

1.1 PENGENALAN

1.1.1 APAKAH KRISA (DEFINISI)?

KRISA merupakan panduan praktikal berbentuk “how-to” yang mengandungi pendekatan dan format dokumen standard bagi pembangunan sistem aplikasi sektor awam. Ia menyediakan kaedah yang berstruktur, selaras dan komprehensif bagi membantu agensi sektor awam melaksanakan perancangan, pembangunan, dokumentasi dan serahan projek secara sistematik untuk mencapai kualiti yang optimum dalam pembangunan sistem.

1.1.2 OBJEKTIF KRISA

Objektif KRISA adalah untuk:

- a) memperkenalkan prosedur standard yang praktikal bagi panduan agensi sektor awam dalam proses pembangunan sistem aplikasi berdasarkan dasar, polisi dan garis panduan yang dikeluarkan oleh Kerajaan, serta piawaian antarabangsa yang berkaitan dan pengetahuan tacit *Subject Matter Expert* (SME) Pembangunan Sistem dalam Sektor Awam;
- b) menjamin kualiti sistem aplikasi yang dibangunkan dan memastikan kesinambungan kerja dalam proses pembangunan, penyelenggaraan dan peningkatan sistem aplikasi;
- c) memastikan konsistensi dalam amalan dan aktiviti pembangunan sistem aplikasi di seluruh agensi sektor awam;
- d) meningkatkan efisiensi pelaksanaan projek pembangunan sistem aplikasi melalui pendekatan yang lebih berstruktur, tersusun dan komprehensif; dan
- e) meningkatkan kesedaran pemegang taruh tentang kepentingan dokumentasi dan serahan sistem berkaitan yang perlu dihasilkan sepanjang pelaksanaan aktiviti-aktiviti pembangunan sistem aplikasi.

1.1.3 SKOP KRISA

Skop KRISA adalah seperti berikut:

- a) Terpakai bagi semua projek pembangunan sistem aplikasi di dalam sektor awam.
- b) Merangkumi kaedah pembangunan secara dalaman, pembangunan bersama (co-development), dan pembangunan secara penyumberluaran (outsourced).
- c) Serasi dengan semua metodologi pembangunan sistem aplikasi yang berasaskan kepada Kitar Hayat Pembangunan Sistem (SDLC).
- d) Tidak terikat kepada mana-mana teknologi, platform, produk dan perkhidmatan yang tertentu.
- e) Tidak bergantung kepada jenis aplikasi yang dibangunkan, sama ada aplikasi web, mudah alih (*mobile*), peranti boleh pakai (*wearables*), dan lain-lain.
- f) Boleh diguna pakai dalam pelbagai skala dan saiz projek, merentas domain serta sektor perkhidmatan awam yang pelbagai.

1.2 PENGENALAN KEPADA KITAR HAYAT PEMBANGUNAN SISTEM APLIKASI

1.2.1 DEFINISI SDLC

Kitar Hayat Pembangunan Sistem Aplikasi atau *Software Development Life Cycle (SDLC)* adalah merujuk kepada aktiviti-aktiviti yang dilaksanakan pada setiap peringkat pembangunan sistem. SDLC merupakan pendekatan pembangunan sistem secara berstruktur yang mentakrifkan fasa-fasa utama pembangunan sistem secara sistematik. Merujuk kepada *ISTQB Certification*¹, model kitar hayat pembangunan sistem aplikasi (SDLC) adalah terdiri 6 fasa utama iaitu fasa pengumpulan keperluan dan analisis, fasa reka bentuk, fasa pembangunan (*coding*), fasa pengujian, fasa pelaksanaan (*deployment*) dan fasa penyelenggaraan. Fasa-Fasa Kitar Hayat Pembangunan Sistem Aplikasi (SDLC) adalah seperti Rajah 1.1 di bawah.



Rajah 1.1: Fasa-Fasa Kitar Hayat Pembangunan Sistem Aplikasi (SDLC)

Pendekatan ini bertujuan untuk memastikan proses pembangunan sistem yang teratur, terkawal dan efisien, bagi mencapai keperluan pengguna yang ditetapkan, serta mengoptimumkan penggunaan sumber dari segi kos dan masa.

SDLC menekankan pendekatan berfasa yang jelas dengan proses dokumentasi yang teratur, mekanisme kawalan kualiti bersepadu, dan penekanan terhadap penyelarasan keperluan bisnes dan sistem. Antara kelebihan utama pelaksanaan SDLC termasuk pengurusan risiko yang lebih baik, penjimatan kos dalam jangka panjang, pematuhan kepada standard kualiti industri, serta kemudahan dalam proses penyelenggaraan sistem. Ciri-ciri ini menjadikan SDLC sebagai kerangka kerja penting dalam pembangunan sistem yang berkualiti tinggi.

¹ <http://istqbexamcertification.com/what-are-the-software-development-life-cycle-sdlc-phases/>

1.2.2 PENDEKATAN TRADISIONAL DAN MODEN DALAM SDLC

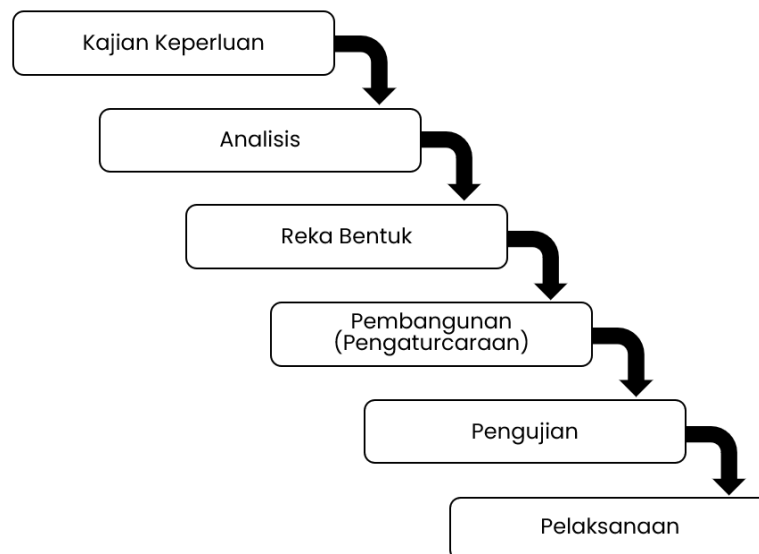
Dalam pelaksanaan SDLC, terdapat dua (2) pendekatan utama yang digunakan secara meluas, iaitu pendekatan SDLC tradisional dan moden. Setiap pendekatan mempunyai ciri-ciri, kelebihan, dan kekuatan tersendiri bergantung kepada objektif, skop dan strategi pelaksanaan sistem yang ingin dibangunkan.

