



**Unit Pemodenan Tadbiran dan Perancangan
Pengurusan Malaysia (MAMPU)
Jabatan Perdana Menteri**

Rangka Kerja Pelaksanaan DevOps dalam Pembangunan Sistem Aplikasi Sektor Awam

Hak Cipta Terpelihara.

© Unit Pemodenan Tadbiran dan Perancangan Pengurusan Malaysia (MAMPU), 2022.

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 MAMPU.

Diterbitkan oleh:

Unit Pemodenan Tadbiran dan Perancangan Pengurusan Malaysia (MAMPU)
Aras 6, Setia Perdana 2
Kompleks Setia Perdana
Pusat Pentadbiran Kerajaan Persekutuan
62502 Putrajaya
Malaysia
Tel : 603 8000 8000
Fax : 603 8872 3163
Web : <http://www.mampu.gov.my>

KANDUNGAN

KANDUNGAN	ii
SENARAI JADUAL	iv
SENARAI RAJAH	v
SENARAI LAMPIRAN	vi
AKRONIM	vii
TAKRIFAN	viii
PRAKATA KETUA PENGARAH MAMPU	xiii
PRAKATA TIMBALAN KETUA PENGARAH (ICT) MAMPU	xvi
RINGKASAN EKSEKUTIF	1
BAB 1 PENDAHULUAN	5
BAB 2 SKOP DOKUMEN	9
BAB 3 PENGENALAN KEPADA DEVOPS	11
3.1. PERSEKITARAN SEMASA PEMBANGUNAN SISTEM APLIKASI	11
3.2. DEFINISI DEVOPS	11
3.3. OBJEKTIF PELAKSANAAN DEVOPS	13
3.4. FASA-FASA DALAM PELAKSANAAN DEVOPS	13
3.5. MANFAAT PELAKSANAAN DEVOPS	15
BAB 4 RANGKA KERJA	19
4.1. TERAS 1: PENERAPAN PRINSIP DAN BUDAYA	23
4.1.1. Prinsip DevOps	23
4.1.2. Budaya DevOps dan Penerapan dalam Pasukan	25
4.2. TERAS 2: PENGADAPTASIAN METODOLOGI	30
4.2.1. Penetapan Matlamat dan Objektif	31
4.2.2. Penilaian Persekitaran Semasa bagi Ketersediaan Pelaksanaan DevOps	33
4.2.3. Proses Peralihan	40
4.2.4. Pengukuran Tahap Pelaksanaan	44
4.3. TERAS 3: PENGUKUHAN TADBIR URUS	46
4.3.1. Struktur Pelaporan Organisasi	47
4.3.2. Struktur Pasukan	48
4.3.3. Ahli Pasukan	52
4.4. TERAS 4: PEMANTAPAN PELAKSANAAN DAN PEMANTAUAN	56
4.4.1. Perancangan	56
4.4.2. Pelaksanaan	66

4.4.3. Kawalan	67
4.4.4. Penyelenggaraan	70
4.4.5. Pemantauan	71
4.5. TERAS 5: PEMERKASAAN TEKNOLOGI	73
4.5.1. Infrastruktur <i>Pipeline</i> DevOps	73
4.5.2. Persekitaran <i>Pipeline</i> DevOps	77
4.5.3. <i>Tools</i> DevOps	81
4.5.4. Penggunaan <i>Tools</i> untuk <i>Pipeline</i> DevOps.....	84
BAB 5 PENUTUP	88
LAMPIRAN.....	91
SUMBER RUJUKAN.....	112
PENGHARGAAN	118

SENARAI JADUAL

Jadual 4-1: Penerangan Berkaitan Aspek-Aspek Budaya dan Cadangan Aktiviti....	26
Jadual 4-2: Contoh Templat Penetapan Matlamat S.M.A.R.T	33
Jadual 4-3: Model Kematangan DevOps	36
Jadual 4-4: Metodologi Pelaksanaan Proses Peralihan kepada Amalan DevOps secara Berperingkat.....	41
Jadual 4-5: Jadual Pengukuran Prestasi Tahap Pelaksanaan DevOps dari Segi Penyampaian dan Operasi Sistem Aplikasi	45
Jadual 4-6: Cadangan Keanggotaan Jawatankuasa Pengurusan Produk	47
Jadual 4-7: Peranan, Tugas dan Kemahiran Ahli Pasukan.....	53
Jadual 4-8: Pengurusan dan Pemantauan dalam DevOps	56
Jadual 4-9: Contoh Pelan Pelepasan.....	58
Jadual 4-10: Aktiviti Pengurusan Konfigurasi.....	60
Jadual 4-11: Aktiviti Pengurusan Pindaan.....	62
Jadual 4-12: Aktiviti Pengurusan Insiden	63
Jadual 4-13: Aktiviti Pengurusan Pelepasan.....	65
Jadual 4-14: Aliran Tugas <i>Tools</i> Mengikut Fasa DevOps	85

SENARAI RAJAH

Rajah 3-1: Fasa-Fasa dalam Pelaksanaan DevOps	14
Rajah 4-1: Rangka Kerja Pelaksanaan DevOps	19
Rajah 4-2: Prinsip Asas Pelaksanaan DevOps	20
Rajah 4-3: Pemboleh Daya Pelaksanaan DevOps	21
Rajah 4-4: Teras Pelaksanaan DevOps.....	22
Rajah 4-5: Prinsip DevOps.....	23
Rajah 4-6: Budaya DevOps untuk Mencapai Enam Prinsip DevOps	25
Rajah 4-7: Metodologi Asas dalam Pelaksanaan DevOps.....	30
Rajah 4-8: Kaedah S.M.A.R.T. dalam Penetapan Matlamat dan Objektif.....	31
Rajah 4-9: Keluk Kematangan Agensi (Sumber: ACT-IAC)	34
Rajah 4-10: Proses Peralihan kepada Pelaksanaan DevOps.....	40
Rajah 4-11: Struktur Pelaporan dalam Pelaksanaan DevOps	47
Rajah 4-12: Topologi Struktur Pasukan DevOps	49
Rajah 4-13: Gambaran Pelaksanaan Topologi 3	50
Rajah 4-14: Gambaran Pelaksanaan Topologi 4	51
Rajah 4-15: Cadangan Keahlian Pasukan	52
Rajah 4-16: Perkembangan Epik ke Cerita Pengguna.....	57
Rajah 4-17: Langkah - Langkah bagi Penyediaan Pelan Pelepasan	64
Rajah 4-18: Contoh Carta Burndown dan Carta Gantt.....	68
Rajah 4-19: Contoh WBS.....	68
Rajah 4-20: Infrastruktur <i>Pipeline</i> DevOps.....	74
Rajah 4-21: Pembinaan Persekitaran ke Arah Pelaksanaan <i>Pipeline</i> DevOps.....	77
Rajah 4-22: <i>Tools</i> untuk Persekitaran <i>Pipeline</i> DevOps	81
Rajah 4-23: Penggunaan <i>Tools</i> Mengikut Fasa DevOps	84

SENARAI LAMPIRAN

Lampiran 1: Metrik Pengukuran Tahap Pelaksanaan DevOps	92
Lampiran 2: Konsep MVP dalam Pembangunan Produk	98
Lampiran 3: Peranan dan Tanggungjawab dalam Pengurusan <i>Tools</i> dan Proses CI/CD	100
Lampiran 4: Konfigurasi dalam Pelaksanaan DevOps	103
Lampiran 5: Kawalan Versi dalam Pelaksanaan DevOps	104
Lampiran 6: Senarai <i>Exit Criteria</i> bagi Aktiviti Utama Pelaksanaan DevOps	105
Lampiran 7: Perbandingan Teknik Pengujian Automatik TDD dan BDD	106
Lampiran 8: Metrik Pemantauan DevOps	107
Lampiran 9: Senarai <i>Tools</i> DevOps	109

AKRONIM

ACT- IAC	<i>American Council for Technology-Industry Advisory Council</i>
DASA	<i>DevOps Agile Skills Association</i>
ICT	<i>Information and Communication Technology</i>
MVP	<i>Minimum Viable Product</i>
PPP	Pelan Pengurusan Projek
WBS	<i>Work Breakdown Structure</i>

TAKRIFAN

Agile	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.
Blameless post mortem	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).
Cerita Pengguna	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.
Container	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.
Definition of done (DoD)	Kriteria dan syarat yang telah dipersetujui oleh pasukan dan pemilik produk sebelum projek atau cerita pengguna, boleh dianggap selesai.
Epik	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 tugasan terperinci berdasarkan keperluan pengguna.
Hutang teknikal	<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.
Infrastructure as a Code (IaC)	Pengautomasian penyediaan infrastruktur melalui konfigurasi skrip atau kod yang boleh diubah dengan pantas bagi mempercepatkan penyediaan infrastruktur dalam pembangunan sistem aplikasi.

<i>Iteration</i>	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 <i>Scrum</i>.
Konvensyen pengekodan	Garis panduan untuk bahasa pengaturcaraan yang merangkumi gaya pengaturcaraan, sintaks, amalan dan kaedah bagi mengekalkan konsistensi pengekodan bahasa tersebut.
<i>Lean</i>	Amalan pengurusan untuk meningkatkan kecekapan dan keberkesanan dengan mengelakkan pembaziran.
<i>Mean Time to Acknowledge (MTTA)</i>	Metrik bagi mengukur purata masa yang diperlukan oleh pasukan untuk bertindak balas terhadap aduan, gangguan atau insiden.
<i>Mean Time to Resolve (MTTR)</i>	Metrik bagi mengukur purata masa yang diperlukan oleh sistem untuk mendapatkan semula prestasi operasi normal setelah berlaku kegagalan sistem.
<i>Microservices</i>	Arkitektur secara modular bagi memastikan setiap perkhidmatan aplikasi tidak bergantung kuat antara satu sama lain.
<i>Minimum Viable Product (MVP)</i>	Konsep pembangunan sistem aplikasi secara iteratif berdasarkan maklum balas berterusan. MVP menghasilkan sistem aplikasi dengan fungsi minimum serta memberikan nilai kepada pengguna.
<i>Orchestration</i>	Proses bagi mengkoordinasi proses automasi dan konfigurasi yang digunakan dalam <i>pipeline</i> CI/CD.
<i>Pair programming</i>	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.
<i>Peer review</i>	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.
Pelan Pengurusan Projek (PPP)	Dokumen yang menjadi rujukan utama dalam mengurus dan mengawal projek sistem aplikasi ICT dalam sektor awam merujuk kepada PPrISA.

Pipeline	Satu set proses dan <i>tools</i> yang membolehkan pembangun dan pasukan operasi bekerjasama dan berkolaborasi dalam DevOps. <i>Pipeline</i> CI/CD berlaku dalam seluruh fasa DevOps dan menitikberatkan pengautomasian proses pembangunan sistem aplikasi.
Produk	Sistem aplikasi yang dibangunkan untuk menyokong fungsi bisnes organisasi.
Projek	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.
Retrospektif	Mesyuarat pasukan pada akhir <i>iteration</i> untuk membincangkan kaedah kerja pasukan. Retrospektif perlu berlaku semasa projek supaya penambahbaikan yang terhasil boleh digunakan dan relevan dengan kerja yang akan datang pada projek yang sama. Aspek retrospektif seperti masa, ahli pasukan dan format mesyuarat harus konsisten.
Semakan analisis statik	Analisis yang digunakan untuk mengenal pasti masalah keselamatan semasa fasa pengkodan. 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.
Telemetry	Log data yang terhasil 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 anomali berlaku.
Tools DevOps	<i>Tools</i> yang digunakan bagi menggalakkan prinsip berterusan DevOps. Setiap fasa DevOps mempunyai <i>tools</i> dengan fungsi yang tersendiri serta terbahagi mengikut kesesuaian dalam kematangan pelaksanaan DevOps. <i>Tools</i> DevOps yang dipilih bergantung kepada model dan infrastruktur dalam pembangunan sistem aplikasi.
Tunggakan tugas	Senarai keutamaan tugas yang perlu diusahakan oleh pasukan dalam <i>iteration</i> untuk memenuhi keperluan pengguna.

PRAKATA



PRAKATA KETUA PENGARAH MAMPU

**Assalamualaikum Warahmatullahi Wabarakatuh
dan Salam Sejahtera.**

Dengan nama Allah yang Maha Pemurah lagi Maha Mengasihani.

Alhamdulillah, syukur ke hadrat Allah yang Maha Esa kerana dengan limpah kurnia dan izin-Nya jua, 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 visi dan misi kerajaan bagi meningkatkan kecekapan penyampaian perkhidmatan awam melalui pendekatan digital dan teknologi terkini.

Saya ingin mengucapkan tahniah dan syabas kepada ahli pasukan projek yang telah bersungguh-sungguh memerah 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.

Arus globalisasi dan persaingan dalam kalangan negara-negara dunia telah memberi kesan kepada landskap penyampaian perkhidmatan sektor awam di Malaysia. Sektor awam kini berdepan dengan pelbagai rintangan ekoran perubahan persekitaran yang dinamik serta peningkatan harapan rakyat dan komuniti perniagaan. Situasi ini memerlukan penjawat awam bertindak dengan lebih responsif dan inovatif dalam menyampaikan perkhidmatan kepada pelanggan.

Dalam usaha untuk mengatasi gelombang pasaran dunia yang tidak menentu dan pada masa yang sama, keperluan sektor awam untuk memenuhi kehendak dan ekspektasi rakyat yang semakin meningkat, kerajaan telah melancarkan Rangka Tindakan Ekonomi Digital Malaysia (MyDigital) bagi mentransformasikan Malaysia menjadi negara berpendapatan tinggi menjelang 2025 melalui inisiatif pendigitalan. Berlandaskan kepada aspirasi tersebut, MAMPU telah membentangkan Pelan Strategik Pendigitalan Sektor Awam (PSPSA) 2021-2025 yang 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.

Sebagai peneraju pemodenan perkhidmatan awam, MAMPU telah mengorak langkah ke hadapan dalam usaha menggerakkan dan menjayakan hasrat tersebut. Pembangunan dan pelaksanaan beberapa inisiatif pendigitalan baharu telah diketengahkan, antaranya ialah inisiatif pelaksanaan DevOps, iaitu salah satu komponen strategik yang dicadangkan dalam Pelan Strategik MAMPU 2021-2025 melalui Teras Strategik 1: Penerokaan Idea Baharu Pemodenan Perkhidmatan Awam.

Pelaksanaan DevOps akan memberi nilai kepada ahli pasukan dan membantu agensi-agensi dalam memperkasakan pembangunan sistem aplikasi sektor awam. Justeru itu, saya menyeru agar kandungan dokumen ini dijadikan rujukan agensi untuk memastikan penyampaian perkhidmatan sentiasa pada tahap yang optimum.

Adalah menjadi harapan saya, agar rangka kerja ini dapat dimanfaatkan sepenuhnya sebagai panduan kepada semua agensi kerajaan dan penjawat awam khususnya, dalam pelaksanaan DevOps bagi pembangunan sistem aplikasi sektor awam. Saya 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.

Datuk Seri Dr. Yusof Bin Ismail

KETUA PENGARAH MAMPU



PRAKATA TIMBALAN KETUA PENGARAH (ICT) MAMPU

**Assalamualaikum Warahmatullahi Wabarakatuh
dan Salam Sejahtera.**

Dengan nama Allah yang Maha Pemurah lagi Maha
Mengasihani.

Segala puji bagi Allah kerana dengan limpah izin-
Nya, dokumen Rangka Kerja Pelaksanaan DevOps
bagi Pembangunan Sistem Aplikasi Sektor Awam
telah dapat disiapkan dan diterbitkan dengan
jayanya. Dokumen ini akan menjadi landasan asas
dan panduan kepada pasukan projek pembangunan
sistem aplikasi kerajaan untuk mengaplikasikan
pendekatan DevOps dalam aktiviti pembangunan
masing-masing.

Penyampaian perkhidmatan yang berkesan dan mesra pengguna adalah penting bagi
memastikan peranan agensi kekal relevan sejajar dengan perubahan dan kehendak semasa.
Menyedari pentingnya transformasi digital bagi meningkatkan kecekapan dalam penyampaian
perkhidmatan kerajaan, MAMPU selaku peneraju agenda pendigitalan sektor awam sentiasa
komited menerajui pelbagai inisiatif pendigitalan dan pemodenan sistem penyampaian
perkhidmatan awam yang bersepadu serta berfokuskan rakyat.

Justeru, Rangka Kerja Pelaksanaan DevOps Dalam Pembangunan Sistem Aplikasi Sektor
Awam telah dibangunkan bagi memastikan agensi sektor awam membangunkan sistem
aplikasi yang berkualiti serta dapat meningkatkan kecekapan penyampaian perkhidmatan
Kerajaan. Di samping itu, melalui pelaksanaan DevOps, manfaat daripada teknologi baharu
dapat dioptimumkan agar perkhidmatan yang disediakan terus relevan seiring dengan arus
perkembangan semasa dan kekal berdaya saing pada masa hadapan.

DevOps ialah anjakan budaya yang memerlukan visi, perancangan, penerimaan pihak pengurusan dan kolaborasi yang erat bagi menjayakan kaedah yang lebih bersepadu dalam membangunkan dan menyediakan sistem aplikasi. Pelaksanaan DevOps bertujuan memudahkan dan mempercepatkan proses kerja serta meningkatkan hubungan antara pasukan berlainan dalam organisasi dengan bekerja sebagai satu pasukan. Pendekatan ini menggalakkan kolaborasi sesebuah pasukan dalam menghasilkan sistem aplikasi dalam masa yang ditetapkan dan memenuhi keperluan pihak berkepentingan dalam usaha meningkatkan kualiti perkhidmatan. Menyedari kelebihan pembangunan sistem melalui pendekatan DevOps, rangka kerja ini telah dibangunkan sebagai rujukan asas bagi membantu agensi melaksanakan projek-projek pembangunan sistem di agensi masing-masing dengan lebih berkesan.

Saya mengambil kesempatan ini untuk merakamkan ucapan tahniah dan setinggi-tinggi penghargaan serta terima kasih kepada semua pihak yang terlibat termasuk Pasukan Projek DevOps dan Pasukan Perunding ICT dari Bahagian Perundingan ICT, MAMPU di atas usaha gigih dan komitmen yang tinggi dalam membangunkan rangka kerja ini. Semoga dengan terhasilnya dokumen rangka kerja DevOps ini, agensi sektor awam dapat memantapkan pembangunan sistem aplikasi kerajaan dan menyumbang kepada peningkatan kualiti penyampaian perkhidmatan sektor awam, secara amnya.

Sekian, terima kasih.

Dr. Fazidah Binti Abu Bakar

Menanggung Tugas

TIMBALAN KETUA PENGARAH (ICT) MAMPU



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.

