOps in the Enterprise (pp. 7-45).

<sup>29</sup> Coyne, B., & Sharma, S. (2015). Technology in DevOps. Dalam B. Coyne, & S. Sharma, DevOps for Dummies (2nd IBM Limited Edition) (hlm. 24-30). John Wiley & Sons, Inc.;

- i. Peratusan sumber (masa, personel dan peruntukan) agensi yang diperlukan untuk mendokumentasikan perancangan dan keperluan,
- ii. jumlah masa dan sumber manusia yang diperlukan untuk melaksanakan pembangunan keperluan pengguna,
- iii. peratusan pindaan keperluan pengguna yang berubah daripada keperluan asal, dan
- iv. peratusan ciri-ciri yang diterima dan digunakan oleh pengguna yang memenuhi kriteria keperluan pengguna.

Proses dokumentasi dan penyusunan keperluan pengguna sehingga berupaya menyokong keputusan yang akan dibuat perlu dilaksanakan bagi mengoptimumkan *pipeline* DevOps.

Pasukan digalakkan untuk memfokuskan kepada perancangan *pipeline* DevOps bagi mengoptimumkan keperluan pengguna agar boleh memberikan nilai tambah kepada pengguna.

### **b. Pengujian dan Pembetulan Ralat**

Peringkat pengujian dan pembetulan ralat adalah untuk memastikan produk mempunyai kualiti yang tinggi dan ralat dapat dikenal pasti serta diperbaiki sebelum memasuki persekitaran produksi.

Metrik berikut digunakan untuk mendapatkan input penilaian dan penambahbaikan:

- i. Penilaian masa yang diperlukan untuk menjalankan set penuh ujian dan perancangan untuk peralihan kepada ujian automatik jika ujian yang dijalankan secara manual,

- ii. penilaian dan perancangan ke atas kebolehulangan kes ujian (repeatability test case) agar boleh dipercayai dan memberikan hasil yang konsisten,
- iii. peratus ralat yang ditemui terhadap kes ujian yang dijalankan semasa ujian unit, ujian sistem automatik dan ujian manual, dan
- iv. penilaian kecekapan kerja dan masa kitaran terhadap saiz kelompok kod dan kekerapan pelepasan.

#### **c. Penyediaan Persekutaran**

Pembangunan persekitaran dapat dilakukan melalui penggunaan IaC, agar perubahan dan konfigurasi persekitaran menjadi lebih cepat, stabil dan konsisten.

Metrik berikut digunakan untuk mengenal pasti isu semasa penyediaan persekitaran dan memulakan proses penambahbaikan:

- i. Jumlah jam yang bermula dari permohonan penyediaan persekitaran hingga persekitaran sedia digunakan,
- ii. kekerapan persekitaran baharu diperlukan,
- iii. peratusan masa diambil untuk penambahbaikan persekitaran ujian dan persekitaran produksi, dan
- iv. peratus ralat yang berkaitan dengan kod, persekitaran serta pangkalan data pada setiap peringkat persekitaran.

#### **d. Pelepasan Produksi**

Pada peringkat pelepasan ke dalam persekitaran produksi, semua ralat perlu dibaiki dan semua isu diselesaikan semasa di peringkat pengujian bagi memastikan proses pada peringkat produksi berjalan lancar.

Metrik berikut digunakan untuk mendapatkan maklumat kepada penambahbaikan produk:

- i. Penilaian masa dan aktiviti pasukan yang diperlukan untuk pemasangan dan pelepasan ke persekitaran produksi dan
- ii. bilangan ralat yang dijumpai semasa pelepasan.

#### e. Pemantauan dan Operasi

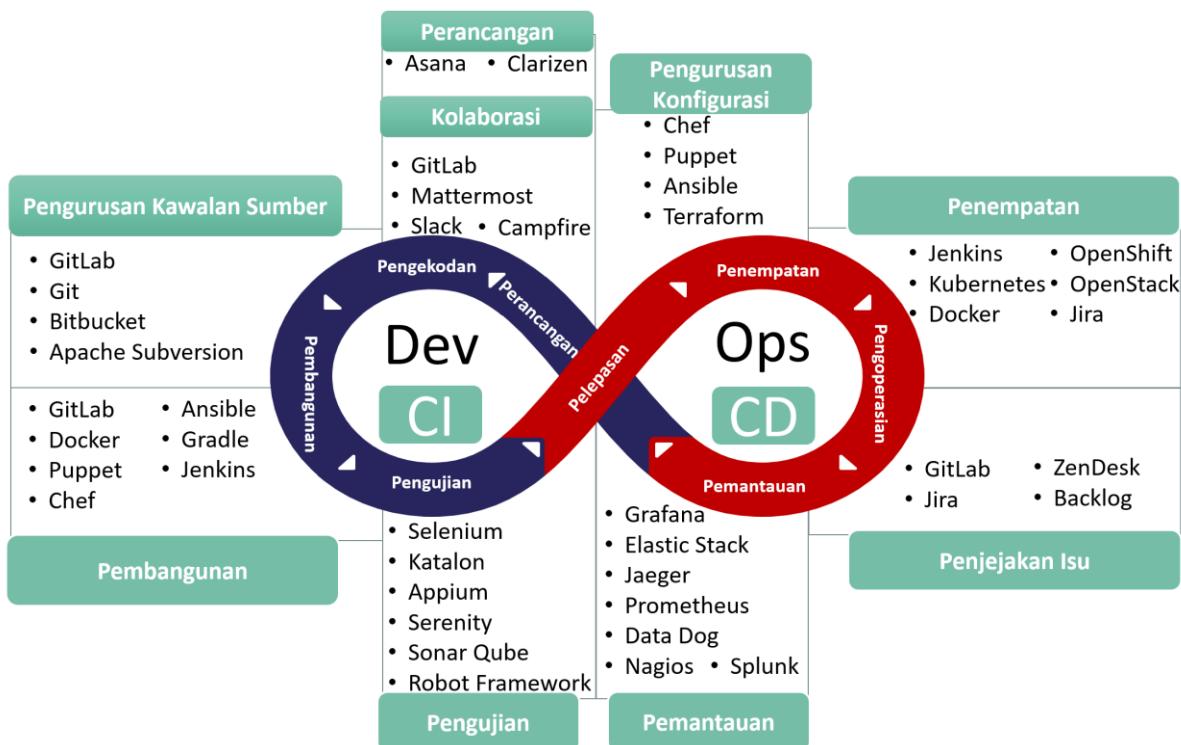
Pemantauan dan operasi dilaksanakan untuk memastikan produk berfungsi secara optimum dalam persekitaran produksi. Metrik utama pemantauan dan operasi adalah:

- i. Mengenal pasti bilangan insiden yang berlaku dalam persekitaran produksi yang menyebabkan kegagalan fungsi produk dan
- ii. mengukur tempoh masa yang diambil oleh pasukan untuk memulihkan insiden perkhidmatan berkaitan produk atau kegagalan memberi kesan kepada pengguna seperti gangguan yang tidak dirancang atau kemerosotan perkhidmatan.

Metrik pengukuran untuk mengoptimumkan *pipeline* DevOps asas boleh dirujuk di para 4.2.4: Mengukur Tahap Pelaksanaan.

### 3.5.3. Tools DevOps

Tools DevOps ialah perisian yang membantu pasukan pembangun dan operasi mengautomasikan proses pembangunan produk dalam *pipeline* DevOps. Penggunaan *tools* DevOps juga menggalakkan komunikasi dan kerjasama yang lebih lancar antara pihak pengurusan, pengurus produk, pasukan pembangun dan pasukan operasi. Kategori rantaian *tools* DevOps dan peranan *tools* bagi setiap peringkat DevOps adalah seperti di Rajah 3-53.



Rajah 3-53: *Tools* untuk Persekutaran *Pipeline* DevOps

Agensi perlu memahami fungsi dan tugas rantaian *tools* DevOps sebelum memilih dan menggunakan *tools* dalam pelaksanaan *pipeline* DevOps.

Penerangan terperinci bagi kategori *tools* adalah seperti berikut:

**a. *Tools* Perancangan (Plan)**

Membantu pasukan merancang, memantau dan mengurus keperluan projek untuk mencapai matlamat dalam tempoh masa yang ditetapkan. Menyediakan fungsi untuk memantau pelbagai tugas dan membuat agihan tugas mengikut sumber yang diperlukan. Contoh *tools* perancangan ialah carta Gantt, Asana dan Clarizen.

**b. *Tools* Kolaborasi (Collaboration)**

Membolehkan ahli pasukan menyelaraskan aktiviti kerja, meningkatkan kerjasama dan berkongsi pengetahuan. *Tools* kolaborasi meningkatkan amalan komunikasi berkesan dalam pasukan dan membantu pasukan dalam mengenal pasti keutamaan aktiviti dengan cepat dan efisien. Contoh *tools* kolaborasi dan komunikasi untuk sidang video adalah Google Meet, Microsoft Teams dan Skype. Manakala *tools* kolaborasi untuk pasukan pembangun adalah seperti GitLab, Mattermost, Slack dan Campfire.

**c. *Tools* Pengurusan Kawalan Sumber (Source Control Management)**

Mengurus dan mengesan perubahan pada repositori kod sumber serta sebagai lokasi storan berpusat untuk semua kod sumber, konfigurasi, dokumentasi dan lain-lain. Kod sumber dalam *tools* pengurusan kawalan sumber boleh diagihkan kepada *branch* yang berbeza untuk dicapai oleh ahli pasukan. Contoh *tools* pengurusan kawalan sumber adalah GitLab, Git, Bitbucket dan Apache Subversion.

**d. *Tools* Pembangunan (Build)**

Menguruskan binaan dan versi kod dengan bantuan alat automatik yang membantu dalam menyusun dan mempakejkan kod untuk keluaran ke persekitaran produksi. Contoh *tools* Docker, Puppet, Chef, Ansible, Gradle, Jenkins.

**e. *Tools* Automasi Pengujian (Testing)**

Mengautomasikan proses ujian seperti ujian unit dan ujian fungsian. Contoh *tools* rangka kerja automasi ujian ialah Selenium, Katalon, Appium, Serenity, Sonar Qube dan Robot Framework.

**f. *Tools* CI/CD**

Mengautomasikan proses aliran kerja untuk pembangunan, pengujian, pelepasan dan penempatan sistem aplikasi. Contoh *tools* CI/CD ialah GitLab, Jenkins, CircleCI, Bamboo dan Argo CD.

**g. *Tools* Pengurusan Konfigurasi (Configuration Management)**

Menguruskan IaC dengan penggunaan fail skrip konfigurasi. *Tools* ini membolehkan infrastruktur dikonfigurasi dan disediakan secara automatik melalui *orchestration*. Contoh *tools* pengurusan konfigurasi adalah Chef, Puppet, Ansible dan Terraform.

**h. *Tools* Penempatan (Deployment)**

Menguruskan, menjadualkan, menyelaraskan dan mengautomasikan keluaran produk (product release) ke persekitaran produksi. Contoh *tools* penempatan adalah Jenkins, Kubernetes, Docker, OpenShift, OpenStack dan Jira.

**i. *Tools* Penjejakan Isu (Issue Tracking)**

Membolehkan pengesanan isu secara telus dalam kitaran hayat pembangunan produk. *Tools* ini mengkatalog dan menjelaki isu bagi membolehkan penyelesaian dibuat dengan pantas dan cekap. Contoh *tools* penjejakan isu adalah GitLab, Jira, ZenDesk dan Backlog.

**j. *Tools* Pemantauan (Monitoring)**

Menganalisis data dan memberi gambaran keseluruhan prestasi sistem kepada pasukan dalam mengenal pasti dan menangani isu berkaitan

sistem. *Tools* pemantauan dapat memastikan masa operasi perkhidmatan dan prestasi sistem berada pada tahap optimum. Contoh *tools* pemantauan adalah Elastic Stack, Jaeger, Prometheus, Grafana, Data Dog, Nagios dan Splunk.

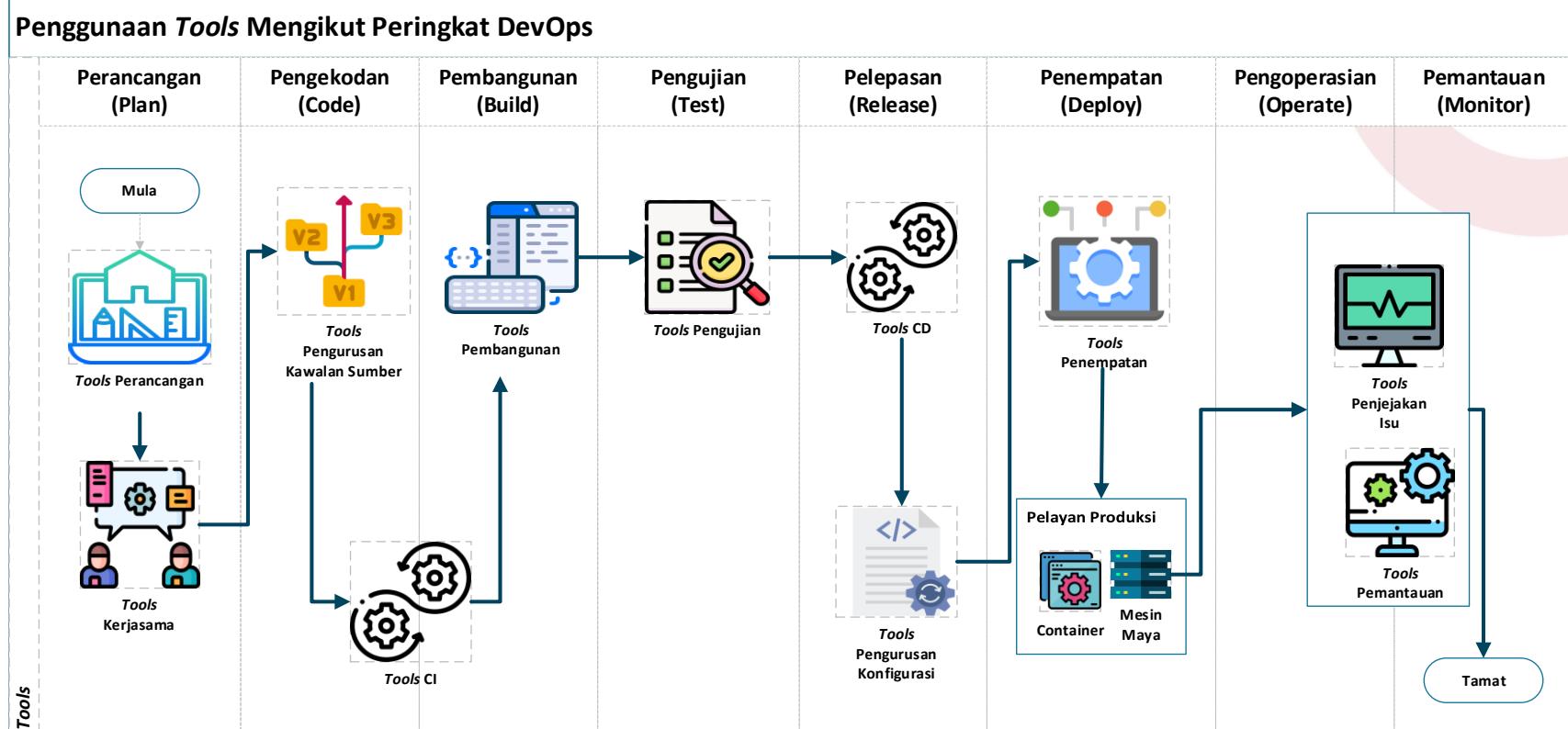
Senarai *tools* DevOps berdasarkan 17 kategori yang dihasilkan oleh pengamal DevOps adalah seperti dalam **Lampiran 21: Senarai Tools DevOps**.<sup>30</sup>

### 3.5.4. Penggunaan *Tools* Mengikut Peringkat DevOps

Aliran proses *pipeline* DevOps membantu pasukan membina (build), menguji (test) dan menempatkan (deploy) produk dengan cepat dan cekap melalui penggunaan rantaian *tools* DevOps. Aliran proses merujuk kepada susunan tugasan proses bagi kitar hayat pembangunan sistem berdasarkan peringkat DevOps dijelaskan seperti di Rajah 3-54.

---

<sup>30</sup> Sumber rujukan adalah dari laman sesawang Periodic Table of DevOps. Digital.ai. (n.d.). Diakses 9 Jun 2022, dari <https://digital.ai/periodic-table-of-devops-tools>.



Rajah 3-54: Penggunaan Tools Mengikut Peringkat DevOps

Jadual 3-49 menyenaraikan aliran tugas bagi *tools* mengikut peringkat DevOps.

**Jadual 3-49: Aliran Tugasan *Tools* Mengikut Peringkat DevOps**

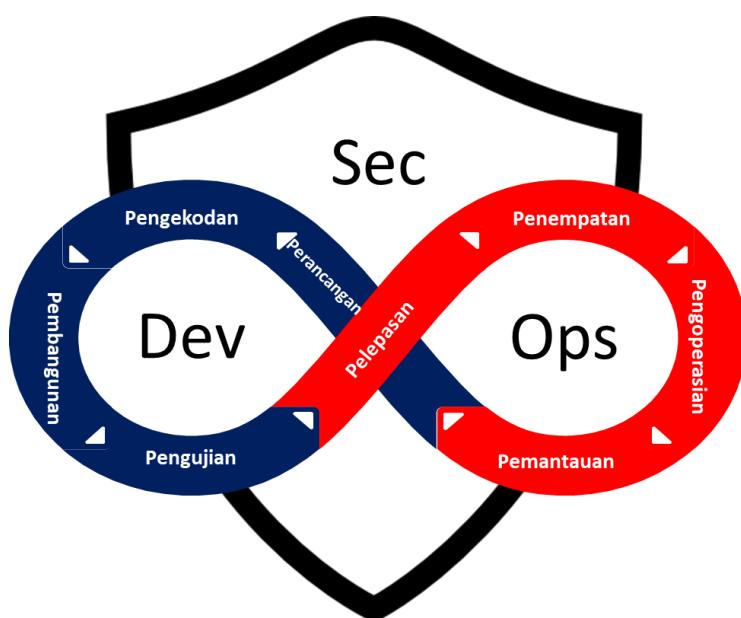
Peringkat DevOps	Tugasan <i>Tools</i>
<b>Perancangan (Plan)</b>	<ul style="list-style-type: none"> <li>i. Pasukan akan menggunakan <i>tools</i> perancangan untuk           <ul style="list-style-type: none"> <li>a. mendokumentasikan kajian keperluan pengguna,</li> <li>b. melaksanakan pengagihan tugas, dan</li> <li>c. menetapkan garis masa jadual pelaksanaan projek.</li> </ul> </li> <li>ii. Penyelarasaran aktiviti pembangunan akan menggunakan <i>tools</i> kolaborasi.</li> </ul>
<b>Pengekodan (Code)</b>	<ul style="list-style-type: none"> <li>i. Pembangun akan mendaftar masuk kod sumber ke dalam <i>tools</i> pengurusan kawalan sumber.</li> <li>ii. <i>Tools</i> pengurusan kawalan sumber akan melaksanakan tugas berikut;           <ul style="list-style-type: none"> <li>a. mengurus perubahan dan kemas kini kod daripada pembangun,</li> <li>b. melaksanakan analisis kod statik, dan</li> <li>c. melaksanakan semakan kod secara automasi dan <i>peer review</i>.</li> </ul> </li> </ul>
<b>Pembangunan (Build)</b>	<ul style="list-style-type: none"> <li>i. <i>Tools CI</i> akan mengesan perubahan pada kod sumber dan dengan integrasi bersama <i>tools</i> pembangunan, langkah berikut berdasarkan skrip konfigurasi;           <ul style="list-style-type: none"> <li>a. <i>Build</i> - penyusunan kod,</li> <li>b. <i>Unit test</i> - pengujian unit, dan</li> <li>c. <i>Release</i> - imej akan dihantar untuk pengujian lanjut.</li> </ul> </li> </ul>
<b>Pengujian (Test)</b>	<ul style="list-style-type: none"> <li>i. Imej yang telah dijana seterusnya akan <i>deploy</i> oleh <i>tools CI</i></li> <li>ii. Satu siri ujian akan dilaksanakan oleh <i>tools</i> pengujian seperti <i>load test</i>, <i>stress test</i>, pengujian UI, dan <i>penetration test</i>.</li> </ul>

Peringkat DevOps	Tugasan <i>Tools</i>
<b>Pelepasan (Release)</b>	<ul style="list-style-type: none"> <li>i. <i>Tools</i> CD akan melaksanakan tugas berikut untuk peringkat pelepasan (release) setelah siri ujian lulus pada peringkat pengujian.           <ul style="list-style-type: none"> <li>a. Penyediaan nota pelepasan (release note), dan</li> <li>b. Penandaan versi (version tagging).</li> </ul> </li> </ul>
<b>Penempatan (Deploy)</b>	<ul style="list-style-type: none"> <li>i. <i>Tools</i> penempatan akan melaksanakan tugas berikut;           <ul style="list-style-type: none"> <li>a. Pengemaskinian infrastruktur,</li> <li>b. Pengesahan untuk penempatan (deployment verification), dan</li> <li>c. Pemasangan imej ke pelayan produksi.</li> </ul> </li> </ul>
<b>Pengoperasian dan Pemantauan (Operate and Monitor)</b>	<ul style="list-style-type: none"> <li>i. <i>Tools</i> penjejakan isu akan merekod dan membantu pasukan dalam memantau perkembangan isu dan log ralat.</li> <li>ii. <i>Tools</i> pemantauan membantu pasukan menganalisis data dan memantau prestasi sistem bagi memastikan prestasi sistem aplikasi berada pada tahap optimum.</li> </ul>

Kesimpulannya, integrasi dan pengautomasian dalam *pipeline* DevOps dengan mengaplikasikan *tools* DevOps dapat membantu pasukan memperkemas pembangunan produk di antara pasukan pembangun dan pasukan operasi. Ini secara tidak langsung dapat memacu keseluruhan proses pengeluaran produk melalui penyepaduan berterusan, pembangunan, maklum balas, pemantauan, ujian, penghantaran dan penggunaan.