a) SDLC Tradisional

Pendekatan SDLC tradisional adalah merujuk kepada kaedah pembangunan sistem yang bersifat prediktif, berturutan (sequential), dan mempunyai struktur yang tetap. Keseluruhan proses pembangunan dirancang dengan teliti pada peringkat awal, dan setiap fasa dilaksanakan mengikut perancangan yang telah ditetapkan. Pendekatan ini menekankan kestabilan keperluan yang diperoleh dan sebarang perubahan selepas fasa kajian keperluan memerlukan penilaian terperinci bagi memastikan keselarasan dengan objektif, skop, kos dan tempoh masa projek. Pendekatan ini paling sesuai untuk projek pembangunan sistem seperti berikut :

- i) mempunyai keperluan bisnes dan sistem yang jelas dan stabil
- ii) boleh mematuhi kepada kerangka tadbir urus projek pembangunan sistem yang jelas dan terperinci.

Pendekatan SDLC tradisional adalah seperti Rajah 1.2 di bawah.



Rajah 1.2: SDLC Tradisional

b) SDLC Moden

Pendekatan SDLC moden pula adalah berlandaskan kepada prinsip pembangunan sistem yang bersifat iteratif, penambahan berperingkat (*incremental*), dan adaptif. Ia menggalakkan pembangunan secara fleksibel/dinamik dengan maklum balas berterusan daripada pengguna bagi memastikan penambahbaikan dapat dilakukan sepanjang kitaran hayat projek. Pendekatan ini membenarkan penyelarasan keperluan bisnes dan sistem

secara lebih dinamik. Pendekatan ini sesuai untuk projek pembangunan sistem seperti berikut:

- i) dijangka akan mengalami perubahan keperluan bisnes dan sistem yang kerap
- ii) mengutamakan fleksibiliti dalam pembangunan dan pelaksanaan.

Contoh pendekatan SDLC moden adalah seperti Rajah 1.3 di bawah.



Rajah 1.3: SDLC Moden

Rujukan

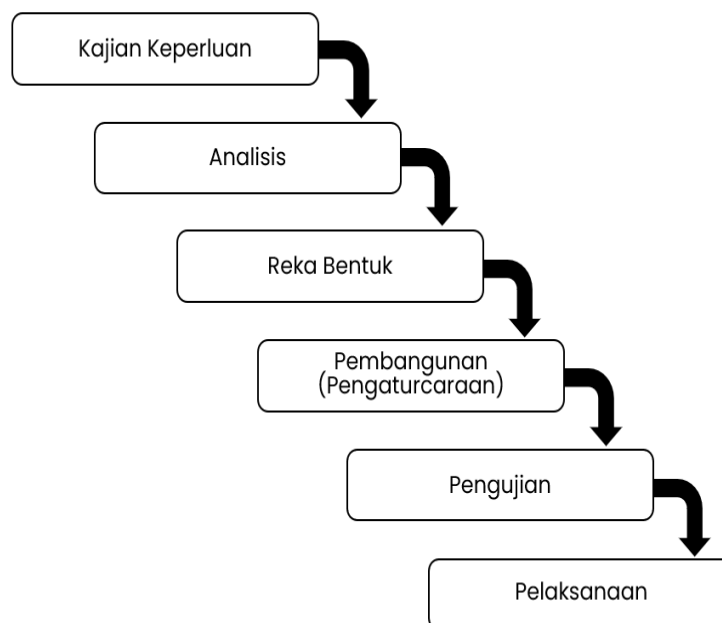
1. IEEE Std 12207: Systems and Software Engineering — Software Life Cycle Processes
2. ISO/IEC/IEEE 24765:2017: Systems and software engineering — Vocabulary
3. ISTQB Glossary – <https://glossary.istqb.org>
4. Sommerville, I. (2016). Software Engineering (10th Edition). Pearson Education
5. IEEE Std 1074-1997: Standard for Developing a Software Project Life Cycle Process

1.3 METODOLOGI PEMBANGUNAN SISTEM APLIKASI

Metodologi Pembangunan Sistem adalah satu pendekatan yang digunakan untuk menstruktur, merancang dan mengawal selia proses dan aktiviti dalam satu-satu projek pembangunan sistem. Setiap metodologi pembangunan sistem aplikasi direka bentuk untuk mematuhi prinsip asas Kitar Hayat Pembangunan Sistem Aplikasi (SDLC), dengan memastikan setiap fasa pembangunan dilaksanakan secara teratur dan terkawal. Terdapat pelbagai metodologi pembangunan sistem yang telah diperkenalkan. Pemilihan metodologi adalah bergantung kepada kesesuaian, keupayaan, keperluan projek dan organisasi. Setiap metodologi mempunyai persamaan dan perbezaan berpanduan kepada kerangka kerja dan prinsip yang tersendiri.

1.3.1 JENIS-JENIS METODOLOGI PEMBANGUNAN SISTEM

a) Metodologi *Waterfall*



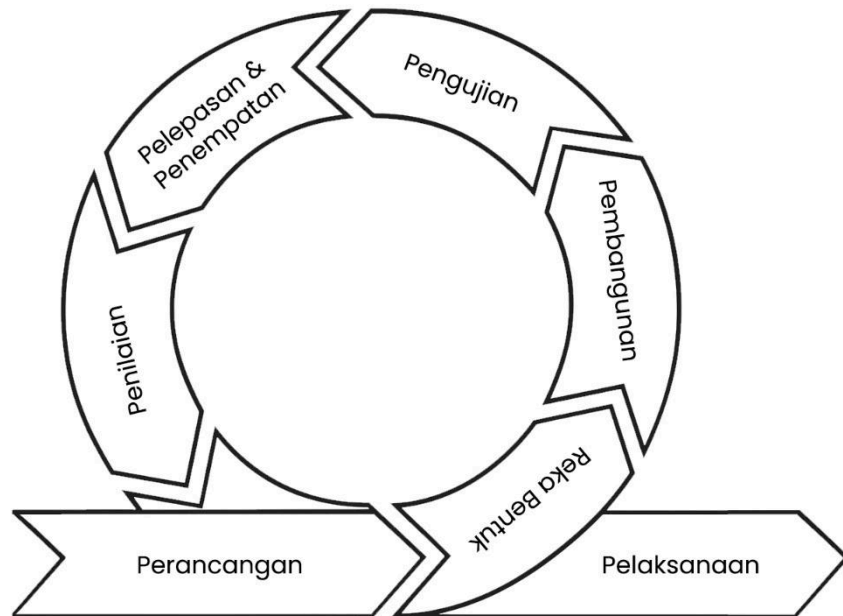
Rajah 1.4: Metodologi *Waterfall*

Metodologi *Waterfall* seperti rajah 1.4 di atas, atau dikenali juga sebagai Metodologi Jujukan Linear merupakan pendekatan pembangunan sistem yang bersifat prediktif dan berturutan. Setiap fasa dalam kitar hayat pembangunan seperti kajian keperluan, analisis, reka bentuk, pembangunan, pengujian, dan pelaksanaan dilaksanakan secara tersusun, dengan semakan dan pengesahan serahan dilakukan secara lebih terkawal di penghujung setiap fasa. Dokumentasi disediakan pada setiap peringkat bagi memastikan ketelusan serta memudahkan pemantauan kemajuan projek dari semasa ke semasa.