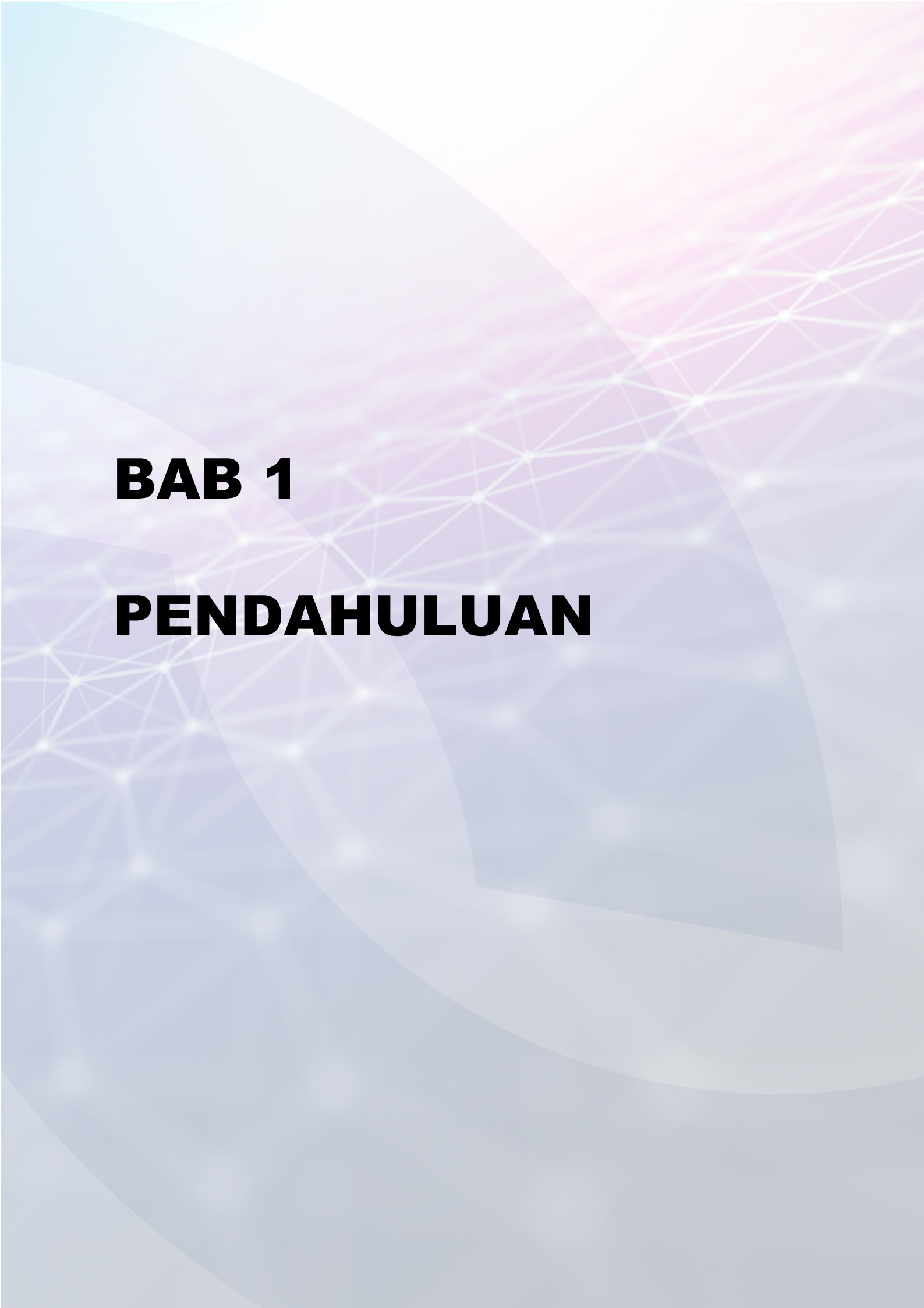
Rangka kerja ini mengariskan komponen yang perlu diambil kira oleh kementerian dan agensi kerajaan dalam melaksanakan DevOps bagi pembangunan sistem aplikasi secara dalaman. Unit Pemodenan Tadbiran dan Perancangan Pengurusan Malaysia (MAMPU) telah membangunkan rangka kerja ini bagi mengurus dan memastikan semua aktiviti pembangunan sistem yang dilaksanakan menggunakan pendekatan DevOps mematuhi panduan yang telah digariskan. Struktur dokumen ini dibahagikan kepada lima bab, seperti berikut:

- a. **Bab 1** menerangkan tujuan dan latar belakang dokumen ini,
- b. **Bab 2** menerangkan skop dokumen rangka kerja pelaksanaan DevOps bagi pembangunan sistem aplikasi sektor awam dan kumpulan sasaran,
- c. **Bab 3** memberikan penerangan terhadap definisi, objektif pelaksanaan dan manfaat DevOps,
- d. **Bab 4** menjelaskan tentang rangka kerja pelaksanaan DevOps bagi pembangunan sistem aplikasi sektor awam yang merangkumi penerapan prinsip dan budaya, pengadaptasian metodologi, pengukuhan tadbir urus, pemantapan pelaksanaan dan pemantauan, serta pemerkasaan teknologi, dan
- e. **Bab 5** merumuskan kandungan dokumen rangka kerja ini, faktor-faktor yang menyumbang kepada kejayaan pelaksanaan DevOps dan harapan terhadap penggunaan dokumen ini.

Rangka kerja pelaksanaan DevOps ini merangkumi lima teras utama yang terdiri daripada penerapan prinsip dan budaya, pengadaptasian metodologi, pengukuhan tadbir urus, pemantapan pelaksanaan dan pemantauan, serta pemeraksanaan teknologi. 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 kolaboratif.

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 *Agile* serta memastikan produk yang terhasil adalah lebih berkualiti dalam memenuhi ekspektasi 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, proses-proses yang terlibat dalam pembangunan sistem aplikasi akan dilaksanakan secara automatik dengan menggunakan *tools* atau perisian tertentu, seterusnya mempercepatkan tempoh masa pembangunan sistem aplikasi dan memudahkan aktiviti penyelenggaraan.

MAMPU 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, perbincangan kumpulan fokus dengan agensi sektor awam yang telah mula melaksanakan DevOps. Selain daripada itu, semakan kualiti dan penyelarasan bersama *Subject Matter Expert* (SME) berkaitan dengan panduan pengurusan projek dan pembangunan sistem aplikasi turut dilaksanakan. Kajian ini lebih menfokuskan 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.

Dokumen ini perlu dibaca bersama dengan dokumen dasar dan garis panduan berkaitan yang telah dikeluarkan oleh 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.

BAB 2

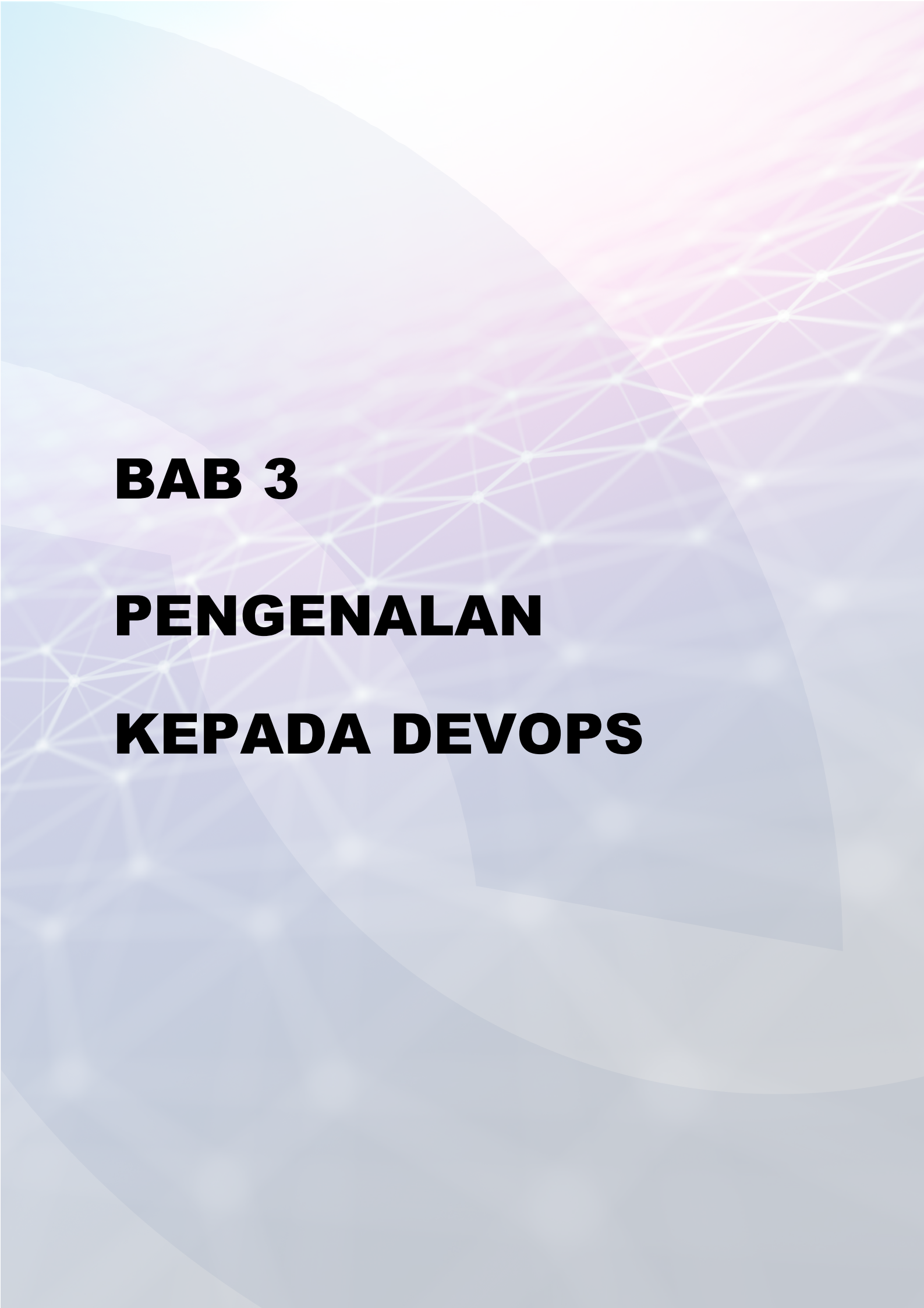
SKOP DOKUMEN

BAB 2 SKOP DOKUMEN

Skop pelaksanaan DevOps adalah khusus untuk pembangunan sistem aplikasi secara dalaman oleh pegawai teknikal sektor awam. Dokumen ini juga boleh digunakan untuk pembangunan sistem aplikasi baharu atau penambahbaikan/peningkatan 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.



BAB 3

PENGENALAN

KEPADA DEVOPS

BAB 3 PENGENALAN KEPADA DEVOPS

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.

3.1. PERSEKITARAN SEMASA PEMBANGUNAN SISTEM APLIKASI

Kajian kebolehlaksanaan DevOps dalam pembangunan sistem aplikasi sektor awam telah dilaksanakan oleh pasukan projek DevOps MAMPU 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, MAMPU turut menganjurkan beberapa siri bengkel pada Januari hingga Julai 2022 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.

3.2. 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, 2020¹)

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, 2020²)

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

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

3.3. OBJEKTIF PELAKSANAAN DEVOPS

Objektif pelaksanaan DevOps adalah seperti berikut:

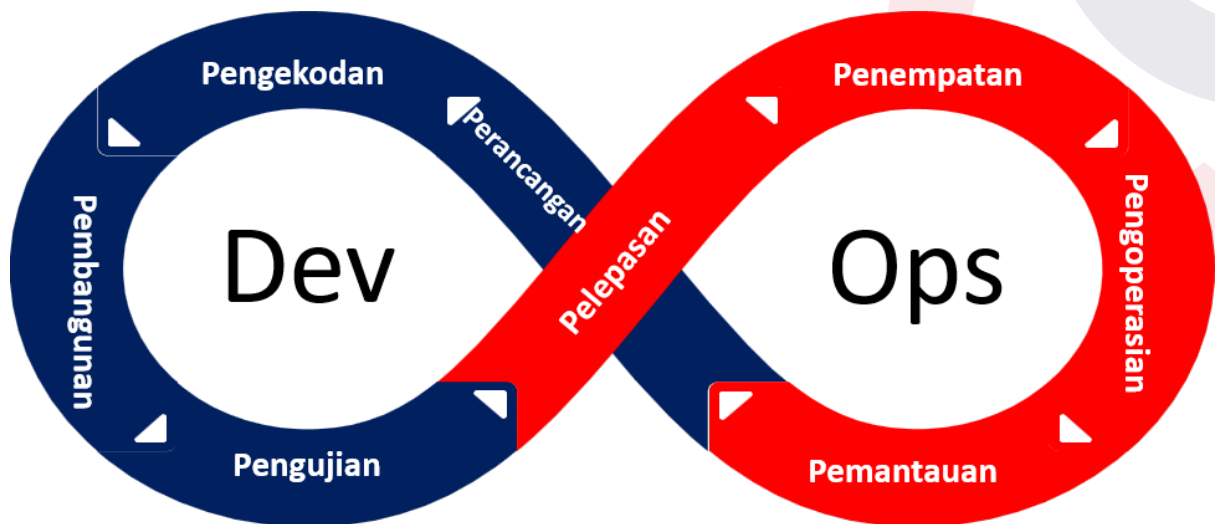
- a. Meningkatkan hubungan kerjasama antara pasukan pembangun dan pasukan operasi menerusi amalan kerja secara kolaboratif dan komunikatif,
- b. Mempercepatkan proses pembangunan sistem aplikasi bagi meningkatkan tahap penyampaian perkhidmatan digital di agensi, dan
- c. Meningkatkan tahap kualiti dan 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'.

3.4. FASA-FASA DALAM PELAKSANAAN DEVOPS

Pelaksanaan DevOps membentuk kitaran dan digunakan untuk mengenal pasti fasa-fasa dalam kitaran hayat pembangunan sistem aplikasi. Terdapat lapan fasa iteratif dalam kitaran hayat DevOps yang merangkumi keupayaan, proses dan teknik yang penting untuk pembangunan sistem aplikasi. Fasa-fasa DevOps adalah seperti di Rajah 3-1.³ Fasa 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 fasa kitar hayat DevOps adalah sangat penting untuk memastikan proses pembangunan sistem aplikasi dapat dilaksanakan dengan lancar dan berkesan.

³ 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>.



Rajah 3-1: Fasa-Fasa dalam Pelaksanaan DevOps

a. Fasa Perancangan (Plan)

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

b. Fasa Pengekoden (Code)

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

c. Fasa Pembangunan (Build)

Fasa ini melibatkan kompilasi dan integrasi kod. Kod yang telah *commit* akan dikompil dan diintegrasikan ke *branch* utama.

d. Fasa Pengujian (Test)

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

e. Fasa Pelepasan (Release)

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

f. Fasa Penempatan (Deploy)

Fasa ini melibatkan penempatan sistem aplikasi ke persekitaran produksi.

g. Fasa Pengoperasian (Operate)

Fasa ini melibatkan sistem aplikasi *go-live* dan aktiviti pengoperasian sistem akan diambil alih sepenuhnya oleh pasukan operasi.

h. Fasa Pemantauan (Monitor)

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

3.5. MANFAAT PELAKSANAAN DEVOPS

Manfaat yang diperolehi daripada pelaksanaan DevOps adalah seperti berikut:⁴

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*.

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.

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

d. Masa penyampaian yang optimum

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

e. Penjimatan 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).



BAB 4

RANGKA KERJA

BAB 4 RANGKA KERJA

Terdapat lima teras dalam rangka kerja ini iaitu Penerapan Prinsip dan Budaya, Pengadaptasian Metodologi, Pengukuhan Tadbir Urus, Pemantapan Pelaksanaan dan Pemantauan serta Pemerkasaan Teknologi. Rajah 4-1 memaparkan komposisi teras dalam melaksanakan amalan DevOps bagi pembangunan sistem aplikasi di sektor awam.

RANGKA KERJA PELAKSANAAN DEVOPS SEKTOR AWAM



Rajah 4-1: Rangka Kerja 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 4-2.

Fokus kepada amalan integrasi, pengujian, penempatan dan penyampaian 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



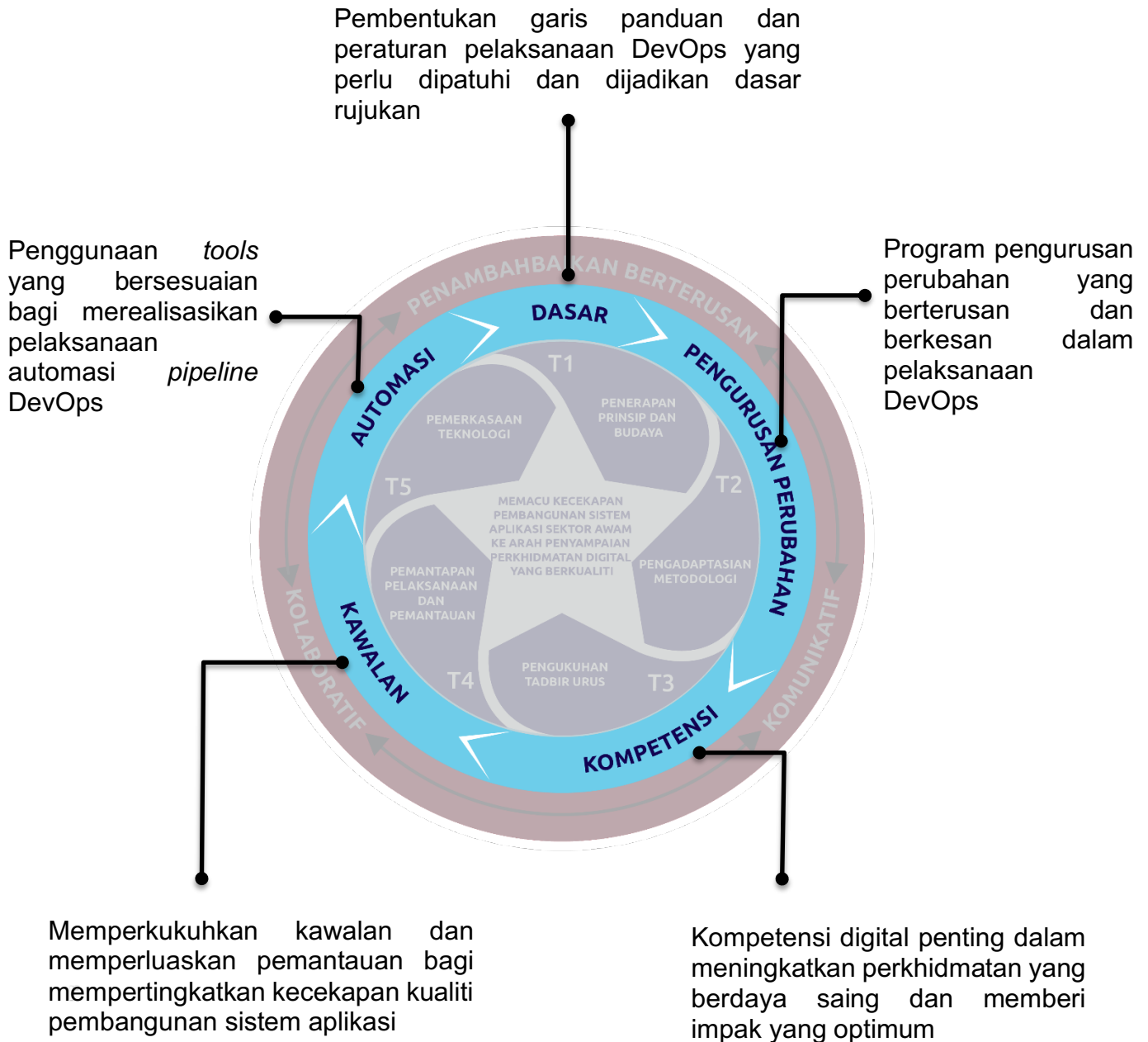
Jalinan usaha sama strategik di antara pasukan bagi tujuan perkongsian sumber, pengalaman, dan kepakaran. Budaya kerja secara kolaboratif akan menggerakkan semua ahli pasukan seperti pemilik produk, pasukan pembangun dan pasukan operasi untuk bertanggungjawab sepenuhnya dalam pembangunan sistem

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 4-2: Prinsip Asas Pelaksanaan DevOps

PEMBOLEH DAYA

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



Rajah 4-3: Pemboleh Daya Pelaksanaan DevOps

TERAS

Lima teras utama bagi pelaksanaan DevOps sektor awam adalah seperti di Rajah 4-4:



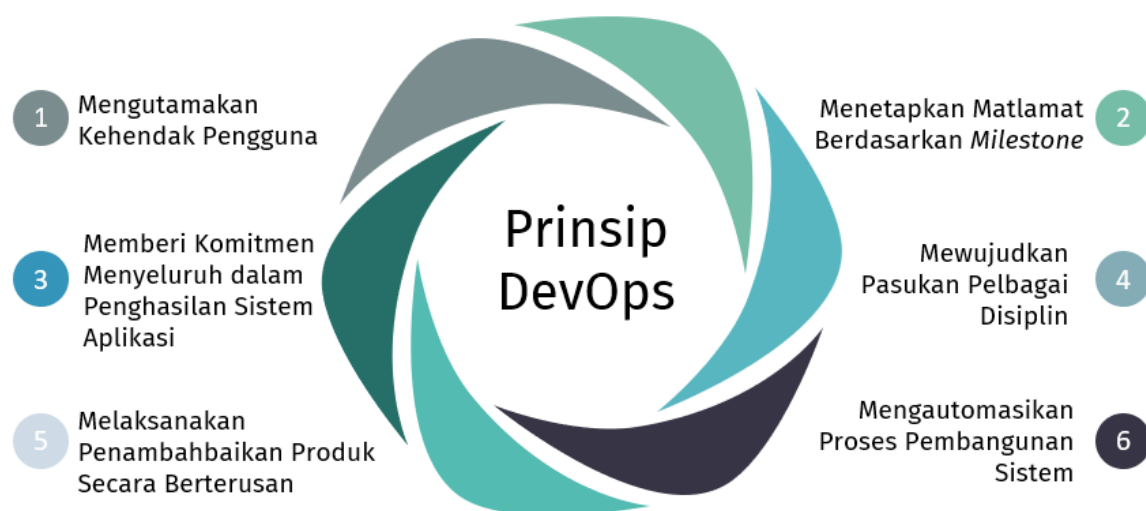
Rajah 4-4: Teras Pelaksanaan DevOps

4.1. TERAS 1: 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.

4.1.1. Prinsip DevOps

Enam prinsip DevOps yang diadaptasi daripada *DevOps Agile Skills Association* (DASA) yang boleh diamalkan bagi membantu memudahkan pelaksanaan DevOps adalah seperti di Rajah 4-5.^{5, 6, 7}



Rajah 4-5: 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.