### 3.6. KOMPONEN 6 (K6): PEMATUHAN KESELAMATAN

Ekosistem keselamatan dalam DevOps merujuk kepada penyepaduan antara amalan keselamatan, *tools*, dan aspek budaya yang diterapkan sepanjang kitaran hayat pembangunan dan pelaksanaan pembangunan sistem aplikasi. Dengan mempunyai ekosistem keselamatan yang mapan dalam DevOps, pasukan dapat menyeimbangkan keperluan pembangunan secara meningkat dan berulang (incremental and iterative development) seterusnya memastikan sistem aplikasi berdaya tahan terhadap ancaman keselamatan siber.

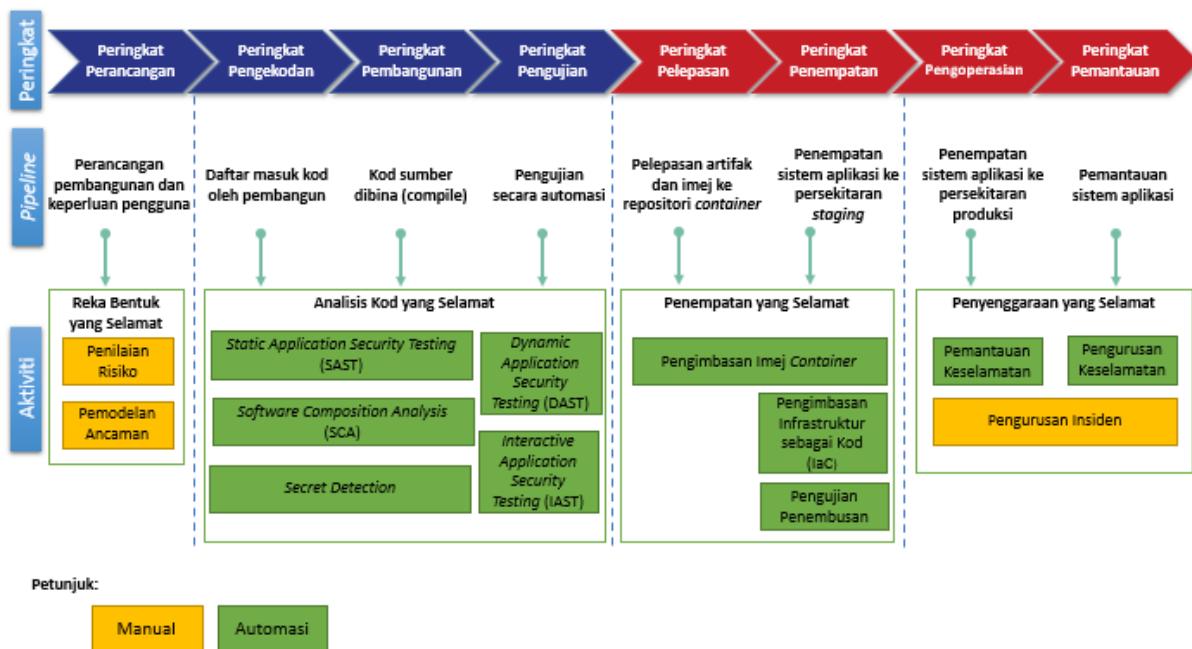


**Rajah 3-55: Penyepaduan Ekosistem Keselamatan dalam DevOps yang membentuk DevSecOps**

DevSecOps adalah lanjutan daripada pendekatan DevOps dengan penekanan pada elemen keselamatan yang disepadukan sepanjang kitaran hayat pembangunan sistem aplikasi. Dalam landskap ancaman keselamatan siber yang sentiasa berkembang, amalan konvensional dalam menangani ancaman keselamatan sebagai langkah berasingan dalam proses pembangunan meningkatkan risiko ancaman terhadap sistem aplikasi. Metodologi DevSecOps menjadikan keselamatan sebagai sebahagian daripada *pipeline* pembangunan dengan menekankan langkah proaktif untuk mengenal pasti dan mengurangkan potensi kelemahan bermula pada peringkat awal.

### 3.6.1. Aktiviti Keselamatan dalam DevSecOps

Penyepaduan aktiviti keselamatan pada keseluruhan peringkat DevOps yang menekankan kepentingan keselamatan dalam reka bentuk, analisis kod, penempatan dan penyelenggaraan secara berterusan dapat mewujudkan persekitaran sistem aplikasi yang lebih berdaya tahan dan selamat terhadap ancaman keselamatan.



Rajah 3-56: Aktiviti Keselamatan DevSecOps.

Rajah 3-56 menunjukkan aktiviti keselamatan pada setiap peringkat DevOps berdasarkan kepada panduan keselamatan pembangunan sistem aplikasi berikut<sup>31</sup>:

#### a. Reka Bentuk yang Selamat (Secure by Design)

Amalan yang menekankan penyepaduan langkah keselamatan ke dalam reka bentuk asas dan seni bina sistem perisian dari peringkat

<sup>31</sup> Cyber Security Malaysia. (2020). Guideline for Secure Software Development Life Cycle (SSDLC). *Secure Software Development Life Cycle (SSDLC) Guideline*. Retrieved February 9, 2024, from <https://www.cybersecurity.my/en/others/announcement/main/detail/2721/index.htm>

awal. Ia melibatkan penilaian aspek keselamatan, menjalankan *threat modeling* dan mewujudkan keperluan keselamatan semasa fasa reka bentuk untuk membina asas yang berdaya tahan dan selamat. Dokumen yang boleh dirujuk adalah seperti dibawah:

- i. *Guideline for Secure Software Development Life Cycle (SSDLC), 5 Phase 2: Security design.*<sup>32</sup>
- ii. *OWASP Checklist for Securing Application Design.*<sup>33</sup>

**b. Analisis Kod yang Selamat (Secure Code Analysis)**

Garis panduan berkaitan pengekodan selamat dan amalan terbaik sepanjang proses pembangunan sistem aplikasi yang melibatkan semakan serta penilaian sistematik kod sumber. Aktiviti ini bertujuan untuk meminimumkan kelemahan dengan menulis kod sumber yang mematuhi piawaian pengekodan yang ditetapkan. Dokumen yang boleh dirujuk adalah seperti dibawah:

- i. *OWASP Secure Coding Practices Quick Reference Guide, Version 2.0.*<sup>34</sup>
- ii. *CERT C Secure Coding Standard.*<sup>35</sup>

---

<sup>32</sup> Cyber Security Malaysia. (2020). Guideline for Secure Software Development Life Cycle (SSDLC). *Secure Software Development Life Cycle (SSDLC) Guideline*. Retrieved February 9, 2024, from <https://www.cybersecurity.my/en/others/announcement/main/detail/2721/index.htm>

<sup>33</sup> OWASP. *OWASP Checklist for Securing Application Design*. [https://owasp.org/www-pdf-archive/Checklist\\_For\\_Design.pdf](https://owasp.org/www-pdf-archive/Checklist_For_Design.pdf)

<sup>34</sup> OWASP (2010). *OWASP SecureCodingPractices Quick ReferenceGuide*, Version 2.0. Retrieved February 9, 2024, from [https://owasp.org/www-pdf-archive/OWASP\\_SCP\\_Quick\\_Reference\\_Guide\\_v2.pdf](https://owasp.org/www-pdf-archive/OWASP_SCP_Quick_Reference_Guide_v2.pdf)

<sup>35</sup> Seacord, R. C. (2014). *The CERT® C Coding Standard, second Edition: 98 rules for developing Safe, Reliable, and Secure systems*. <https://dl.acm.org/citation.cfm?id=2636993>

### c. Penempatan yang Selamat (Secure Deployment)

Merujuk kepada amalan dan langkah yang dilaksanakan untuk memastikan sistem aplikasi ditempatkan dan dikonfigurasikan dengan selamat. Pengurusan konfigurasi selamat, kawalan akses, penyulitan dan kawalan keselamatan lain yang berkaitan dengan penempatan bertujuan untuk meminimumkan ancaman keselamatan dan mengurangkan risiko eksploitasi semasa fasa penempatan kitaran hayat pembangunan perisian. Dokumen yang boleh dirujuk adalah seperti *Guideline for Secure Software Development Life Cycle (SSDLC), 8 Phase 5: Security deployment.*<sup>31</sup>

### d. Penyelenggaraan yang Selamat (Secure Maintenance)

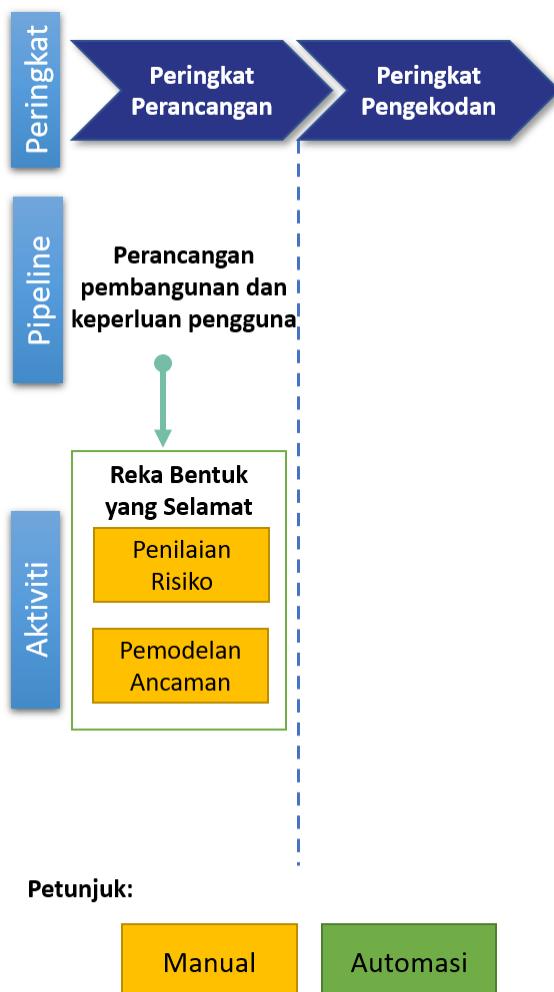
Amalan berterusan untuk memantau, mengemas kini dan menangani ancaman keselamatan dalam sistem aplikasi pada persekitaran produksi. Aktiviti seperti melaksanakan *security patches*, menjalankan pemeriksaan integriti kod dan memastikan sistem aplikasi kekal berdaya tahan terhadap ancaman keselamatan yang sentiasa berevolusi. Penyelenggaraan keselamatan adalah penting untuk keselamatan jangka panjang dan integriti sistem aplikasi yang digunakan. Dokumen yang boleh dirujuk adalah seperti *Guideline for Secure Software Development Life Cycle (SSDLC), 9 Phase6: Security maintenance.*<sup>36</sup>

---

<sup>36</sup> Cyber Security Malaysia. (2020). Guideline for Secure Software Development Life Cycle (SSDLC). *Secure Software Development Life Cycle (SSDLC) Guideline*. Retrieved February 9, 2024, from <https://www.cybersecurity.my/en/others/announcement/main/detail/2721/index.htm>

### **3.6.2. Reka Bentuk yang Selamat**

Reka bentuk yang selamat memberi tumpuan kepada komponen keselamatan siber dalam DevSecOps iaitu penilaian risiko dan pemodelan ancaman seperti pada Rajah 3-57. Komponen ini penting dalam mengenal pasti dan menangani kelemahan yang berpotensi memberi kesan kepada sistem aplikasi. Penilaian risiko dan pemodelan ancaman perlu menjadi tanggungjawab dan melibatkan semua ahli dalam pasukan DevOps. Matlamatnya adalah untuk mengintegrasikan keselamatan dengan lancar ke dalam kitaran hayat pembangunan sistem aplikasi.



### Rajah 3-57: Aktiviti untuk Reka Bentuk yang Selamat

### **3.6.2.1. Penilaian Risiko (Risk Assessment)**

Penilaian risiko ialah proses mengenal pasti, menilai dan mengutamakan risiko yang boleh memberi kesan kepada operasi, aset, individu atau organisasi. Dalam konteks DevSecOps, penilaian risiko melibatkan pemahaman tentang potensi ancaman terhadap perisian dan sistem aplikasi serta kelemahan yang mungkin dieksloitasi oleh ancaman dan kesan eksloitasi tersebut terhadap organisasi.

#### **a. Langkah Melaksanakan Penilaian Risiko**

Penilaian risiko merangkumi sembilan langkah seperti penerangan pada Rajah 3-58. Dokumen yang boleh dirujuk berkaitan pelaksanaan penilaian risiko sebagai garis panduan lanjut adalah seperti berikut:

- i. *Guidelines for Secure Software Development Life Cycle (SSDLC)* oleh Cyber Security Malaysia
- ii. Arahan, pekeliling, rangka kerja, garis panduan dan amalan terbaik penilaian risiko keselamatan maklumat yang sedang berkuat kuasa.



**Rajah 3-58: Langkah-Langkah untuk Melaksanakan Penilaian Risiko ke Atas Sistem Aplikasi**

### b. Kepentingan Penilaian Risiko

Berikut adalah faedah pelaksanaan penilaian risiko:

- i. Menyediakan cara berstruktur untuk mengenal pasti dan memahami risiko yang dihadapi oleh organisasi.
- ii. Membantu organisasi mengutamakan sumber dengan memfokuskan pada risiko yang paling ketara terlebih dahulu.
- iii. Menyokong dalam membuat keputusan berkaitan peruntukan sumber untuk pengurangan risiko.
- iv. Membantu sebagai garis panduan pematuhan terhadap keperluan kawal selia dan piawaian industri.

### 3.6.2.2. Pemodelan Ancaman (Threat Modeling)

Pemodelan Ancaman dalam DevSecOps adalah lanjutan dari penilaian risiko yang membentuk aktiviti berterusan bertujuan untuk mengukuhkan postur keselamatan sepanjang kitaran hayat pembangunan sistem aplikasi.

Berdasarkan hasil analisis yang diperoleh daripada penilaian risiko, pemodelan ancaman memfokuskan dengan lebih mendalam tentang pengenalpastian dan analisis ancaman, kelemahan dan vektor serangan yang berpotensi terhadap sistem aplikasi. Dengan menyepakatkan pemodelan ancaman sebagai aktiviti berterusan dalam DevSecOps, pasukan boleh memastikan penilaian berterusan dan dinamik terhadap cabaran keselamatan yang sentiasa berevolusi.

#### a. Langkah Melaksanakan Pemodelan Ancaman

Rajah 3-59 menerangkan secara ringkas langkah permodelan ancaman yang boleh dilaksanakan oleh pasukan. Dokumen yang boleh dirujuk berkaitan pelaksanaan permodelan ancaman sebagai garis panduan lanjut adalah seperti berikut:

- i. *Guidelines for Secure Software Development Life Cycle (SSDLC)* oleh Cyber Security Malaysia dan
- ii. Arahan, pekeliling, rangka kerja, garis panduan dan amalan terbaik penilaian risiko keselamatan maklumat yang sedang berkuat kuasa.



### Mengenal Pasti Komponen Sistem Aplikasi

1. Menentukan dan menyenaraikan komponen aplikasi atau sistem yang terlibat
2. Merujuk gambarajah aliran data (Data Flow Diagram(DFD)) untuk mendapatkan gambaran proses interaksi sistem seperti entiti luaran, proses, storan data dan aliran data.
3. Mengenal pasti dan mengklasifikasikan komponen dalam sistem mengikut kategori seperti data sensitif, fungsi kritis atau harta intelek.



### Menentukan Peringkat dan Mengelaskan Ancaman

1. Mengkategorikan potensi ancaman menggunakan metodologi seperti STRIDE, DREAD atau melaksanakan sesi *brainstorming* dalam pasukan.
2. Mengelas dan memberi keutamaan terhadap potensi ancaman berdasarkan faktor kebarangkalian ancaman berlaku, kesan ancaman dan kerumitan proses tindak balas.
3. Menentukan bentuk ancaman yang perlu diberi keutamaan serta fokus dalam memperuntukkan sumber seperti tenaga, masa dan peralatan.



### Menentukan Tindak Balas dan Mitigasi

1. Mereka bentuk strategi mitigasi dan langkah tindak balas untuk menangani ancaman.
2. Mendokumentkan model ancaman dan berkongsi dengan pihak berkepentingan untuk memastikan semua pihak memahami cara pelaksanaan.
3. Menyemak dan mengemas kini model ancaman secara kerap seiring perkembangan sistem aplikasi, perubahan dalam teknologi, keperluan bisnes atau ancaman yang wujud.

Rajah 3-59: Gambaran Berkanaan Langkah Pemodelan Ancaman

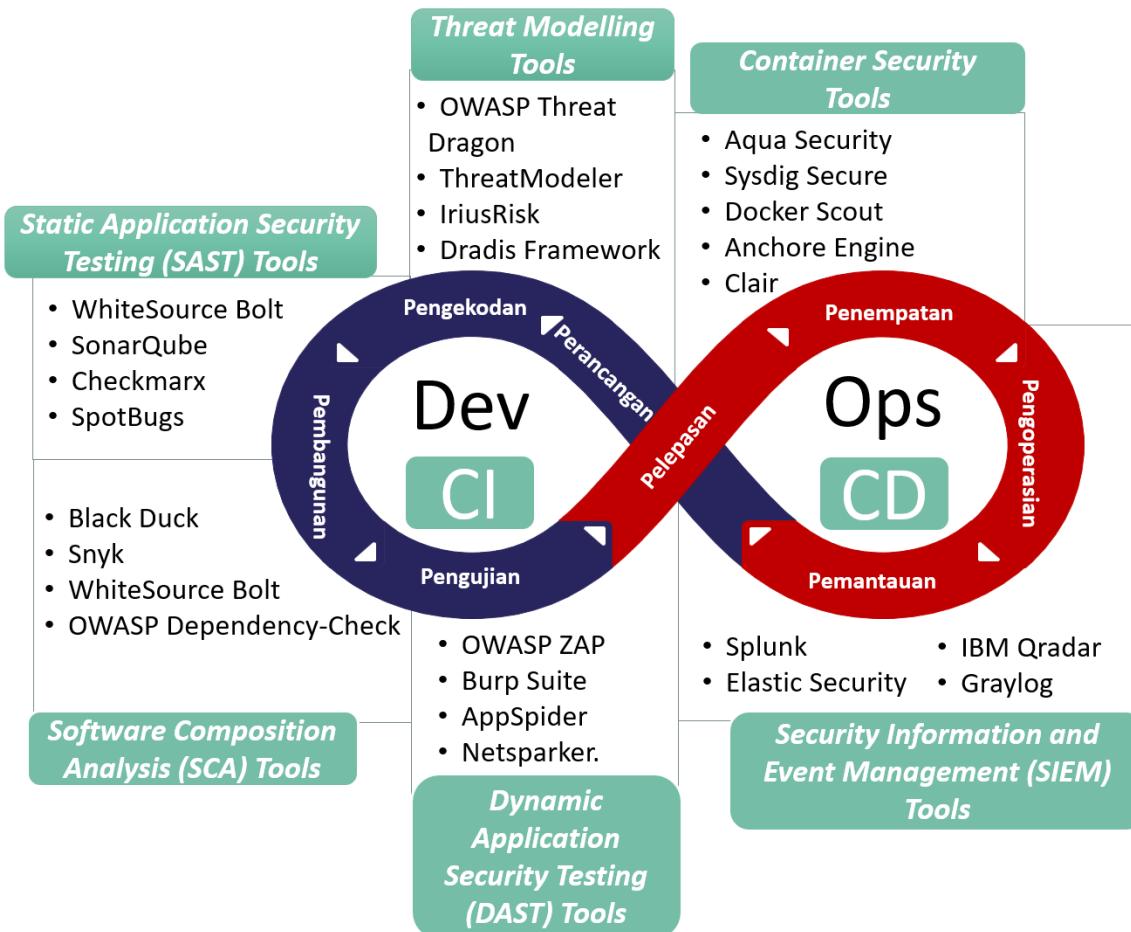
### 3.6.3. Tools Keselamatan dalam DevSecOps

Terdapat beberapa *tools* keselamatan yang direka untuk memudahkan amalan DevSecOps dengan mengintegrasikan keselamatan ke dalam proses pembangunan sistem aplikasi. *Tools* ini membantu mengautomasi pemeriksaan keselamatan, membolehkan kerjasama antara pasukan pembangunan dan keselamatan, dan memastikan keselamatan diterapkan sepanjang kitar hidup pembangunan sistem aplikasi.

#### 3.6.3.1. Pengintegrasian Tools Keselamatan dalam Rantaian Tools DevOps

*Tools* dan fungsi memainkan peranan penting dalam menyepadukan amalan keselamatan secara automasi dalam proses pembangunan sistem aplikasi.

Penggunaan *tools* yang sesuai bukan sahaja memudahkan dan mengautomasikan tugas berulang tetapi memberikan maklum balas berkesan yang boleh meningkatkan serta menambahbaik proses pembangunan sistem aplikasi. Rajah 3-60 adalah beberapa *tools* keselamatan yang popular dalam melaksanakan automasi keselamatan.