Selain itu, metodologi *waterfall* juga memberi penekanan kepada perancangan dan penetapan keperluan secara menyeluruh sebelum pembangunan dimulakan.

Pendekatan ini sesuai digunakan dalam situasi yang memerlukan dokumentasi lengkap serta kawalan proses yang formal dan teratur. Walau bagaimanapun, keberkesanan metodologi ini bergantung kepada kesesuaian dengan konteks projek dan organisasi, khususnya apabila keperluan adalah jelas dan stabil sepanjang tempoh pembangunan sistem.

i. Metodologi Agile



Rajah 1.5: Metodologi Agile

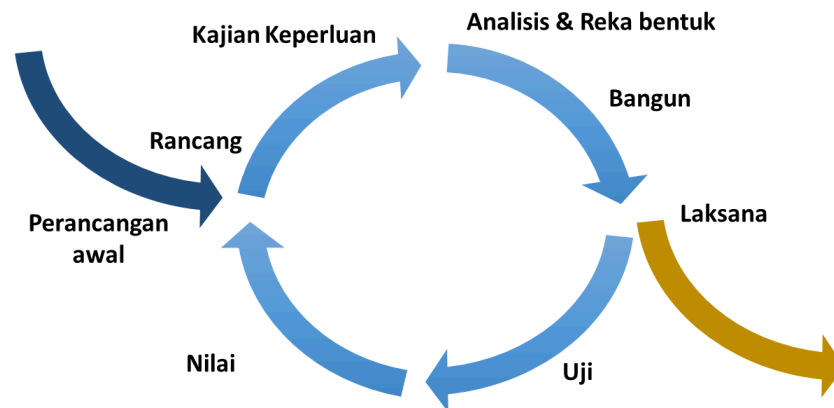
Metodologi *agile* seperti rajah 1.5 di atas merupakan pendekatan pembangunan sistem secara iteratif dan *incremental* yang memberi tumpuan kepada pelaksanaan sistem yang berfungsi secara berperingkat, integrasi berterusan antara komponen, serta pengesahan kualiti melalui pembangunan dan pengujian secara berterusan. Ia membantu mengurus pembangunan secara lebih fleksibel dengan memfokuskan kepada keperluan semasa, tanpa bergantung sepenuhnya kepada rekabentuk sistem berskala besar pada peringkat awal.

Selain itu, metodologi *agile* menekankan kepada penglibatan aktif di kalangan pemegang taruh bagi mendapatkan maklum balas secara berkala di sepanjang proses dan aktiviti pembangunan sistem. Dengan menyediakan versi sistem yang boleh diuji dengan lebih awal, sebarang perubahan atau penyesuaian dapat dikenalpasti dengan segera dan ditangani secara berperingkat. Walau bagaimanapun, keberkesanan metodologi ini bergantung kepada kesesuaian dengan konteks projek dan organisasi, khususnya apabila keperluan sistem dijangka berubah dan memerlukan penyesuaian secara berterusan sepanjang tempoh pembangunan sistem.

ii. Metodologi Rapid Application Development (RAD)

Metodologi *RAD* seperti rajah 1.6 di bawah merangkumkan fasa pembangunan sistem kepada empat (4) fasa utama iaitu Analisis dan Reka bentuk,

Pembangunan secara prototaip sehinggalah prototaip menjadi sistem aplikasi yang lengkap mengikut kehendak pengguna.



Rajah 1.8: Metodologi *Incremental Development*

1.3.2 FAKTOR-FAKTOR PENILAIAN PEMILIHAN METODOLOGI

Pemilihan metodologi yang sesuai adalah bergantung kepada keperluan bagi satu-satu projek, keupayaan pasukan pembangun sistem, dan penetapan sasaran serta matlamat organisasi. Penilaian yang teliti perlu dititikberatkan bagi memastikan metodologi yang dipilih adalah benar-benar sesuai dengan situasi dan keperluan projek pembangunan sistem yang ingin dilaksanakan. Antara faktor-faktor penilaian pemilihan metodologi adalah seperti berikut:

a) Menilai Keperluan Projek dan Dinamik Pasukan Pembangun Sistem

Keperluan projek dan dinamik pasukan perlu dinilai secara objektif bagi menentukan pendekatan yang paling sesuai diguna pakai, sama ada berbentuk fleksibel ataupun berstruktur. Projek yang mempunyai keperluan bisnes yang telah ditentukan secara jelas pada peringkat awal adalah lebih sesuai untuk menggunakan metodologi *waterfall*. Sementara itu, projek yang melibatkan perubahan keperluan bisnes secara berterusan adalah lebih cenderung untuk menerima guna metodologi yang lebih adaptif seperti *agile*. Pemilihan metodologi juga hendaklah mengambil kira kepada tahap pengetahuan, pengalaman dan ketersediaan pasukan projek pembangun sistem dalam mengaplikasikan metodologi tersebut dengan berkesan.

b) Mengambil Kira Jangka Masa, Kewangan dan Skop Projek

Faktor jangka masa, kewangan dan skop projek perlu diperhalusi secara holistik oleh kerana ketiga-tiga elemen ini akan memberi kesan secara langsung terhadap pemilihan metodologi yang sesuai. Metodologi *waterfall* adalah lebih serasi bagi projek yang mempunyai skop dan peruntukan kewangan yang telah ditetapkan secara terperinci pada peringkat awal dan setiap fasa pembangunan sistem bagi metodologi ini disusun secara sistematik. Sebaliknya, metodologi *agile* menyediakan kaedah yang lebih fleksibel dalam menguruskan elemen jangka masa, kewangan dan skop projek. Pendekatan ini membolehkan aplikasi dapat dibina terlebih dahulu secara *minimum viable product* (MVP) bagi mencapai tempoh *time-to-market* yang lebih singkat berbanding dengan metodologi lain.

c) Menyesuaikan dengan Matlamat Organisasi dan Keperluan Bisnes

Pemilihan metodologi hendaklah diselaraskan dengan matlamat organisasi dan keperluan bisnes. Projek-projek yang memerlukan struktur pelaksanaan yang lebih terkawal serta bergantung kepada pendekatan berfasa yang tersusun dan terancang adalah lebih sesuai untuk mempraktikkan metodologi *waterfall*. Manakala, organisasi yang menekankan pembangunan dan penambahbaikan sistem secara iteratif melalui penyelidikan dan maklum balas pelanggan adalah lebih berpatutan untuk mengguna pakai metodologi *agile*.