⁵ Coupland, M. (2021). *DevOps Adoption Strategies: Principles, Processes, Tools, and Trends: Embracing DevOps through effective culture, people, and processes*. Packt Publishing.

⁶ Devops Agile Skills Association LLC. (2020, September). DASA DevOps Fundamentals (2.0.1 ed.) [E-book]. Devops Agile Skills Association LLC.

⁷ Deloitte. (2020). DevOps Point of View. *An Enterprise Architecture Perspective*, 17.

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 fasa kitar hayat pembangunan sistem aplikasi. Setiap ahli pasukan akan terlibat dari fasa awal hingga fasa 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. Mengautomatiskan Proses Pembangunan Produk

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

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.

4.1.2. Budaya DevOps dan Penerapan dalam Pasukan

Pendekatan DevOps dapat dilaksanakan dengan lebih berkesan, melalui penerapan budaya seperti di Rajah 4-6, 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 4-6: Budaya DevOps untuk Mencapai Enam Prinsip DevOps

Jadual 4-1 menyenaraikan penerangan berkaitan aspek budaya dan cadangan aktiviti yang boleh dilaksanakan oleh agensi berdasarkan prinsip DevOps.

Jadual 4-1: Penerangan Berkaitan Aspek-Aspek Budaya dan Cadangan Aktiviti

Aspek Budaya	Penerangan	Cadangan Aktiviti
Menggalakkan Pembelajaran dan Peningkatan Kemahiran yang Berterusan	Membangunkan tahap kemahiran dan daya kompeten ahli pasukan dengan menekankan budaya pembelajaran secara berterusan dan perkongsian pengetahuan berkaitan idea dan teknologi baharu	i. Melaksanakan sesi perkongsian pengetahuan terhadap teknologi atau penyelesaian baharu dan menggalakkan setiap ahli pasukan untuk terlibat dalam sesi ini, dan ii. menyediakan program latihan dan coaching untuk ahli pasukan bagi menambahkan dan meningkatkan tahap kemahiran.
Menggalakkan Eksperimentasi dan Keberanian Mengambil Risiko	Mewujudkan pengembangan idea-idea baharu dengan kaedah menggalakkan ahli pasukan meneroka dan bereksperimentasi dengan idea mereka	i. Menyediakan platform bagi melaksanakan sesi perkongsian pengetahuan untuk ahli pasukan membentangkan idea-idea mereka, ii. memberi kepercayaan kepada ahli pasukan dan menghargai setiap idea-idea baharu yang dicadangkan oleh setiap pasukan, dan iii. menyediakan kemudahan yang diperlukan oleh setiap ahli pasukan bagi mencuba dan melaksanakan idea-idea baharu mereka.

Aspek Budaya	Penerangan	Cadangan Aktiviti
Mengutamakan Kualiti Produk	Menerapkan tanggungjawab bersama dalam memastikan kualiti produk yang dihasilkan berada pada tahap tertinggi	i. Menggalakkan setiap ahli pasukan untuk bertanggungjawab dan terlibat dalam setiap fasa kitar hayat pembangunan sistem aplikasi, 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, iii. mengautomasi proses-proses yang dilaksanakan secara manual bagi meminimumkan ralat yang terhasil daripada kelalaian ahli pasukan serta meningkatkan kualiti produk yang dihasilkan, dan iv. melaksanakan penambahbaikan secara berterusan terhadap produk berdasarkan setiap maklum balas yang diterima daripada pengguna.
Menyokong Pemikiran Kreatif dan Inovatif	Menerapkan pemikiran secara kreatif dan inovatif dalam menyelesaikan masalah yang dihadapi bagi mengoptimumkan proses dan masa pembangunan produk	i. Melaksanakan program-program inovasi dan menggalakkan penyertaan oleh setiap ahli pasukan bagi membolehkan mereka berfikir secara kreatif dan inovatif, ii. menyediakan platform dan kemudahan yang dapat membantu setiap ahli pasukan untuk berinovasi bagi menghasilkan penyelesaian baharu, dan

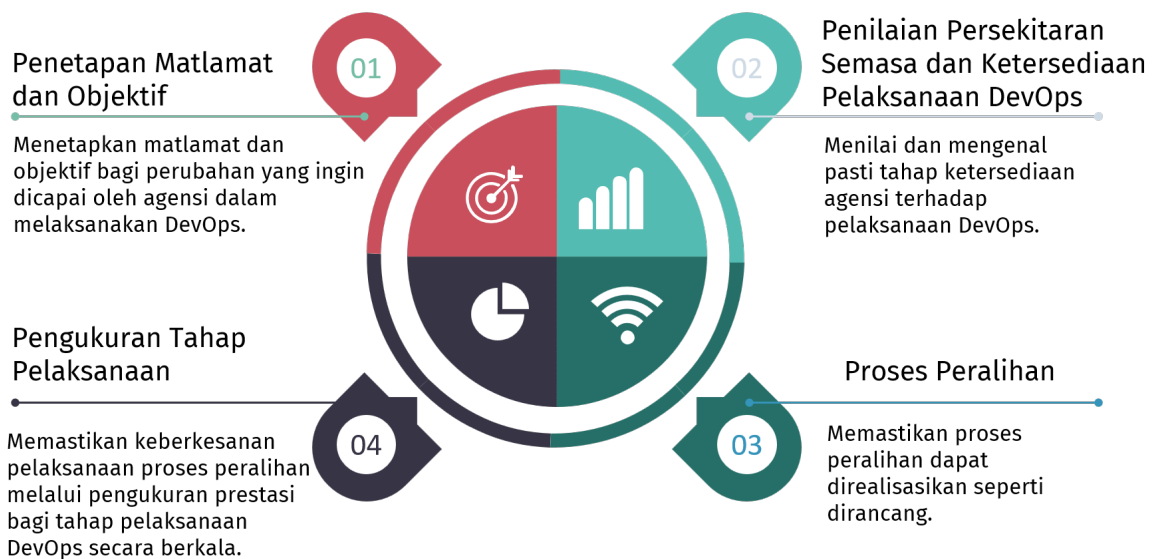
Aspek Budaya	Penerangan	Cadangan Aktiviti
		iii. membentuk komuniti pengamal (communities of practices - COP) untuk peningkatan dan perkongsian pengetahuan dan pengalaman. Ini akan memberi keyakinan kepada setiap ahli pasukan untuk berinovasi dalam menyelesaikan sesuatu tugas berdasarkan kreativiti mereka.
Menerapkan Budaya Pemikiran Berasaskan Produk	Menggalakkan penyelesaian isu dengan melihat produk sebagai satu penyelesaian lengkap tanpa menumpukan kepada fungsi-fungsi khusus sahaja	i. Menggalakkan pengguna untuk memberikan maklum balas tentang produk dan memastikan pasukan memahami serta bertindak balas secara cepat dan efisien, dan ii. menggalakkan ahli pasukan untuk menyampaikan taklimat berkaitan produk yang dibangunkan di seminar dan penerbitan di laman sesawang.
Mewujudkan Persekitaran Kerja yang Ideal dan Kondusif	Mewujudkan persekitaran kerja yang selesa dari segi fizikal dalam memperkukuhkan budaya organisasi berprestasi tinggi dan inovatif	i. Melaksanakan bengkel bagi membangun dan menetapkan peranan dan tanggungjawab ahli pasukan, ii. mewujudkan matlamat yang ingin dicapai oleh ahli pasukan, dan iii. meningkatkan komunikasi secara berkesan bagi mewujudkan kepercayaan dan ketelusan antara ahli pasukan.

Aspek Budaya	Penerangan	Cadangan Aktiviti
Mengutamakan Nilai dan Kepuasan Pengguna	Memberi tumpuan bagi memastikan produk yang dibangunkan memenuhi kehendak dan ekspektasi pengguna secara keseluruhan	i. Menggalakkan setiap ahli pasukan untuk berinteraksi dengan pengguna bagi memahami dengan lebih mendalam terhadap keperluan dan kehendak pengguna, ii. melaksanakan sesi demonstrasi kepada pengguna di setiap peringkat penyampaian produk untuk mendapatkan maklum balas terhadap produk yang dibangunkan, dan 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.

4.2. TERAS 2: PENGADAPTASIAN METODOLOGI

Peralihan kepada pendekatan DevOps perlu dilaksanakan secara berperingkat kerana proses tersebut memerlukan masa yang sesuai, tahap ketersediaan dan persekitaran semasa agensi bagi memastikan perubahan dapat diterima dan diamalkan secara berkesan.

Rajah 4-7 menjelaskan metodologi asas dalam pelaksanaan DevOps yang merangkumi empat aktiviti iaitu menetapkan matlamat dan objektif, menilai persekitaran semasa, melaksana proses peralihan dan mengukur tahap pelaksanaan DevOps.^{8, 9, 10}



Rajah 4-7: Metodologi Asas dalam Pelaksanaan DevOps

⁸ Coupland, M. (2021). *DevOps Adoption Strategies: Principles, Processes, Tools, and Trends: Embracing DevOps through effective culture, people, and processes*. Packt Publishing.

⁹ Devops Agile Skills Association LLC. (2020, September). DASA DevOps Fundamentals (2.0.1 ed.) [E-book]. Devops Agile Skills Association LLC.

¹⁰ Forsgren, Nicole & Tremblay, Monica & Vander Meer, Debra & Humble, Jez. (2017). DORA Platform: DevOps Assessment and Benchmarking. 436-440. 10.1007/978-3-319-59144-5_27.

4.2.1. Penetapan Matlamat dan Objektif

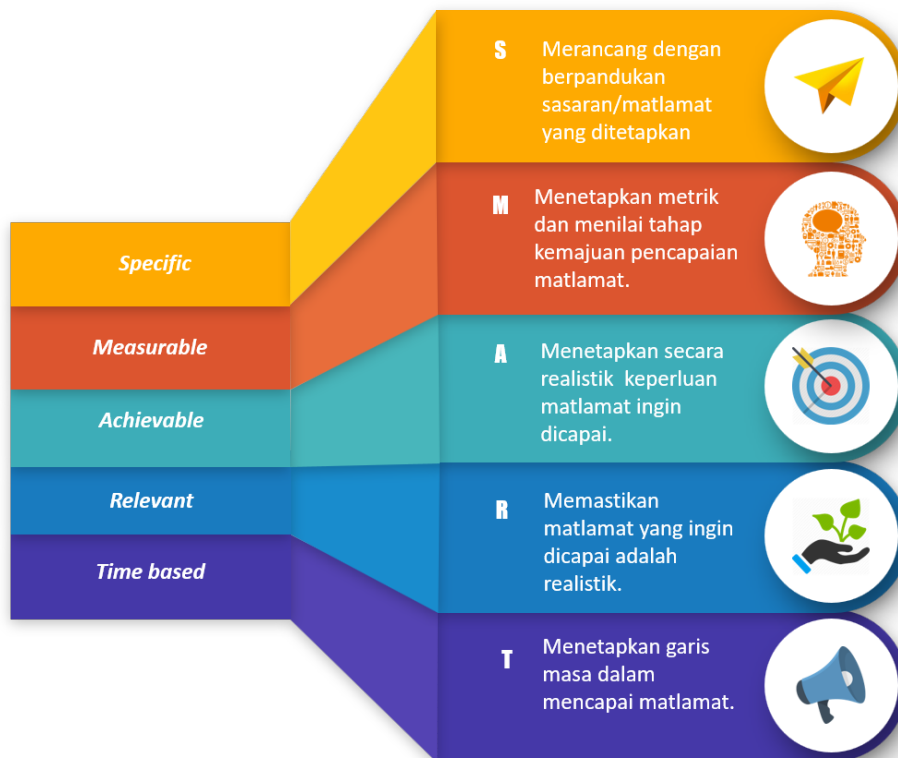
Matlamat merupakan idea umum dan menyeluruh terhadap hasil akhir yang ingin dicapai. Manakala, objektif pula merupakan tindakan atau langkah khusus yang dirancang dan dilaksanakan untuk mencapai matlamat yang telah ditetapkan. Objektif membantu pihak-pihak yang terlibat untuk kekal fokus dan bergerak ke arah matlamat yang ingin dicapai. Penetapan matlamat dan objektif dilaksanakan dalam fasa perancangan.

4.2.1.1. Tujuan

Menetapkan dan melengkapkan maklumat berkenaan matlamat dan objektif tentang perubahan yang ingin dicapai oleh agensi dalam melaksanakan DevOps.

4.2.1.2. Metodologi

Salah satu kaedah yang boleh digunakan dalam penetapan matlamat dan objektif adalah kaedah S.M.A.R.T. Penetapan objektif menggunakan kaedah S.M.A.R.T. mempunyai ciri-ciri seperti di Rajah 4-8.



Rajah 4-8: Kaedah S.M.A.R.T. dalam Penetapan Matlamat dan Objektif

a. Khusus (Specific)

Objektif pelaksanaan DevOps dalam pembangunan produk perlu ditetapkan secara terperinci supaya hasil, kualiti dan nilai dapat dicapai. Pihak berkepentingan yang terlibat perlu dikenal pasti dan masa yang diperlukan untuk mencapai objektif yang akan ditetapkan.

b. Boleh Diukur (Measurable)

Setiap objektif pelaksanaan DevOps perlu mempunyai penanda aras (indicator) yang boleh dijadikan sebagai penentu kepada tahap pencapaiannya.

c. Boleh Dicapai (Achievable)

Penetapan sesuatu objektif pelaksanaan DevOps perlu memastikan bahawa hasil yang ingin dicapai adalah realistik dan munasabah dengan mengambil kira faktor-faktor seperti masa, kekangan dan kemahiran pihak-pihak yang terlibat.

d. Relevan (Relevant)

Objektif pelaksanaan DevOps perlu dibentuk selari dengan matlamat akhir yang ingin dicapai dan boleh dilaksanakan oleh semua pihak-pihak yang terlibat.

e. Berasaskan Masa (Time Based)

Setiap objektif pelaksanaan DevOps perlu dibentuk bersama tempoh masa yang bersesuaian bagi mencapai matlamat yang ditetapkan. Tempoh masa ditetapkan bagi memastikan tindakan-tindakan yang perlu diambil dapat disusun dan diuruskan dengan berkesan.

Contoh template seperti di Jadual 4-2 boleh digunakan sebagai panduan oleh agensi untuk menetapkan matlamat pelaksanaan DevOps menggunakan metodologi S.M.A.R.T.

Jadual 4-2: Contoh Templat Penetapan Matlamat S.M.A.R.T

Matlamat		Penetapan Matlamat
S	<i>Specific</i>	i. Siapakah yang terlibat? ii. Apakah matlamat yang ingin dicapai? iii. Bilakah matlamat ini perlu dicapai?
M	<i>Measurable</i>	iv. Bagaimanakah tahap kemajuan diukur? v. Bagaimanakah agensi menentukan bahawa matlamat telah dicapai?
A	<i>Achievable</i>	vi. Adakah masa dan sumber yang diperuntukkan dapat menyelesaikan matlamat dalam masa yang ditetapkan?
R	<i>Relevant</i>	vii. Adakah matlamat yang ditetapkan ini berkaitan dengan perubahan yang ingin dicapai oleh keseluruhan agensi?
T	<i>Time Based</i>	viii. Berapa lamakah masa yang diperlukan untuk mencapai matlamat ini? ix. Bilakah semakan terhadap status pencapaian matlamat akan dilakukan?

4.2.2. Penilaian Persekitaran 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.

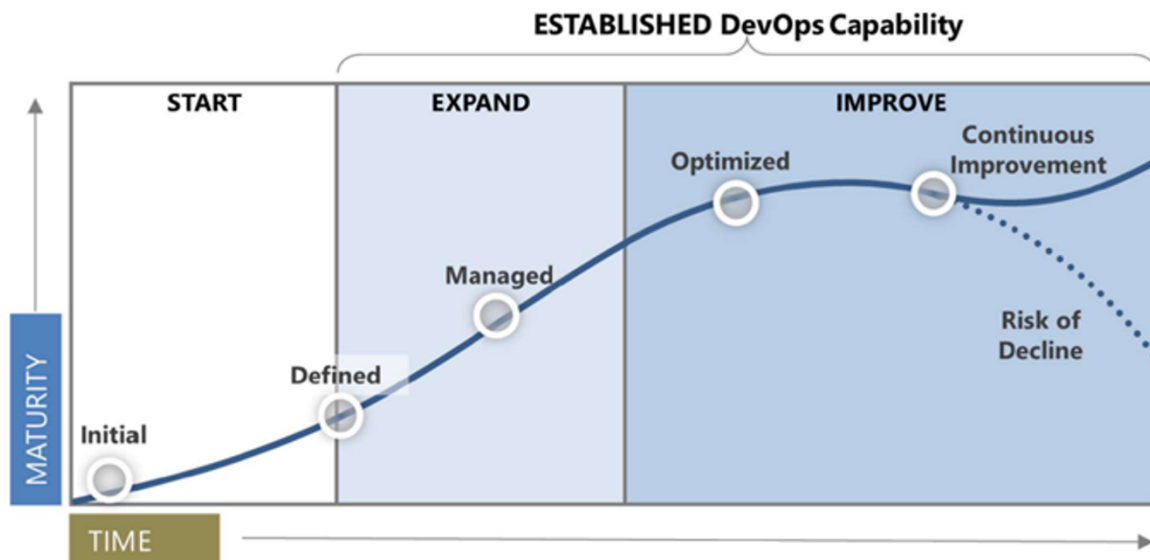
4.2.2.1. Tujuan

Menilai dan mengenal pasti tahap ketersediaan agensi terhadap pelaksanaan DevOps.

4.2.2.2. Metodologi

Penilaian terhadap persekitaran semasa DevOps boleh dilaksanakan berpanduan kepada Model Keluk Kematangan Keupayaan Agensi yang dikeluarkan oleh American

Council for Technology-Industry Advisory Council (ACT-IAC).¹¹ 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 4-9: Keluk Kematangan Agensi (Sumber: ACT-IAC)

Rajah 4-9 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.