Rajah 3-60: *Tools* Keselamatan dalam Membentuk DevSecOps

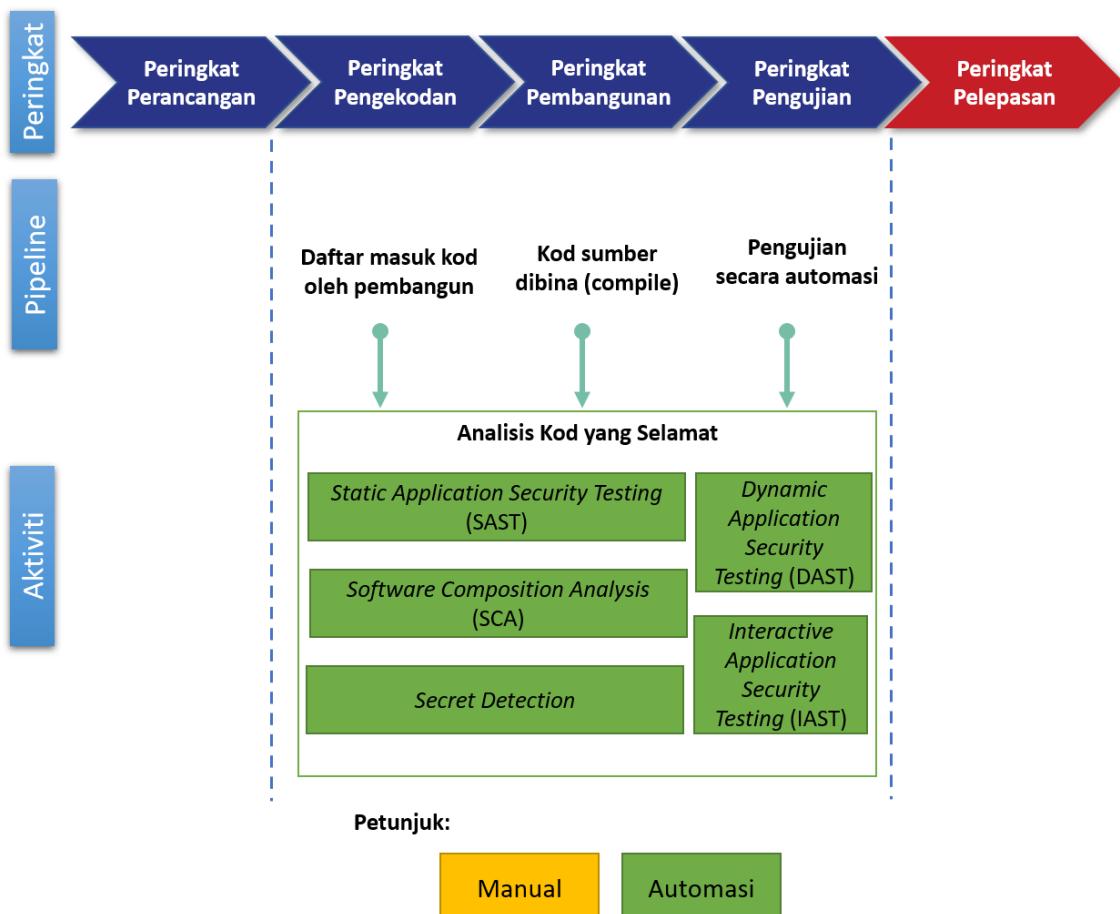
Jadual 3-50 menerangkan fungsi untuk setiap *tools* keselamatan yang merangkumi pelbagai aspek keselamatan dan boleh disepadukan dalam rantaian *pipeline* DevOps untuk meningkatkan amalan keselamatan sepanjang kitaran hayat pembangunan sistem aplikasi. Pasukan boleh menggabungkan *tools* berdasarkan keperluan khusus dan keutamaan aliran kerja masing-masing.

### Jadual 3-50: Penerangan berkaitan *Tools* Keselamatan

Tools	Fungsi	Contoh Tools
<b>Threat Modeling Tools</b>	Mengenal pasti dan menganalisis potensi ancaman dan kelemahan keselamatan dalam peringkat reka bentuk dan perancangan.	OWASP Threat Dragon, ThreatModeler, IriusRisk, Dradis Framework.
<b>Static Application Security Testing (SAST) Tools</b>	Menganalisis kod sumber untuk kelemahan keselamatan, ralat pengekodan dan pematuhan kepada amalan pengekodan selamat.	SonarQube, Checkmarx, SpotBugs, WhiteSource Bolt.
<b>Software Composition Analysis (SCA) Tools</b>	Mengenal pasti dan menguruskan kelemahan keselamatan dalam library dan komponen pihak ketiga yang digunakan dalam aplikasi.	OWASP Dependency-Check, WhiteSource Bolt, Snyk, Black Duck
<b>Dynamic Application Security Testing (DAST) Tools</b>	Menguji sistem aplikasi dalam masa nyata untuk mengenal pasti kelemahan keselamatan seperti ralat konfigurasi keselamatan.	OWASP ZAP, Burp Suite, AppSpider, Netsparker.
<b>Container Security</b>	Menyediakan keselamatan untuk aplikasi container dengan mengimbas imej untuk mencari kelemahan, menguatkuasakan dasar keselamatan dan memantau <i>runtime behavior</i> .	Aqua Security, Clair, Sysdig Secure, Docker Scout, Anchore Engine.
<b>Security Information and Event Management (SIEM) Tools</b>	Mengumpul, menganalisis dan mengabungkan data peristiwa keselamatan daripada pelbagai sumber untuk mengesan dan bertindak balas terhadap insiden keselamatan.	Splunk, Elastic Security, IBM QRadar, Graylog.

### 3.6.4. Analisis Kod yang Selamat

Pengujian keselamatan dalam DevSecOps melibatkan penyepaduan *tools* dan proses pengujian secara automasi pada sepanjang kitaran hayat pembangunan sistem aplikasi untuk mengenal pasti dan mengurangkan kelemahan keselamatan secara proaktif. Dengan mengintegrasikan pengujian keselamatan dalam *pipeline CI/CD*, pasukan boleh menilai secara sistematik postur keselamatan aplikasi mereka pada setiap peringkat. Automasi dalam pengujian keselamatan mempercepatkan pengenalpastian potensi ancaman seperti *code vulnerabilities*, ralat konfigurasi dan isu pematuhan seterusnya membolehkan penyelesaian segera. Rajah 3-61 menggambarkan automasi pengujian pada setiap peringkat dan *pipeline CI/CD*.



**Rajah 3-61: Automasi Pengujian Keselamatan**

Jadual 3-51 menjelaskan tentang automasi pengujian keselamatan dalam DevSecOps yang merangkumi pelbagai jenis pengujian seperti pengujian *Static Application Security Testing* (SAST), *Software Composition Analysis* (SCA), *Dynamic Application Security Testing* (DAST), *Interactive Application Security Testing* (IAST), & *Secret Detection*.

**Jadual 3-51: Penjelasan Berkennaan Automasi Pengujian Keselamatan**

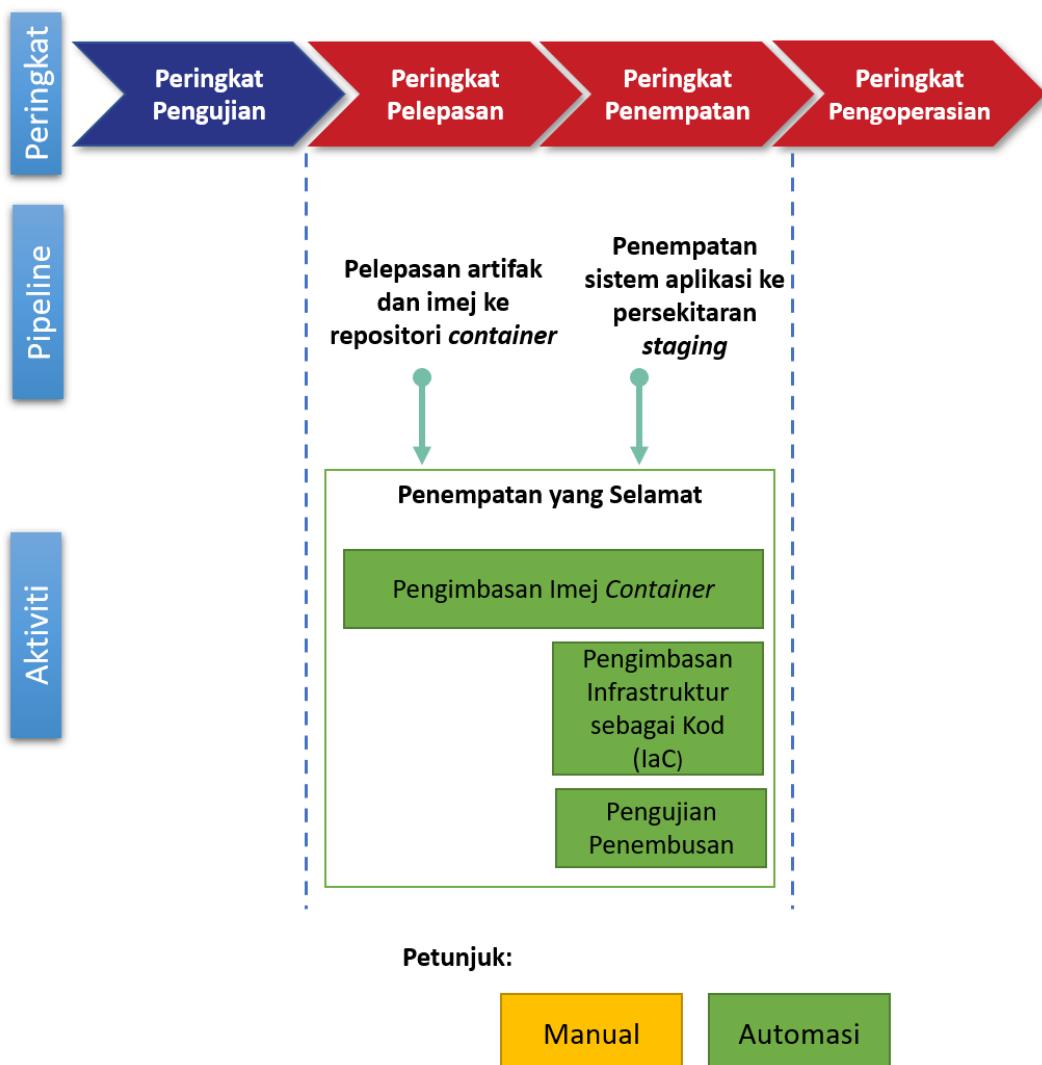
Pengujian Keselamatan	Keterangan
<b>Static Application Security Testing (SAST)</b>	Menganalisis kod sumber secara white-box dan kod binari untuk kelemahan keselamatan, ralat pengekodan dan isu pematuhan tanpa melaksanakan program.
<b>Software Composition Analysis (SCA)</b>	Mengimbas dan menganalisis komponen pihak ketiga, <i>libraries</i> , dan <i>dependencies</i> yang digunakan dalam sistem aplikasi untuk mengurangkan potensi kelemahan keselamatan, isu pelesenan, dan potensi risiko lain
<b>Dynamic Application Security Testing (DAST)</b>	Pengujian keselamatan yang berinteraksi dengan aplikasi yang sedang berjalan untuk mengesan potensi kelemahan keselamatan. Kaedah ini tidak memerlukan akses kepada kod sumber aplikasi. Sebaliknya, DAST mensimulasikan serangan terhadap aplikasi dan menganalisis respons aplikasi untuk mengenal pasti ancaman keselamatan.
<b>Interactive Application Security Testing (IAST)</b>	IAST ialah pengujian keselamatan yang menggabungkan aspek SAST dan DAST untuk menganalisis aplikasi dari dalam semasa masa jalan. IAST disepadukan ke dalam sistem aplikasi dan beroperasi dengan memberikan maklum balas terperinci tentang operasi sistem aplikasi, termasuk butiran tentang aliran data, laluan kawalan dan juga maklumat konfigurasi.

Pengujian Keselamatan	Keterangan
<b>Secret Detection</b>	Mengimbas dan menganalisis pangkalan kod dan fail teks dalam repositori untuk mengenalpasti corak yang sepadan dengan format maklumat sensitif seperti kunci API, token akses, kata laluan dan credential. Maklumat yang terdedah berpotensi mendedahkan kelemahan keselamatan atau akses tanpa kebenaran terhadap sistem aplikasi.

Kesimpulannya, pelbagai jenis pengujian keselamatan memainkan peranan yang berbeza dalam mengenal pasti dan mengurangkan risiko berkaitan keselamatan dalam pembangunan sistem aplikasi. Setiap pengujian mempunyai kekuatan dan fungsi berbeza bergantung pada keperluan khusus organisasi, fungsi sistem aplikasi dan proses pembangunan. Mengintegrasikan pelbagai jenis pengujian keselamatan mengikut keperluan organisasi dalam pipeline CI/CD boleh membantu mengenal pasti dan memulihkan kelemahan keselamatan bermula pada peringkat awal, mengurangkan risiko insiden keselamatan serta memupuk budaya kesedaran keselamatan dalam pasukan seterusnya di seluruh organisasi.

### 3.6.5. Penempatan yang Selamat

Pemantauan aktiviti dan log keselamatan adalah komponen penting yang diintegrasikan dalam *pipeline* DevSecOps yang bermatlamat untuk meningkatkan daya tahan keselamatan sistem aplikasi dalam keseluruhan persekitaran pembangunan dan persekitaran pengoperasian. Pemantauan keselamatan melibatkan dua aktiviti utama seperti Rajah 3-62 iaitu penempatan keselamatan dan penyenggaraan keselamatan.



Rajah 3-62: Aktiviti Berkaitan Penempatan yang Selamat

### **3.6.5.1. Pengimbasan Imej Container**

Tools automasi digunakan untuk menyemak kelemahan keselamatan imej *container* yang disimpan dalam repositori *container* sebelum penempatan ke persekitaran *staging*. Langkah keselamatan proaktif yang disepadukan dalam *pipeline CI/CD* membantu mengenal pasti kelemahan yang berpotensi pada peringkat awal dan meminumkan insiden terhadap pelanggaran keselamatan pada peringkat pengoperasian.

### **3.6.5.2. Pengimbasan Infrastruktur sebagai Kod (IaC)**

Pengimbasan infrastruktur sebagai kod (IaC) ialah proses menganalisis skrip yang menyediakan dan mengkonfigurasi persekitaran infrastruktur secara automatik. Pengimbasan IaC menyasarkan sintaks dan struktur khusus yang digunakan dalam mengkonfigurasi persekitaran infrastruktur yang bertujuan untuk mengekalkan pengoperasian infrastruktur yang selamat dan cekap.

### **3.6.5.3. Pengujian Penembusan**

Tools automasi digunakan untuk melaksanakan simulasi ancaman serangan terhadap langkah keselamatan yang sedang diuji. Pengujian ini membantu mengenal pasti sebarang potensi ancaman yang boleh dieksloitasi serta mengesan sebarang salah konfigurasi atau perubahan yang tidak dibenarkan pada sistem

### 3.6.6. Penyenggaraan yang Selamat

Penyenggaraan yang selamat adalah amalan keselamatan siber yang merangkumi pemantauan berterusan terhadap sistem aplikasi untuk mengesan ancaman, penetapan prosedur untuk mengurus risiko dan insiden serta memastikan konfigurasi dan *patches* sentiasa dikemas kini.

Penyenggaraan keselamatan melibatkan tiga aktiviti utama seperti Rajah 3-63 iaitu pemantauan keselamatan, pengurusan keselamatan dan pengurusan insiden.



**Rajah 3-63: Aktiviti Berkaitan Penyenggaraan yang Selamat**

### 3.6.6.1. Pemantauan Keselamatan (Security Monitoring)

Memberi tumpuan kepada pemantauan, pengesahan dan tindak balas terhadap ancaman serta kelemahan keselamatan apabila sistem aplikasi digunakan pada persekitaran produksi.

Jadual 3-52 menjelaskan aktiviti lanjut yang dilaksanakan untuk setiap aktiviti keselamatan.

**Jadual 3-52: Penerangan Aktiviti Pemantauan Keselamatan**

Aktiviti Keselamatan	Penerangan Aktiviti Keselamatan
Pemantauan Keselamatan	<ul style="list-style-type: none"> <li>i. <b>Pemantauan Berterusan</b> <ul style="list-style-type: none"> <li>• Melibatkan pengesahan dan tindak balas pada aktiviti keselamatan di persekitaran produksi.</li> <li>• <i>Tools CI/CD</i> disepadukan dengan <i>tools</i> pemantauan yang memberikan input pada masa nyata berkaitan prestasi sistem, aktiviti pengguna dan insiden keselamatan.</li> </ul> </li> <li>ii. <b>Pemantauan Log dan Analisis</b> <ul style="list-style-type: none"> <li>• <i>Tools CI/CD</i> dan infrastruktur yang berkaitan menjana log yang merekod pelbagai aktiviti.</li> <li>• Pengauditan keselamatan melibatkan aktiviti mengumpul, memantau dan menganalisis log untuk mengenal pasti sebarang tingkah laku yang mencurigakan atau anomali yang mungkin menunjukkan insiden keselamatan.</li> </ul> </li> <li>iii. <b>Kemaskini Patches Keselamatan</b> <ul style="list-style-type: none"> <li>• Kemaskini yang bertujuan untuk menangani kerentanan yang diketahui dalam komponen perisian dan infrastruktur.</li> <li>• <i>Tools CI/CD</i> mengautomasikan proses kemaskini <i>patches</i> keselamatan untuk mengekalkan keselamatan sistem dan mencegah eksploitasi.</li> </ul> </li> </ul>

### 3.6.6.2. Pengurusan Keselamatan (Security Management)

Melibatkan pemerhatian secara aktif kelakuan masa jalan (runtime behavior) sistem aplikasi dan infrastruktur untuk mengesan isu keselamatan yang berpotensi serta memastikan prestasi keseluruhan sistem berada pada tahap optimum.

Jadual 3-53 menjelaskan aktiviti lanjut yang dilaksanakan untuk setiap aktiviti keselamatan.

**Jadual 3-53: Penerangan Aktiviti Penyenggaraan Keselamatan**

Aktiviti Keselamatan	Penerangan Aktiviti Keselamatan
Pengurusan Keselamatan	<ul style="list-style-type: none"> <li>i. <b>Security Information and Event Management (SIEM)</b> <ul style="list-style-type: none"> <li>• Penyepadan dengan <i>tools</i> SIEM membolehkan analisis log secara terpusat dan terhubung.</li> <li>• Aktiviti keselamatan berkaitan aktiviti CI/CD dipantau untuk mengenal pasti corak yang mungkin menunjukkan ancaman keselamatan.</li> <li>• Insiden keselamatan boleh dikesan melalui analisis masa nyata data pemantauan.</li> </ul> </li> <li>ii. <b>Tindak Balas Insiden</b> <ul style="list-style-type: none"> <li>• <i>Tools</i> CI/CD dikonfigurasikan untuk mencetuskan makluman berdasarkan peraturan keselamatan yang telah ditetapkan atau anomali dalam data yang dipantau.</li> <li>• Insiden akan dimaklumkan kepada ketua pasukan devops sebagai <i>product backlog</i> baharu untuk dikendalikan secara dalaman oleh ahli pasukan.</li> </ul> </li> <li>iii. <b>Pemantauan Prestasi</b> <ul style="list-style-type: none"> <li>• Analisis keselamatan semasa fasa pemantauan termasuk pemantauan prestasi untuk memastikan langkah keselamatan tidak memberi kesan negatif kepada prestasi sistem.</li> <li>• <i>Tools</i> CI/CD dikonfigurasikan untuk merekod metrik prestasi dan memberikan paparan data berkaitan sebarang kemerosotan yang disebabkan oleh proses berkaitan keselamatan.</li> </ul> </li> </ul>

### 3.6.6.3. Pengurusan Insiden Keselamatan

Pengurusan insiden keselamatan dalam sektor awam adalah penting dalam menghadapi ancaman siber. Pasukan DevOps perlu mempunyai perancangan serta pelaksanaan tindakan yang sistematik dan berstruktur untuk memastikan pasukan dapat mengurus serta menangani insiden keselamatan siber dengan cekap serta berkesan. Pengurusan insiden pada peringkat pengoperasian dan pemantauan seperti Rajah 3-64 dilaksanakan secara manual apabila terdapat pengesanan insiden keselamatan semasa aktiviti penyenggaraan keselamatan dilaksanakan.



**Rajah 3-64: Pengurusan Insiden Keselamatan**

### a. Mengenal Pasti Jenis Insiden Keselamatan

Pengenalpastian jenis insiden keselamatan membolehkan pasukan memahami ancaman yang dihadapi dan menyesuaikan strategi tindak balas dengan lebih berkesan. Proses ini membolehkan tindak balas insiden yang pantas dan tepat serta meminimumkan kesan gangguan terhadap sistem aplikasi.

Jenis insiden keselamatan siber adalah seperti berikut<sup>37</sup>:

#### i. Penafian Perkhidmatan (Denial of Service, DoS) atau Penafian Perkhidmatan Teragih (Distributed Denial of Service, DDoS)

Serangan DoS atau DDoS merupakan serangan terhadap sistem atau rangkaian komputer yang menyebabkan ketidakupayaan sistem atau rangkaian tersebut untuk memberikan perkhidmatan kepada pengguna.

#### ii. Pencerobohan (Intrusion)

Pencerobohan merujuk kepada capaian tanpa kebenaran/tidak sah yang berjaya menembusi sistem atau rangkaian. Insiden ini boleh mengakibatkan akaun pentadbir sistem diambil alih, laman web diceroboh, kerosakan pada sistem, data atau konfigurasi sistem dipinda dan/atau pemasangan kod hasad seperti backdoor atau trojan.

#### iii. Jangkitan Perisian Hasad (Malicious Software, Malware)

Perisian hasad adalah perisian yang direka untuk memasuki sistem komputer tanpa kebenaran dan berpotensi membahayakan mesin atau rangkaian.

---

<sup>37</sup> Jabatan Perdana Menteri. (2022). *Pekeling Am Bilangan 4 Tahun 2022 Pengurusan dan Pengendalian Insiden Keselamatan Siber Sektor Awam*, 8-10.

#### **iv. Pengehosan Perisian Hasad (Malware Hosting)**

Pengehosan perisian hasad merujuk kepada keadaan di mana perisian hasad berada dalam pelayan atau komputer pengguna secara tidak sah seterusnya dijadikan sebagai sumber untuk dimuat turun atau diakses oleh siri serangan perisian hasad yang lain.

#### **v. Percubaan Pencerobohan (Intrusion Attempt)**

Percubaan dengan hasrat untuk menceroboh atau mengambil alih sistem secara tidak sah melalui aktiviti imbasan port rangkaian, akses sistem secara brute force atau mengenal pasti kerentanan sistem.

#### **vi. Potensi Serangan (Potential Attack)**

Potensi serangan adalah ancaman yang berkemungkinan berlaku akibat daripada kerentanan yang terdapat pada sistem/rangkaian atau kelemahan pada proses kerja sesebuah agensi. Serangan ini boleh memusnah, mendedah, meminda, melumpuh, mencuri atau mendapatkan akses yang tidak sah bagi tujuan menggunakan aset yang tidak dibenarkan. Serangan ini dikenal pasti berdasarkan maklumat risikan atau hasil pemantauan terhadap agensi.

### **b. Proses Pelaporan, Pengurusan dan Pengendalian Insiden Keselamatan**

Insiden keselamatan seperti pelanggaran data, serangan siber atau akses tanpa kebenaran boleh menimbulkan risiko yang besar kepada maklumat dan sistem aplikasi organisasi. Mewujudkan Proses Pelaporan, Pengurusan dan Pengendalian Insiden Keselamatan yang mantap adalah penting untuk mengenal pasti, bertindak balas dan mengurangkan kesan insiden keselamatan dengan kadar segera.