1.3.3 GABUNGAN METODOLOGI (HIBRID)

Pendekatan hibrid dalam pembangunan sistem adalah pendekatan yang semakin diiktiraf sebagai kaedah yang lebih seimbang dan praktikal, khususnya bagi projek-projek semasa yang kian kompleks dan dinamik. Melalui gabungan elemen-elemen daripada metodologi *waterfall* dan *agile*, organisasi dapat memanfaatkan kelebihan pada kedua-dua metodologi ini dalam satu-satu projek pembangunan sistem. Ciri-ciri utama pendekatan hibrid adalah seperti berikut:

- a) Keseimbangan di antara fleksibiliti dan struktur;
- b) Pendekatan berfasa dan iteratif;
- c) Pengurusan kawalan perubahan yang lebih terancang;
- d) Strategi mitigasi risiko yang fleksibel dan terkawal; dan
- e) Pengurusan sumber yang lebih cekap

Sebagai contoh, metodologi *waterfall* boleh diubahsuai dan diterapkan secara modular (*modular waterfall*) dengan mengelompokkan modul atau submodul sistem kepada kumpulan-kumpulan yang berbeza. Setiap kumpulan modul ini boleh dibangun dan dilaksanakan mengikut fasa-fasa yang berasingan bagi membolehkan aktiviti-aktiviti pembangunan dan pelaksanaan sistem dilakukan secara berperingkat tanpa kebergantungan tinggi di antara satu sama lain. Pendekatan ini membolehkan aktiviti-aktiviti pembangunan sistem dapat dilaksanakan dengan lebih fleksibel, sambil mengekalkan ciri-ciri metodologi *waterfall* yang berstruktur, sistematik dan terkawal.

Sementara itu, *Scaled Agile Framework* (SAFe) merupakan rangka kerja *agile* yang dipinda untuk mengurus pembangunan sistem secara lebih teratur dan selaras. SAFe menggabungkan prinsip-prinsip asas *agile* seperti pembangunan secara iteratif serta mengekalkan kerjasama erat di antara sesama pasukan. Rangka kerja ini memfokuskan kepada penerapan elemen-elemen perancangan strategik, kawalan berstruktur dan tadbir urus (*governance*) yang sinonim dan mirip dengan metodologi *waterfall*. Pendekatan ini membolehkan pembangunan sistem mampu untuk dilaksanakan secara terancang dan selaras tanpa mengabaikan ciri-ciri metodologi *agile* yang bersifat fleksibel dan responsif.

1.4 EKOSISTEM KEJURUTERAAN SISTEM APLIKASI SEKTOR AWAM



Rajah 1.9: Ekosistem Kejuruteraan Sistem Aplikasi Sektor Awam

Ekosistem Kejuruteraan Sistem Aplikasi Sektor Awam seperti rajah 1.9 di atas dibangunkan secara berstruktur bermula daripada rangka kawal selia tertinggi sehingga kepada panduan teknikal khusus. Ekosistem ini merangkumi **Akta dan Dasar Pendigitalan Sektor Awam** sebagai tonggak utama, diikuti dengan **Rancangan Pendigitalan Sektor Awam** dan **Pelan Strategik Organisasi** yang memacu hala tuju pelaksanaan inisiatif pendigitalan sektor awam. Di samping itu, **Garis Panduan Pengurusan Projek ICT (PPrISA)** menjadi panduan yang sistematik dalam pengurusan projek ICT khususnya bidang pembangunan sistem di agensi sektor awam. Pada peringkat pelaksanaan teknikal, panduan kejuruteraan sistem sektor awam seperti **Panduan Kejuruteraan Sistem Aplikasi Sektor Awam (KRISA)**, **Panduan Pelaksanaan DevOps dalam Pembangunan Sistem Aplikasi Sektor Awam**, **Garis Panduan Pembangunan Aplikasi Mudah Alih Sektor Awam (GP PAMASA)** dan **Buku Panduan Pelaksanaan *Independent Verification and Validation* (IV&V Handbook)** menyediakan kaedah dan amalan standard dalam aktiviti pembangunan sistem aplikasi. Keseluruhan ekosistem ini disediakan supaya pembangunan sistem aplikasi sektor awam dilaksanakan secara sistematik, berstruktur dan selaras dengan dasar, perancangan, piawaian, pelaksanaan serta kawal selia yang telah ditetapkan oleh Kerajaan.

1.5 METODOLOGI KRISA

Metodologi KRISA adalah berasaskan kepada tinjauan, penglibatan dan pengalaman pasukan perunding terhadap kaedah yang lazim diamalkan dalam projek pembangunan sistem sama ada di sektor awam mahupun swasta. Senarai ini merangkumi enam (6) fasa utama mengikut kerangka SDLC, iaitu:

- a) Pemulaan;
- b) Analisis;
- c) Reka Bentuk;
- d) Pembangunan (Pengaturcaraan);
- e) Pengujian; dan
- f) Pelaksanaan.

Rajah 1.10 di bawah menunjukkan Metodologi KRISA yang dikelompokkan mengikut fasa-fasa SDLC.