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

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 di 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 4-3 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 4-9, merangkumi lima aspek pelaksanaan iaitu Proses Pembinaan, Pelepasan dan Penempatan, Proses Pengujian, Keselamatan, Infrastruktur, Budaya serta Tadbir Urus.¹²

Jadual 4-3: Model Kematangan DevOps

Aspek	Tahap Kematangan DevOps				
	<i>Initial</i>	<i>Defined</i>	<i>Managed</i>	<i>Optimised</i>	<i>Continuous Improvement</i>
1. Pembinaan, Pelepasan dan Penempatan	Melaksanakan aktiviti pembinaan, pelepasan dan penempatan secara manual	i. Menenal pasti dan merancang proses yang boleh diautomasikan, dan ii. mengenal pasti <i>tools</i> integrasi bagi proses automasi	Melaksanakan sebahagian aktiviti pembinaan, pelepasan dan penempatan secara automasi	Melaksanakan semua aktiviti pembinaan, pelepasan dan penempatan secara automasi	i. Melaksanakan semua aktiviti pembinaan, pelepasan dan penempatan secara automasi ii. Melaksanakan strategi pemantauan berterusan terhadap aktiviti pembinaan, pelepasan dan penempatan

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

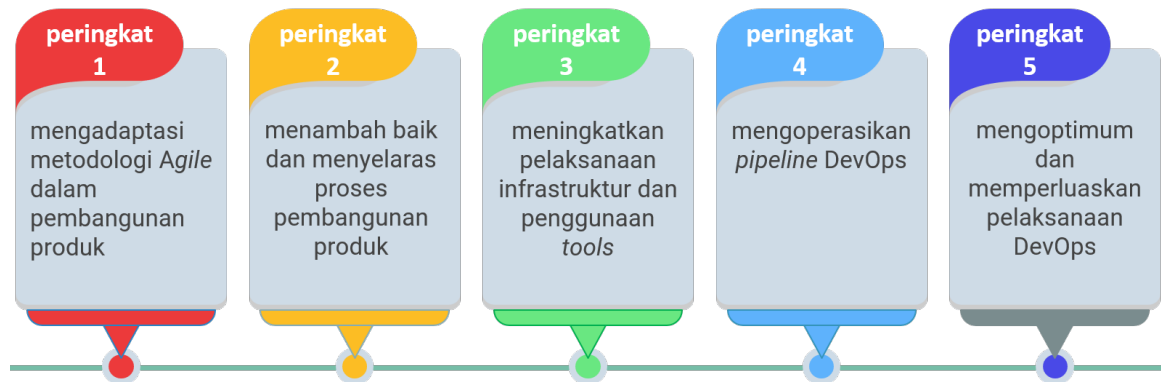
Aspek	Tahap Kematangan DevOps				
	<i>Initial</i>	<i>Defined</i>	<i>Managed</i>	<i>Optimised</i>	<i>Continuous Improvement</i>
2. Pengujian	Melaksanakan pengujian secara manual	i. Mengetahui pasti dan merancang strategi pengujian, ii. mengetahui pasti <i>tools</i> pengujian bagi proses automasi, dan iii. memulakan proses automasi dengan pengujian unit	i. Melaksanakan proses pengujian unit dan integrasi secara automasi, dan ii. melaksanakan analisis kod statik secara automasi	i. Melaksanakan pengujian unit, integrasi, penerimaan, prestasi dan sistem integrasi secara automasi, dan ii. melaksanakan analisis kod statik secara automasi	Melaksanakan proses pengujian berterusan sebagai sebahagian daripada proses pemantauan berterusan
3. Keselamatan	i. Melaksanakan aspek keselamatan di akhir kitar hayat pembangunan sistem aplikasi, ii. melaksanakan pengujian kawalan keselamatan secara manual	Merancang dan membangunkan polisi dan piawaian keselamatan	Menekankan dan melaksanakan aspek keselamatan di awal kitar hayat pembangunan sistem aplikasi untuk mengetahui 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				
	<i>Initial</i>	<i>Defined</i>	<i>Managed</i>	<i>Optimised</i>	<i>Continuous Improvement</i>
4. Infrastruktur	i. Membangunkan piawaian penyediaan persekitaran infrastruktur, dan 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	i. Melaksanakan integrasi berterusan dan pengautomasia n <i>pipeline</i> serahan sistem aplikasi, dan ii. meningkatkan penggunaan repositori kod sumber berpusat berdasarkan amalan terbaik	i. Mengintegrasikan <i>tools</i> pembangunan dan penempatan ke persekitaran sedia ada berdasarkan <i>containerization</i> , dan ii. migrasi sistem legasi kepada persekitaran berasaskan seni bina modular (microservices)	Menyediakan infrastruktur yang boleh melaksanakan proses penskalaan secara automasi berdasarkan keperluan melalui pelaksanaan IaC	i. Memperluaskan pelaksanaan infrastruktur di premis (on-premises) ke pengkomputeran awan (cloud computing), dan ii. melaksanakan strategi pemantauan berterusan terhadap persekitaran infrastruktur

Aspek	Tahap Kematangan DevOps				
	<i>Initial</i>	<i>Defined</i>	<i>Managed</i>	<i>Optimised</i>	<i>Continuous Improvement</i>
5. Budaya dan Tadbir Urus	i. Proses membuat keputusan melalui pelbagai peringkat, ii. pasukan pembangun dan pasukan operasi bekerja secara berasingan dalam pembangunan sistem aplikasi, dan iii. terdapat jurang komunikasi di antara pasukan pembangun dan pasukan operasi	i. Merancang dan mengenal pasti strategi peralihan budaya ke arah DevOps, ii. mewujudkan pasukan pengamal DevOps, dan iii. mengenal pasti <i>tools</i> komunikasi dan kolaborasi	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	i. Proses membuat keputusan diagihkan kepada ahli pasukan, dan ii. melaksanakan pemantauan berterusan dan mengukur tahap keberkesanan pelaksanaan DevOps

4.2.3. Proses Peralihan

Amalan terbaik peralihan kepada pelaksanaan DevOps adalah secara pendekatan berperingkat seperti di Rajah 4-10.¹³ 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 4-10: Proses Peralihan kepada Pelaksanaan DevOps

4.2.3.1. Tujuan

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

¹³ 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.

4.2.3.2. Metodologi

Metodologi pelaksanaan proses peralihan secara berperingkat seperti di Rajah 4-10 dijelaskan secara lanjut dalam Jadual 4-4.

Jadual 4-4: Metodologi Pelaksanaan Proses Peralihan kepada Amalan DevOps secara Berperingkat

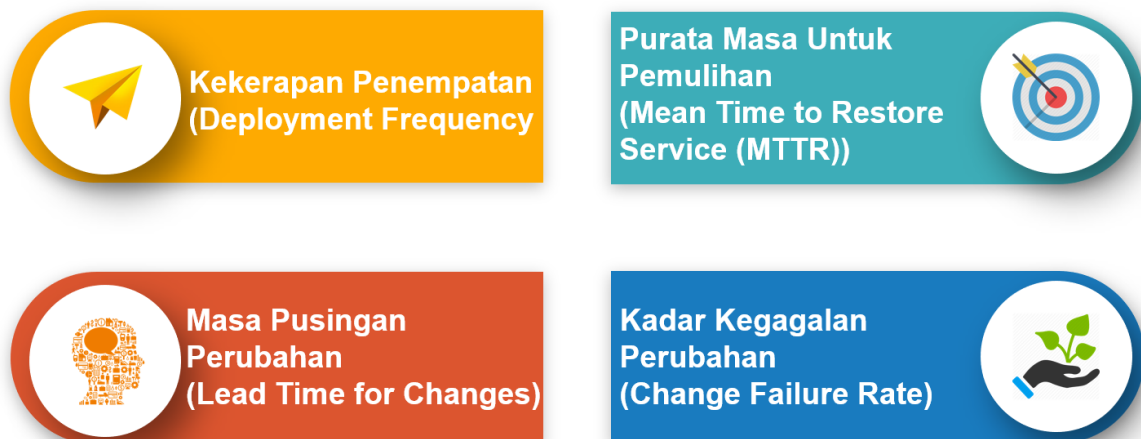
Peringkat	Penjelasan	Aktiviti
1. Mengadaptasi Metodologi Agile dalam Pembangunan Produk	Mengadaptasi metodologi <i>Agile</i> ke dalam pembangunan sistem. Kesedaran terhadap manifesto <i>Agile</i> merangkumi budaya, proses kerja dan <i>tools</i> membantu memberikan hala tuju yang jelas terhadap proses peralihan yang dilaksanakan	i. Membina budaya dan minda pasukan ke arah metodologi pembangunan berasaskan <i>Agile</i>, ii. mempelajari dan mengadaptasi budaya DevOps secara berperingkat ke dalam pembangunan sistem aplikasi, dan 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</i> CI/CD.
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	i. Memperkenalkan dan mengguna pakai konsep <i>Agile</i> dalam pasukan, ii. menyelaraskan proses pembangunan produk, dan iii. menambah baik amalan DevOps pembangunan sedia ada seperti: a. Integrasi berterusan dan pengautomasian <i>pipeline</i> serahan sistem aplikasi, b. Meningkatkan penggunaan repositori kod sumber berpusat berdasarkan amalan terbaik, c. Mengenal pasti proses dokumentasi yang boleh diminimumkan supaya pasukan lebih fokus kepada pembangunan produk, dan d. Merancang dan melaksana proses pelepasan ke persekitaran produksi dan menambah baik proses kerja yang terlibat.

Peringkat	Penjelasan	Aktiviti
3. Meningkatkan Pelaksanaan Infrastruktur dan Penggunaan Tools	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	i. Mengintegrasikan <i>tools</i> pembangunan dan penempatan melalui pengaplikasian pendekatan <i>containerization</i>, ii. migrasi persekitaran sistem legasi kepada persekitaran infrastruktur berasaskan DevOps, dan iii. mengoptimumkan penggunaan <i>tools</i> dan infrastruktur DevOps dalam proses pembangunan dan operasi ke persekitaran produksi.
4. Mengoperasikan Pipeline DevOps	Mengoperasikan <i>pipeline</i> DevOps pada tahap yang optimum bagi menyokong kadar penempatan yang lebih kerap dan berterusan. Dengan pendekatan ini, ralat dalam setiap fasa kitar hayat pembangunan produk dapat dikenal pasti dengan lebih awal	i. Mengintegrasikan <i>tools</i> CI/CD dengan <i>tools</i> perisian pembangunan sedia ada atau <i>tools</i> perisian pihak ketiga, ii. mengautomasi persekitaran infrastruktur melalui pelaksanaan IaC, dan iii. menyesuaikan proses kerja berasaskan prinsip DevOps.

Peringkat	Penjelasan	Aktiviti
5. Mengoptimum dan Memperluaskan Pelaksanaan DevOps	Memberi tumpuan kepada mengoptimumkan dan memperluaskan pelaksanaan DevOps serta menggalakkan ahli pasukan untuk mempertingkatkan tahap pelaksanaan DevOps	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, ii. melaksanakan sesi perkongsian teknologi terkini, program latihan dan <i>coaching</i> mengenai <i>tools</i> CI/CD dalam proses pembangunan aplikasi di agensi, iii. memperkenalkan insentif untuk ahli pasukan bagi meningkatkan pelaksanaan DevOps seperti memberi penghargaan dan pengiktirafan atau insentif yang dipilih secara bersama oleh ahli pasukan, iv. meningkatkan keupayaan infrastruktur DevOps di premis (on-premises) atau menggalakkan penggunaan perkhidmatan pengkomputeran awan (cloud services), dan v. merancang strategi bagi mengoptimumkan pelaksanaan DevOps melalui penempatan dan pemantauan secara berterusan.

4.2.4. Pengukuran Tahap Pelaksanaan

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



Rajah 4-11: Metrik Pengukuran Prestasi Pelaksanaan DevOps

Metrik Pengukuran Prestasi Pelaksanaan DevOps seperti di Rajah 4-11 digunakan sebagai penanda aras pengukuran tahap pelaksanaan 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).¹⁴

Empat profil tahap pengukuran prestasi seperti di Jadual 4-1 iaitu elit, tinggi, sederhana dan rendah digunakan bagi menentukan tahap pencapaian agensi dalam pelaksanaan DevOps.¹⁵

4.2.4.1. Tujuan

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

¹⁴ 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>.

¹⁵ 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.

4.2.4.2. Metodologi

Tahap pengukuran prestasi pelaksanaan DevOps menggunakan empat metrik adalah seperti di Jadual 4-5:

Jadual 4-5: Jadual Pengukuran Prestasi Tahap Pelaksanaan DevOps dari Segi Penyampaian dan Operasi Sistem Aplikasi

Metrik/Tahap Prestasi	Elit	Tinggi	Sederhana	Rendah
Kekerapan Penempatan (Deployment Frequency)	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
Masa Pusingan Perubahan (Lead Time for Changes)	Kurang dari satu hari	Satu hari hingga satu minggu	Satu minggu hingga satu bulan	Satu minggu hingga satu bulan
Purata Masa Untuk Pemulihan (Mean Time to Restore Service (MTTR))	Kurang dari satu jam	Kurang dari satu jam	Kurang dari satu hari	Lebih dari Satu hari
Kadar Kegagalan Perubahan (Change Failure Rate)	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 didaftarkan masuk (commit) sehingga ke fasa 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 menjejak 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 pembaikan segera, atau
- iii. Penempatan ke produksi yang memerlukan pemulihan semula ke versi sebelumnya.

Pengukuran prestasi tahap pelaksanaan DevOps berdasarkan empat metrik adalah seperti Lampiran 1: Metrik Pengukuran Tahap Pelaksanaan DevOps.

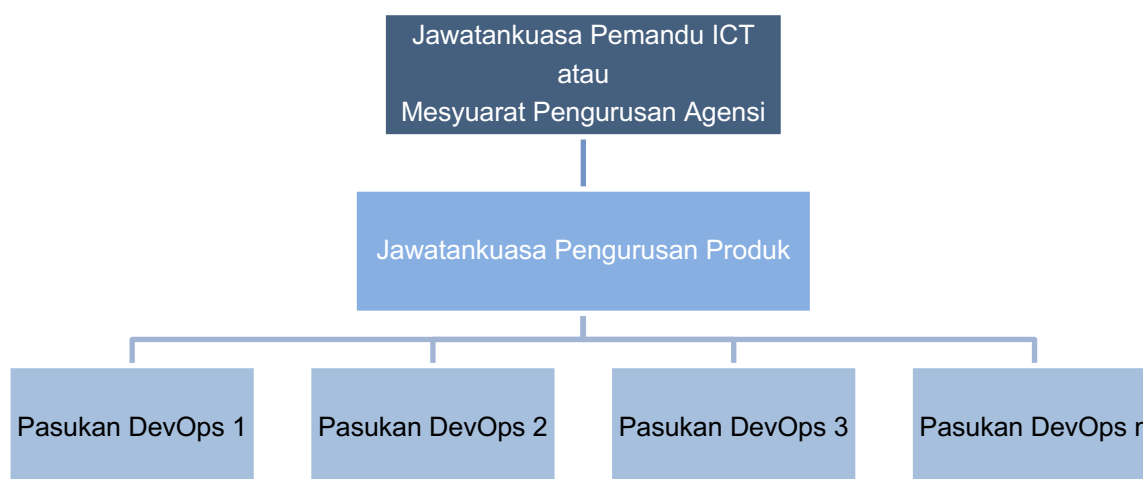
4.3. TERAS 3: PENGUKUHAN TADBIR URUS

Tadbir urus dalam pelaksanaan DevOps penting bagi menyelaraskan 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.

4.3.1. Struktur Pelaporan Organisasi

Cadangan struktur pelaporan dalam pelaksanaan DevOps dengan Jawatankuasa Pengurusan Produk adalah seperti di Rajah 4-12.



Rajah 4-12: Struktur Pelaporan dalam Pelaksanaan DevOps

Cadangan bagi keanggotaan Jawatankuasa Pengurusan Produk adalah seperti di Jadual 4-6.

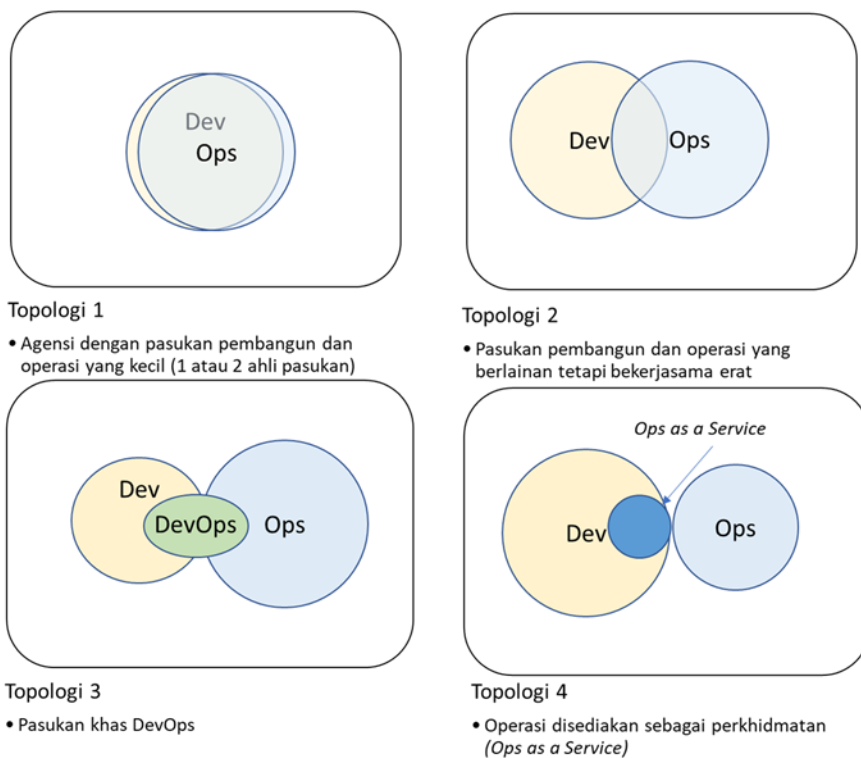
Jadual 4-6: Cadangan Keanggotaan Jawatankuasa Pengurusan Produk

Keahlian	Cadangan
Pengerusi	Ketua Pegawai Digital atau Ketua Agensi/Pengarah ICT Agensi
Ahli-ahli	Pemilik-pemilik produk di agensi yang terlibat dalam pelaksanaan DevOps
Kekerapan Mesyuarat	Sebulan sekali atau mengikut keperluan
Terma Rujukan	i. Menentukan hala tuju dan strategi dalam hal ehwal produk di agensi,

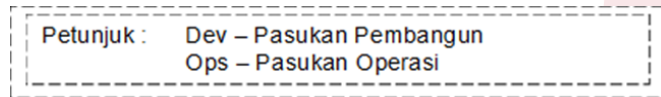
Keahlian	Cadangan
	ii. Membincang dan mengesyorkan penyelesaian isu-isu berkaitan produk, dan iii. Memantau pembangunan dan pelaksanaan produk.

4.3.2. 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 pasukan¹⁸ yang menyokong prinsip DevOps adalah seperti di Rajah 4-13.



¹⁸ Skelton, M., & Pais, M. (2019). Team topologies: organizing business and technology teams for fast flow. It Revolution.