Tindakan terhadap insiden keselamatan hendaklah dibuat berdasarkan tahap keutamaan sesuatu insiden. Tahap keutamaan tindakan terhadap insiden keselamatan siber dijelaskan pada Jadual 3-54.<sup>38</sup>

---

<sup>38</sup> Jabatan Perdana Menteri. (2022). *Pekeliling Am Bilangan 4 Tahun 2022 Pengurusan dan Pengendalian Insiden Keselamatan Siber Sektor Awam*, 10-11.

### Jadual 3-54: Tindakan Terhadap Insiden

Keutamaan	Tindakan
<b>Keutamaan 1 –</b> Insiden keselamatan siber yang memberikan impak tinggi terhadap pertahanan dan keselamatan negara, kestabilan ekonomi negara, imej negara, keupayaan Kerajaan untuk berfungsi, kesihatan dan keselamatan awam serta privasi individu.	Agensi hendaklah melaporkan insiden kepada NC4 bagi tujuan penyelarasan dan memaklumkan kepada agensi yang menyelianya dalam tempoh 24 jam selepas insiden dikesan serta mengaktifkan Pelan Kesinambungan Perkhidmatan (Business Continuity Plan, BCP) dan Pelan Pemulihan Bencana (Disaster Recovery Plan, DRP) sekiranya perlu.
<b>Keutamaan 2 –</b> Insiden keselamatan siber yang tidak memberikan impak seperti mana yang dinyatakan dalam Keutamaan 1.	Pasukan hendaklah melaporkan insiden kepada ketua pasukan DevOps bagi tujuan penyelarasan tindakan. Pasukan DevOps akan melaksanakan pengendalian insiden secara kendiri sebagai <i>product backlog</i> baharu dan seterusnya memaklumkan kepada NC4 dan agensi yang menyelianya setelah proses pengendalian insiden dan pemulihan pada peringkat agensi selesai.

Pasukan boleh merujuk kepada dokumen berkaitan pengurusan dan pengendalian insiden keselamatan siber sektor awam sebagai garis panduan lanjut, khususnya Pekeliling Am Bilangan 4 Tahun 2022 Pengurusan dan Pengendalian Insiden Keselamatan Siber Sektor Awam.

Carta alir proses pelaporan insiden keselamatan siber sektor awam boleh dirujuk pada **Lampiran 22: Pelaporan Insiden**.

### 3.6.7. Pengukuran Metrik Keselamatan

Metrik keselamatan dalam DevSecOps adalah untuk mengukur keberkesanan dan kecekapan pelaksanaan prinsip-prinsip keselamatan dalam pembangunan sistem aplikasi yang selamat untuk sesebuah pasukan atau agensi. Rajah 3-65 adalah beberapa metrik keselamatan yang boleh digunakan sebagai permulaan untuk pengukuran dalam pelaksanaan DevSecOps.



**Rajah 3-65: Metrik Utama Pengukuran Pelaksanaan DevSecOps**

#### a. Kerentenan Keselamatan (Security Vulnerabilities)

Mengira bilangan kelemahan yang dikenal pasti dalam kod, infrastruktur dan konfigurasi, *dependencies* atau komponen lain berkaitan sistem aplikasi. Metrik ini bertujuan sebagai tanda pengukuran kesihatan keselamatan proses pembangunan sistem aplikasi, membantu pasukan dalam mengenal pasti ruang untuk penambahbaikan dan memastikan penyelesaian isu keselamatan tepat pada masanya.

**b. Metrik Pematuhan (Compliance Metrics)**

Mengukur pematuhan organisasi atau pasukan terhadap keperluan peraturan dan dasar keselamatan. Antaranya termasuk bilangan aset tidak patuh, masa yang diambil untuk mencapai pematuhan selepas perubahan atau bilangan insiden pematuhan dari semasa ke semasa.

**c. Liputan Automasi Pengujian (Automated Test Coverage)**

Mengukur peratusan kod yang dilaksanakan oleh pengujian secara automasi seperti pengujian SAST dan pengujian DAST serta penyepaduan pengujian dalam pipeline CI/CD. Liputan automasi pengujian yang lebih tinggi membantu hasil keluaran kod yang berkualiti lebih tinggi dan mengurangkan isu atau insiden keselamatan.

Penggunaan dan pemilihan metrik perlu selaras dengan matlamat pasukan dalam memberikan manfaat yang dapat mendorong kepada tindakan susulan yang berkesan. Metrik juga digunakan untuk memupuk budaya penambahbaikan berterusan dan sebagai penanda aras keberkesanan pelaksanaan DevSecOps pada pasukan.

## **BAB 4**

## **PENUTUP**

## BAB 4 : PENUTUP

Dokumen ini secara keseluruhannya telah memberi penjelasan mengenai Rangka Kerja Pelaksanaan DevOps dalam Pembangunan Sistem Aplikasi Sektor Awam. Beberapa metodologi kajian telah dilaksanakan sepanjang penghasilan dokumen ini antaranya adalah kajian literatur, perbincangan kumpulan fokus dan semakan oleh pasukan pakar. *Output* utama dalam penghasilan dokumen ini telah dapat dikenal pasti iaitu komponen-komponen utama rangka kerja, penyelesaian terhadap permasalahan dan penyesuaianya terhadap pelaksanaan DevOps di sektor awam.

Dokumen ini secara asasnya merangkumi matlamat dan manfaat pelaksanaan DevOps, dan enam komponen yang merupakan elemen utama dalam melaksanakan amalan DevOps iaitu Penerapan Prinsip dan Budaya, Pengukuhan Tadbir Urus, Pengadaptasian Metodologi, Pengukuran Kematangan, Pemerkasaan Teknologi dan Pematuhan Keselamatan. Setiap komponen disokong dengan penjelasan dan huraian terperinci serta beberapa cadangan pelaksanaan DevOps di agensi. Rangka Kerja DevOps merupakan komponen utama bagi dokumen ini yang menjelaskan secara terperinci pelaksanaan DevOps serta mengandungi langkah-langkah untuk melaksanakan DevOps dalam pembangunan sistem aplikasi di sektor awam.

Kejayaan pelaksanaan DevOps di sektor awam bergantung kepada faktor-faktor berikut:

- a. Pengurusan atasan sektor awam perlu komited dalam menyokong pelaksanaan DevOps melalui pengadaptasian rangka kerja yang dibangunkan dan amalan-amalan yang digariskan. Pemahaman yang jelas tentang visi dan objektif bagi pelaksanaan DevOps amat penting bagi memastikan pengurusan atasan agensi dapat merealisasikan pembangunan produk melalui pendekatan ini dengan lebih berkesan,
- b. Penglibatan sumber manusia yang kompeten dan mencukupi meliputi pelbagai disiplin adalah penting untuk meningkatkan kecekapan dalam pembangunan produk melalui pendekatan DevOps. Dengan adanya tahap kemahiran dan pengetahuan yang tinggi dalam penggunaan *tools* DevOps, pembangunan

produk dapat dilaksanakan dengan efisien, seterusnya berupaya menghasilkan produk yang berkualiti tinggi,

- c. Pembudayaan DevOps perlu dilaksanakan secara berterusan dan tidak terhad kepada tempoh masa tertentu. Langkah pembudayaan ini sentiasa perlu diterapkan dalam amalan kerja pembangunan produk di agensi dengan melaksanakan penambahbaikan berterusan melalui pendekatan yang lebih kolaboratif dan menekankan aspek komunikatif yang tinggi sepanjang proses pembangunan produk,
- d. Pemilihan infrastruktur dan *tools* DevOps yang bersesuaian adalah penting bagi memastikan proses pembangunan produk dapat dilaksanakan secara automasi, dan
- e. Pemantauan yang berterusan pada setiap peringkat dalam kitaran DevOps untuk memastikan kejayaan pelaksanaan DevOps di agensi.

Dokumen ini diharap dapat menjadi rujukan kepada agensi sektor awam dalam memantapkan pembangunan produk dan meningkatkan tahap penyampaian perkhidmatan melalui pendekatan DevOps.



# **LAMPIRAN**

**Lampiran 1: Templat Artifak *Product Vision*****PRODUCT VISION BOARD UNTUK SISTEM TEMPAHAN BILIK MESUARAT****VISI**

Halatuju dan visi utama produk secara umum. Visi ini boleh diekstrak dari dokumen *Business Requirement Specification* (\*BRS) sekiranya ada.

**KUMPULAN SASARAN**

Memberi penerangan kepada sasaran pengguna yang akan menggunakan produk yang akan dibangunkan. (\*\*PPS)

**JUSTIFIKASI**

Memberi justifikasi kepada permasalahan yang dihadapi oleh pengguna buat masa sekarang. (\*\*PPS)

**PRODUK**

Penerangan kepada produk yang akan dibangunkan beserta modul-modulnya. (\*BRS)

**TUJUAN**

Penerangan berkenaan matlamat pembangunan produk. (\*BRS)

**Lampiran 2: Templat Artifak *User Story***

ARTIFAK USER STORY				
USER STORY ID	PERANAN	FUNGSI	IMPAK	ACCEPTANCE CRITERIA
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

## Lampiran 3: Templat Artifak *Product Backlog*

#### Lampiran 4: Templat Artifak ***Definition of Done***

Komponen	Senarai Semak	Status	Catatan
<b>Pengujian Keperluan Fungsian</b>			
<b>Pengujian Keperluan Bukan Fungsian</b>			
<b>Penempatan</b>			
<b>Dokumentasi</b>			

### Lampiran 5: Templat Artifak Perancangan Kapasiti

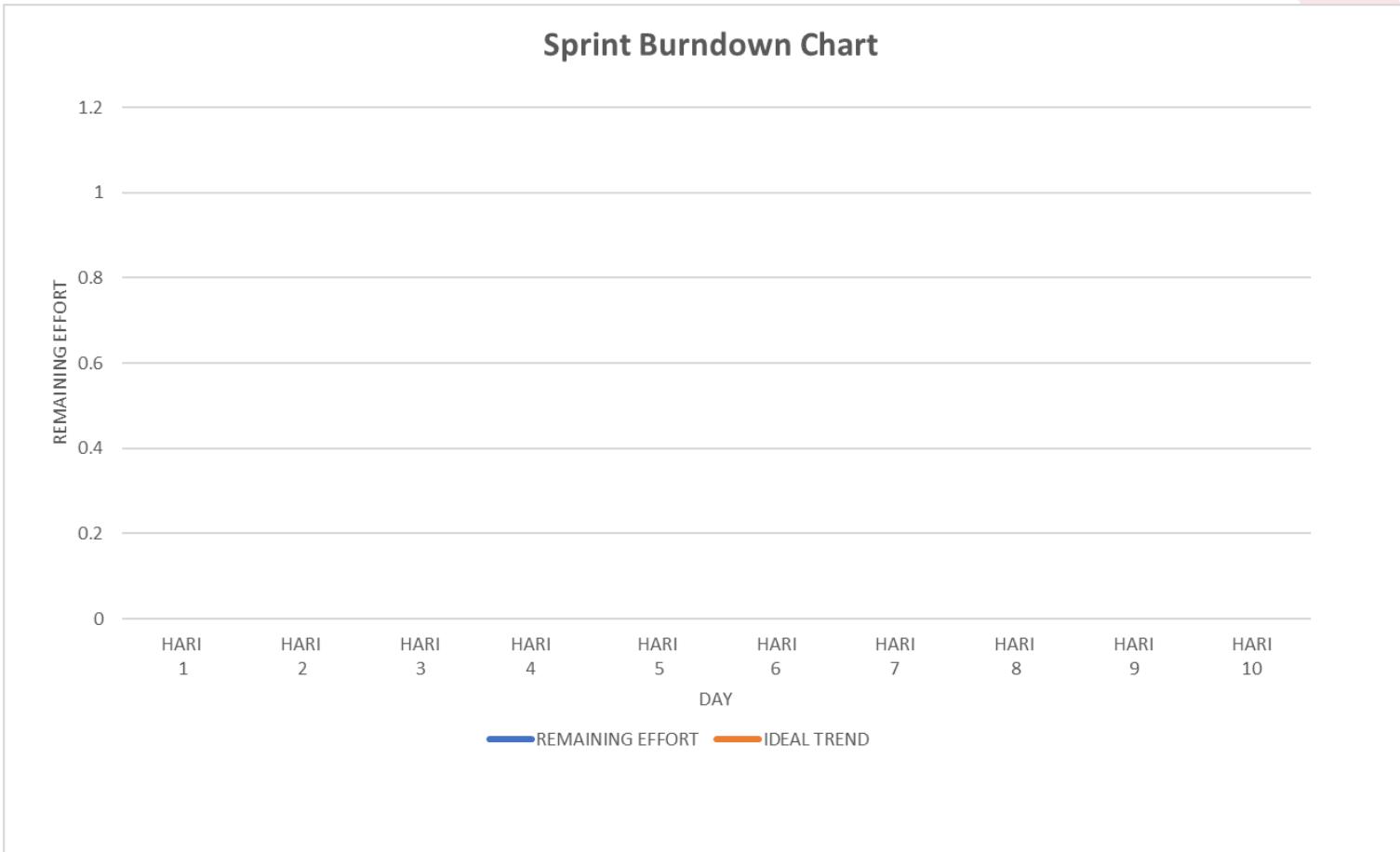
#### Perancangan Kapasiti *Sprint 1*

Tempoh Sprint =

Nama	Jumlah Hari Bekerja Semasa <i>Sprint</i> (Hari) (a)	Jumlah Masa Bekerja Sehari (Jam) (b)	Jumlah Masa yang Diperuntukkan (Jam) (a*b)
<b>Jumlah Masa Keseluruhan (Jam) (c)</b>			
(-)	<b>5-10% Interrupt Buffer(Jam) (d)</b>		
(-)	<b>5-10% PB refinement (Jam) (e)</b>		
<b>Jumlah Masa Bersih (Jam) <math>= (c - (d+e))</math></b>			

## Lampiran 6: Templat Artifak *Sprint Backlog*

### Lampiran 7: Templat Artifak **Burndown Chart**



**Lampiran 8: Templat Artifak *Sprint Retrospective***

ARTIFAK SPRINT RETROSPECTIVE			
ID	Perkara Yang Berjalan Dengan Lancar	Perkara Yang Tidak Berjalan Seperti Dijangka	Perkara Yang Perlu Ditambah Baik

### Lampiran 9: Kajian Kes Agile Scrum

#### PRODUCT VISION BOARD UNTUK SISTEM TEMPAHAN BILIK MESUARAT

VISI	JUSTIFIKASI	PRODUK	TUJUAN
<p><b>KUMPULAN SASARAN</b></p> <p>1. Warga agensi (MAMPU) bagi membuat permohonan 2. Pegawai tadbir bagi peringkat kelulusan - pentadbir bilik mesyuarat 3. Peringkat pengurusan agensi bagi tujuan perancangan keperluan.</p> <p>(Pelan Pembangunan Sistem)</p>	<p><b>JUSTIFIKASI</b></p> <p>1. Kebergantungan sepenuhnya kepada Pegawai tadbir bagi semakan kekosongan bilik mesyuarat. 2. Pertindihan tempahan bilik mesyuarat disebabkan oleh human error. 3. Proses pembatalan tidak boleh dilakukan dalam kadar segera. 4. Status kesedian bilik mesyuarat sukar dipantau. 5. Laporan penggunaan bilik mesyuarat tidak tepat bagi perancangan keperluan pejabat.</p> <p>(Pelan Pembangunan Sistem)</p>	<p><b>PRODUK</b></p> <p>Sistem yang mempunyai fungsi-fungsi bisnes yang mengurus penggunaan bilik mesyuarat seperti berikut:</p> <ol style="list-style-type: none"><li>1. Mengurus Pengguna</li><li>2. Mengurus Bilik Mesyuarat</li><li>3. Mengurus Tempahan</li><li>4. Paparan dashboard dan Jana Laporan</li></ol> <p>(BRS)</p>	<p><b>TUJUAN</b></p> <p>Sistem yang dibangunkan adalah untuk mengganti sistem ada dalam menyokong proses mengurus penggunaan bilik mesyuarat oleh Bahagian Khidmat Pengurusan</p> <p>(BRS)</p>

Rajah 4-1: Templat Artifikat *Product Vision*

ARTIFAK USER STORY				
USER STORY ID	PERANAN	FUNGSI	IMPAK	ACCEPTANCE CRITERIA
<b>Epic 1 : Pengurusan Pengguna</b>				
BF-BM-EP01-US01	Pengguna (warga MAMPU)	BOLEH mendaftar profil pengguna baru	SUPAYA boleh mendaftar masuk sistem tempahan bilik mesyuarat	Semasa warga melengkapkan butiran pendaftaran perlu pastikan: i.Format nombor kad pengenalan 12 digit dimasukan dan tidak termasuk simbol '-'. ii.Hanya alamat e-mel rasmi agensi sahaja yang boleh diterima. iii.Semak dan pastikan tiada pertindihan untuk kad pengenalan dan e-emel
BF-BM-EP01-US02	Pengguna (warga MAMPU)	BOLEH mengemaskini maklumat profil pengguna yang telah didaftarkan	SUPAYA akaun dapat disahkan dan selamat.	i.Saiz gambar dicadangkan tidak boleh melebih dari 800 x 800 piksel. ii.Format gambar yang dicadangkan adalah .jpeg dan .png sahaja.
<b>Epic 2 : Pengurusan Bilik Mesyuarat</b>				
BF-BM-EP02-US01	Pentadbir Bilik Mesyuarat	BOLEH menyelenggara maklumat bilik mesyuarat dengan mewujudkan rekod bilik yang baharu atau mengemaskini rekod sedia ada.	SUPAYA mendapatkan maklumat bilik mesyuarat yang tepat	i. rekod maklumat bilik mesyuarat dapat dilihat dari tahun 2019
BF-BM-EP02-US02	Pentadbir Bilik Mesyuarat	BOLEH menyemak maklum balas penggunaan bilik mesyuarat yang diterima daripada pengguna (warga Agensi)	SUPAYA mendapatkan sumber rujukan untuk melaporkan kerosakan dan perlukan pembaikan yang berkaitan.	i. statistik laporan kerosakan akan dijana di
BF-BM-EP02-US03	Pentadbir Bilik Mesyuarat	BOLEH menyediakan aduan kerosakan	SUPAYA hantar aduan kerosakan tersebut ke Sistem Selenggara Aset untuk tindakan seterusnya.	i. aduan kerosakan dihantar ke sistem selenggara asset mengikut batch harian
BF-BM-EP02-US04	Pentadbir Bilik Mesyuarat	BOLEH menerima notifikasi status aduan pembaikan kerosakan dari Sistem Selenggara Aset	SUPAYA kemaskini maklumat status bilik mesyuarat.	i. pengemaskinian boleh dilakukan sehari sebelum tarikh tempahan sahaja. ii. Pengemaskinian hanya boleh dilakukan oleh pengguna yang melakukan tempahan sahaja.

<b>Epic 3 : Pengurusan Tempahan</b>				
BF-BM-EP03-US01	Pengguna (warga MAMPU)	BOLEH menyemak kekosongan bilik mesyuarat	SUPAYA melakukan permohonan tempahan penggunaan bilik mesyuarat yang ingin digunakan pada tempoh masa tertentu.	i. memaparkan kekosongan bilik mesyuarat dalam bentuk dashboard dan infografik.
BF-BM-EP03-US02	Pentadbir Bilik Mesyuarat	BOLEH menyemak permohonan bilik mesyuarat yang memerlukan tindakan kelulusan	SUPAYA memberi kelulusan bagi setiap permohonan tempahan	i. notifikasi ke alamat emel dihantar
BF-BM-EP03-US03	Pengguna (warga MAMPU)	BOLEH melihat status pemohonan bilik mesyuarat	SUPAYA tahu berkenaan status permohonan yang telah dilakukan.	i. boleh melihat dan menerima notifikasi pemohonan
BF-BM-EP03-US04	Pengguna (warga MAMPU)	BOLEH melihat status pemohonan bilik mesyuarat	SUPAYA membatalkan permohonan sekiranya perlu.	i. pembatalan boleh dilakukan sekali sahaja
BF-BM-EP03-US05	Pengguna (warga MAMPU)	BOLEH menerima notifikasi untuk memberikan maklum balas penggunaan bilik	SUPAYA melengkapkan maklum balas penggunaan bilik mesyuarat setelah selesai penggunaan bilik.	i. notifikasi emel diberikan sejurus selepas selesai penggunaan bilik
<b>Epic 4: Paparan Dashboard dan Jana Laporan</b>				
BF-BM-EP04-US01	Pentadbir Bilik Mesyuarat	BOLEH melihat dashboard maklumat status rekod terkinin dan rumusan statistik bilik, tempahan, penggunaan dan status bilik mesyuarat	SUPAYA gunakan maklumat tersebut untuk tujuan pelaporan	i. dahboard dan laporan boleh dijana dalam bentuk xlsx,jpeg, png
BF-BM-EP04-US02	Pentadbir Bilik Mesyuarat	BOLEH menjana dan memaparkan laporan statistik terperinci tempahan, penggunaan dan penyelenggaraan bilik mesyuarat	SUPAYA mencetak laporan yang dijana atau dieksport laporan untuk disimpan	i. laporan boleh dijana dalam bentuk xlsx,jpeg, png

**Rajah 4-2: Artifak User Story**

ARTIFAK PRODUCT BACKLOG								
PRODUCT BACKLOG ID	KEUTAMAAN	TURUTAN KEUTAMAAN	USER STORY			STORY POINT	SPRINT NO	STATUS
<b>Epic 1 : Pengurusan Pengguna</b>								
BF-BM-EP01-PB01	Mesti Ada	1	Pengguna (warga MAMPU)	<b>BOLEH</b> mendaftar profil pengguna baru	<b>SUPAYA</b> boleh mendaftar masuk sistem tempahan bilik mesyuarat	3	Sprint 1	Dalam Tindakan
BF-BM-EP01-PB02	Mesti Ada	2	Pengguna (warga MAMPU)	<b>BOLEH</b> mengemaskini maklumat profil pengguna yang telah didaftarkan	<b>SUPAYA</b> akaun dapat disahkan dan selamat.	5	Sprint 1	Baharu
<b>Epic 2 : Pengurusan Bilik Mesyuarat</b>								
BF-BM-EP02-PB01	Mesti Ada	4	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menyelenggara maklumat bilik mesyuarat dengan mewujudkan rekod bilik yang baharu atau mengemaskini rekod sedia ada.	<b>SUPAYA</b> mendapatkan maklumat bilik mesyuarat yang tepat.	13		
BF-BM-EP02-PB02	Mesti Ada	3	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menyemak maklum balas penggunaan bilik mesyuarat yang diterima daripada pengguna (warga Agensi)	<b>SUPAYA</b> mendapatkan sumber rujukan untuk melaporkan kerosakan dan perlukan pembaikan yang berkaitan.	8	Sprint 1	
BF-BM-EP02-PB03	Patut ada	5	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menyediakan aduan kerosakan	<b>SUPAYA</b> hantar aduan kerosakan tersebut ke Sistem Selenggara Aset untuk tindakan seterusnya.	13		
BF-BM-EP02-PB04	Patut ada	6	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menerima notifikasi status aduan pembaikan kerosakan dari Sistem Selenggara Aset	<b>SUPAYA</b> kemaskini maklumat status bilik mesyuarat.	8		