METODOLOGI KEJURUTERAAN PEMBANGUNAN SISTEM SEKTOR AWAM					
PERMULAAN	ANALISIS	REKA BENTUK	PEMBANGUNAN	PENGUJIAN	PELAKSANAAN
1.1 Penyediaan Perancangan Pembangunan Sistem 1.2 Penentuan Keperluan Bisnes 1.3 Pemodelan Fungsi Bisnes 1.4 Pemodelan Proses Bisnes 1.5 Penyediaan Spesifikasi Keperluan Bisnes	2.1 Pemodelan Use Case (Fungsian) 2.2 Pemodelan Fungsi Sistem 2.3 Pemodelan Keperluan Data 2.4 Pemodelan Proses Sistem 2.5 Penentuan Keperluan Bukan Fungsian 2.6 Penyediaan Spesifikasi Keperluan Sistem	3.1 Reka Bentuk Arkitektur 3.2 Reka Bentuk Teknologi 3.3 Reka Bentuk pangkalan Data 3.4 Reka Bentuk Antaramuka Pengguna 3.5 Reka Bentuk Transaksi Sistem 3.6 Penyediaan Spesifikasi Reka bentuk Sistem 3.7 Pelan Pelan Migrasi Data 3.8 Reka Bentuk Migrasi Data 3.9 Penyediaan Pelan Integrasi Sistem 3.10 Ujian Penerimaan Akhir (FAT)	4.1 Pembangunan Pangkalan Data 4.2 Pengaturcara Aplikasi 4.3 Pengujian Sistem	5.1 Penyediaan Pelan Induk Pengujian 5.2 Penyediaan Dokumentasi Persediaan Ujian 5.3 Penyediaan Pelan Ujian Penerimaan 5.4 Ujian Penerimaan Pengguna (UAT) 5.5 Ujian Penerimaan Provisional (PAT) 5.6 Penyediaan Laporan Ujian Penerimaan	6.1 Pelaksanaan Migrasi Data 6.2 Ujian Penerimaan Akhir (FAT) 6.3 Penyediaan Manual Pengguna 6.4 Serahan Sistem Aplikasi

Rajah 1.10: Senarai Model dan Pendekatan KRISA Mengikut Fasa SDLC

1.5.1 FASA PERMULAAN

Fasa Permulaan memberikan penekanan kepada komunikasi dan komitmen pemegang taruh dan organisasi yang bertanggungjawab membangunkan sistem selepas satu-satu permintaan diterima. Fasa ini dibahagikan kepada 2 aktiviti utama, iaitu Perancangan pembangunan Sistem dan Kajian Keperluan Bisnes.

1.5.2 FASA ANALISIS

Fasa Analisis bertujuan untuk menyediakan Spesifikasi Keperluan Sistem (SRS) bagi sistem yang akan dibangunkan supaya memenuhi kehendak bisnes dan jangkaan pemegang taruh. Proses-proses yang dijalankan dalam fasa ini merangkumi Pemodelan Fungsi Sistem, Pemodelan Use Case, Pemodelan Keperluan Data, Pemodelan Proses Sistem, Penentuan Keperluan Bukan Fungsian, serta penyediaan Pelan Migrasi Data dan Integrasi Sistem. Melalui proses-proses ini, pasukan projek dapat mengenal pasti fungsi utama sistem, perancangan migrasi data dan integrasi sistem, serta aspek-aspek bukan fungsian seperti keselamatan, prestasi dan kebolehsenggaraan sistem secara lebih jelas, tersusun dan menyeluruh.

1.5.3 FASA REKA BENTUK

Berdasarkan kepada keperluan sistem yang diperolehi dalam Fasa Analisis, reka bentuk keseluruhan sistem yang merangkumi reka bentuk antara muka, aplikasi dan pangkalan data akan dihasilkan. Fasa ini juga akan beralih dari 'apa' yang sistem perlukan kepada penerangan 'bagaimana' sistem ini akan dihasilkan. Fasa ini merangkumi model/metode seperti Reka Bentuk Arkitektur, Reka Bentuk Antaramuka Pengguna, Reka Bentuk Pangkalan Data, Reka Bentuk Transaksi Sistem, Reka Bentuk Migrasi dan Reka Bentuk Integrasi Data.

1.5.4 FASA PEMBANGUNAN

Di dalam Fasa Pembangunan, proses penghasilan sistem aplikasi adalah merupakan matlamat utama. Aktiviti-aktiviti yang dilaksanakan dalam Fasa Pembangunan merangkumi Pembangunan Pangkalan Data, Pengaturcaraan Aplikasi dan Pengujian Sistem oleh pasukan pembangun. Fasa Pembangunan merealisasikan elemen-elemen yang telah disediakan dan direkodkan pada Fasa Reka Bentuk. Komponen dan fungsi sistem dibangunkan melalui proses pengaturcaraan dan kemudiannya diintegrasikan di antara satu sama lain serta diuji secara dalaman pasukan pembangun bertujuan menghasilkan sesuatu sistem yang lengkap, mampan dan sedia untuk diteruskan ke fasa seterusnya.

1.5.5 FASA PENGUJIAN

Aktiviti-aktiviti yang dilaksanakan di dalam Fasa Pengujian adalah berkaitan dengan Penyediaan Pelan Pengujian, Penyediaan Dokumentasi Persediaan Ujian, pelaksanaan pengujian penerimaan oleh pengguna ke atas sistem, serta penyediaan Laporan Ujian Penerimaan. Pengujian yang dimaksudkan adalah Ujian Penerimaan Pengguna (UAT) dan Ujian Penerimaan Provisional (PAT). Ujian ini dilaksanakan sebagai validasi ke atas sistem aplikasi yang dibangunkan berdasarkan kepada keperluan bisnes dan sistem bagi memastikan keperluan-keperluan tersebut dipenuhi sebelum sistem aplikasi dilaksanakan. Di samping itu, kaedah Pengujian Awal dan Penyediaan Traceability Matrix dilaksanakan bagi memastikan setiap keperluan sistem telah diuji dan dapat dijejak semula kepada hasil pengujian yang berkaitan.

1.5.6 FASA PELAKSANAAN

Aktiviti-aktiviti dalam Fasa Pelaksanaan merangkumi aktiviti-aktiviti berkaitan persediaan akhir sebelum sistem dilaksanakan sepenuhnya. Aktiviti utama dalam fasa ini termasuk Pelaksanaan Migrasi Data, Ujian Penerimaan Akhir, Penyediaan Manual Pengguna, Penyediaan Persekitaran Pembangunan Sistem, dan Serahan Sistem Aplikasi kepada pihak pemilik sistem.

Semua pelaksanaan aktiviti yang dinyatakan pada setiap fasa adalah tertakluk kepada skop projek pembangunan sistem aplikasi yang akan dilaksanakan. Setiap fasa boleh disesuaikan mengikut tahap kompleksiti, sumber, serta objektif organisasi bagi memastikan pembangunan sistem dilaksanakan secara efisien, berstruktur dan menepati matlamat projek.