Rajah 4-13: Topologi Struktur Pasukan DevOps

a. Topologi 1 - Perkongsian Tanggungjawab dan Peranan

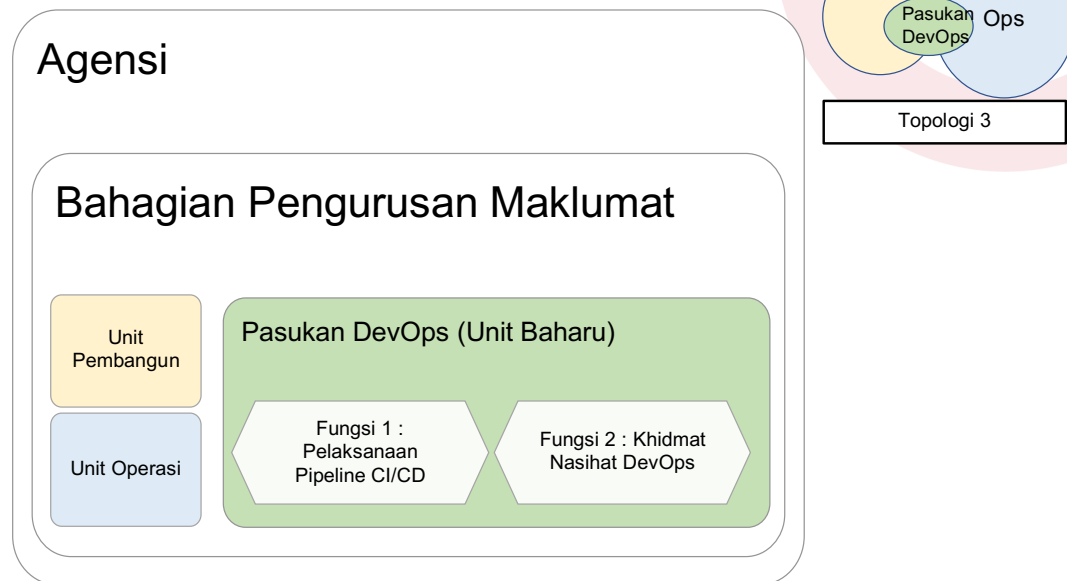
Topologi ini berlaku kepada agensi yang mempunyai penjawatan yang kecil dimana peranan pembangunan dan operasi dilaksanakan oleh pegawai yang sama. Struktur pasukan DevOps ini menggalakkan kerjasama yang erat dan cekap antara pasukan pembangun dan pasukan operasi (termasuk pasukan sokongan seperti pasukan 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 *silos*. Gambaran pelaksanaan topologi ini adalah seperti di Rajah 4-14.

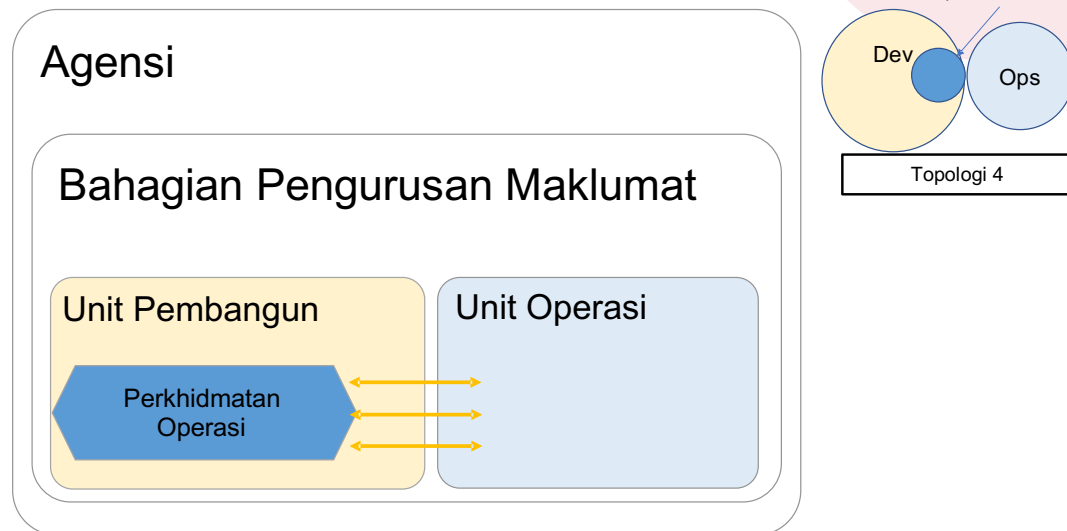


Rajah 4-14: Gambaran Pelaksanaan Topologi 3

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

Struktur ini memastikan pasukan operasi mengautomatiskan 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 memahirkannya dalam

proses pengoperasian. Gambaran pelaksanaan topologi ini adalah seperti di Rajah 4-15.

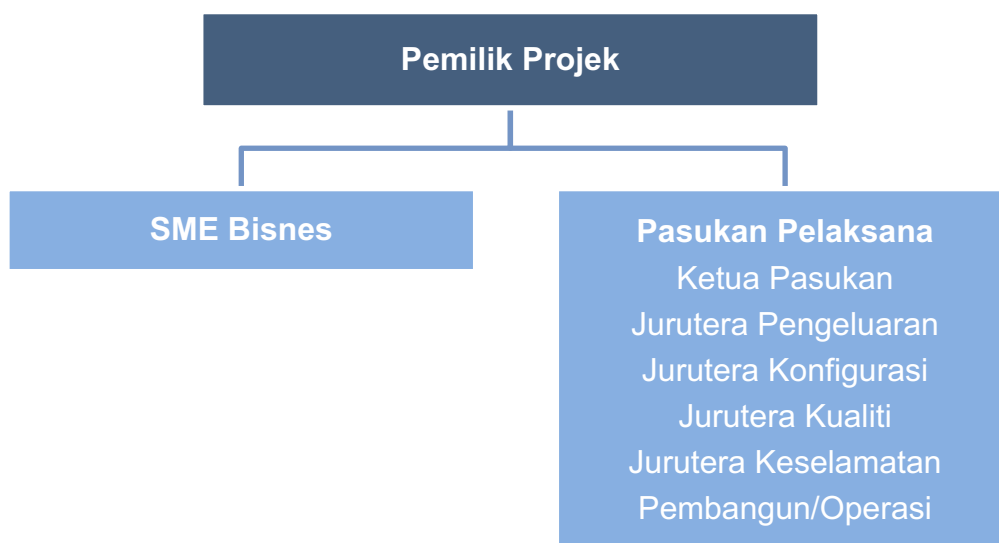


Rajah 4-15: Gambaran Pelaksanaan Topologi 4

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

4.3.3. Ahli Pasukan

Bilangan ahli pasukan DevOps yang dicadangkan adalah antara lima hingga sepuluh orang dalam satu pasukan dan bilangan ahli pasukan yang minimum adalah digalakkan.¹⁹ Beberapa pasukan DevOps boleh diwujudkan jika terdapat ramai ahli pasukan dalam pembangunan produk. Cadangan keahlian pasukan DevOps adalah seperti di Rajah 4-16.



Rajah 4-16: Cadangan Keahlian Pasukan

¹⁹ Cagle, R., Rice, T., & Kristan, M. (2018). *DevOps for federal acquisition*. MITRE CORP BEDFORD MA.

Peranan, tugas dan kemahiran ahli pasukan DevOps adalah seperti di Jadual 4-7.

Jadual 4-7: Peranan, Tugas dan Kemahiran Ahli Pasukan

Peranan	Tugas	Kemahiran
Pemilik Produk	i. Memastikan produk yang dihasilkan menepati keperluan pengguna seperti keperluan <i>User Interface</i> (UI) dan fungsi sistem, ii. menjadi penyelarar bagi pihak pemegang taruh dalam memahami nilai produk yang bakal dibangunkan, dan iii. mengutamakan penghasilan produk jangka panjang berdasarkan keperluan pengguna.	i. Berpengetahuan tentang sistem pengurusan yang digunakan oleh pasukan, dan ii. pantas dan dinamik dalam menangani perubahan fungsi produk.
Ketua Pasukan	i. Merancang strategi pelaksanaan DevOps, ii. mengenal pasti set <i>tools</i> yang boleh mengoptimumkan produktiviti pasukan, iii. memastikan pasukan mempunyai sumber yang diperlukan bagi <i>pipeline</i> CI/CD yang berterusan, dan iv. memastikan penghasilan produk berdasarkan keperluan pengguna dalam tempoh masa dan kos yang ditetapkan.	i. Perlu mewujudkan hubungan yang baik dalam pasukan dan menyokong prinsip dan amalan DevOps secara proaktif, dan ii. boleh menerangkan keperluan teknikal dan keperluan pengguna dengan mudah kepada pihak pemegang taruh.
Jurutera Pelepasan	i. Menyelia integrasi, pembangunan, ujian dan penggunaan <i>tools</i> untuk menyokong pelepasan berterusan, ii. merancang dan melaksanakan proses pelepasan dengan efektif dan bertanggungjawab bagi setiap pelepasan produk, dan	i. Berpengetahuan dan mahir dalam penggunaan <i>tools</i> pelepasan, dan ii. mampu mengkoordinasi aktiviti-aktiviti pelepasan dengan pihak pemegang taruh.

Peranan	Tugasan	Kemahiran
	iii. memantau penggunaan <i>tools</i> DevOps serta pelepasan produk secara menyeluruh.	
Jurutera Konfigurasi/Automasi	i. Mengautomatiskan proses pembangunan dan operasi serta berkongsi maklumat dengan ahli pasukan, dan ii. memastikan proses konfigurasi dilakukan secara teratur dan selamat.	i. Berpengetahuan dan mahir dalam penggunaan <i>tools</i> konfigurasi, ii. pantas mengenal pasti punca masalah dalam isu konfigurasi, dan iii. berpengetahuan luas mengenai konfigurasi <i>tools</i> DevOps.
Jurutera Kualiti	i. Menentukan prosedur ujian dan bekerjasama rapat dengan ahli pasukan untuk mengautomatiskan ujian yang diperlukan, ii. mereka bentuk dan melaksanakan ujian yang memerlukan pengujian secara manual, iii. mengesan ralat atau pepijat dan membetulkan produk seawal mungkin, dan iv. meningkatkan kualiti produk yang dibangunkan.	i. Mahir dalam pengurusan kualiti, ii. berpengetahuan dan mahir dalam pengujian sistem, dan iii. teliti dalam proses semakan kualiti.
Jurutera Keselamatan	i. Menyelenggara alat dan teknik pengesanan keselamatan, ii. memastikan <i>pipeline</i> CI/CD di integrasi dan diurus secara selamat dalam semua persekitaran, iii. memantau insiden keselamatan dan bekerjasama dengan pasukan untuk memastikan proses pembangunan dan operasi berjalan lancar, dan	i. Berpengetahuan dan mahir dalam penggunaan <i>tools</i> keselamatan DevOps, ii. arif dalam perkembangan terkini teknologi keselamatan.

Peranan	Tugasan	Kemahiran
	iv. menyediakan laporan terperinci insiden kepada pihak berkepentingan.	
Pembangun/Operasi	i. Membangunkan kod serta melaksanakan pengujian <i>back-end</i> dan <i>front-end</i>, ii. memastikan penggunaan, penyelenggaraan dan pemantauan <i>pipeline</i> CI/CD sentiasa tersedia, iii. memastikan kualiti produk dikekalkan dari peringkat awal, dan iv. memberi perkhidmatan sokongan produk.	i. Mahir dalam bahasa pengaturcaraan yang digunakan, ii. berpengetahuan mengenai <i>tools</i> DevOps dan <i>pipeline</i> CI/CD, dan iii. berpengalaman dalam proses pembangunan produk.
SME Bisnes	i. Menjadi pakar rujuk berkaitan bisnes, dan ii. mengesahkan produk yang dihasilkan menepati keperluan bisnes.	Mahir dalam proses bisnes.

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

4.4. TERAS 4: PEMANTAPAN PELAKSANAAN DAN PEMANTAUAN

Pengurusan dan pemantauan dalam fasa-fasa DevOps adalah seperti di Jadual 4-8.

Jadual 4-8: Pengurusan dan Pemantauan dalam DevOps

Aspek Pengurusan dan Pemantauan	Fasa DevOps							
	Perancangan	Pengekoden	Pembangunan	Pengujian	Pelepasan	Penempatan	Pengoperasian	Pemantauan
Perancangan	√							
Pelaksanaan		√	√	√				
Kawalan	√	√	√	√	√	√	√	√
Penyelenggaraan							√	
Pemantauan								√

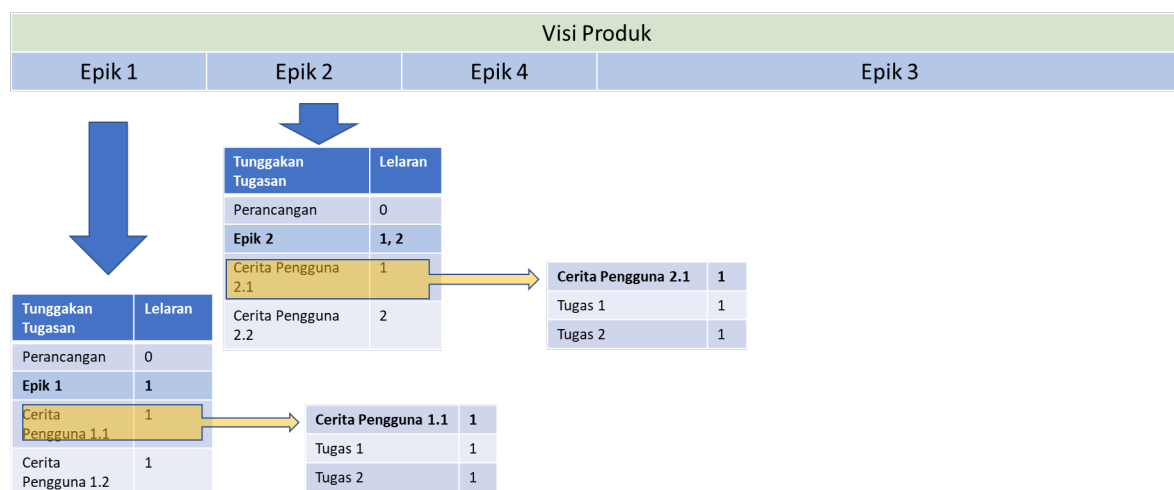
4.4.1. Perancangan

Perancangan DevOps melibatkan aktiviti-aktiviti sebelum pengekodan bermula seperti memperhalusi keperluan pengguna dalam skop produk serta merancang pengurusan bagi *tools* dan proses CI/CD.²⁰ Pelaksanaan DevOps perlu menepati keperluan yang terdapat di dalam PPrISA dan KRISA.

²⁰ 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);

4.4.1.1. Pengurusan Skop Produk

Skop produk ialah fungsi dan ciri-ciri bagi penghasilan produk yang ditetapkan. Skop produk diperincikan melalui epik dan cerita pengguna. Pasukan akan mengenal pasti aktiviti-aktiviti dan tempoh masa yang diperlukan serta menyenaraikan ciri-ciri cerita pengguna yang akan dihasilkan dalam pelan pelepasan. Perkembangan epik kepada cerita pengguna adalah seperti di Rajah 4-17 di bawah:



Rajah 4-17: Perkembangan Epik ke Cerita Pengguna

Pasukan akan menaakrif 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 2: Konsep MVP dalam Pembangunan Produk.

Konsep MVP memastikan produk dihasilkan dengan ciri-ciri dan fungsi asas bagi pengesahan awal produk. MVP membantu pemilik 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 pemilik produk akan mengesahkan *Definition of Done* (DoD) bagi setiap pelepasan berdasarkan *iteration* bagi mengesahkan ciri-ciri produk (berdasarkan cerita pengguna) yang telah

selesai. Ciri-ciri produk yang tidak diperlukan akan dikeluarkan daripada pelan pelepasan.

Jadual 4-9 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 4-9: Contoh Pelan Pelepasan

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

Senario pelan pelepasan yang merujuk kepada Jadual 4-9 adalah seperti berikut:

- Suku pertama tahun 2022 (Q1 2022),
- 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.

4.4.1.2. 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.²¹ Peranan dan tanggungjawab ahli pasukan dalam pengurusan-pengurusan ini adalah seperti di Lampiran 3: Peranan dan Tanggungjawab dalam Pengurusan *Tools* dan Proses CI/CD.

4.4.1.2.1. Pengurusan Konfigurasi

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

²¹ Cagle, R., Rice, T., & Kristan, M. (2018). *DevOps for federal acquisition*. MITRE CORP BEDFORD MA;

Tiga aktiviti yang terlibat di bawah pengurusan konfigurasi adalah seperti di Jadual 4-10.

Jadual 4-10: Aktiviti Pengurusan Konfigurasi

Aktiviti Pengurusan Konfigurasi	Huraian Aktiviti
Mengenal pasti Konfigurasi	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 fasa DevOps.
Mengawal Versi dan Konfigurasi	Perubahan dalam maklumat konfigurasi dan versi perlu dikawal selia dan aktiviti ini perlu mengambil kira perkara berikut: i. Penguatkuasaan, ii. Keupayaan Integrasi, dan iii. Kawalan Versi.
Perakuan Pematuan Konfigurasi dan Semakan Audit	Konfigurasi akan sentiasa berubah dan aktiviti ini memastikan pematuan konfigurasi dilaksanakan dengan teliti melalui prosedur yang betul.

Maklumat konfigurasi hendaklah diuruskan sepanjang kitaran hayat pembangunan produk. Konfigurasi dalam pelaksanaan DevOps adalah seperti di Lampiran 4: Konfigurasi dalam Pelaksanaan DevOps.

Tiga komponen utama dalam pengurusan konfigurasi adalah seperti berikut:

- a. **Repositori** - Repositori digunakan untuk menguruskan versi dan menyimpan item konfigurasi dengan selamat. Tiga jenis repositori pengurusan konfigurasi adalah:
 - i. **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),

- ii. **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
 - iii. **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.
- b. **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*,²² dan
- c. **Konfigurasi sebagai Kod** - Kod atau skrip yang menyeragamkan konfigurasi dalam sumber tertentu, seperti pelayan atau rangkaian. Kod atau skrip ini sentiasa digunakan sepanjang fasa pembangunan DevOps.

4.4.1.2.2. 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 4-11.

²² 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.

Jadual 4-11: Aktiviti Pengurusan Pindaan

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

- a. Penambahan dan penghapusan epik dari MVP atau produk yang akan datang,
- b. penambahan epik baru bagi ciri-ciri produk yang tidak disenaraikan dalam epik sedia ada,
- c. perubahan terhadap DoD bagi *iteration* semasa yang memerlukan penambahan masa projek,
- d. penyusunan semula keutamaan serahan epik dalam *roadmap* produk yang telah dipersetujui, dan

- e. perubahan pada garis masa kerana berlaku penambahan *iteration* yang perlu dilaksanakan bagi pembangunan MVP produk yang dipersetujui.

4.4.1.2.3. 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 rangkaian *tools* CI/CD secara proaktif dan berterusan.

Lima aktiviti yang terlibat dalam pengurusan insiden adalah seperti di Jadual 4-12.

Jadual 4-12: Aktiviti Pengurusan Insiden

Aktiviti-aktiviti Pengurusan Insiden	Huraian Aktiviti
Merekod Insiden	Pengesanan insiden berlaku lebih awal melalui penggunaan <i>tools</i> DevOps dan akan direkodkan secara automatik.
Berkomunikasi dengan Pihak Pemegang Taruh	<i>Tools</i> komunikasi digunakan untuk memaklumkan insiden dengan segera serta melaporkan situasi terkini insiden dari semasa ke semasa kepada pihak pemegang taruh. <i>Tools</i> komunikasi adalah penting dan perlu diintegrasikan dengan <i>tools</i> pengurusan insiden.
Mengenal pasti Resolusi Insiden	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.
Melaksanakan Kajian Penutupan Insiden	<i>Tools</i> pengurusan insiden mengeluarkan laporan insiden secara terperinci untuk makluman dan perhatian pemegang taruh. <i>Blameless post-mortem</i> dilakukan sebaik sahaja insiden selesai oleh pasukan.
Menyediakan Dokumentasi Insiden	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.