<b>Epic 3 : Pengurusan Tempahan</b>								
BF-BM-EP03-PB01	Patut ada	7	Pengguna (warga MAMPU)	<b>BOLEH</b> menyemak kekosongan bilik mesyuarat	<b>SUPAYA</b> melakukan permohonan tempahan penggunaan bilik mesyuarat yang ingin digunakan pada tempoh masa tertentu.	8		
BF-BM-EP03-PB02	Patut ada	8	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menyemak permohonan bilik mesyuarat yang memerlukan tindakan kelulusan	<b>SUPAYA</b> memberi kelulusan bagi setiap permohonan tempahan	8		
BF-BM-EP03-PB03	Patut ada	9	Pengguna (warga MAMPU)	<b>BOLEH</b> melihat status semasa pemohonan tempahan yang telah dilakukan	<b>SUPAYA</b> tahu berkenaan status dan maklumat permohonan permohonan yang telah dilakukan.	5		
BF-BM-EP03-PR04	Patut ada	10	Pengguna (warga MAMPU)	<b>BOLEH</b> melihat status pemohonan bilik mesyuarat	<b>SUPAYA</b> membatalkan permohonan sekiranya perlu.	5		
BF-BM-EP03-PB05	Patut ada	11	Pengguna (warga MAMPU)	<b>BOLEH</b> menerima notifikasi untuk memberikan maklum balas penggunaan bilik mesyuarat	<b>SUPAYA</b> melengkapkan maklum balas penggunaan bilik mesyuarat setelah selesai penggunaan bilik.	8		
<b>Epic 4 : Paparan Dashbord dan Jana Laporan</b>								
BF-BM-EP04-PB01	Mungkin Ada	12	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> melihat dashboard maklumat status rekod terkinin dan rumusan statistik bilik, tempahan, penggunaan dan status bilik	<b>SUPAYA</b> gunakan maklumat tersebut untuk tujuan pelaporan	8		
BF-BM-EP04-PB02	Mungkin Ada	13	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menjana dan memaparkan laporan statistik terperinci tempahan, penggunaan dan penyelenggaraan bilik mesyuarat	<b>SUPAYA</b> mencetak laporan yang dijana atau dieksport laporan untuk disimpan	8		

Rajah 4-3: Artifak *Product Backlog*

Perancangan Kapasiti <i>Sprint 1</i>			
Tempoh <i>Sprint</i> , 2 minggu = 10 hari bekerja			
Nama	Jumlah Hari Bekerja Semasa <i>Sprint</i> (Hari) (a)	Jumlah Masa Bekerja Sehari (Jam) (b)	Jumlah Masa yang Diperuntukkan (Jam) (a*b)
Ali	8	5	40
Azhim	9	5	45
Ahmad	7	5	35
Raziman	6	2	12
Fauzi	4	4.5	18
Jumlah Masa Keseluruhan (Jam) (c)			150
(-)	5-10% <i>Interrupt Buffer</i> (Jam) (d)		15
(-)	5-10% <i>PB Refinement</i> (Jam) (e)		15
Jumlah Masa Bersih (Jam) = (c - (d+e))			120

Rajah 4-4: Artifak Perancangan Kapasiti

PRODUCT BACKLOG ID	PRODUCT BACKLOG ITEM	ID TUGASAN	TUGASAN	AHLI PASUKAN	STATUS	STORY POINT	ANGGARAN MASA (JAM)	HARI 1	HARI 2	HARI 3	HARI 4	HARI 5	HARI 6	HARI 7	HARI 8	HARI 9	HARI 10	
<b>Epic 1: Pengurusan Pengguna</b>																		
BF-BM-EP01-PB01	Pengguna (warga MAMPU) BOLEH mendaftar profil pengguna baru SUPAYA boleh mendaftar masuk sistem tempahan bilik mesyuarat	US-001-ST01	<i>new table format</i>	Azhim	Selesai	3	8											
		US-001-ST02	<i>new table UI</i>	Ahmad	Selesai		4											
		US-001-ST03	<i>implement new db format</i>	Fauzi	Selesai		6											
		US-001-ST04	<i>template setup</i>	Raziman	Selesai		2											
		US-001-ST05	<i>creation of auto approve</i>	Ali	Selesai		4											
BF-BM-EP01-PB02	Pengguna (warga MAMPU) BOLEH mengemaskini maklumat profil pengguna yang telah didaftarkan SUPAYA akan dapat disahkan dan selamat.	US-002-ST01	<i>Install contact page.</i>	Ali	Selesai	5	5											
		US-002-ST02	<i>new table UI</i>	Raziman	Selesai		6											
		US-002-ST03	<i>implement new db format</i>	Ahmad	Selesai		8											
		US-002-ST04	<i>template setup</i>	Azhim	Selesai		8											
		US-002-ST05	<i>export configuration</i>	Ahmad	Selesai		7											
		US-002-ST06	<i>creation of auto approve</i>	Ali	Selesai		8											
		US-002-ST07	<i>audit trail</i>	Azhim	Selesai		5											
		US-002-ST08	<i>creation of new table</i>	Raziman	Selesai		5											
<b>Epic 2: Pengurusan Bilik Mesyuarat</b>																		
BF-BM-EP02-PB02	Pentadbir Bilik Mesyuarat BOLEH menyemak maklum balas penggunaan bilik mesyuarat yang diterima daripada pengguna (warga Agensi) SUPAYA mendapatkan sumber rujukan untuk melaporkan kerosakan dan perlukan pembaharuan yang berkaitan.	US-004-ST01	<i>Install contact page</i>	Azhim	Selesai	8	8											
		US-004-ST02	<i>new table UI</i>	Raziman	Selesai		6											
		US-004-ST03	<i>implement new db format</i>	Ali	Selesai		7											
		US-004-ST04	<i>template setup</i>	Ahmad	Selesai		7											
		US-004-ST05	<i>table implementation</i>	Ali	Selesai		6											
		US-004-ST06	<i>export configuration</i>	Fauzi	Selesai		5											
		US-004-ST07	<i>creation of auto approve</i>	Raziman	Selesai		6											
<b>JUMLAH STORY POINT</b>						16												
<b>MASA SEBENAR (Jam) (a)</b>						0												
<b>REMAINING EFFORT (Jam) (b)</b> (b=b hari sebelum – a hari semasa)						121												
<b>IDEAL TREND (jam) (c)</b> (c = c rancang – (c rancang/jumlah hari x hari semasa))						121												

Rajah 4-5: Artifak Sprint Backlog Hari 1 bagi Sprint 1

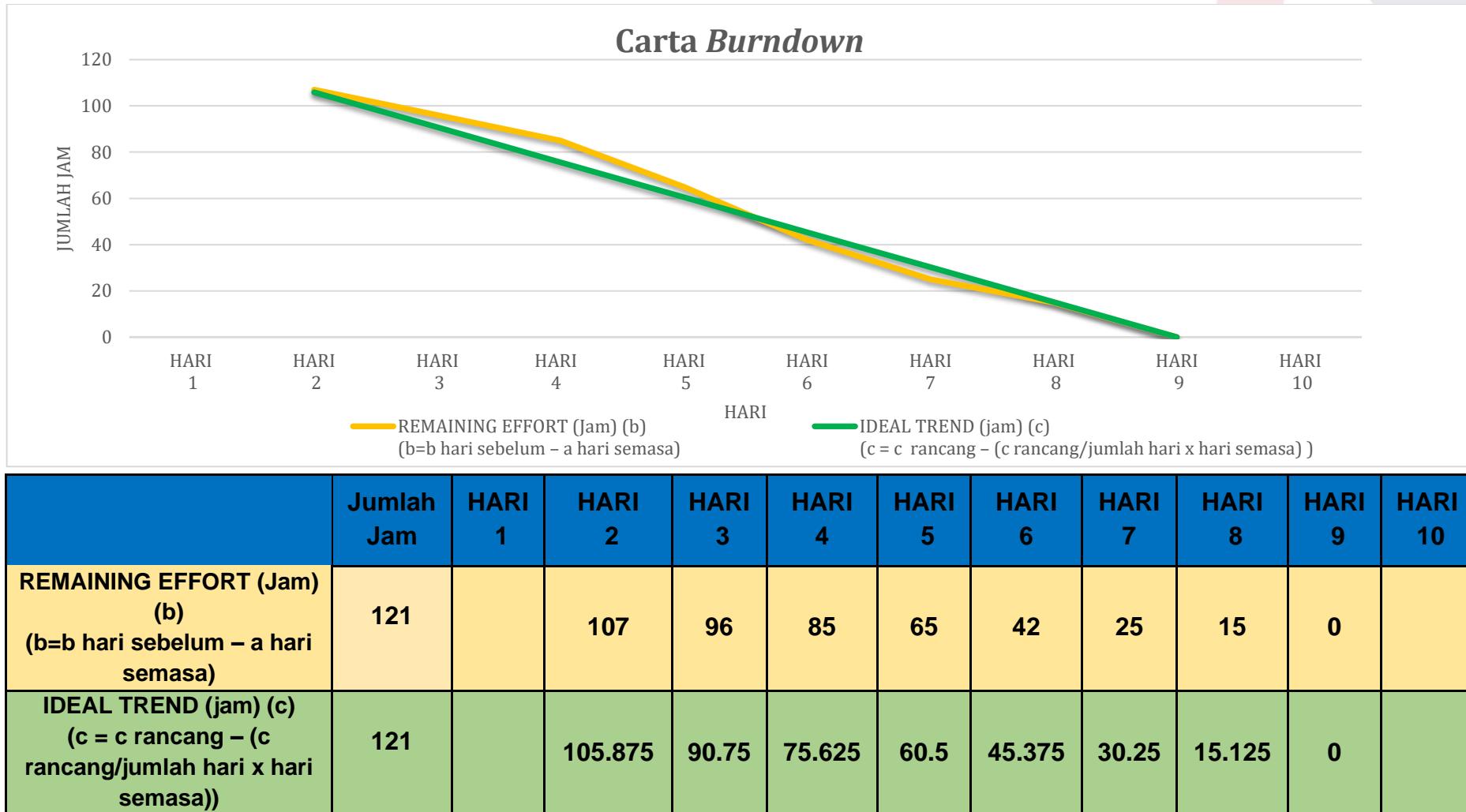
DAILY SCRUM LOG						
Ahli Pasukan	Soalan	Hari 2 Selasa	Hari 3 Rabu	Hari 4 Khamis	Hari 5 Jumaat	Hari 6 Isnin
Ali	Apakah yang anda lakukan semalam?	menyediakan auto approve fungsi	penyediaan fungsian auto approve (50% siap)	bercuti	bercuti	penyediaan fungsian auto approve (100% siap)
	Tugasan apakah yang akan dilakukan hari ini?	sambungan penyediaan fungsian auto approve	sambungan penyediaan fungsian auto approve	bercuti	sambungan penyediaan fungsian auto approve (80% siap)	mengelakkan pengujian
	Adakah terdapat sebarang permasalahan?	memerlukan maklumat lanjut berkenaan db table auto approve	tiada	bercuti	tiada	tiada
Azhim	Apakah yang anda lakukan semalam?	menyelesaikan new table format untuk User story 1	bercuti	table formatting untuk user story 1 (65% siap)	table formatting untuk user story 1 (90% siap)	table formatting untuk user story 1 (100% siap)
	Tugasan apakah yang akan dilakukan hari ini?	bercuti	sambungan kepada menyediakan formatting table bagi user story 1	sambungan kepada menyediakan formatting table bagi user story 1	sambungan kepada menyediakan formatting table bagi user story 1 dan mengelakkan pengujian	mengelakkan pengujian
	Adakah terdapat sebarang permasalahan?	memerlukan maklumat lanjut dari DBA berkenaan struktur db table	tiada	tiada	tiada	tiada
Ahmad	Apakah yang anda lakukan semalam?	menyediakan UI kepada modul pendaftaran pengguna	bercuti	bercuti	bercuti	
	Tugasan apakah yang akan dilakukan hari ini?	sambungan kepada penyediaan UI bagi modul pendaftaran pengguna	bercuti	sambungan kepada penyediaan UI bagi modul pendaftaran pengguna	bercuti	penyediaan UI bagi modul pendaftaran pengguna (90% siap)
	Adakah terdapat sebarang permasalahan?	memerlukan perbincangan bersama pemilik produk untuk mengesahkan UI yang diingini.	bercuti	UI telah mencapai peratusan siap 75%. Perlukan input dari pemilik produk mengenai UI lain yang diperlukan.	bercuti	duplicasi of UI yang mempunyai sama UI, memerlukan perbincangan bersama pemilik produk

Raziman	Apakah yang anda lakukan semalam?	menyediakan template bagi modul pendaftaran pengguna	bercuti	penyediaan template (85% siap)	penyediaan template setup (100% siap)	pengujian kepada template setup yang telah dikonfigurasi
	Tugasan apakah yang akan dilakukan hari ini?	bercuti	sambungan kepada penyediaan template setup	sambungan kepada penyediaan template dan melakukan pengujian	mengelakkan pengujian	sambungan pengujian
	Adakah terdapat sebarang permasalahan?	tiada	memerlukan pengesahan pemilik produk berkenaan template yang telah dibangunkan	tiada	memerlukan pengesahan pemilik produk berkenaan template yang telah disediakan	tiada
Fauzi	Apakah yang anda lakukan semalam?	bercuti	bercuti	bercuti	implement new db table (50% siap)	implement new db table 80% siap)
	Tugasan apakah yang akan dilakukan hari ini?	bercuti	bercuti	implement new db table	sambungan kepada implementation new db table	sambungan kepada implementation new db table
	Adakah terdapat sebarang permasalahan?	bercuti	bercuti	perlukan maklumat lajut dari DBA berkenaan struktur db modul	tiada	perlukan sokongan dan maklumat lanjut DBA

**Rajah 4-6: Artifak Log *Daily Scrum***

PRODUCT BACKLOG ID	PRODUCT BACKLOG ITEM	ID TUGASAN	TUGASAN	AHLI PASUKAN	STATUS	STORY POINT	ANGGARAN MASA (JAM)	HARI 1	HARI 2	HARI 3	HARI 4	HARI 5	HARI 6	HARI 7	HARI 8	HARI 9	HARI 10	
<b>Epic 1: Pengurusan Pengguna</b>																		
BF-BM-EP01-PB01	Pengguna (warga MAMPU) BOLEH mendaftar profil pengguna baru SUPAYA boleh mendaftar masuk sistem tempahan bilik mesyuarat	US-001-ST01	<i>new table format</i>	Azhim	Selesai	3	8		3	0	2	2	1	0	0	0	0	
		US-001-ST02	<i>new table UI</i>	Ahmad	Selesai		4		3	1	0	0	0	0	0	0	0	
		US-001-ST03	<i>implement new db format</i>	Fauzi	Selesai		6		2	0	0	4	0	0	0	0	0	
		US-001-ST04	<i>template setup</i>	Raziman	Selesai		2		0	0	2	1	0	0	0	0	0	
		US-001-ST05	<i>creation of auto approve</i>	Ali	Selesai		4		2	3	0	0	0	0	0	0	0	
BF-BM-EP01-PB02	Pengguna (warga MAMPU) BOLEH mengemaskini maklumat profil pengguna yang telah didaftarkan SUPAYA akaun dapat disahkan dan selamat.	US-002-ST01	<i>Install contact page.</i>	Ali	Selesai	5	5		0	3	0	3	0	0	0	0	0	
		US-002-ST02	<i>new table UI</i>	Raziman	Selesai		6		0	0	0	2	2	0	0	2		
		US-002-ST03	<i>implement new db format</i>	Ahmad	Selesai		8		0	2	3	0	3	1	0	0		
		US-002-ST04	<i>template setup</i>	Azhim	Selesai		8		0	0	4	2	3	0	0	0		
		US-002-ST05	<i>export configuration</i>	Ahmad	Selesai		7		0	0	0	0	4	2	0	0		
		US-002-ST06	<i>creation of auto approve</i>	Ali	Selesai		8		0	0	0	0	4	4	1	0		
		US-002-ST07	<i>audit trail</i>	Azhim	Selesai		5		0	0	0	1	2	3	1	0		
		US-002-ST08	<i>creation of new table</i>	Raziman	Selesai		5		0	0	0	0	2	0	0	2		
<b>Epic 2: Pengurusan Bilik Mesyuarat</b>																		
BF-BM-EP02-PB02	Pentadbir Bilik Mesyuarat BOLEH menyemak maklum balas penggunaan bilik mesyuarat yang diterima daripada pengguna (warga Agensi) SUPAYA mendapatkan sumber rujukan untuk melaporkan kerosakan dan perlukan pembaharuan yang berkaitan.	US-004-ST01	<i>Install contact page</i>	Azhim	Selesai	8	8		0	0	0	0	0	0	2	3		
		US-004-ST02	<i>new table UI</i>	Raziman	Selesai		6		0	0	0	0	0	0	0	0	3	
		US-004-ST03	<i>implement new db format</i>	Ali	Selesai		7		1	0	0	0	0	0	2	3	1	
		US-004-ST04	<i>template setup</i>	Ahmad	Selesai		7		0	2	0	0	0	2	0	0		
		US-004-ST05	<i>table implementation</i>	Ali	Selesai		6		3	0	0	0	0	0	3	1		
		US-004-ST06	<i>export configuration</i>	Fauzi	Selesai		5		0	0	0	2	0	3	0	3		
		US-004-ST07	<i>creation of auto approve</i>	Raziman	Selesai		6		0	0	0	3	2	0	0	0		
<b>JUMLAH STORY POINT</b>						16												
MASA SEBENAR (Jam) (a)							121		14	11	11	20	23	17	10	15		
REMAINING EFFORT (Jam) (b) (b=b hari sebelum – a hari semasa)							121		107	96	85	65	42	25	15	0		
IDEAL TREND (jam) (c) (c = c rancang – (c rancang/jumlah hari x hari semasa))							121		105.875	90.75	75.625	60.5	45.375	30.25	15.125	0		

Rajah 4-7: Artifak Sprint Backlog sehingga Hari 10



Rajah 4-8: Carta Burndown sehingga Hari ke 10

ARTIFAK SPRINT RETROSPECTIVE					
SPRINT ID	MULA	BERHENTI	TERUSKAN	LAKUKAN LEBIH LAGI	DIKURANGKAN
SP001	Tambah user demo	Memanjangkan tempoh Daily Scrum Meeting lebih daripada 15 minit.	Mengedar nota Daily Scrum Meeting untuk rujukan	Komunikasi kepada Pihak Pengurusan Tertinggi	Tiada
SP001	Menyediakan deployment plan diakhir setiap sprint	Tiada	Mengadakan demo	Tiada	Mengurangkan mesyuarat/ perbincangan yang memerlukan penglibatan pasukan pembangunan

**Rajah 4-9: Artifak Sprint Retrospective**

Komponen	Senarai Semak	Status	Catatan
Pengujian Keperluan Fungsian	1. Kod sumber selesai dibangunkan	✓	Sprint 1
	2. Kod sumber disemak	✓	Sprint 1
	3. Kod sumber digabungkan (merged)	✓	Sprint 1
	4. Melepas pengujian unit	✓	Sprint 1
	5. Melepas pengujian integrasi	✓	Sprint 1
	6. Melepas pengujian sistem dan memenuhi <i>acceptance criteria</i> bagi setiap <i>user story</i>	✓	Sprint 1
	7. Melepas pengujian penerimaan pengguna		Sprint 3
Pengujian Keperluan Bukan Fungsian	1. Melepas pengujian kualiti kod	✓	Sprint 1
	2. Melepas pengujian SAST	✓	Sprint 1
	3. Melepas pengujian prestasi	✓	Sprint 1
Penempatan	1. Penempatan ke persekitaran <i>staging</i> pada <i>sprint</i> yang melibatkan <i>product release</i> sahaja		Sprint 3
Dokumentasi	1. Artifik <i>product vision</i>	✓	Sprint 1

Komponen	Senarai Semak	Status	Catatan
	2. Dokumen laporan UAT		<b>Sprint 3</b>
	3. Artifak <i>sprint backlog</i>	✓	<b>Sprint 1</b>
	4. Artifak carta <i>burndown</i>	✓	<b>Sprint 1</b>
	5. Artifak perancangan kapasiti	✓	<b>Sprint 1</b>
	6. Artifak log <i>daily scrum</i>	✓	<b>Sprint 1</b>
	7. Artifak <i>sprint retrospective</i>	✓	<b>Sprint 1</b>

Rajah 4-10: Artifak *Definition of Done*

### Lampiran 10: Templat Kes Pengujian

ID Kes Ujian	TC-BF-BM-MP-PR-01		
Nama Kes Ujian	Mendaftar profil pengguna baharu.		
Keterangan Kes Ujian	Sebagai Pengguna (warga Jabatan Digital Negara), saya BOLEH mendaftar profil pengguna baharu SUPAYA pengguna boleh mendaftar masuk sistem tempahan bilik mesyuarat.		
ID Product Backlog	PB-BF-BM-MP-PR-01		
Pra-Syarat	Nama pengguna dan alamat email yang sah untuk didaftarkan.		
Input/ Langkah-Langkah Ujian	Jangkaan Hasil	Hasil Sebenar	Status (Lulus/Gagal)
Daftar nama dan email pengguna.	Antara muka daftar pengguna baharu dipaparkan.  Sistem akan memaparkan mesej pop-up “Pendaftaran pengguna baharu berjaya”.		