1.6 FAKTOR KEJAYAAN DALAM PEMBANGUNAN SISTEM APLIKASI

Ekosistem Kejuruteraan Sistem Aplikasi Sektor Awam telah menggariskan beberapa faktor kejayaan dalam pembangunan sistem iaitu Tadbir Urus Pengurusan Projek ICT, Pengurusan Kawalan Pindaan, Jaminan Kualiti Perisian (SQA), Faktor Keselamatan ICT, Pengukuran Saiz Sistem Aplikasi dan Penglibatan Pemegang Taruh. Aspek-aspek utama ini perlu dititik berat dalam melaksana metodologi kitar hayat pembangunan sistem ke arah mencapai matlamat bisnes yang ditetapkan.

1.6.1 TADBIR URUS PENGURUSAN PROJEK ICT YANG JELAS

Tadbir Urus dan Struktur Organisasi yang jelas dan bersesuaian adalah penting bagi kejayaan projek pembangunan sistem aplikasi. Tadbir urus meliputi penubuhan struktur organisasi projek dan jawatankuasa pemantauan projek yang bersesuaian dengan jenis, saiz, kompleksiti, bilangan sumber tenaga manusia di agensi. Sila rujuk Garis Panduan Pengurusan Projek ICT Sektor Awam (PPrISA) 2.0 berhubung tadbir urus dan struktur organisasi projek ICT.

1.6.2 PEMILIHAN STRATEGI PEMBANGUNAN SISTEM APLIKASI YANG BERSESUAIAN

Usaha pembangunan sistem aplikasi boleh dilaksanakan melalui pelbagai cara bergantung kepada kemampuan kewangan, sumber sedia ada serta tahap kompleksiti projek. Terdapat tiga (3) jenis usaha pembangunan sistem yang lazim dipraktikkan, iaitu pembangunan secara dalaman (*in-house*), pembangunan secara kerjasama (*co-development*), dan pembangunan sepenuhnya oleh pihak luar (*outsourcing*).

a) Pembangunan Sistem Secara *In-House*

Jenis usaha ini adalah merujuk kepada pembangunan sistem yang dilaksanakan sepenuhnya oleh pasukan dalaman organisasi. Kaedah ini membolehkan organisasi memiliki kawalan penuh terhadap keseluruhan proses pembangunan sistem memandangkan tidak terikat secara perundangan serta kos pembangunan sistem yang lebih rendah. Pasukan dalaman lebih mudah memahami keperluan bisnes berbanding dengan pihak luar dan boleh membuat perubahan skop atau menambah ciri baharu dengan lebih mudah seterusnya membolehkan organisasi kekal relevan dalam pasaran yang dinamik. Namun begitu, pembangunan secara *in-house* ini

lazimnya memerlukan masa yang lebih lama, mempunyai tahap fleksibiliti yang terhad (bergantung kepada kompetensi dan komitmen masa yang terhad) serta berisiko tinggi sekiranya sumber dalaman yang tersedia adalah tidak mencukupi dari segi kekuatan dan kepakaran serta tidak mencapai tahap kualiti yang dikehendaki.

b) Pembangunan Sistem Secara *Co-Development*

Jenis usaha ini adalah merujuk kepada pembangunan sistem yang dilaksanakan secara kolaborasi di antara pasukan dalaman Kerajaan dengan pihak swasta. Kaedah ini bukan sahaja mampu menggabungkan keupayaan dan kepakaran kedua-dua pihak, malah turut mengurangkan tahap kebergantungan yang tinggi kepada pihak swasta. Pada masa yang sama, kaedah ini turut memberi peluang kepada pasukan Kerajaan untuk mempertingkatkan pengetahuan dan kemahiran melalui pemindahan ilmu serta pendedahan kepada amalan dan teknologi semasa yang diperkenalkan oleh pihak syarikat. Walau bagaimanapun, pembangunan melalui usaha kerjasama ini lazimnya memerlukan koordinasi yang lebih kompleks, dan berpotensi untuk menimbulkan konflik kepentingan (*conflict of interest*) di antara pihak Kerajaan dan syarikat sekiranya tadbir urus dan pengurusan projek tidak dikendalikan secara profesional dan sistematik.

c) Pembangunan Sistem Secara *Outsourcing*

Pembangunan sistem secara penyumberluaran (*outsourcing*) merujuk kepada kaedah pelaksanaan projek pembangunan sistem aplikasi yang dilaksanakan melalui pihak ketiga atau pembekal, di mana agensi kerajaan melantik penyedia perkhidmatan ICT yang berkecualan untuk membangunkan, melaksana dan menyerahkan sistem mengikut keperluan yang telah ditetapkan.

Walaupun bagaimanapun, usaha pembangunan secara penyumberluaran bukan sahaja melibatkan kos, malah ia juga berisiko mewujudkan ketergantungan yang tinggi dan berterusan dengan pihak syarikat, terutamanya dari segi penyelenggaraan, sokongan teknikal dan keselamatan sistem.

d) Faktor-Faktor Pemilihan Jenis Usaha Pembangunan Sistem

Pemilihan jenis usaha pembangunan sistem perlu dibuat dengan mengambil kira beberapa faktor utama yang mempengaruhi keberkesanan dan kejayaan sesuatu projek pembangunan sistem aplikasi. Antara faktor-faktor tersebut adalah seperti berikut:

i. Kompleksiti Projek

Merujuk kepada tahap kerumitan sistem yang ingin dibangunkan dari aspek skop, fungsi, teknologi, dan integrasi dengan sistem lain. Semakin kompleks sesuatu projek, semakin besar keperluan terhadap kepakaran dan sumber yang sesuai.

ii. Sumber

Melibatkan ketersediaan kos, tenaga kerja, dan infrastruktur yang dimiliki oleh organisasi. Faktor ini menentukan sama ada pembangunan boleh dilaksanakan sepenuhnya secara dalaman, melalui kerjasama, atau diserahkan kepada pihak luar.

iii. Jadual Masa

Tempoh masa yang ditetapkan untuk menyiapkan pembangunan sistem. Pemilihan usaha pembangunan perlu diselaraskan dengan keperluan organisasi sama ada memerlukan serahan dan pelaksanaan sistem dalam masa yang singkat, atau mempunyai kemudahan ruang masa yang lebih panjang.

iv. Risiko

Tahap risiko yang perlu diuruskan oleh organisasi, sama ada dari aspek kawalan, keselamatan data, kos, atau kebergantungan kepada pihak luar. Setiap jenis usaha pembangunan mempunyai tahap risiko yang berbeza dan perlu dipertimbang berdasarkan kepada kepentingan organisasi.