4.4.1.2.4. Pengurusan Pelepasan

Pengurusan pelepasan melibatkan aktiviti menyelaraskan 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 4-18.



Rajah 4-18: 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 mengautomatiskan 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 4-13.

Jadual 4-13: Aktiviti Pengurusan Pelepasan

Aktiviti-aktiviti Pengurusan Pelepasan	Huraian Aktiviti
Perancangan	i. Menentukan keperluan dan kriteria khusus untuk pelepasan. ii. Memastikan kriteria penerimaan produk ditakrifkan dengan jelas dalam pelan pelepasan.
Pembinaan	Menentukan kriteria penerimaan menepati keperluan yang ditetapkan bagi membina pelepasan.
Pengujian	Menyemak persekitaran pengujian untuk memastikan semua kriteria penerimaan pelepasan dipenuhi.
Penyediaan	Membolehkan pindaan dan kemas kini produk dilakukan tanpa memberi kesan langsung kepada versi semasa pengguna sedia ada.
Penempatan	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.

4.4.2. Pelaksanaan

Pelaksanaan dalam DevOps melibatkan fasa-fasa pengekodan, pembangunan, pengujian, pelepasan dan penempatan.

Pelaksanaan fasa-fasa DevOps berlaku secara iteratif sepanjang pembangunan sistem menggunakan *tools*. Ahli pasukan DevOps menggunakan sistem kawalan versi untuk mendaftar masuk kod dan membina 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.

Sebelum proses pelepasan dilaksanakan, pemilik 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 fasa 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.²³

Persekitaran-persekitaran 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

²³ 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.

mengurus dan menyelaraskan teknologi-teknologi²⁴ ini, antaranya seperti Terraform, Ansible dan Puppet.

Pasukan DevOps bertanggungjawab mengurus *tools* serta menambah baik proses kerja. Penggunaan *tools* komunikasi yang diintegrasikan dengan *pipeline* CI/CD memudahkan perkongsian maklumat antara ahli pasukan. 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, dan
- f. membincangkan maklum balas yang diterima secara bersama.

Amalan dalam pelaksanaan DevOps yang disenaraikan di atas tidak terhad dan boleh ditambah baik berdasarkan kepentingan dan keperluan agensi.

4.4.3. Kawalan

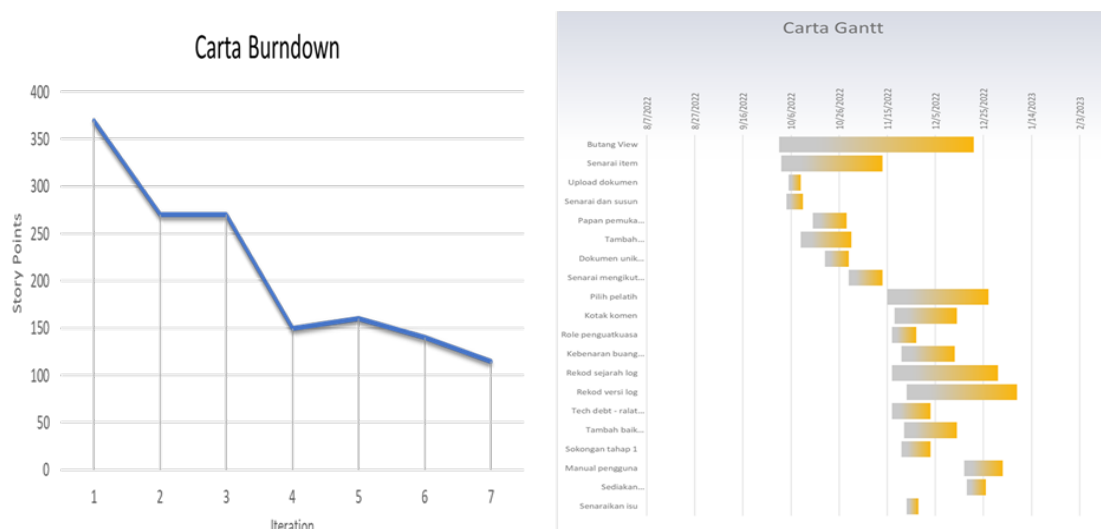
Perkara yang perlu dikawal bagi kejayaan sesuatu projek DevOps adalah pencapaian kemajuan projek dan kualiti.²⁵ Kawalan dalam DevOps diintegrasikan dalam *pipeline* CI/CD secara automatik menggunakan *tools* DevOps.

²⁴ 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.

²⁵ 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).

4.4.3.1. Kawalan Kemajuan Projek

Carta Burndown digunakan bagi menjejak pencapaian kemajuan projek dalam pelaksanaan *Agile* namun mekanisme seperti *Work Breakdown Structure* (WBS) dan carta Gantt masih boleh digunakan. WBS dalam *Agile* merujuk kepada tunggakan tugas iteration yang dijadualkan berdasarkan pelepasan, serta diperhalusi secara iteratif sepanjang pembangunan produk. Contoh carta Burndown dan carta Gantt serta WBS bagi memantau prestasi tunggakan tugas *iteration* adalah seperti di Rajah 4-19 dan Rajah 4-20.



Rajah 4-19: Contoh Carta Burndown dan Carta Gantt

Nama Projek: Projek ABC								
Jumlah Tugas		20	Tempoh Masa		91	Peratus Selesai		
Selesai		10	Tarikh Mula		10/1/2022	55%		
Dalam Proses		2	Tarikh Akhir		12/30/2022			
Belum Bermula		8						
40%								
ITERATION	TUGAS	AHLI	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 pemuka 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	Miss 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 4-20: Contoh WBS

4.4.3.2. Kawalan Kualiti

Kawalan kualiti dalam pelaksanaan DevOps antaranya adalah melalui kawalan versi, *exit criteria* dan pengujian automatik.

a. Kawalan Versi

Kawalan versi memainkan peranan penting bagi melaksanakan pembangunan produk yang berkualiti. Sistem kawalan versi yang sentiasa tersedia, membantu kepada pelaksanaan pembangunan yang berterusan. Lampiran 5: Kawalan Versi dalam Pelaksanaan DevOps adalah antara kawalan versi yang dilakukan dalam pelaksanaan DevOps.

b. *Exit Criteria*

Exit criteria merupakan satu set syarat yang telah ditetapkan dan dipenuhi untuk menamatkan aktiviti dalam fasa-fasa DevOps. *Exit criteria* yang ditetapkan perlu dibincang dan dipersetujui bersama oleh pemilik produk dan ahli pasukan.

Kriteria ini perlu dirangka semasa fasa perancangan dan perlu ada dalam setiap aktiviti di dalam pelaksanaan DevOps. Penambahbaikan ke atas *exit criteria* berlaku sepanjang proses pembangunan dan perlu dipatuhi. *Exit criteria* yang ringkas dan jelas akan membantu proses pelepasan berlaku secara berterusan dan pantas. Sebagai contoh, *exit criteria* bagi pengekodan adalah kod mesti didaftar masuk dengan semua kebergantungan yang ada ke dalam sistem kawalan versi.

Exit criteria adalah berbeza mengikut keperluan. Persetujuan semua ahli pasukan dalam mengenal pasti dan mematuhi *exit criteria* penting bagi menghasilkan tugas yang seragam dan berkualiti. *Exit criteria* dijadikan skrip *checklist* dalam *tools* DevOps bagi tugas yang telah diautomasi.

Senarai *Exit Criteria* bagi aktiviti utama pelaksanaan DevOps adalah seperti di Lampiran 6: Senarai *Exit Criteria* bagi Aktiviti Utama Pelaksanaan DevOps.

c. 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 7: Perbandingan Teknik Pengujian Automatik TDD dan BDD.

4.4.4. Penyelenggaraan

Fasa operasi DevOps menekankan pengautomasian proses penyelenggaraan dalam persekitaran produksi. Objektif utama fasa ini adalah memastikan *tools* dan sistem berfungsi dengan prestasi optimum semasa pelepasan, *update*, *patch*, peningkatan versi dan penyelenggaraan berjadual seperti pembaharuan lesen dan sijil (certificate).

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. Pengautomasian penyelenggaraan meningkatkan kebolehpercayaan (reliability) dan ketersediaan sistem, mengurangkan masa henti serta mengukuhkan keselamatan dan tadbir urus.

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.

4.4.5. Pemantauan

Pemantauan ialah fasa bagi memantau dan mengesan isu pematuhan serta ancaman keselamatan pengoperasian. Fasa ini membantu pasukan projek memantau dan mengkaji metrik penting yang berkaitan untuk menyelesaikan isu tersebut dengan pantas. Tujuan pemantauan adalah bagi:

- a. meningkatkan ketelusan dan keterlihatan operasi ICT dan rangkaian, terutamanya yang boleh mencetuskan insiden keselamatan melalui sistem amaran secara masa nyata,
- b. mengenal pasti isu prestasi pengoperasian dan mencari penyelesaian yang sesuai sebelum kerosakan yang ketara dalam infrastruktur dan rangkaian, dan
- c. membantu menjejaki maklum balas pengguna selepas aplikasi dikemas kini ke persekitaran ujian atau produksi bagi melihat kesan positif, negatif atau neutral pada pengalaman pengguna.

4.4.5.1. Elemen Pemantauan

Tools pemantauan DevOps terdiri daripada tiga elemen utama iaitu pemantauan prestasi aplikasi, pemantauan infrastruktur dan pemantauan rangkaian seperti berikut:

a. Pemantauan prestasi aplikasi

Produk yang diguna pakai oleh pengguna perlu bebas daripada sebarang gangguan. Metrik yang digunakan untuk aktiviti pemantauan tersebut adalah termasuk masa aplikasi beroperasi, masa dan jumlah transaksi, tindak balas sistem, tindak balas API, dan kestabilan aplikasi dari segi *back-end* dan *front-end*.

b. Pemantauan infrastruktur

Infrastruktur platform CI/CD seperti *tools*, persekitaran, sistem pengendalian, storan, peranti dan aplikasi perlu dipantau. Semasa pemantauan dilaksanakan, data dari *pipeline* CI/CD yang merangkumi ketersediaan, keselarasan, tugasan dan sebagainya perlu dikumpul dan diperiksa bagi memastikan infrastruktur berada di tahap optimum.

c. Pemantauan rangkaian

Aktiviti pemantauan adalah seperti menjejaki rangkaian, memantau status dan fungsi *firewall*, *routers*, *switches*, *server*, *virtual machine* dan sebagainya. Pemantauan rangkaian yang proaktif berupaya mengesan kebarangkalian gangguan rangkaian. Pemantauan proaktif yang cekap juga boleh mengelakkan masa henti (downtime) atau kegagalan rangkaian.

Contoh metrik-metrik pemantauan prestasi aplikasi, infrastruktur dan rangkaian adalah seperti di Lampiran 8: Metrik Pemantauan DevOps.

4.4.5.2. Kaedah Pemantauan

Sistem pengurusan yang berkesan mengandungi alat pemantauan *built-in* yang boleh membantu pasukan DevOps mengurangkan tenaga kerja dan mengautomasikan teknik asas penyelesaian isu. Tiga kaedah pemantauan secara berterusan adalah seperti berikut:

a. Pemantauan melalui *tools*

Tools pemantauan yang perlu dan sesuai bagi persekitaran produksi perlu dikenal pasti. *Tools* pemantauan prestasi aplikasi, infrastruktur serta rangkaian mempunyai fungsi berbeza mengikut jenis pemantauan yang diperlukan.

b. Pemantauan melalui pemberitahuan

Tools pemantauan mempunyai keupayaan pemberitahuan sekiranya terdapat sebarang anomali atau ralat. Pemberitahuan ini berlaku secara masa nyata dan boleh dikonfigurasi untuk disampaikan kepada ahli pasukan berdasarkan peranan, jenis sistem dan tahap kritikal setiap insiden.

c. Pemantauan berdasarkan tahap *threshold*

Tahap *threshold* yang ditetapkan membantu dalam pemantauan sumber dan perkhidmatan infrastruktur serta rangkaian secara proaktif. Setiap peranti mempunyai tahap *threshold* yang ditetapkan berdasarkan keutamaan dan keperluan. Pengesanan awal membantu tindak balas serta merta dibuat sekiranya tahap *threshold* melebihi kadar yang ditetapkan.

4.5. TERAS 5: PEMERKASAAN TEKNOLOGI

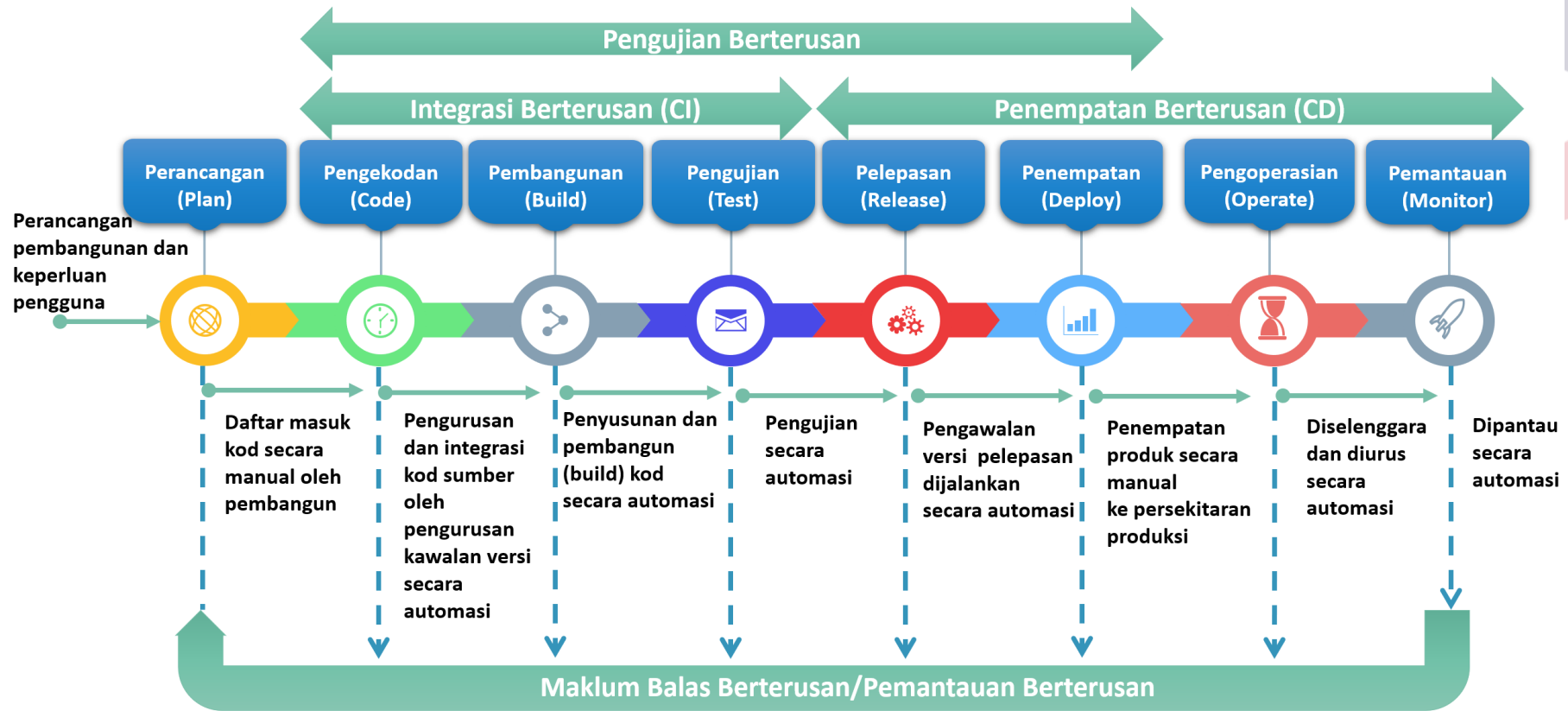
Infrastruktur DevOps diperlukan bagi mewujudkan persekitaran yang menyokong pembangunan dan penyampaian produk. Dua elemen penting dalam infrastruktur DevOps bagi meningkatkan pembangunan produk adalah pengintegrasian dan pengautomasian tugas manual melalui penggunaan *tools*. Seni bina DevOps yang menggabungkan 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.

4.5.1. 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 4-21.^{26, 27}

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

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



Rajah 4-21: Infrastruktur *Pipeline* DevOps

Berikut adalah penerangan proses yang berlaku dalam infrastruktur *pipeline* DevOps berdasarkan Rajah 4-21.

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 fasa 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.

4.5.1.1. Pembentukan Infrastruktur *Pipeline* DevOps

Berdasarkan Rajah 4-21, 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 mengautomatiskan 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 fasa produksi, maklumat balas dari pengguna terhadap penggunaan produk dapat diperolehi untuk penambahbaikan berterusan.

4.5.2. Persekitaran *Pipeline* DevOps

Persekitaran yang menyokong pelaksanaan *pipeline* DevOps perlu diwujudkan seperti yang diterangkan dalam Rajah 4-22.^{28, 29}



Rajah 4-22: Pembinaan Persekitaran ke Arah Pelaksanaan *Pipeline* DevOps

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

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

²⁹ 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;.

a. Perancangan dan Keperluan Pengguna

Agensi perlu menilai ketidakcekan semasa perancangan dan keperluan pengguna sebelum memulakan penambahbaikan.

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

- i. peratusan sumber (masa, personel dan peruntukan) agensi yang diperuntukkan 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 fasa 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 Persekitaran

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

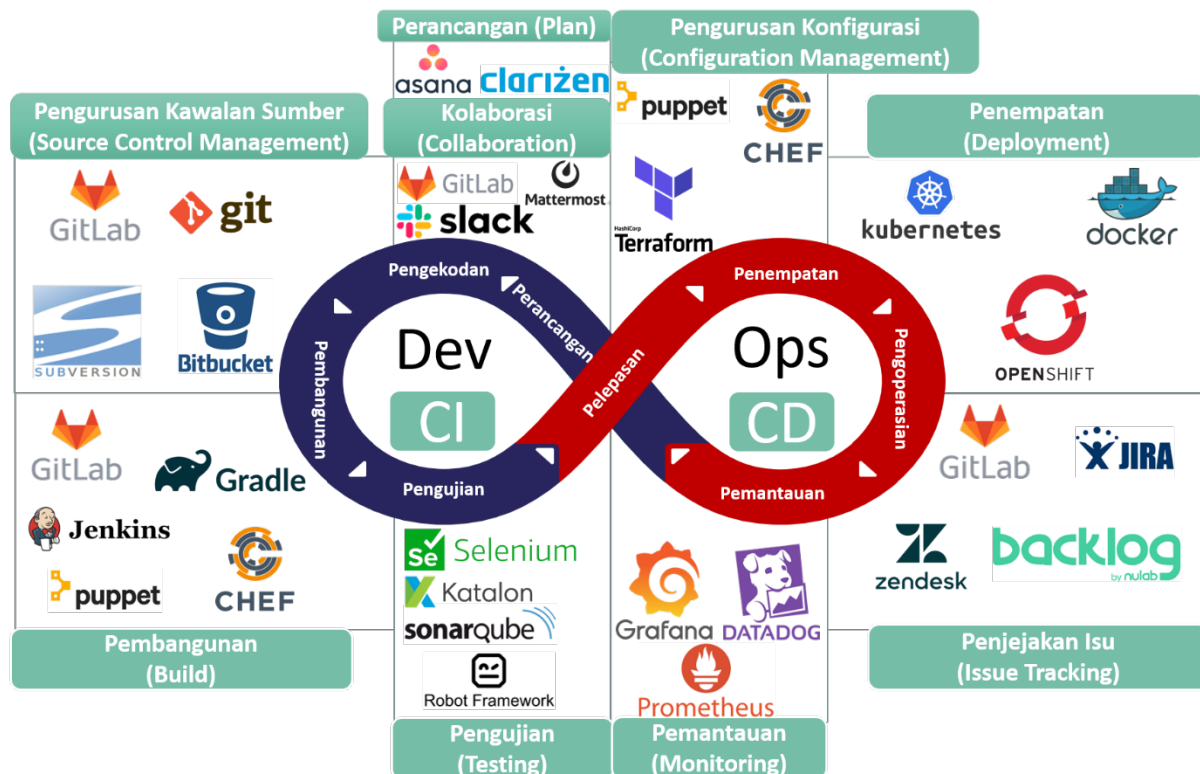
Pemantauan dan operasi dilaksanakan untuk memastikan produk berfungsi secara optimum dalam persekitaran produksi. Metrik utama pemantauan dan operasi adalah:

- 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.