## Lampiran 11: Tahap Kritikal

**Jadual 4-1: Tahap Kritikal.**<sup>39</sup>

Bil.	Tahap Kritikal	Keterangan
1.	Tahap 1 – Tinggi	<ul style="list-style-type: none"> <li>i. <i>Defects</i> yang memberi kesan yang besar kepada fungsi utama, data atau operasi pengguna yang tiada jalan penyelesaian.</li> <li>ii. Ujian manual atau automatik tidak boleh diteruskan lagi dalam kes ujian yang berkenaan (tiada penyelesaian).</li> </ul>
2.	Tahap 2 – Sederhana	<ul style="list-style-type: none"> <li>i. <i>Defects</i> yang memberi kesan yang sederhana kepada fungsi sistem, data atau operasi pengguna dan mempunyai jalan penyelesaian.</li> <li>ii. Kes ujian atau prosedur boleh dilengkapkan, tetapi menghasilkan <i>output</i> yang salah apabila maklumat yang sah atau tidak sah dimasukkan.</li> </ul>
3.	Tahap 3 – Rendah	<ul style="list-style-type: none"> <li>i. <i>Defects</i> yang tidak mempengaruhi fungsi atau data, tidak memerlukan jalan penyelesaian dan tidak memberi kesan kepada produktiviti dan kecekapan pengguna.</li> <li>ii. Semua kes ujian dan prosedur diluluskan seperti yang tertulis, tetapi mungkin terdapat semakan kecil, perubahan kosmetik, format laporan. Kecacatan ini tidak menjelaskan pelaksanaan fungsi sistem.</li> </ul>

<sup>39</sup>

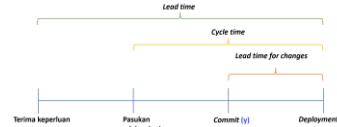
**Lampiran 12: Templat *Traceability Matrix Ujian***

ID Senario Ujian	ID Use Case	ID Kes Ujian	Keterangan Kes Ujian

### Lampiran 13: Pengukuran Tahap Pelaksanaan DevOps<sup>40</sup>

Metrik	Kekerapan Penempatan (Deployment Frequency)	Masa Pusingan Perubahan (Lead Time for Change)	Purata Masa untuk Pemulihan (Mean Time to Restore Service (MTTR))	Kadar Kegagalan Perubahan (Change Failure Rate)
<b>Metrik yang Diukur</b>	Mengukur kekerapan penempatan produk ke persekitaran produksi	Mengukur tempoh masa yang diambil untuk kod sumber didaftar (commit) sehingga ke peringkat penempatan produk	Mengukur tempoh masa yang diambil untuk memulihkan perkhidmatan apabila terdapat insiden	Mengukur peratusan kekerapan penempatan yang memerlukan pemulihan semula ke versi sebelumnya (rollback) dan pembaikan terkini (hotfix)

<sup>40</sup> Forsgren, N., Smith, D., Humble, J., & Frazelle, J. (2019). Accelerate State of DevOps 2019. [linea]. Available: <https://services.google.com/fh/files/misc/state-of-devops-2019.pdf>.

Metrik	Kekerapan Penempatan (Deployment Frequency)	Masa Pusingan Perubahan (Lead Time for Change)	Purata Masa untuk Pemulihian (Mean Time to Restore Service (MTTR))	Kadar Kegagalan Perubahan (Change Failure Rate)
<b>Formula Pengiraan</b>	Masa Pusingan Perubahan = Bilangan Penempatan Setiap Hari	<p>Masa Perubahan  <math>= x - y</math>  <math>= (\text{Masa Penempatan} - \text{Masa Perubahan})</math></p>  <p>i. Masa perubahan ialah apabila kod didaftar masuk ke dalam repositori (commit)  ii. Masa penempatan ialah apabila penempatan produk ke persekitaran produksi</p>	<p>MTTR  <math>= \text{Jumlah Masa Penyelesaian Insiden} / \text{Jumlah Insiden Berlaku}</math></p> <p>MTTR - menjumlahkan semua masa penyelesaian insiden dan membahagikannya dengan bilangan kejadian insiden. Contoh, sistem tidak berfungsi selama 30 minit dalam dua insiden berasingan.</p> $MTTR = \frac{30}{2} = 15 \text{ minit}$	<p>Kadar Perubahan  <math>= \text{Bilangan Penempatan yang Menyebabkan Insiden} / \text{Jumlah Bilangan Penempatan}</math></p> <p>Kegagalan</p>

Metrik		Kekerapan Penempatan (Deployment Frequency)	Masa Pusingan Perubahan (Lead Time for Change)	Purata Masa untuk Pemulihan (Mean Time to Restore Service (MTTR))	Kadar Kegagalan Perubahan (Change Failure Rate)
Tahap Pengukuran Prestasi	Elit	Atas permintaan (beberapa penempatan dalam sehari)	Kurang dari satu hari	Kurang dari satu jam	0-15%
	Tinggi	Minimum sehari sekali Maksimum seminggu sekali	Satu hari hingga satu minggu		
	Sederhana	Minimum seminggu sekali Maksimum sebulan sekali	Satu minggu hingga satu bulan	Kurang dari satu hari	
	Rendah	Minimum sebulan sekali Maksimum tiga bulan sekali		Lebih dari satu hari	46-60%

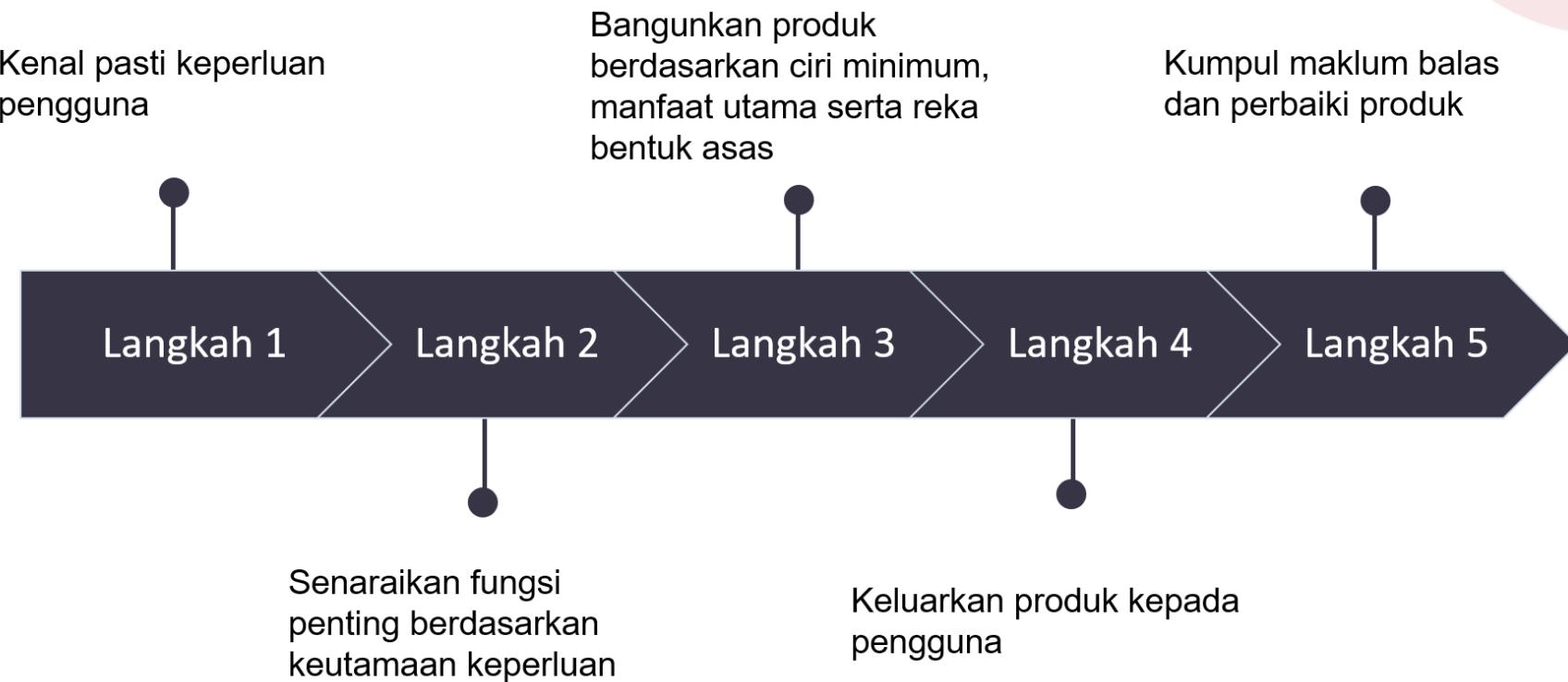
Metrik	Kekerapan Penempatan (Deployment Frequency)	Masa Pusingan Perubahan (Lead Time for Change)	Purata Masa untuk Pemulihan (Mean Time to Restore Service (MTTR))	Kadar Kegagalan Perubahan (Change Failure Rate)
<b>Faktor Penurunan Prestasi</b>	<p>i. Perubahan yang besar diperkenalkan dalam kod, kekurangan sumber manusia atau perubahan dalam struktur organisasi, dan ketidakcekapan dalam proses pembangunan.</p>	<p>i. Perubahan yang besar diperkenalkan dalam kod, keperluan pengguna yang tidak jelas, pengujian dijalankan secara manual, dan ketidakcekapan dalam proses pembangunan.</p>	<p>i. Pengujian dijalankan secara manual, kekurangan sumber manusia dalam struktur organisasi, dan ketidakcekapan dalam proses pengurusan insiden.</p>	<p>i. Kebergantungan pada proses penempatan secara manual meningkatkan risiko kesilapan,</p> <p>ii. pengujian dijalankan secara manual,</p> <p>iii. kualiti kod yang lemah sehingga sukar diselenggara,</p> <p>iv. pengenalan kod baharu membawa kepada pengujian yang lebih kompleks, dan</p> <p>v. ralat sistem aplikasi yang tidak dijangka.</p>

Metrik	Kekerapan Penempatan (Deployment Frequency)	Masa Pusingan Perubahan (Lead Time for Change)	Purata Masa untuk Pemulihan (Mean Time to Restore Service (MTTR))	Kadar Kegagalan Perubahan (Change Failure Rate)
<b>Cara Meningkatkan Prestasi</b>	<p>i. Melaksanakan amalan terbaik CI/CD secara berterusan - membolehkan peningkatan kadar penyampaian sistem aplikasi</p> <p>ii. Melaksanakan versi perubahan pada skala kecil - memudahkan pasukan mengambil tindakan penambahbaikan produk berdasarkan maklum balas dan menyelesaikan isu lebih cepat mengikut versi perubahan</p>	<p>i. Melaksanakan versi perubahan pada skala kecil - memudahkan pasukan mengambil tindakan penambahbaikan produk berdasarkan maklum balas dan menyelesaikan isu lebih cepat mengikut versi perubahan</p> <p>ii. Mengautomasikan pengujian - menggabungkan ujian automatik pada setiap peringkat pipeline CI/CD membantu mengurangkan masa penyampaian</p> <p>iii. Mengautomasikan semakan kod -</p>	<p>i. Meningkatkan pengetahuan dan kemahiran ahli pasukan - sumber manusia yang berpengetahuan boleh mengenal pasti punca masalah dengan cepat dan menyelesaikannya dengan segera</p> <p>ii. Menggunakan automasi ujian - menggabungkan ujian automatik pada setiap peringkat pipeline CI/CD membantu mengurangkan masa penyampaian dan membantu proses baik pulih dengan lebih cepat</p>	<p>i. Melaksanakan versi perubahan pada skala kecil - memudahkan pasukan mengambil tindakan penambahbaikan produk berdasarkan maklum balas dan menyelesaikan isu lebih cepat mengikut versi perubahan</p> <p>ii. Mengautomasikan pengujian - menggabungkan ujian automatik pada setiap peringkat pipeline CI/CD membantu mengurangkan masa penyampaian</p>

Metrik	Kekerapan Penempatan (Deployment Frequency)	Masa Pusingan Perubahan (Lead Time for Change)	Purata Masa untuk Pemulihan (Mean Time to Restore Service (MTTR))	Kadar Kegagalan Perubahan (Change Failure Rate)
		menggunakan semakan automatik SonarQube dan GitHub	<p>tools kod seperti dan</p> <p>iii. <b>Menyediakan/mengemas kini dokumentasi</b> - mendokumentasikan proses pengurusan insiden dengan lengkap dan teratur</p> <p>iv. <b>Menyediakan latihan</b> - Melatih ahli pasukan secara berterusan tentang proses dan cara bertindak sekiranya berlaku insiden</p> <p>v. <b>Mengautomasikan proses pengurusan insiden</b> - Mengenal pasti langkah yang berpotensi dalam proses pengurusan insiden untuk diautomasi</p>	<p>iii. <b>Mengautomasikan semakan kod</b> - menggunakan tools semakan kod automatik membantu agensi meningkatkan kualiti kod dan menjimatkan masa</p>

### Lampiran 14: Konsep MVP dalam Pembangunan Produk

a. Langkah-langkah bagi pembangunan MVP adalah seperti berikut:<sup>41</sup>



<sup>41</sup> Sabharwal, N., Rathore, R., & Agrawal, U. (2022). Introduction to Agile and DevOps. In *Hands-On Guide to AgileOps* (pp. 29-40). Apress, Berkeley, CA.

b. Contoh MVP bagi Pembangunan Platform Mesej

### Pembangunan Platform Mesej secara MVP

Mesej Asas

Gambar,  
Emoji,  
Dokumen

Video, Audio

**MVP:**

Keperluan: berkomunikasi dengan pantas

**Produk yang diperbaiki:**

Keperluan: berkomunikasi dan berkongsi maklumat dengan pantas

**Produk akhir:**

Keperluan: berkomunikasi dan berkongsi maklumat *rich format* dengan pantas

**Ciri-ciri:**

- ✓ Berfungsi
- ✓ Boleh digunakan
- ✓ Reka bentuk yang menarik

### Lampiran 15: Peranan dan Tanggungjawab dalam Pengurusan Tools dan Proses CI/CD

Peranan dan tanggungjawab dalam pengurusan *tools* dan proses CI/CD adalah seperti di bawah:

Pengurusan	Peranan	Tanggungjawab
<b>Pengurusan Konfigurasi</b>	Ketua Pasukan	<ul style="list-style-type: none"> <li>i. Menguruskan pelan dan dasar pengurusan konfigurasi, dan</li> <li>ii. mengesahkan pindaan konfigurasi tidak bercanggah dengan konfigurasi lain dalam <i>pipeline</i> CI/CD.</li> </ul>
	Jurutera Konfigurasi Automasi /	<ul style="list-style-type: none"> <li>i. Mendapatkan maklumat bagi perubahan konfigurasi,</li> <li>ii. menilai aktiviti pelaksanaan pindaan konfigurasi melalui risiko dan impak,</li> <li>iii. menilai keadaan konfigurasi sebelum dan selepas pengubahsuaian,</li> <li>iv. mengesahkan bahawa butiran konfigurasi bagi <i>pipeline</i> CI/CD adalah betul dan lengkap,</li> <li>v. mengesahkan bahawa pengubahsuaian konfigurasi yang dicadangkan mematuhi dasar pengurusan konfigurasi, jika ada,</li> <li>vi. mengkonfigurasi item ke dalam pangkalan data pengurusan konfigurasi, dan</li> <li>vii. mengesahkan bahawa butiran konfigurasi dikemas kini dengan teratur.</li> </ul>
<b>Pengurusan Pindaan</b>	Pengurus Produk	<ul style="list-style-type: none"> <li>i. Memastikan skop pindaan tidak bercanggah dengan ciri-ciri keperluan dan fungsi produk sistem aplikasi, dan</li> <li>ii. memastikan kriteria penerimaan dikemas kini setelah pindaan diluluskan.</li> </ul>
	Ketua Pasukan	<ul style="list-style-type: none"> <li>i. Mereka bentuk pendekatan strategi pengurusan pindaan,</li> <li>ii. menilai kesan serta risiko pindaan, dan</li> <li>iii. menjalankan semakan untuk menilai keputusan dan prestasi yang berkaitan dengan permintaan pindaan.</li> </ul>
	Pembangun	<ul style="list-style-type: none"> <li>i. Mengurus permintaan pindaan,</li> <li>ii. menilai impak pindaan secara berpasukan, dan</li> </ul>

Pengurusan	Peranan	Tanggungjawab
<b>Pengurusan Pelepasan</b>	Pengurus Produk	<ul style="list-style-type: none"> <li>iii. melaksanakan pindaan dan pengujian pindaan.</li> </ul>
	Pengurus Projek	<ul style="list-style-type: none"> <li>i. Menentukan skop pelepasan yang bersesuaian, dan</li> <li>ii. menakrif skop pelepasan berdasarkan MVP bersama pasukan.</li> </ul>
	Jurutera Kualiti	<ul style="list-style-type: none"> <li>i. Memastikan ciri-ciri penerimaan skop bagi pelepasan adalah seperti yang dipersetujui oleh pengurus produk,</li> <li>ii. mengesan ralat semasa proses pelepasan dan melakukan pembetulan awal, dan</li> <li>iii. memastikan pengurangan risiko pengulangan ralat dalam persekitaran produksi.</li> </ul>
	Jurutera Pelepasan	<ul style="list-style-type: none"> <li>i. Memastikan penciptaan rantaian <i>tools</i> pelepasan bagi menjayakan pelepasan,</li> <li>ii. menyelaras skop pelepasan automatik, jika ada,</li> <li>iii. memastikan pematuhan proses pelepasan,</li> <li>iv. mengawal dan mengatasi permasalahan berkaitan pelepasan, dan</li> <li>v. menyelia proses dan <i>tools</i> pelepasan bagi menyokong pelepasan berterusan.</li> </ul>
<b>Pengurusan Insiden</b>	Pengurus Projek	<ul style="list-style-type: none"> <li>i. Memastikan pengurusan insiden dilakukan mengikut proses yang dipersetujui, dan</li> <li>ii. mengkoordinasi komunikasi ahli pasukan, pengurus produk serta pihak berkepentingan.</li> </ul>
	Pembangun	<ul style="list-style-type: none"> <li>i. Mengurus insiden atau isu yang timbul semasa proses pembangunan,</li> <li>ii. mengurus insiden atau isu yang timbul dalam persekitaran produksi, dan</li> <li>iii. bekerjasama dalam menyelesaikan insiden.</li> </ul>

### Lampiran 16: Konfigurasi dalam Pelaksanaan DevOps

Antara maklumat yang dikonfigurasi dalam pelaksanaan DevOps adalah seperti berikut:

- a. Maklumat berkaitan persekitaran pembangunan, pengujian dan produksi,
- b. Maklumat berkaitan infrastruktur dan rangkaian,
- c. Senarai sumber yang diperlukan seperti *tools*, fail, dokumen dan sebagainya,
- d. Kes ujian, keperluan spesifikasi dan data ujian,
- e. Maklumat kebergantungan aplikasi terhadap mana-mana sistem atau perkhidmatan, dan
- f. Matriks capaian bagi pasukan projek, sistem yang akan dicapai, tahap keselamatan dan kawalan peringkat penguatkuasaan.

Maklumat yang dikonfigurasi perlu sentiasa dikemas kini bagi memastikan maklumat adalah terkini. Contohnya, metrik capaian perlu dikemas kini sekiranya ahli pasukan bertukar peranan semasa projek berlangsung.

### Lampiran 17: Kawalan Versi dalam Pelaksanaan DevOps

- a. Kod dan data yang diperlukan untuk membina sistem aplikasi, konfigurasi infrastruktur dan *pipeline* CI/CD hendaklah disimpan dalam repositori kawalan versi,
- b. Maklumat pengujian dan skrip ujian yang diperlukan untuk mengesahkan aplikasi, konfigurasi infrastruktur dan *pipeline* CI/CD hendaklah disimpan dalam repositori kawalan versi,
- c. Keputusan ujian dan proses semakan yang digunakan untuk menilai kualiti aplikasi, konfigurasi infrastruktur dan *pipeline* CI/CD hendaklah disimpan dalam repositori kawalan versi,
- d. Perubahan kod dan data hendaklah ditandakan dengan *metadata* bagi memudahkan carian,
- e. Papan pemuka dan sistem pemantauan hendaklah digunakan bagi menjamin keselamatan repositori pengurusan versi,
- f. Sekurang-kurangnya satu sandaran repositori perlu bersedia pada bila-bila masa,
- g. Sejarah versi yang disimpan perlu tersedia berdasarkan dua atau tiga pelepasan yang sebelumnya,
- h. Sekurang-kurangnya satu replika repositori pengurusan versi perlu dikemas kini secara berperingkat,
- i. Proses replikasi dan sandaran hendaklah disahkan secara berkala pada setiap kali pelepasan,
- j. Capaian kepada repositori perlu dikawal dengan sistem pengurusan capaian berdasarkan peranan,
- k. Pembangun perlu mendaftar masuk perubahan aplikasi ke dalam repositori kawalan versi sekurang-kurangnya sekali sehari, dan
- l. Sistem pengesahan bencana yang berupaya untuk memulihkan sistem pengurusan kawalan versi perlu tersedia dan berfungsi pada bila-bila masa.