1.6.3 PENGLIBATAN PEMEGANG TARUH

Penglibatan pemegang taruh merujuk kepada proses melibatkan semua pihak yang berkepentingan atau terkesan dalam sesuatu projek pembangunan. Ia bertujuan memastikan projek dirancang, dilaksana dan dinilai dengan lebih berkesan melalui kerjasama, maklum balas serta sokongan daripada pihak berkaitan. Di samping menyumbang kepada kualiti sistem yang dibangunkan, penglibatan pemegang taruh juga dapat menyelesaikan strategi pengurusan perubahan ke arah penggunaan sistem aplikasi yang dibangunkan.

a) Faedah Yang Diperolehi

- i) Meningkatkan kualiti sistem dan ketetapan kepada kehendak dan keperluan pengguna dalam menyokong peranan dan bisnes agensi
- ii) Meningkatkan kefahaman dan penerimaan sistem aplikasi yang dibangunkan
- iii) Dengan penglibatan SME dalam mengenalpasti keperluan sistem dalam konteks fungsi organisasi, ini dapat mengurangkan masa pembangunan
- iv) Memupuk perasaan pemilikan sistem baru, dengan penglibatan pengguna dalam proses pembangunan akan menyebabkan penggunaan lebih komited untuk menggunakan sistem dan untuk memastikan kejayaannya semasa pelaksanaan.

Berikut adalah matriks penglibatan pemegang taruh berdasarkan fasa dan serahan yang berkaitan serta keterangan bagi setiap pemegang taruh.

Jadual 1.1. Penglibatan Pemegang Taruh Berdasarkan Fasa Pembangunan

	PERANCANGAN	PERMULAAN	ANALISIS	REKA BENTUK	PEMBANGUNAN	PENGUJIAN PENERIMAAN	PELAKSANAAN
PENGANJUR PROJEK	X						
SUBJECT MATTER EXPERT (SME)	X		X	X		X	
AHLI PASUKAN PROJEK	X						
JURU ANALISA SISTEM	X		X	X			
JURU ANALISA BISNES			X				
PEMILIK PROJEK	X		X				
PEMILIK SISTEM		X					
PENGGUNA SISTEM			X				
PENGURUS PROJEK					X		X
PENGATURCARA PROGRAM					X		
PENGUJI SISTEM					X		
PEMILIK PROSES/ PENGUNA					X		
PENGURUS UJIAN						X	
KETUA UJIAN						X	
PASUKAN PENGUJI						X	
PASUKAN PEMBANGUN SISTEM APLIKASI						X	
PENGUNA AKHIR						X	
PELANGGAN							X
ORGANISASI PELAKSANA							X
PEMBIAYA							X
PEMILIK PROSES						X	

Jadual 1.2. Keterangan Peranan Pemegang Taruh

PEMEGANG TARUH	KETERANGAN PERANAN
PENGANJUR PROJEK	Ketua jabatan atau agensi yang meluluskan projek serta menyediakan peruntukan kewangan dan sokongan strategik bagi pembangunan sistem aplikasi.
SUBJECT MATTER EXPERT (SME)	Memberi kepakaran domain, menyemak dan mengesahkan keperluan fungsi, data serta aliran proses agar selaras dengan operasi sebenar.
AHLI PASUKAN PROJEK	Wakil pengguna yang terlibat dalam mengenal pasti keperluan sistem, memberi input reka bentuk, serta terlibat dalam pengujian dan penilaian sistem.
JURU ANALISA SISTEM	Menjalankan analisis keperluan sistem, menyediakan model, spesifikasi teknikal dan dokumentasi untuk menyokong fasa reka bentuk dan pembangunan sistem.
JURU ANALISA BISNES	Menterjemahkan keperluan bisnes pengguna kepada keperluan sistem bagi memastikan penyelesaian yang dibangunkan sejajar dengan objektif bisnes.
PEMILIK PROJEK	Ketua bahagian yang bertanggungjawab terhadap hala tuju projek serta memastikan sistem yang dibangunkan memenuhi keperluan jabatan.
PEMILIK SISTEM	Bertanggungjawab menyemak, mengesahkan dan meluluskan keperluan fungsi dan bukan fungsi sistem agar selaras dengan keperluan organisasi.
PENGGUNA SISTEM	Merancang, memantau dan mengawal pelaksanaan projek termasuk skop, jadual, kos dan risiko serta menyelaraskan komunikasi antara pihak berkepentingan.
PENGURUS PROJEK	Membangunkan sistem aplikasi berdasarkan spesifikasi yang diluluskan termasuk penulisan kod dan integrasi sistem.
PENGATURCARA PROGRAM	Menjalankan ujian sistem bagi memastikan fungsi, prestasi dan keselamatan sistem memenuhi keperluan sebelum pelancaran.
PENGUJI SISTEM	Melaksanakan dan menyertai Ujian Penerimaan Pengguna (UAT) bagi memastikan sistem memenuhi keperluan operasi sebenar.
PEMILIK PROSES/ PENGGUNA	Merancang, menyelaraskan dan mengurus keseluruhan aktiviti pengujian termasuk penyediaan laporan keputusan ujian.
PENGURUS UJIAN	Mengendalikan dan memimpin pasukan pengujian serta memastikan aktiviti ujian dilaksanakan mengikut pelan yang ditetapkan.
KETUA UJIAN	Menyediakan kes ujian, melaksanakan ujian dan mendokumenkan keputusan serta isu yang dikenal pasti semasa pengujian.
PASUKAN PENGUJI	Membuat pembetulan, penambahbaikan dan sokongan teknikal berdasarkan laporan ujian yang diterima.
PASUKAN PEMBANGUN SISTEM APLIKASI	Menggunakan sistem dan memberi maklum balas terhadap kefungsiannya serta kebolehgunaan sistem.
PENGGUNA AKHIR	Individu atau organisasi yang menerima manfaat atau menggunakan produk sistem yang dibangunkan.
PELANGGAN	Pihak yang melaksanakan kerja-kerja pembangunan sistem termasuk perancangan, pembangunan, pengujian dan pelaksanaan.
ORGANISASI PELAKSANA	Individu atau organisasi yang menyediakan sumber kewangan atau sokongan dalam bentuk tunai atau barangan bagi projek.
PEMBAIAYA	Bertanggungjawab memastikan proses bisnes yang disokong oleh sistem dilaksanakan dengan betul serta menyokong aktiviti UAT.