4.5.3. Tools DevOps

Tools DevOps ialah perisian yang membantu pasukan pembangun dan operasi mengautomatiskan proses pembangunan produk di dalam *pipeline* DevOps. Penggunaan *tools* DevOps juga menggalakkan komunikasi dan kerjasama yang lebih lancar antara pihak pengurusan, pemilik produk, pasukan pembangun dan pasukan operasi. Kategori rantaian *tools* DevOps dan peranan *tools* bagi setiap fasa DevOps adalah seperti di Rajah 4-23.



Rajah 4-23: *Tools* untuk Persekitaran *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 menjejaki isu bagi membolehkan penyelesaian dibuat dengan pantas dan cekap. Contoh *tools* penjejakan isu adalah GitLab, Jira, ZenDesk dan Backlog.

j. *Tools* Pemantauan (Monitoring)

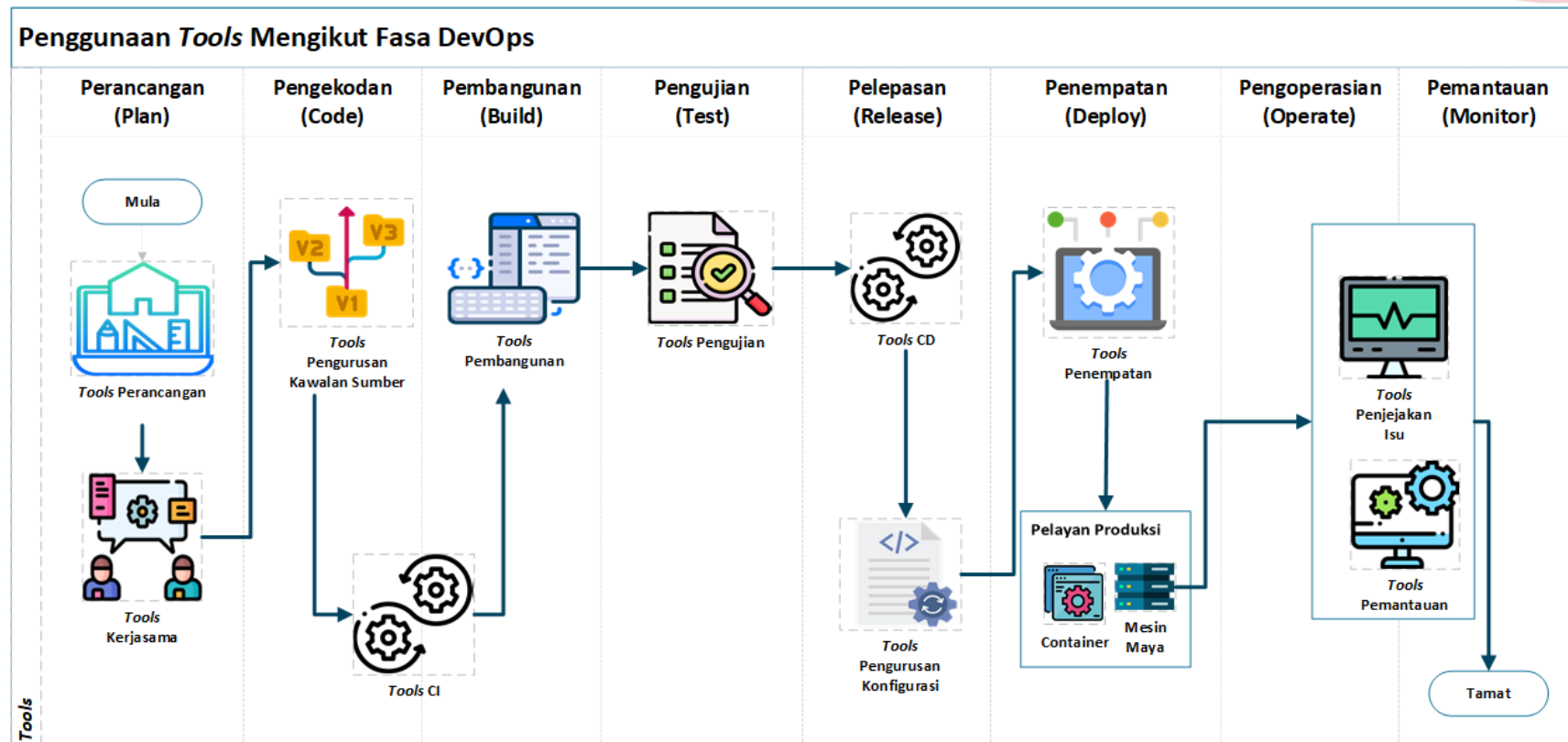
Menganalisis data dan memberi gambaran keseluruhan prestasi sistem kepada pasukan dalam mengenal pasti dan menangani isu berkaitan sistem. *Tools* pemantauan dapat memastikan masa operasi perkhidmatan dan prestasi sistem berada pada tahap optimum. Contoh *tools* pemantauan adalah Elastic Stack, Jaeger, Prometheus, Grafana, Data Dog, Nagios dan Splunk.

Senarai *tools* DevOps berdasarkan 17 kategori yang dihasilkan oleh pengamal DevOps adalah seperti dalam Lampiran 9: Senarai *Tools* DevOps.³⁰

³⁰ 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>.

4.5.4. Penggunaan *Tools* untuk *Pipeline* 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 tugas proses bagi kitar hayat pembangunan sistem berdasarkan fasa DevOps dijelaskan seperti di Rajah 4-24.



Rajah 4-24: Penggunaan *Tools* Mengikut Fasa DevOps

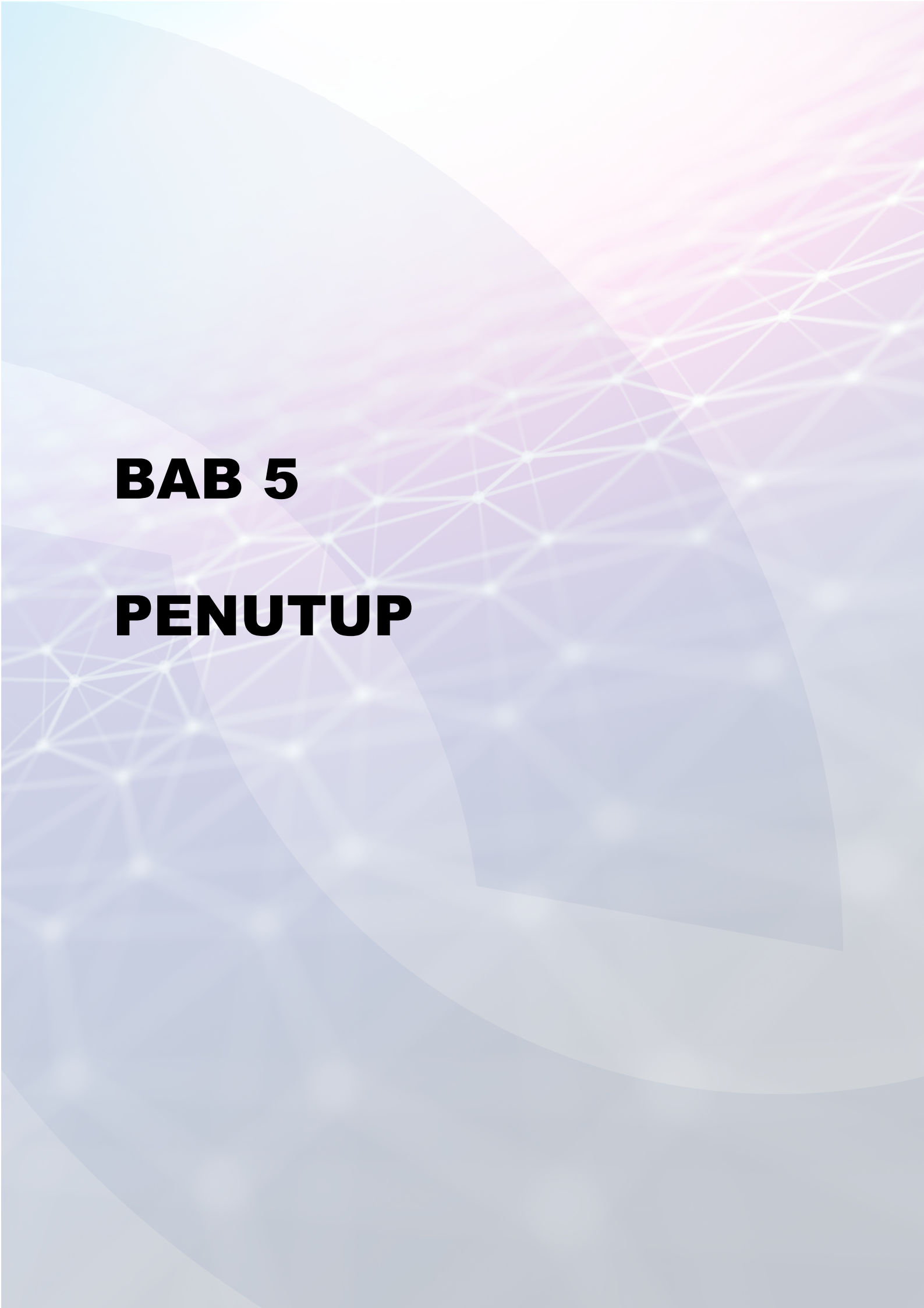
Jadual 4-14 menyenaraikan aliran tugas bagi *tools* mengikut fasa DevOps.

Jadual 4-14: Aliran Tugas *Tools* Mengikut Fasa DevOps

Fasa DevOps	Tugas <i>Tools</i>
Perancangan (Plan)	i. Pasukan akan menggunakan <i>tools</i> perancangan untuk a. mendokumentasikan kajian keperluan pengguna, b. melaksanakan pengagihan tugas, dan c. menetapkan garis masa jadual pelaksanaan projek. ii. Penyelarasan aktiviti pembangunan akan menggunakan <i>tools</i> kolaborasi.
Pengekodan (Code)	i. Pembangun akan mendaftar masuk kod sumber ke dalam <i>tools</i> pengurusan kawalan sumber. ii. <i>Tools</i> pengurusan kawalan sumber akan melaksanakan tugas berikut; a. mengurus perubahan dan kemas kini kod daripada pembangun, b. melaksanakan analisis kod statik, dan c. melaksanakan semakan kod secara automasi dan <i>peer review</i>.
Pembangunan (Build)	i. <i>Tools</i> CI akan mengesan perubahan pada kod sumber dan dengan integrasi bersama <i>tools</i> pembangunan, langkah berikut berdasarkan skrip konfigurasi; a. <i>Build</i> - penyusunan kod, b. <i>Unit test</i> - pengujian unit, dan c. <i>Release</i> - imej akan dihantar untuk pengujian lanjut.
Pengujian (Test)	i. Imej yang telah dijana seterusnya akan <i>deploy</i> oleh <i>tools</i> CI 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>.
Pelepasan (Release)	i. <i>Tools</i> CD akan melaksanakan tugas berikut untuk fasa pelepasan (release) setelah siri ujian lulus pada fasa pengujian. a. Penyediaan nota pelepasan (release note), dan b. Penandaan versi (version tagging).

Fasa DevOps	Tugasan <i>Tools</i>
Penempatan (Deploy)	i. <i>Tools</i> penempatan akan melaksanakan tugas berikut; a. Pengemaskinian infrastruktur, b. Pengesahan untuk penempatan (deployment verification), dan c. Pemasangan imej ke pelayan produksi.
Pengoperasian (Operate) dan Pemantauan (Monitor)	i. <i>Tools</i> penjejakan isu akan merekod dan membantu pasukan dalam memantau perkembangan isu dan log ralat. ii. <i>Tools</i> pemantauan membantu pasukan menganalisis data dan memantau prestasi sistem bagi memastikan prestasi sistem aplikasi berada pada tahap optimum.

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.



BAB 5

PENUTUP

BAB 5 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 penyesuaiannya terhadap pelaksanaan DevOps di sektor awam.

Dokumen ini secara asasnya merangkumi matlamat dan manfaat pelaksanaan DevOps, dan lima teras yang merupakan elemen utama dalam melaksanakan amalan DevOps iaitu Penerapan Prinsip dan Budaya, Pengadaptasian Metodologi, Pengukuhan Tadbir Urus, Pemantapan Pelaksanaan dan Pemantauan serta Pemerkasaan Teknologi. Setiap teras 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 di dalamnya. 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 fasa 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.

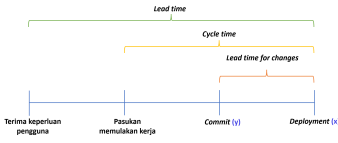


LAMPIRAN

Lampiran 1: Metrik Pengukuran Tahap Pelaksanaan DevOps³¹

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)
Metrik yang Diukur	Mengukur kekerapan penempatan produk ke persekitaran produksi	Mengukur tempoh masa yang diambil untuk kod sumber didaftar (commit) sehingga ke fasa 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)

³¹ 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 Pemulihan (Mean Time to Restore Service (MTTR))	Kadar Kegagalan Perubahan (Change Failure Rate)
Formula Pengiraan	Masa Pusingan Perubahan = Bilangan Penempatan Setiap Hari	Masa Pusingan Perubahan $= x - y$ $= (\text{Masa Penempatan} - \text{Masa Perubahan})$  i. Masa perubahan ialah apabila kod didaftar masuk ke dalam repositori (commit) ii. Masa penempatan ialah apabila penempatan produk ke persekitaran produksi	MTTR $= \frac{\text{Jumlah Masa Penyelesaian Insiden}}{\text{Jumlah Insiden Berlaku}}$ MTTR - menjumlahkan semua masa penyelesaian insiden dan membahagikannya dengan bilangan kejadian insiden. Contoh, sistem tidak berfungsi selama 30 minit dalam dua insiden berasingan. $MTTR = \frac{30}{2} = 15 \text{ minit}$	Kadar Kegagalan Perubahan $= \frac{\text{Bilangan Penempatan yang Menyebabkan Insiden}}{\text{Jumlah Bilangan Penempatan}}$

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	46-60%
	Rendah	Minimum sebulan sekali Maksimum tiga bulan sekali		Lebih dari satu hari	

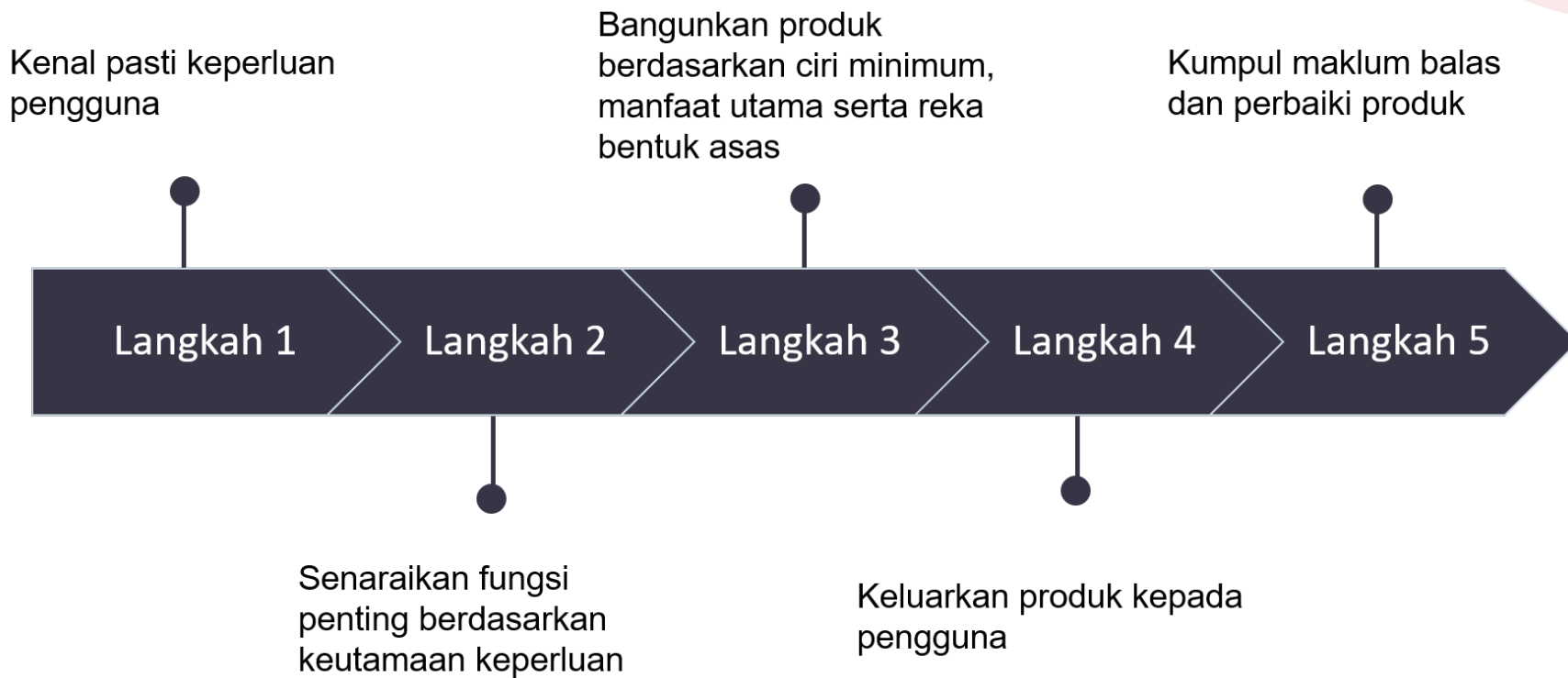
Metrik	Kekerapan Penempatan (Deployment Frequency)	Masa Pusingan Perubahan (Lead Time for Change)	Purata Masa untuk Pemulihan (Mean Time to Restore Service (MTTR))	Kadar Kegagalan Perubahan (Change Failure Rate)
Faktor Prestasi Penurunan	i. Perubahan yang besar diperkenalkan dalam kod, ii. kekurangan sumber manusia atau perubahan dalam struktur organisasi, dan iii. ketidakcekapan dalam proses pembangunan.	i. Perubahan yang besar diperkenalkan dalam kod, ii. keperluan pengguna yang tidak jelas, iii. pengujian dijalankan secara manual, dan iv. ketidakcekapan dalam proses pembangunan.	i. Pengujian dijalankan secara manual, ii. kekurangan sumber manusia dalam struktur organisasi, dan iii. ketidakcekapan dalam proses pengurusan insiden.	i. Kebergantungan pada proses penempatan secara manual meningkatkan risiko kesilapan, ii. pengujian dijalankan secara manual, iii. kualiti kod yang lemah sehingga sukar diselenggara iv. pengenalan kod baharu membawa kepada pengujian yang lebih kompleks, dan v. ralat sistem aplikasi yang tidak dijangka.