## Lampiran 18: Senarai *Exit Criteria* bagi Aktiviti Utama Pelaksanaan DevOps

Aktiviti-aktiviti dan contoh *exit criteria* yang boleh digunakan adalah seperti berikut:

Aktiviti-aktiviti	<i>Exit criteria</i>
<b>Iteration</b>	<ul style="list-style-type: none"> <li>i. Semua cerita pengguna dalam tunggakan <i>iteration</i> telah diselesaikan,</li> <li>ii. Semua tugas reka bentuk dan tugas khas dalam <i>iteration</i> telah diselesaikan,</li> <li>iii. Isu yang kritikal dengan keutamaan tinggi telah diselesaikan, dan</li> <li>iv. Ujian regresi bagi semua ciri-ciri yang dibangunkan dalam <i>iteration</i> sebelumnya telah dilaksanakan.</li> </ul>
<b>Cerita Pengguna</b>	<ul style="list-style-type: none"> <li>i. Semua tugas yang ditakrifkan untuk cerita pengguna telah diselesaikan,</li> <li>ii. Isu berkaitan cerita pengguna telah dibincangkan, diselesaikan dan diuji semula,</li> <li>iii. Semua isu kritikal berkaitan cerita pengguna telah diselesaikan,</li> <li>iv. Ujian untuk cerita pengguna telah ditambah ke dalam ujian regresi, dan</li> <li>v. Data ujian dan konfigurasi bagi ujian automasi untuk cerita pengguna telah ditambah ke sistem pengujian automatik.</li> </ul>
<b>Pengekodan</b>	<ul style="list-style-type: none"> <li>i. Semakan kod telah dilakukan oleh rakan pembangun,</li> <li>ii. Semakan analisis statik telah dijalankan pada kod dan semua ralat dalam kod telah diselesaikan,</li> <li>iii. Kod yang ditulis mesti memenuhi kriteria penerimaan pelepasan,</li> <li>iv. Kod mesti menggunakan piawaian yang <i>standard</i>, menggunakan konvensyen pengekodan yang dipersetujui ahli pasukan, dan</li> <li>v. Kod telah disemak masuk dengan semua kebergantungan terkini.</li> </ul>
<b>Pengujian unit</b>	<ul style="list-style-type: none"> <li>i. Skrip ujian ditulis dan dikemas kini dalam sistem pengurusan ujian,</li> <li>ii. Ujian dilaksanakan dan ralat direkodkan berdasarkan cerita pengguna, dan</li> <li>iii. Ujian yang berkaitan telah ditambah ke pek pengujian regresi.</li> </ul>
<b>Semakan Kod</b>	<ul style="list-style-type: none"> <li>i. Semakan kod telah dilakukan berdasarkan cerita pengguna, dan</li> <li>ii. Maklum balas telah diambil kira dan kod diperbaiki.</li> </ul>
<b>Pengujian Automatik</b>	<ul style="list-style-type: none"> <li>i. Skrip ujian yang disediakan telah dimasukkan ke dalam ujian automasi bagi cerita pengguna yang berkaitan, dan</li> <li>ii. Skrip ujian dilaksanakan sekurang-kurangnya sekali sebelum pelepasan.</li> </ul>

## Lampiran 19: Perbandingan Teknik Pengujian Automatik TDD dan BDD

Perbandingan bagi teknik pengujian automatik TDD dan BDD adalah seperti berikut:<sup>42</sup>

Aktiviti	<b>Test-Driven Development (TDD)</b>	<b>Behaviour-Driven Development (BDD)</b>
Penerangan	Pembangun membina kes-kes ujian berdasarkan cerita pengguna, sebelum fungsi produk dibina.	Pengurus produk menerangkan tingkah laku pengguna dan penerangan ini digunakan bagi membina kes-kes ujian.
Kaedah pelaksanaan	<ul style="list-style-type: none"> <li>i. Pembangun menulis kes ujian berdasarkan keperluan pengguna,</li> <li>ii. Pengujian dilakukan ke atas aplikasi produk. Oleh kerana fungsi dalam skop ini belum dibina, keputusan ujian akan menjadi negatif atau gagal,</li> <li>iii. Pengekodan dan pengujian dilakukan oleh pembangun sehingga hasil ujian positif atau berjaya,</li> <li>iv. Pembangun <i>refactor</i> kod untuk membaiki kualiti kod, dan</li> <li>v. Fungsi produk yang berjaya diuji dihasilkan.</li> </ul>	<ul style="list-style-type: none"> <li>i. Tingkah laku pengguna dalam penggunaan produk dijelaskan kepada pembangun,</li> <li>ii. Pembangun menulis tingkah laku pengguna yang diterangkan ke dalam kes-kes ujian,</li> <li>iii. Pengekodan dan pengujian dilakukan sehingga tingkah laku pengguna seperti dalam kes ujian diperolehi,</li> <li>iv. Pembangun <i>refactor</i> kod untuk membaiki kualiti kod, dan</li> <li>v. Fungsi produk yang berjaya diuji dihasilkan.</li> </ul>

<sup>42</sup> Devops Agile Skills Association LLC. (2020, September). DASA DevOps Fundamentals (2.0.1 ed.) [E-book]. Devops Agile Skills Association LLC.

## Lampiran 20: Metrik Pemantauan DevOps

Metrik yang boleh digunakan bagi aktiviti-aktiviti pemantauan adalah seperti berikut:

Aktiviti Pemantauan	Metrik Pemantauan
<b>Prestasi Aplikasi</b>	<ul style="list-style-type: none"> <li>i. Skor Kepuasan Pengguna - mengukur tahap kepuasan pengguna berdasarkan masa tindak balas permintaan apabila berinteraksi dengan aplikasi,</li> <li>ii. Masa Tindak Balas - jangka masa yang diambil oleh pengguna aplikasi untuk menerima tindak balas daripada aplikasi,</li> <li>iii. Kadar Ralat - bilangan atau jumlah ralat aplikasi dalam tempoh masa tertentu,</li> <li>iv. Bilangan/Kadar Permintaan Transaksi - bilangan permintaan transaksi aplikasi dalam tempoh masa tertentu,</li> <li>v. CPU (<i>Central Processing Unit</i>) – penggunaan CPU asas, dan</li> <li>vi. Ketersediaan/Masa Aktif Aplikasi – tempoh masa aplikasi beroperasi dan berfungsi dalam memenuhi permintaan pengguna.</li> </ul>
<b>Pemantauan Infrastruktur</b>	<ul style="list-style-type: none"> <li>i. Aliran penggunaan keadaan infrastruktur - aliran penggunaan keadaan hos seperti penggunaan cakera, CPU, memori, <i>container</i>, penggunaan perkhidmatan sistem, sistem maya dan storan,</li> <li>ii. Garis dasar dan perbandingan penggunaan - garis dasar dan perbandingan sebelum dan selepas penggunaan infrastruktur,</li> <li>iii. Profil penggunaan pangkalan data - memprofilkan penggunaan pangkalan data infrastruktur,</li> <li>iv. Penjejak prestasi - penjejakan prestasi melalui perubahan dalam infrastruktur,</li> <li>v. Fail log amaran - fail log amaran dan rujukan ralat yang senang dan boleh dicari,</li> <li>vi. Keselamatan dan masa tindak balas infrastruktur, dan</li> <li>vii. Daya pemprosesan aplikasi (<i>throughput</i>).</li> </ul>

Aktiviti Pemantauan	Metrik Pemantauan
Pemantauan Rangkaian	<ul style="list-style-type: none"><li>i. Kesesakan rangkaian (<i>latency</i>),</li><li>ii. Metrik <i>port</i> yang berbeza,</li><li>iii. Penggunaan jalur lebar (<i>bandwidth</i>), dan</li><li>iv. Aliran pakej rangkaian (<i>network packet flow</i>).</li></ul>

## Lampiran 21: Senarai Tools DevOps

Senarai tools DevOps berdasarkan 17 kategori. Sumber rujukan adalah dari laman sesawang **Periodic Table of DevOps**. Digital.ai. (n.d.). Dimuat turun pada June 9, 2022, dari <https://digital.ai/periodic-table-of-devops-tools>.

The Periodic Table of DevOps displays 17 categories of tools, each represented by a colored square containing a tool name and its acronym. The categories are:

- AIOps/Analytics**: Aja (Atlassian Jira Align)
- Artifact/Package Management**: Daa (Digital AI Agility)
- Cloud**: Pv (Planview)
- Collaboration**: Br (Broadcom Rally)
- Configuration Automation**: In (Instana)
- Containers**: Dd (Datadog)
- Continuous Integration**: Ja (JFrog Artifactory)
- Database Management**: Aws (AWS)
- Deployment**: Sl (Slack)
- Enterprise Agile Planning**: Mt (Microsoft Teams)
- Issue Tracking/ITSM**: Rha (Red Hat Ansible)
- Release Management**: Ht (HashiCorp Terraform)
- Security**: Rho (Red Hat OpenShift)
- Source Control Management**: Lb (Liquidbase)
- Testing**: Dk (Docker)
- Value Stream Management**: Dp (Delphix)
- Git**: Ow (OWASP ZAP)
- Serverless/PaaS**: Dad (Digital AI Deploy)
- Sonatype Nexus IQ**: Sni (Sonatype Nexus IQ)
- Aqua Security**: Aq (Aqua Security)
- CloudBees Flow**: Cfr (CloudBees Flow)
- BMC RLM**: Brl (BMC RLM)
- Github**: Gls (GitLab SCM)
- AWS CodePipeline**: Acp (AWS CodePipeline)
- AWS Lambda**: Hv (HashiCorp Vault)
- UrbanCode Release**: Ur (UrbanCode Release)
- Altlassian Bitbucket**: Abb (Altlassian Bitbucket)
- Micro Focus Fortify SCA**: Ff (Micro Focus Fortify SCA)
- Azure Functions**: Al (AWS Lambda)
- Compuware ISPW**: Ci (Compuware ISPW)
- Harvest**: Vc (Veracode)
- SonarQube**: Sr (SonarQube)
- Octopus Deploy**: Od (Octopus Deploy)
- Synopsys Black Duck**: Sk (Synopsys Black Duck)
- Checkmarx SAST**: Od (Checkmarx SAST)
- Heroku**: Cx (Checkmarx SAST)
- Subversion**: He (Heroku)
- PortSwigger Burp Suite**: Sv (Subversion)
- Google Firebase**: Gf (Google Firebase)
- Cloud Foundry**: Cf (Cloud Foundry)

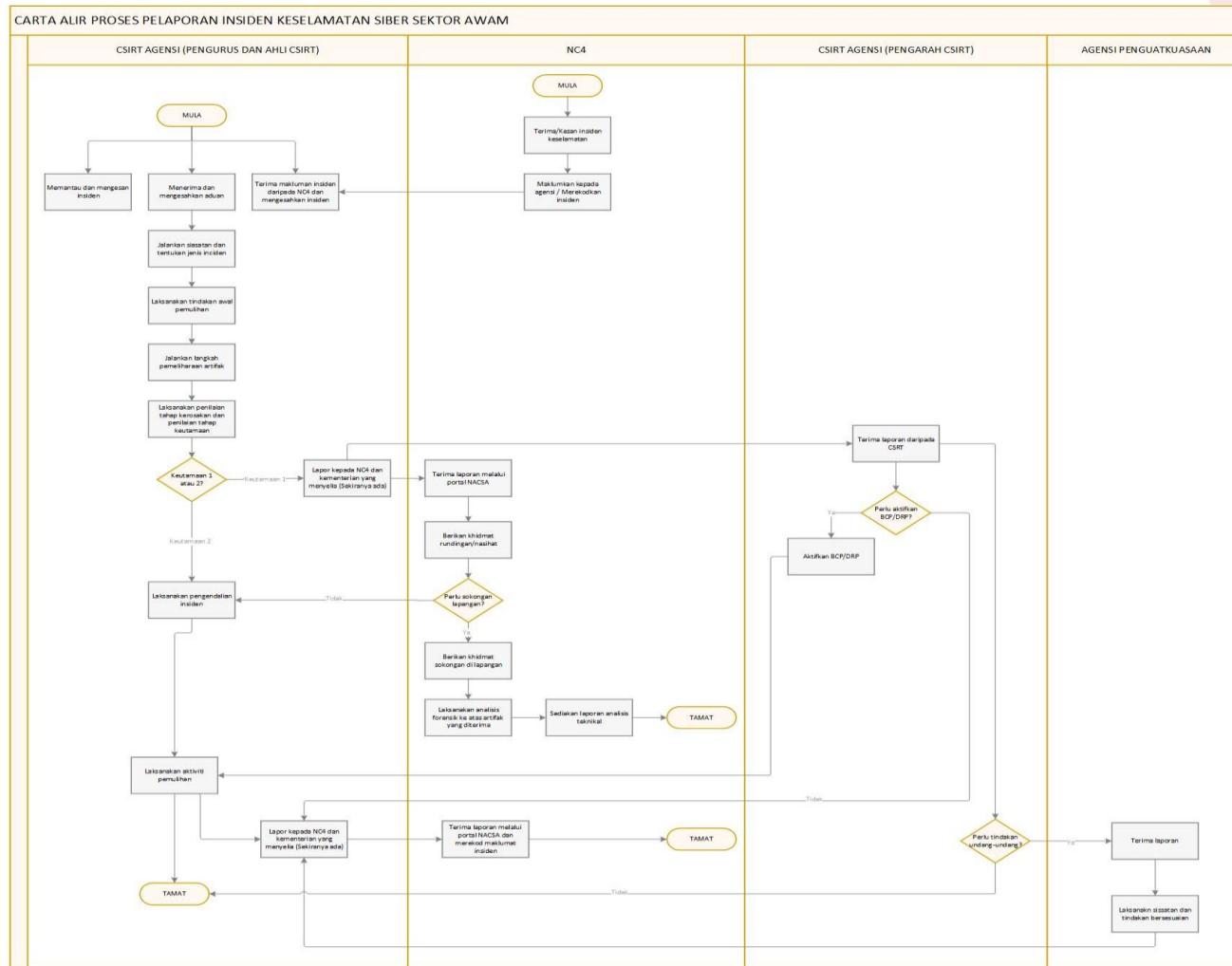
Legend for tool types:

- Os: Open Source
- Fr: Free
- Fm: Freemium
- Pd: Paid
- En: Enterprise

91) Os	92) En	93) Os	94) Os	95) Fm	96) Os	97) Pd	98) Os	99) En	100) Os	101) En	102) Pd	103) En	104) Pd	105) Os
Jn (Jenkins)	Azc (Azure DevOps Code)	Glc (GitLab CI)	Tr (Travis CI)	Cc (CircleCI)	Mv (Maven)	Ab (Atlassian Bamboo)	Gd (Gradle)	Acb (AWS CodeBuild)	Aj (Atlassian Jira)	Bi (BMC Helix ITSM)	At (Atlassian Triflow)	Sw (ServiceNow)	Td (TOPdesk)	Pd (PagerDuty)
106) Fr	107) Pd	108) Fr	109) Fr	110) Pd	111) En	112) En	113) Os	114) Fr	115) Fr	116) Pd	117) En	118) En	119) En	120) Os
Tt (Tricentis Tosca)	Nn (Neotys NeoLoad)	Se (Selenium)	Ju (JUnit)	Sl (Sauce Labs)	Ct (Compuware Topaz)	Ap (Appium)	Sq (Squash TM)	Cu (Cucumber)	Jm (JMeter)	Pa (Parasoft)	Dai (Digital.ai)	Tp (Tasktop)	Pr (Plutora)	Gl (GitLab)

ColabNetVersionOne, XebiaLabs, Arvan, Numerify & Expertise are now Digital.ai

## Lampiran 22: Pelaporan Insiden



**Lampiran 23: Templat Pelaporan Status Kemajuan Pembangunan Produk**



# **LAPORAN STATUS KEMAJUAN PRODUK**

## **NAMA PRODUK**

(Sertakan nama modul di bawah nama sistem sekiranya dokumen disediakan secara berasingan bagi setiap modul di bawah sistem yang sama)

<b>NAMA AGENSI</b>	:	
<b>NAMA AGENSI INDUK</b>	:	
<b>TARIKH DOKUMEN</b>	:	
<b>VERSI DOKUMEN</b>	:	

**DOKUMEN DISEDIAKAN OLEH**

Nama	Jawatan	Tandatangan	Tarikh

**DOKUMEN DISEMAK OLEH**

Nama	Jawatan	Tandatangan	Tarikh

**DOKUMEN DISAHKAN OLEH**

Nama	Jawatan	Tandatangan	Tarikh

## KANDUNGAN

<b>KANDUNGAN.....</b>	<b>1</b>
<b>SENARAI JADUAL.....</b>	<b>1</b>
<b>SENARAI RAJAH.....</b>	<b>1</b>
1. <b>Ringkasan Status Keseluruhan.....</b>	<b>2</b>
2. <b>Carta Kemajuan Kerja (Carta Release Burn-up).....</b>	<b>4</b>
3. <b>Kemajuan <i>Product Backlog</i>.....</b>	<b>8</b>
3.1. <b>Ringkasan Status Kemajuan <i>Product Backlog</i>.....</b>	<b>8</b>

## SENARAI JADUAL

<b>Jadual 1: Ringkasan Status Aktiviti Utama.....</b>	<b>3</b>
<b>Jadual 2: Status Kemajuan <i>Story Points</i> untuk Setiap <i>Sprints</i>.....</b>	<b>5</b>
<b>Jadual 3: Formula dan Contoh Pengiraan.....</b>	<b>6</b>
<b>Jadual 4: Ringkasan Status Senarai <i>Product Backlog</i>.....</b>	<b>6</b>

## SENARAI RAJAH

<b>Rajah 1: Carta Release Burn-Up.....</b>	<b>4</b>
--	----------

## 1. Ringkasan Status Keseluruhan

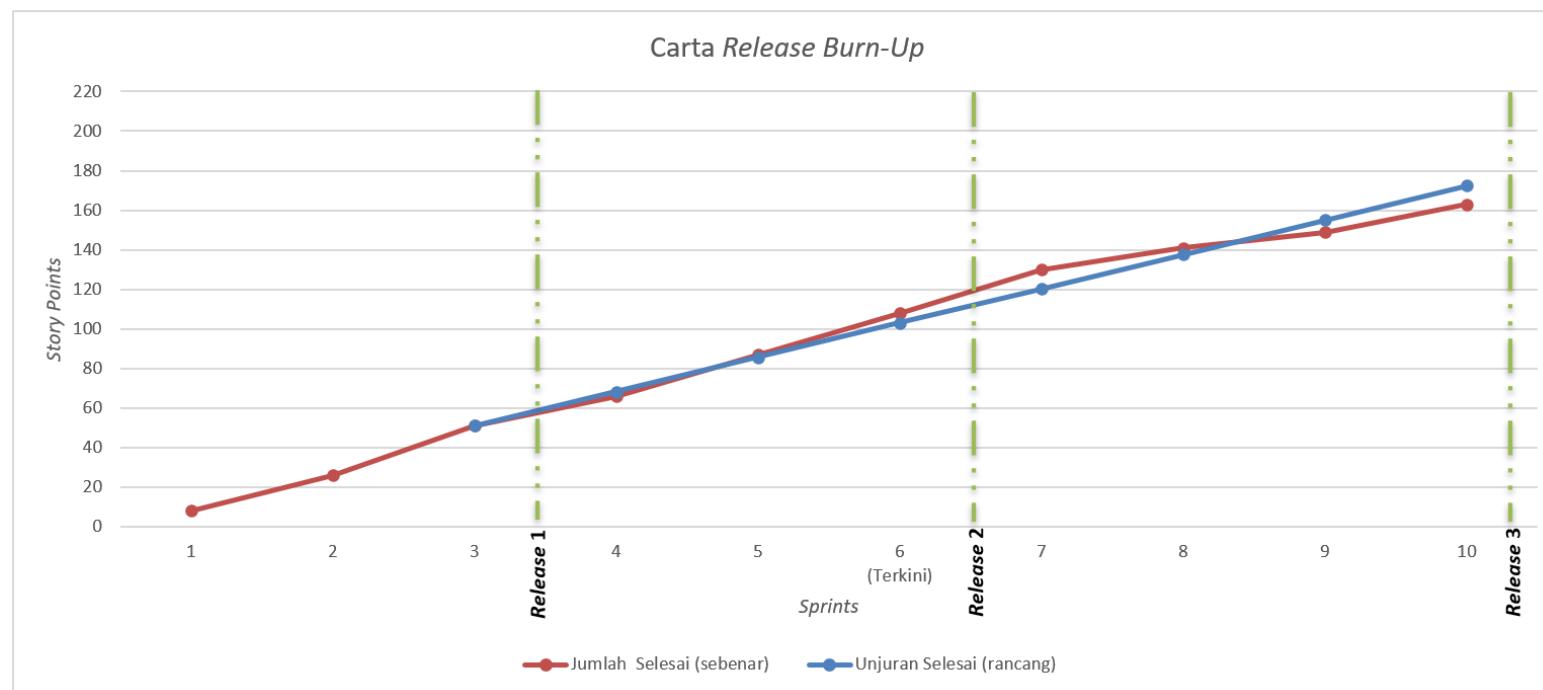
**Jadual 1: Ringkasan Status Aktiviti Utama**

Bil.	Aktiviti Utama	Penerangan	Status*	Tarikh Dirancang Mula	Tarikh Mula Sebenar	Tarikh Dirancang Tamat	Tarikh Tamat Sebenar	Pencapaian %	Catatan
1.	Kajian Keperluan Bisnes	Penyedian BRS, <i>product vision</i> , hierarki fungsi bisnes, <i>epics</i> , <i>user stories</i> , serta <i>product backlogs</i>	Ikut Jadual	03/07/2023	03/07/2023	21/07/2023	21/07/2023	100%	
2.	<i>Sprint 0</i>	Penyediaan persekitaran pembangunan serta penetapan peranan ahli pasukan.	Ikut Jadual	24/07/2023	24/07/2023	04/08/2023	04/08/2023	100%	
3.	<i>Sprint 1</i>	Pembangunan Modul Pengurusan Pengguna	Ikut Jadual	07/08/2023	07/08/2023	18/08/2023	18/08/2023	100%	
4.	<i>Sprint 2</i>	Pembangunan Modul Pengurusan Bilik Mesyuarat	Ikut Jadual	21/08/2023	21/08/2023	01/09/2023	01/09/2023	85%	
5.	<i>Sprint 3</i>	Pembangunan Modul Pengurusan Tempahan	Ikut Jadual	04/09/2023	04/09/2023	15/09/2023	15/09/2023		
6.	<i>Release 1</i>	UAT, Penempatan ke persekitaran staging	Ikut Jadual	18/09/2023	18/09/2023	29/09/2023	29/09/2023		
7.	<i>Sprint 4</i>	Pembangunan Modul Paparan Dashboard dan Jana Laporan	Ikut Jadual	02/10/2023	02/10/2023	13/10/2023	13/10/2023		
8.	<i>Sprint n</i>	Pembangunan Modul xx	Ikut Jadual	06/11/2023	06/11/2023	17/11/2023	17/11/2023		
9.	<i>Release n</i>	UAT, Penempatan ke persekitaran produksi dan Go Live	Ikut Jadual						

\*Status: █ Lewat Jadual    █ Ikut Jadual    █ Perlu Kawalan

## 2. Carta Kemajuan Kerja (Carta Release Burn-up)

Carta *release burn-up* pada Rajah 1 digunakan untuk menjak status kemajuan pasukan dengan membandingkan kerja yang dirancang berbanding kerja yang telah dilaksanakan. Jadual 2 menunjukkan jumlah tugas yang telah disiapkan oleh pasukan dalam jangka masa tertentu. Tugasan merujuk kepada *story points* atau boleh dirujuk sebagai *number of hours* yang digunakan.