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)
Cara Meningkatkan Prestasi	i. Melaksanakan amalan terbaik CI/CD secara berterusan - membolehkan peningkatan kadar penyampaian sistem aplikasi ii. Melaksanakan versi perubahan pada skala kecil - memudahkan pasukan mengambil tindakan penambahbaikan produk berdasarkan maklum balas dan menyelesaikan isu lebih cepat mengikut versi perubahan	i. Melaksanakan versi perubahan pada skala kecil - memudahkan pasukan mengambil tindakan penambahbaikan produk berdasarkan maklum balas dan menyelesaikan isu lebih cepat mengikut versi perubahan ii. Mengautomasikan pengujian - menggabungkan ujian automatik pada setiap peringkat <i>pipeline</i> CI/CD membantu mengurangkan masa penyampaian	i. Meningkatkan pengetahuan dan kemahiran ahli pasukan - sumber manusia yang berpengetahuan boleh mengenal pasti punca masalah dengan cepat dan menyelesaikannya dengan segera ii. Menggunakan automasi ujian - menggabungkan ujian automatik pada setiap peringkat <i>pipeline</i> CI/CD membantu mengurangkan masa penyampaian dan membantu proses baik pulih dengan lebih cepat	i. Melaksanakan versi perubahan pada skala kecil - memudahkan pasukan mengambil tindakan penambahbaikan produk berdasarkan maklum balas dan menyelesaikan isu lebih cepat mengikut versi perubahan ii. Mengautomasikan pengujian - menggabungkan ujian automatik pada setiap peringkat <i>pipeline</i> CI/CD membantu mengurangkan masa penyampaian

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)
		iii. Mengautomasikan semakan kod - menggunakan <i>tools</i> semakan kod automatik seperti SonarQube dan GitHub	iii. Menyediakan/mengemas kini dokumentasi - mendokumentasikan proses pengurusan insiden dengan lengkap dan teratur iv. Menyediakan latihan - Melatih ahli pasukan secara berterusan tentang proses dan cara bertindak sekiranya berlaku insiden v. Mengautomasikan proses pengurusan insiden - Mengenal pasti langkah yang berpotensi dalam proses pengurusan insiden untuk diautomasikan	iii. Mengautomasikan semakan kod - menggunakan <i>tools</i> semakan kod automatik membantu agensi meningkatkan kualiti kod dan menjimatkan masa

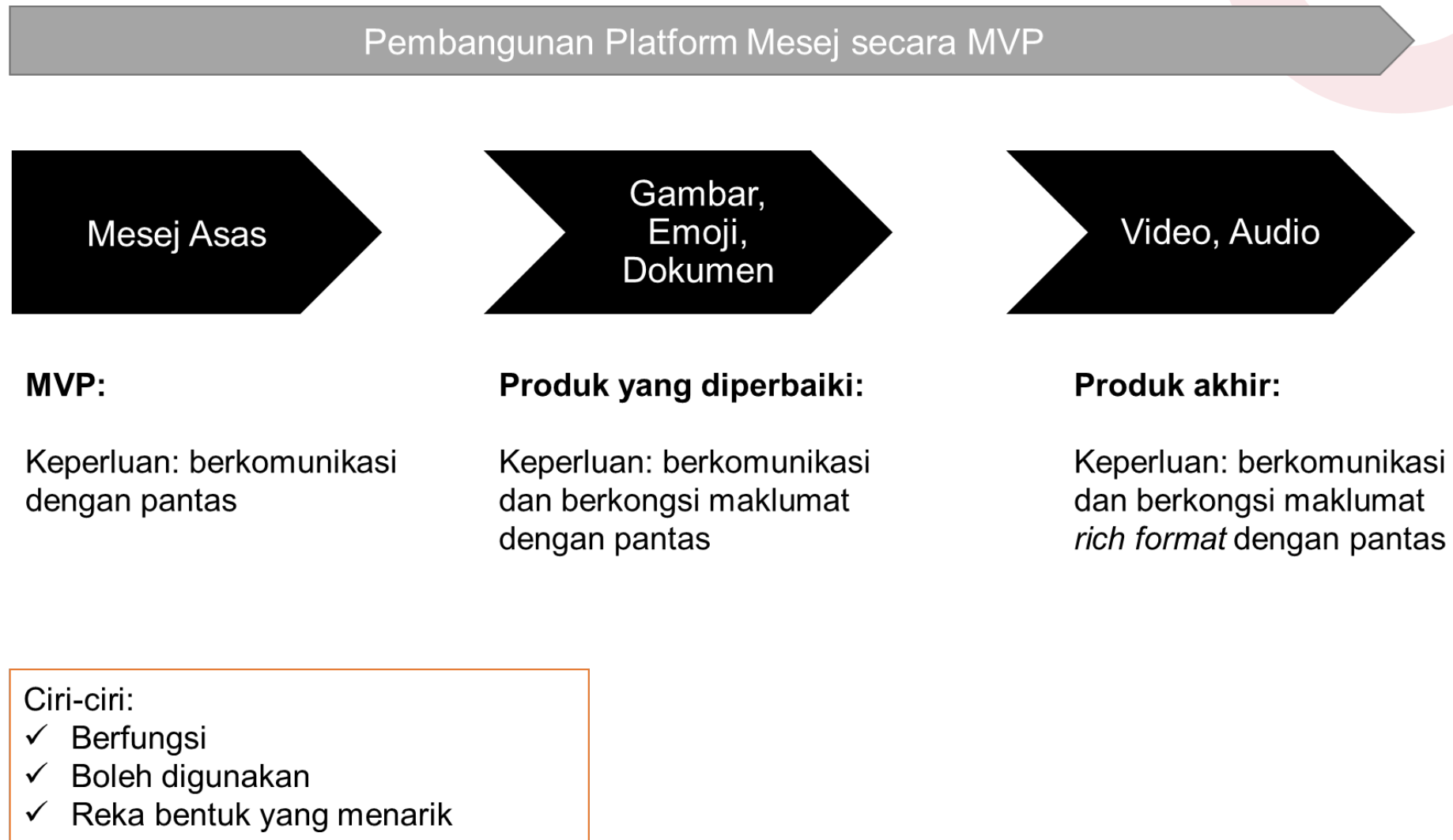
Lampiran 2: Konsep MVP dalam Pembangunan Produk

a. Langkah-langkah bagi pembangunan MVP adalah seperti berikut:³²



³² 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



Lampiran 3: 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
Pengurusan Konfigurasi	Ketua Pasukan	i. Menguruskan pelan dan dasar pengurusan konfigurasi, dan ii. mengesahkan pindaan konfigurasi tidak bercanggah dengan konfigurasi lain dalam <i>pipeline</i> CI/CD.
	Jurutera Konfigurasi Automasi /	i. Mendapatkan maklumat bagi perubahan konfigurasi, ii. menilai aktiviti pelaksanaan pindaan konfigurasi melalui risiko dan impak, iii. menilai keadaan konfigurasi sebelum dan selepas pengubahsuaian, iv. mengesahkan bahawa butiran konfigurasi bagi <i>pipeline</i> CI/CD adalah betul dan lengkap, v. mengesahkan bahawa pengubahsuaian konfigurasi yang dicadangkan mematuhi dasar pengurusan konfigurasi, jika ada, vi. mengkonfigurasi item ke dalam pangkalan data pengurusan konfigurasi, dan vii. mengesahkan bahawa butiran konfigurasi dikemas kini dengan teratur.
Pengurusan Pindaan	Pemilik Produk	i. Memastikan skop pindaan tidak bercanggah dengan ciri-ciri keperluan dan fungsi produk sistem aplikasi, dan ii. memastikan kriteria penerimaan dikemas kini setelah pindaan diluluskan.
	Ketua Pasukan	i. Mereka bentuk pendekatan strategi pengurusan pindaan, ii. menilai kesan serta risiko pindaan, dan iii. menjalankan semakan untuk menilai keputusan dan prestasi yang berkaitan dengan permintaan pindaan.

Pengurusan	Peranan	Tanggungjawab
	Pembangun	i. Mengurus permintaan pindaan, ii. menilai impak pindaan secara berpasukan, dan iii. melaksanakan pindaan dan pengujian pindaan.
Pengurusan Pelepasan	Pemilik Produk	i. Menentukan skop pelepasan yang bersesuaian, dan ii. menakrif skop pelepasan berdasarkan MVP bersama pasukan.
	Pengurus Projek	i. Memastikan pelepasan dilakukan berdasarkan jangka masa yang telah dipersetujui, dan ii. memastikan sumber bagi pelepasan mencukupi dan mematuhi proses pelepasan.
	Jurutera Kualiti	i. Memastikan ciri-ciri penerimaan skop bagi pelepasan adalah seperti yang dipersetujui oleh pemilik produk, ii. mengesan ralat semasa proses pelepasan dan melakukan pembetulan awal, dan iii. memastikan pengurangan risiko pengulangan ralat dalam persekitaran produksi.
	Jurutera Pelepasan	i. Memastikan penciptaan rangkaian <i>tools</i> pelepasan bagi menjayakan pelepasan, ii. menyelaraskan skop pelepasan automatik, jika ada, iii. memastikan pematuan proses pelepasan, iv. mengawal dan mengatasi permasalahan berkaitan pelepasan, dan v. menyelia proses dan <i>tools</i> pelepasan bagi menyokong pelepasan berterusan.
Pengurusan Insiden	Pengurus Projek	i. Memastikan pengurusan insiden dilakukan mengikut proses yang dipersetujui, dan ii. mengkoordinasi komunikasi ahli pasukan, pemilik produk serta pemegang taruh.
	Pembangun	i. Mengurus insiden atau isu yang timbul semasa proses pembangunan,

Pengurusan	Peranan	Tanggungjawab
		ii. mengurus insiden atau isu yang timbul dalam persekitaran produksi, dan iii. bekerjasama dalam menyelesaikan insiden.

Lampiran 4: 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 5: Kawalan Versi dalam Pelaksanaan DevOps

- a. Kod dan data yang diperlukan untuk membina sistem aplikasi, konfigurasi infrastruktur dan *pipeline* CI/CD hendaklah disimpan di dalam repositori kawalan versi,
- b. Maklumat pengujian dan skrip ujian yang diperlukan untuk mengesahkan aplikasi, konfigurasi infrastruktur dan *pipeline* CI/CD hendaklah disimpan di dalam repositori kawalan versi,
- c. Keputusan ujian dan proses semakan yang digunakan untuk menilai kualiti aplikasi, konfigurasi infrastruktur dan *pipeline* CI/CD hendaklah disimpan di 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 pengesanan bencana yang berupaya untuk memulihkan sistem pengurusan kawalan versi perlu tersedia dan berfungsi pada bila-bila masa.

Lampiran 6: Senarai *Exit Criteria* bagi Aktiviti Utama Pelaksanaan DevOps

Aktiviti-aktiviti dan contoh *exit criteria* yang boleh diguna pakai adalah seperti berikut:

Aktiviti-aktiviti	<i>Exit criteria</i>
Iteration	i. Semua cerita pengguna dalam tunggakan <i>iteration</i> telah diselesaikan, ii. Semua tugas reka bentuk dan tugas khas dalam <i>iteration</i> telah diselesaikan, iii. Isu yang kritikal dengan keutamaan tinggi telah diselesaikan, dan iv. Ujian regresi bagi semua ciri-ciri yang dibangunkan dalam <i>iteration</i> sebelumnya telah dilaksanakan.
Cerita Pengguna	i. Semua tugas yang ditakrifkan untuk cerita pengguna telah diselesaikan, ii. Isu berkaitan cerita pengguna telah dibincangkan, diselesaikan dan diuji semula, iii. Semua isu kritikal berkaitan cerita pengguna telah diselesaikan, iv. Ujian untuk cerita pengguna telah ditambah ke dalam ujian regresi, dan v. Data ujian dan konfigurasi bagi ujian automasi untuk cerita pengguna telah ditambah ke sistem pengujian automatik.
Pengekodan	i. Semakan kod telah dilakukan oleh rakan pembangun, ii. Semakan analisis statik telah dijalankan pada kod dan semua ralat dalam kod telah diselesaikan, iii. Kod yang ditulis mesti memenuhi kriteria penerimaan pelepasan, iv. Kod mesti menggunakan piawaian yang <i>standard</i>, menggunakan konvensyen pengekodan yang dipersetujui ahli pasukan, dan v. Kod telah disemak masuk dengan semua kebergantungan terkini.
Pengujian unit	i. Skrip ujian ditulis dan dikemas kini dalam sistem pengurusan ujian, ii. Ujian dilaksanakan dan ralat direkodkan berdasarkan cerita pengguna, dan iii. Ujian yang berkaitan telah ditambah ke pek pengujian regresi.
Semakan Kod	i. Semakan kod telah dilakukan berdasarkan cerita pengguna, dan ii. Maklum balas telah diambil kira dan kod diperbaiki.
Pengujian Automatik	i. Skrip ujian yang disediakan telah dimasukkan ke dalam ujian automasi bagi cerita pengguna yang berkaitan, dan ii. Skrip ujian dilaksanakan sekurang-kurangnya sekali sebelum pelepasan.

Lampiran 7: Perbandingan Teknik Pengujian Automatik TDD dan BDD

Perbandingan bagi teknik pengujian automatik TDD dan BDD adalah seperti berikut:³³

Aktiviti	<i>Test-Driven Development (TDD)</i>	<i>Behaviour-Driven Development (BDD)</i>
Penerangan	Pembangun membina kes-kes ujian berdasarkan cerita pengguna, sebelum fungsi produk dibina.	Pemilik produk menerangkan tingkah laku pengguna dan penerangan ini digunakan bagi membina kes-kes ujian.
Kaedah pelaksanaan	i. Pembangun menulis kes ujian berdasarkan keperluan pengguna, ii. Pengujian dilakukan ke atas aplikasi produk. Oleh kerana fungsi dalam skop ini belum dibina, keputusan ujian akan menjadi negatif atau gagal, iii. Pengekodan dan pengujian dilakukan oleh pembangun sehingga hasil ujian positif atau berjaya, iv. Pembangun <i>refactor</i> kod untuk membaiki kualiti kod, dan v. Fungsi produk yang berjaya diuji dihasilkan.	i. Tingkah laku pengguna dalam penggunaan produk dijelaskan kepada pembangun, ii. Pembangun menulis tingkah laku pengguna yang diterangkan ke dalam kes-kes ujian, iii. Pengekodan dan pengujian dilakukan sehingga tingkah laku pengguna seperti dalam kes ujian diperolehi, iv. Pembangun <i>refactor</i> kod untuk membaiki kualiti kod, dan v. Fungsi produk yang berjaya diuji dihasilkan.

³³ Devops Agile Skills Association LLC. (2020, September). DASA DevOps Fundamentals (2.0.1 ed.) [E-book]. Devops Agile Skills Association LLC.

Lampiran 8: Metrik Pemantauan DevOps

Metrik yang boleh digunakan bagi aktiviti-aktiviti pemantauan adalah seperti berikut:

Aktiviti Pemantauan	Metrik Pemantauan
Prestasi Aplikasi	i. Skor Kepuasan Pengguna - mengukur tahap kepuasan pengguna berdasarkan masa tindak balas permintaan apabila berinteraksi dengan aplikasi, ii. Masa Tindak Balas - jangka masa yang diambil oleh pengguna aplikasi untuk menerima tindak balas daripada aplikasi, iii. Kadar Ralat - bilangan atau jumlah ralat aplikasi dalam tempoh masa tertentu, iv. Bilangan/Kadar Permintaan Transaksi - bilangan permintaan transaksi aplikasi dalam tempoh masa tertentu, v. CPU (<i>Central Processing Unit</i>) – penggunaan CPU asas, dan vi. Ketersediaan/Masa Aktif Aplikasi – tempoh masa aplikasi beroperasi dan berfungsi dalam memenuhi permintaan pengguna.
Pemantauan Infrastruktur	i. Aliran penggunaan keadaan infrastruktur - aliran penggunaan keadaan hos seperti penggunaan cakera, CPU, memori, <i>container</i>, penggunaan perkhidmatan sistem, sistem maya dan storan, ii. Garis dasar dan perbandingan penggunaan - garis dasar dan perbandingan sebelum dan selepas penggunaan infrastruktur, iii. Profil penggunaan pangkalan data - memprofilkan penggunaan pangkalan data infrastruktur, iv. Penjejak prestasi - penjejakan prestasi melalui perubahan dalam infrastruktur, v. Fail log amaran - fail log amaran dan rujukan ralat yang senang dan boleh dicari, vi. Keselamatan dan masa tindak balas infrastruktur, dan vii. Daya pemprosesan aplikasi (<i>throughput</i>).

Aktiviti Pemantauan	Metrik Pemantauan
Pemantauan Rangkaian	i. Kesesakan rangkaian (<i>latency</i>), ii. Metrik <i>port</i> yang berbeza, iii. Penggunaan jalur lebar (<i>bandwidth</i>), dan iv. Aliran pakej rangkaian (<i>network packet flow</i>).



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.
- Chrissis 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.
- 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 toasca,” 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, Gumaei A (2020). Readiness model for DevOps implementation in software organizations. *Journal of Software Evolution and Process*. 33. 10.1002/smr.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 | 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

MAMPU merakamkan setinggi-tinggi penghargaan dan ucapan terima kasih kepada semua pihak yang telah memberikan komitmen dan kerjasama dalam usaha merangka, menyelaraskan 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

Datuk Seri Dr. Yusof bin Ismail
Dr. Fazidah binti Abu Bakar

PENGARAH PROJEK

Hanissull Jalis binti Md Yusof

PENGURUS PROJEK

Zuraidah binti Mat Harun

SME/JAMINAN KUALITI

Nik Zalbiha binti Nik Mat
Subramani a/l Nagaiah
Dr. Ali bin Seman
Ts. Mohd Hasiady bin Yasin
Ts. Roshaimieza binti Mat Adam
Ts. Ibrahim bin Abdulah
Ts. Dr. Mohamed Hairul bin Othman
Chua Swee Suan
Harisfazilah bin Jamel
Surazi bin Mohd Jais

PASUKAN PROJEK

Zamzuri bin Abdullah
Shahiruddin bin Taib
Haslidah binti Halim
Elisah Satim
Khairulbariah binti Mohammad Lukman
Rohani binti Haji Yusoff
Ts. Khairul Ashraf bin Basul Hak
Kholijah binti Norbib
Mohd Syukor bin Abdul
Nurul Ayuni binti Hj. Juhari
Farid Iqbal bin Ibrahim
Roth Camdessus Anak Robert
Adam Muhammad Richard
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
Andrew Yap Kok Keong
Mohamad Hanif bin Khodit
Haslana binti Ramlan
Hanizah binti Baharum
Mohd Zaki bin Zubahir
Abdullah Fathi bin Muhamad Azmi
Hisham bin Aderis
Azmi bin Md Nor
Nur Fazhila Syafwan Effendi
Nurul Shazira binti Saifuzzaman
Wan Normimi Farhana binti Wan Mansor
Zudiana binti Zulkapli

MUNSYI MUDA BAHASA

Siti Noorasmah binti Hashim

AGENSI TERLIBAT

KEMENTERIAN PENDIDIKAN MALAYSIA

Harlina binti Othman
Mohd Fauzie bin Mokhtar

KEMENTERIAN PERTAHANAN MALAYSIA

Nani Lee Yer Fui
Mohd Arif bin Jamal

KEMENTERIAN PENGANGKUTAN MALAYSIA

Ts. Subramani a/l Paiduthaly
Nurul Bariah binti Abu Bakar

JABATAN IMIGRESEN MALAYSIA

Azurawati binti Abdul Halim

JABATAN PERANGKAAN MALAYSIA

Ts. Sabrina binti Abdul Rahim
Khirum Anisa binti Ahmad Samuri

JABATAN PERKHIDMATAN AWAM

Norazwana binti Mohamed Kamari
Ts. Hariadi bin Hinta
Ahmad Naufal bin Jamri

UNIT PENYELARASAN PELAKSANAAN

Abdullah bin Muhammad

INDUSTRI

Mohamad Zulfahmy bin Roslan
Muhammad Fauzilkamil bin Zainuddin
Iszuddin Ismail
Mohd Saidy bin Mohd Yusof
Ts. Fariq bin Abdullah