Rajah 1: Carta *Release Burn-Up*

**Jadual 2: Status Kemajuan *Story Points* untuk Setiap *Sprints***

	Sprints	Story Points Completed	Jumlah Selesai (sebenar)	Unjuran Selesai (rancang)
Selesai	1	9	8	
	2	18	26	
	3	25	51	51
	4	15	66	68
	5	21	87	86
	6 (Terkini)	21	108	103
Akan Dilaksanakan	7	22	130	120
	8	11	141	138
	9	8	149	155
	10	14	163	172

Purata Story Points Selesai untuk 3 Sprints  
Permulaan (*Velocity of Sprint*):

17

### Jadual 3: Formula dan Contoh Pengiraan

Perkara	Formula Pengiraan
<b>Jumlah Selesai (sebenar)</b>	i. Diplot berdasarkan jumlah <i>story points</i> yang disiapkan pada setiap <i>sprint</i> .
<b>Unjuran Selesai (rancang)</b>	i. Diplot berdasarkan jumlah <i>story points</i> yang diunjurkan untuk setiap <i>sprint</i> . ii. Unjuran diperoleh daripada <i>velocity of sprint</i> . iii. <i>Velocity of sprint</i> digunakan untuk menganggarkan unjuran berapa <i>story points</i> yang dapat disiapkan pada setiap <i>sprints</i> seterusnya.
<b>Velocity of Sprint</b>	$\text{Sprint Velocity} = \frac{\text{Jumlah Story Points Selesai}}{\text{Bilangan Sprints}}$ $\text{Sprint Velocity} = \frac{9(\text{Sprint 1}) + 18(\text{Sprint 2}) + 25(\text{Sprint 3})}{3}$ $\text{Sprint Velocity} = 17$ <p>Penerangan formula:</p> <ul style="list-style-type: none"> <li>• <b>Jumlah Story Points Selesai</b> merujuk kepada jumlah <i>story points</i> untuk semua <i>user story</i> atau tugas yang diselesaikan dalam <i>sprint</i>.</li> <li>• <b>Bilangan Sprints</b> merujuk kepada jumlah bilangan <i>sprints</i> yang diselesaikan.</li> </ul>

### 3. Kemajuan Product Backlog

#### 3.1. Ringkasan Status Kemajuan *Product Backlog*

**Jadual 4: Ringkasan Status Senarai *Product Backlog***

PRODUCT BACKLOG ID	TURUTAN KEUTAMAAN	USER STORY			STORY POINT	SPRINT NO	STATUS	SERAHAN %
<b>Epic 1 : Pengurusan Pengguna</b>								
BF-BM-EP01-PB01	1	Pengguna (warga JDN)	<b>BOLEH</b> mendaftar profil pengguna baru	<b>SUPAYA</b> boleh mendaftar masuk sistem tempahan bilik mesyuarat	3	Sprint 1	Selesai	100%
BF-BM-EP01-PB02	2	Pengguna (warga JDN)	<b>BOLEH</b> mengemaskini maklumat profil pengguna yang telah didaftarkan	<b>SUPAYA</b> akaun dapat disahkan dan selamat.	5	Sprint 1	Selesai	100%
<b>Epic 2 : Pengurusan Bilik Mesyuarat</b>								
BF-BM-EP02-PB01	4	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menyelenggara maklumat bilik mesyuarat dengan mewujudkan rekod bilik yang baharu atau mengemaskini rekod sedia ada.	<b>SUPAYA</b> mendapatkan maklumat bilik mesyuarat yang tepat.	13	Sprint 2	Selesai	100%

PRODUCT BACKLOG ID	TURUTAN KEUTAMAAN	USER STORY			STORY POINT	SPRINT NO	STATUS	SERAHAN %
BF-BM-EP02-PB02	3	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menyemak maklum balas penggunaan bilik mesyuarat yang diterima daripada pengguna (warga Agensi)	<b>SUPAYA</b> mendapatkan sumber rujukan untuk melaporkan kerosakan dan perlukan pembaikan yang berkaitan.	8	Sprint 2	Selesai	100%
BF-BM-EP02-PB03	5	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menyediakan aduan kerosakan	<b>SUPAYA</b> hantar aduan kerosakan tersebut ke Sistem Selenggara Aset untuk tindakan seterusnya.	13	Sprint 2	Selesai	100%
BF-BM-EP02-PB04	6	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menerima notifikasi status aduan pembaikan kerosakan dari Sistem Selenggara Aset	<b>SUPAYA</b> kemaskini maklumat status bilik mesyuarat.	8	Sprint 2	Dalam Tindakan	20%
<b>Epic 3 : Pengurusan Tempahan</b>								
BF-BM-EP03-PB01	7	Pengguna (warga JDN)	<b>BOLEH</b> menyemak kekosongan bilik mesyuarat	<b>SUPAYA</b> melakukan permohonan tempahan penggunaan bilik mesyuarat yang	8	Sprint 3		

PRODUCT BACKLOG ID	TURUTAN KEUTAMAAN	USER STORY			STORY POINT	SPRINT NO	STATUS	SERAHAN %
				ingin digunakan pada tempoh masa tertentu.				
BF-BM-EP03-PB02	8	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menyemak permohonan bilik mesyuarat yang memerlukan tindakan kelulusan	<b>SUPAYA</b> memberi kelulusan bagi setiap permohonan tempahan	8	Sprint 3		
BF-BM-EP03-PB03	9	Pengguna (warga JDN)	<b>BOLEH</b> melihat status semasa pemohonan tempahan yang telah dilakukan	<b>SUPAYA</b> tahu berkenaan status dan maklumat permohonan permohonan yang telah dilakukan.	5	Sprint 3		
BF-BM-EP03-PB04	10	Pengguna (warga JDN)	<b>BOLEH</b> melihat status pemohonan bilik mesyuarat	<b>SUPAYA</b> membatalkan permohonan sekiranya perlu.	5	Sprint 3		
BF-BM-EP03-PB05	11	Pengguna (warga JDN)	<b>BOLEH</b> menerima notifikasi untuk memberikan maklum balas penggunaan bilik mesyuarat setelah selesai penggunaan bilik.	<b>SUPAYA</b> melengkapkan maklum balas penggunaan bilik mesyuarat setelah selesai penggunaan bilik.	8	Sprint 3		
<b>Epic 4 : Paparan Dashbord dan Jana Laporan</b>								

PRODUCT BACKLOG ID	TURUTAN KEUTAMAAN	USER STORY			STORY POINT	SPRINT NO	STATUS	SERAHAN %
BF-BM-EP04-PB01	12	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> melihat dashboard maklumat status rekod terkinin dan rumusan statistik bilik, tempahan, penggunaan dan status bilik mesyuarat	<b>SUPAYA</b> gunakan maklumat tersebut untuk tujuan pelaporan	8	Sprint 4		
BF-BM-EP04-PB02	13	Pentadbir Bilik Mesyuarat	<b>BOLEH</b> menjana dan memaparkan laporan statistik terperinci tempahan, penggunaan dan penyelenggaraan bilik mesyuarat	<b>SUPAYA</b> mencetak laporan yang dijana atau dieksport laporan untuk disimpan	8	Sprint 4		



# **SUMBER RUJUKAN**

## SUMBER RUJUKAN

- Amaradri, A. S., & Nutalapati, S. B. (2016). Continuous Integration, Deployment and Testing in DevOps Environment.
- Artac, M., Borovssak, T., Di Nitto, E., Guerriero, M., & Tamburri, D. A. (2017, May). DevOps: introducing infrastructure-as-code. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)* (pp. 497-498). IEEE.
- B. S. Farroha and D. L. Farroha, "A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment," in 2014 IEEE Military Communications Conference, 2014, pp. 288–293.
- Bahadori, K., & Vardanega, T. (2018, March). DevOps meets dynamic orchestration. In *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment* (pp. 142-154). Springer, Cham.
- Bolhuis, W. T. C. (2021). *How Can (Large Scale) Agile be Effectively Adopted and Scaled Up in Dutch Public Sector Organisations* (Master's thesis, University of Twente).
- Bucena, I., & Kirikova, M. (2017). Simplifying the DevOps Adoption Process. *BIR Workshops*, 1-15.
- Cagle, R., Rice, T., & Kristan, M. (2018). *DevOps for federal acquisition*. MITRE CORP BEDFORD MA.
- Cherinka, R., Foote, S., Burgo, J., & Prezzama, J. (2022). The Impact of Agile Methods and “DevOps” on Day 2+ Operations for Large Enterprises. In *Intelligent Computing* (pp. 1068-1081). Springer, Cham.
- Chrassis MB, Konrad M, Shrum S. CMMI for development: guidelines for process integration and product improvement. Pearson Education 2011.
- CNCF. (Diakses Mac 01, 2022). *CNCF Cloud Native Interactive Landscape*. Retrieved from CNCF Cloud Native Interactive Landscape: <https://landscape.cncf.io/>
- D. K. Taft, "Rackspace Survey Spotlights DevOps Business Benefits: Top 6 Findings," eWeek, pp. 1–1, Nov. 2014.
- E. Diel, S. Marczak, and D. S. Cruzes, "Communication Challenges and Strategies in Distributed DevOps," in 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), 2016, pp. 24–28.
- Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *Ieee Software*, 33(3), 94-100.
- Faustino, J., Pereira, R., Alturas, B., & Da Silva, M. M. (2020). Agile information technology service management with DevOps: An incident management case study. *Agile information technology service management with DevOps: An incident management case study*, (4), 339-389.
- Garcia VC. RiSE reference model for software reuse adoption in Brazilian companies. From web site [http://ivanmachado.com.br/research/rise/thesis/files/2010\\_ViniciusGarcia\\_phd.pdf](http://ivanmachado.com.br/research/rise/thesis/files/2010_ViniciusGarcia_phd.pdf).
- Government Accountability Office. Powner, D. (2012). Software Development: Effective Practices and Federal Challenges in Applying Agile Methods, (GAO Publication No. 12-681). Washington, D.C.: U.S. Government Printing Office.

- Government Accountability Office. (2020). *Agile Assessment Guide: Best Practices for Agile Adoption and Implementation*, (GAO Publication No. 20-590G). Washington, D.C.: U.S. Government Printing Office.
- Gruver, G. (2016). Starting and Scaling DevOps in the Enterprise. In G. Gruver, *Starting and Scaling DevOps in the Enterprise* (pp. 7-45).
- Hannah Moss, Francesca El-Attrash, Joshua Hill - Your Guide to DevOps in Government [Report]. <https://docs.broadcom.com/doc/your-guide-to-devops-in-government>
- Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016, May). What is DevOps? A systematic mapping study on definitions and practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016* (pp. 1-11).
- Johannes Wettinger, Uwe Breitenbücher, and Frank Leymann, DevOpSlang – Bridging the Gap between Development and Operations
- J. Wettinger, U. Breitenbücher, and F. Leymann, “DevOpSlang—bridging the gap between development and operations,” in European Conference on Service-Oriented and Cloud Computing, 2014, pp. 108–122.
- J. Wettinger, U. Breitenbücher, and F. Leymann, “Standards-based devops automation and integration using tosca,” in Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, 2014, pp. 59–68.
- Khan SU. Software outsourcing vendors' readiness model (SOVRM). Ph.D. dissertation, School Comput. Math., Keele Univ., Keele, U.K: 2011.
- Kupiainen, E., Mäntylä, M. V., & Itkonen, J. (2015). Using metrics in Agile and Lean Software Development—A systematic literature review of industrial studies. *Information and software technology*, 62, 143-163.
- L. Evenstad, “Delivering Success with Devops,” Computer Weekly, pp. 23–26, Dec. 2015.
- Lappi, T., Karvonen, T., Lwakatare, L. E., Aaltonen, K., & Kuvaja, P. (2018). Toward an improved understanding of agile project governance: A systematic literature review. *Project Management Journal*, 49(6), 39-63.
- Leite, L., Kon, F., Pinto, G., & Meirelles, P. (2020, June). Platform teams: An organizational structure for continuous delivery. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops* (pp. 505-511).
- M. A. McCarthy, L. M. Herger, S. M. Khan, and B. M. Belgodere. Composable devops: Automated ontology based devops maturity analysis. In Services Computing (SCC), 2015 IEEE International Conference on, pages 600–607, June 2015.
- M. Hüttermann, *DevOps for Developers*. Apress, 2012.
- Mantovani Fontana, R., & Marczak, S. (2020). Characteristics and Challenges of Agile Software Development Adoption in Brazilian Government. *Journal of technology management & innovation*, 15(2), 3-10.
- Merkow, M. S. (2022). *Practical Security for Agile and DevOps*. Auerbach Publications.
- Morales, J. A., Yasar, H., & Volkman, A. (2018, May). Implementing DevOps practices in highly regulated environments. In *Proceedings of the 19th International Conference on Agile Software Development: Companion* (pp. 1-9).

- Niazi M, Wilson D, Zowghi D. A maturity model for the implementation of software process improvement: an empirical study. *J Syst Softw.* 2005;74:155-172.
- Nicole Blake Johnson, Isaac Constans, Mark Hensch, Katie Malone, Catherine Andrews - Your Guide to DevOps in Government Today [Report]. <https://go.govloop.com/rs/231-DWB-776/images/DevOps-in-Government-Today.pdf>
- Northern, C., Mayfield, K., Benito, R., & Casagni, M. (2010). *Handbook for implementing agile in department of defense information technology acquisition*. MITRE CORP MCLEAN VA.
- Ogala, J. O. (2022). A Complete Guide to DevOps Best Practices. *International Journal of Computer Science and Information Security (IJCSIS)*, 20(2).
- Plant, O. H. (2019). *DevOps under control: development of a framework for achieving internal control and effectively managing risks in a DevOps environment* (Master's thesis, University of Twente).
- Prestes, M., Parizi, R., Marczak, S., & Conte, T. (2020, June). On the use of design thinking: A survey of the Brazilian agile software development community. In *International Conference on Agile Software Development* (pp. 73-86). Springer, Cham.
- Project Management Institute. (2021). *A guide to the Project Management Body of Knowledge (PMBOK guide)* (7th ed.). Project Management Institute.
- Rafi S, Yu W, Akbar MA. RMDevOps: a road map for improvement in DevOps activities in context of software organizations. In Proceedings of the Evaluation and Assessment in Software Engineering. 2020:413-418.
- Rafi S, Yu W, Akbar MA, Mahmood S, Alsanad A, Gumaie A (2020). Readiness model for DevOps implementation in software organizations. *Journal of Software Evolution and Process*. 33. 10.1002/smri.2323.
- Riungu-Kalliosaari, L. M. (2016, November). DevOps adoption benefits and challenges in practice: A case study. In *International conference on product-focused software process improvement*, pp. 590-597.
- Rodríguez, P., Mäntylä, M., Oivo, M., Lwakatare, L. E., Seppänen, P., & Kuvaja, P. (2019). Advances in using agile and lean processes for software development. In *Advances in Computers* (Vol. 113, pp. 135-224). Elsevier.
- S. Jones, J. Noppen, and F. Lettice, "Management Challenges for DevOps Adoption Within UK SMEs," in Proceedings of the 2Nd International Workshop on Quality-Aware DevOps, New York, NY, USA, 2016, pp. 7–11.
- S. W. Hussaini, "Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through systems approach," in 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), 2014, pp. 178–183.
- Sharma, S. (2017). *The DevOps adoption playbook: a guide to adopting DevOps in a multi-speed IT enterprise*. John Wiley & Sons.
- Skelton, M., & Pais, M. (2019). *Team topologies: organizing business and technology teams for fast flow*. IT Revolution.
- S7 C. Preimesberger, "10 Essential Steps to Mapping Your DevOps Journey," eWeek, pp. 1–1, Mar. 2016
- Taft, Darryl K. Rackspace Survey Spotlights DevOps Business Benefits: Top 6 Findings
- W. B.S. Farroha, D.L. Farroha, A Framework for Managing Mission Needs, Compliance and Trust in the DevOps Environment

- Wiedemann, A., Wiesche, M., & Krcmar, H. (2019, June). Integrating development and operations in cross-functional teams-toward a devops competency model. In *Proceedings of the 2019 on Computers and People Research Conference* (pp. 14-19).
- Yarlagadda, R. T. (2018). How Public Sectors Can Adopt the DevOps Practices to Enhance the System. *International Journal of Emerging Technologies and Innovative Research* ([www.jetir.org/](http://www.jetir.org/) UGC and issn Approved), ISSN, 2349-5162.
- DevOps Implementation Plan: Benefits, Guide, Definition [Report]. <https://codeit.us/blog/devops-implementation-plan#developing-a-devops-implementation-plan>
- Enterprise Architect Framework for DevOps Implementation Strategies [Report].  
<https://www.veritis.com/solutions/devops/implementation-strategy-tools-collaboration/>
- Dr. Gopala Krishna Behara, Enterprise Architect Framework for DevOps Adoption [Report].  
<https://www.wipro.com/blogs/dr-gopala-krishna-behara/ea-framework-for-devops-adoption/>



# **PENGHARGAAN**

## PENGHARGAAN

JDN merakamkan setinggi-tinggi penghargaan dan ucapan terima kasih kepada semua pihak yang telah memberikan komitmen dan kerjasama dalam usaha merangka, menyelaras dan menerbitkan dokumen Rangka Kerja DevOps dalam Pembangunan Sistem Aplikasi Sektor Awam.

Semoga penerbitan dokumen ini akan dapat meningkatkan lagi keberkesanan pembangunan sistem melalui inisiatif pendigitalan dalam sektor perkhidmatan awam di Malaysia.

## PENASIHAT

Ts. Dr. Fazidah binti Abu Bakar

## PENGARAH PROJEK

Nuha binti Musa

## PENGURUS PROJEK

Dr. Rozaida binti Mohd Darus

### SME/JAMINAN KUALITI

Ts. Nik Zalbiha binti Nik Mat  
Subramani a/l Nagaiah  
Dr. Ali bin Seman  
Ts. Mohd Nawawi bin Mustafa  
Ts. Dr. Nurul Aisyah Sim binti Abdullah  
Ts. Dr. Mohamed Hairul bin Othman  
Ts. Mohd Hasiady bin Yasin  
Ts. Rosaimieza binti Mat Adam  
Ts. Ibrahim bin Abdulah  
Ts. Dr. Mohamed Hairul bin Othman  
Chua Swee Suan

### MUNSYI MUDA BAHASA

Siti Noorasmah binti Hashim

## AGENSI TERLIBAT

### KEMENTERIAN PENDIDIKAN MALAYSIA

Harlina binti Othman  
Mohd Fauzie bin Mokhtar

### KEMENTERIAN PERTAHANAN MALAYSIA

Dr. Mariayee a/p Doraisamy  
Nani Lee Yer Fui  
Mohd Arif bin Jamal  
Muhammad Amirul bin Zulkapri

### KEMENTERIAN PENGANGKUTAN MALAYSIA

Ts. Subramani a/l Paiduthaly  
Nurul Bariah binti Abu Bakar

### KEMENTERIAN KESIHATAN MALAYSIA

Siti Zaileha binti A Salam  
Mohammad Amir Ashraaf bin Amran

### KEMENTERIAN DALAM NEGERI

Shuhaida binti Yahaya

### KEMENTERIAN KERJA RAYA

Zalina binti Mohamed Fauzi @ Jazuli  
Raja Azizah binti Raja Shariffuddin

### JABATAN IMIGRESEN MALAYSIA

Azurawati binti Abdul Halim

### JABATAN PERANGKAAN MALAYSIA

Ts. Sabrina binti Abdul Rahim  
Khairum Anisa binti Ahmad Samuri

## PASUKAN PROJEK

Zamzuri bin Abdullah  
Shahiruddin bin Taib  
Rasidah binti Ghazali  
Suhaila binti Norsham  
Elisah Satim  
Haslidah binti Halim  
Khairulbariah binti Mohammad Lukman  
Rohani binti Haji Yusoff  
Normaizura binti Rosani  
Ts. Khairul Ashraf bin Basul Hak  
Ts. Mohd Syukor bin Abdul  
Ts. Farid Iqbal bin Ibrahim  
Nurul Ayuni binti Hj. Juhari  
Kholijah binti Norbib  
Adam Muhammad Richard  
Ts. Roth Camdessus Anak Robert  
Azmil bin Abdul Rahman  
Mohd Nizam bin Kamaruddin  
Siti Aminah Hanum binti Che Kob  
Shahrizan bin Md Razak  
Sharifah Afizan binti Syed Azizan  
Mazri bin Abu Hassan  
Lailati binti Mohd  
Harisfazilah bin Jamel  
Surazi bin Mohd Jais

Andrew Yap Kok Keong  
Mohamad Hanif bin Khodit  
Hasliana binti Ramlan  
Hanizah binti Baharum  
Mohd Zaki bin Zubahir  
Abdullah Fathi bin Muhamad Azmi  
Hisham bin Aderis  
Azmi bin Md Nor  
Nur Fazhila binti Syafwan Effendi  
Nurul Shazira binti Saifuzzaman  
Wan Normimi Farhana binti Wan Mansor  
Zudiana binti Zulkapli  
Muhammad Zulhairi bin Zakaria

**JABATAN PERKHIDMATAN AWAM**  
Norazwana binti Mohamed Kamari  
Ts. Hariadi bin Hinta  
Ahmad Naufal bin Jamri

**UNIT PENYELARASAN PELAKSANAAN**  
Abdullah bin Muhammad

**KEMENTERIAN PEMBANGUNAN  
USAHAWAN DAN KOPERASI**  
Rasyida binti Zainun

**UNIVERSITI TEKNOLOGI MARA**  
Habsah binti Mihat

## INDUSTRI

**TENANG KOMUNITI SDN. BHD.**  
Mohamad Zulfahmy bin Roslan

**APOGEEK CREATIVE SOLUTIONS**  
Muhammad Fauzilkamil bin Zainuddin

**TAMING TECH SDN. BHD.**  
Khairil Iszuddin bin Ismail

**BESPOKE INTEGRATED SDN. BHD.**  
Mohd Saidy bin Mohd Yusof

**SOLSIS (M) SDN. BHD.**  
Ts. Fariq bin Abdullah

**UNIVERSITI KUALA LUMPUR**  
Dr. Shafiza binti Mohd Shariff

**GERMAN MALAYSIAN INSTITUTE**  
Ts. Dahlia Asyiqin binti Ahmad Zainaddin

**GEMAS LESTARI SDN. BHD.**  
Ts. Dr. Mohd Fazli Azran

**KEMENTERIAN DIGITAL  
JABATAN DIGITAL NEGARA**

Bangunan MKN Embassy Techzone,  
Blok B, No. 3200 Jalan Teknokrat 2,  
63000 Cyberjaya, Sepang, Selangor  
Darul Ehsan  
Malaysia



[www.osdec.gov.my](http://www.osdec.gov.my)