1.6.4 JAMINAN KUALITI PERISIAN (SQA)

SQA adalah satu pendekatan yang terancang dan sistematik untuk menilai kualiti dan pematuhan kepada standard aplikasi, proses dan prosedur. Ini termasuklah proses bagi menjamin bahawa standard dan prosedur yang diwujudkan dipatuhi sepanjang aktiviti kitar hayat pembangunan sistem. SQA berbeza daripada ujian kerana ia adalah berorientasikan pencegahan manakala ujian adalah berorientasikan pengesanan. SQA adalah proses semakin bermula dari peringkat awal pembangunan, dengan ini dapat membantu dalam mengenal pasti masalah awal seterusnya menyumbang ke arah peningkatan kualiti sistem. Terdapat 2 item utama yang perlu dibincangkan dalam pelaksanaan SQA iaitu ciri-ciri sistem aplikasi yang berkualiti diukur berdasarkan atribut kualiti perisian dan aktiviti pengujian verifikasi dan validasi bagi memastikan sistem aplikasi yang dihasilkan berkualiti dan memenuhi ciri-ciri kualiti perisian.

a) Atribut Kualiti Perisian

Kualiti sistem aplikasi ditentukan oleh atribut kualiti seperti **Jadual 3**.

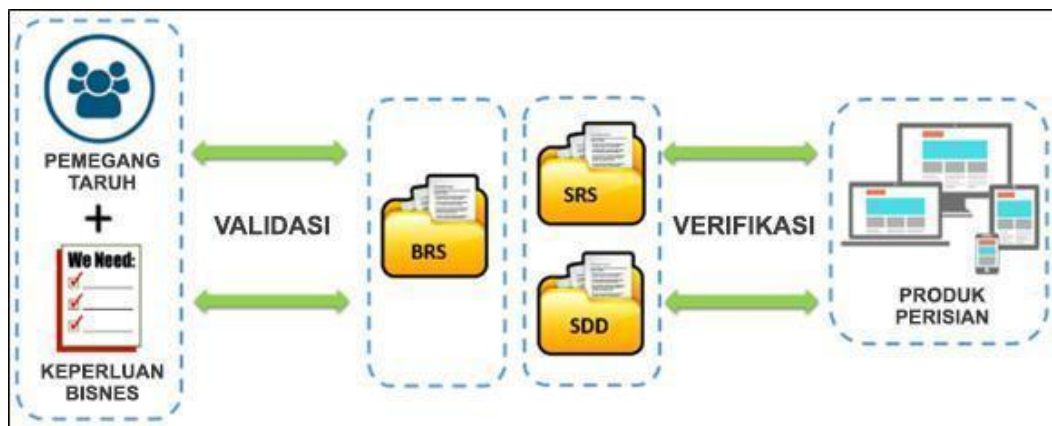
Jadual 1.3 : Atribut Kualiti Perisian

Atribut	Keterangan	Contoh
Atribut fungsian (Functional attributes)	Input dan output produk perisian	• Sistem dapat memenuhi keperluan pengguna dengan tepat (<i>correctness</i>) . • Sistem mengawal capaian modul mengikut peranan pengguna (<i>authentication</i>). • Sistem memaparkan maklumat sulit kepada pengguna tertentu sahaja (<i>integrity</i>).Aspek keselamatan telah diambil berat semasa pembangunan sistem (<i>security</i>)
Atribut operasi (Operational attributes)	Syarat operasi sesuatu produk perisian	• Sistem berupaya memberikan tindak balas (latency or response time) dalam masa yang ditetapkan • Sistem dapat menampung kapasiti (capacity) pengguna serentak yang ditetapkan • Sistem dapat beroperasi walaupun melebihi kapasiti pengguna yang ditetapkan.
Atribut kebolegunaan (Usability attributes)	Sejauh mana produk perisian boleh digunakan dan disesuaikan	• Sistem mudah digunakan (ease of use) oleh pengguna • Sistem mudah dipelajari (ease of learn) oleh pengguna

Atribut	Keterangan	Contoh
	dengan produk perisian	• Sistem mudah disesuaikan (customizable) mengikut keperluan operasi pengguna • Sistem sedia berkolaborasi dengan aplikasi lain (interoperability)
Atribut perniagaan (Business attributes)	Kos pembangunan, penggunaan serta perubahan produk perisian	• Kos pembangunan sistem • Kos penyelenggaraan system • Penjimatan kos apabila produk atau komponennya boleh diguna semula (reusability) • Penjimatan kos apabila produk boleh digunakan pada platform berbeza (portability)
Atribut struktur (Structural attributes)	Struktur dalaman produk perisian	• Sistem yang dibangunkan berupaya untuk diuji (testability) • Sistem mudah diadaptasi (adaptability) mengikut perubahan keperluan pengguna • Pembangunan sistem mengambil kira struktur k modularan (modularity)

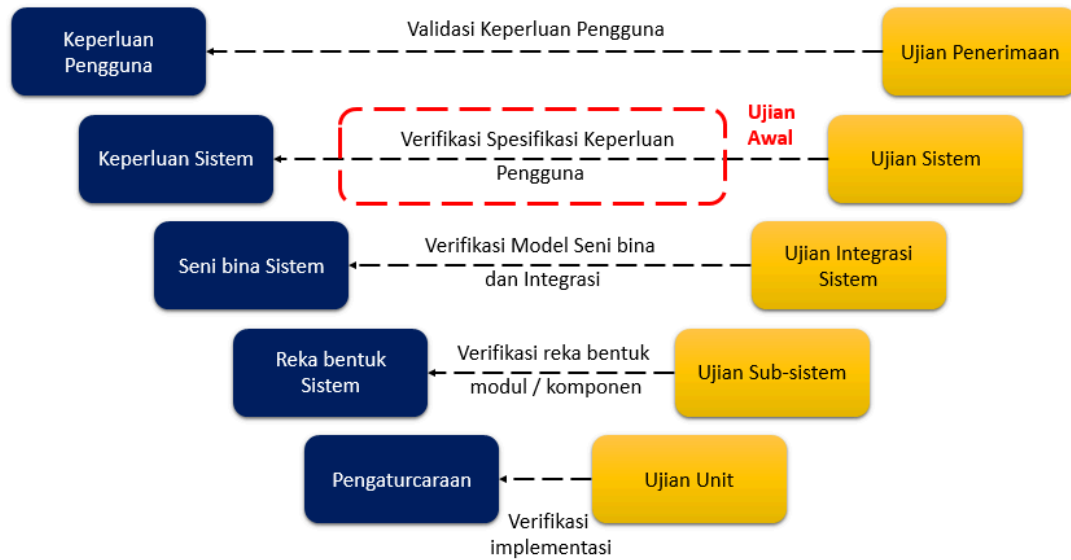
b) Verifikasi dan Validasi

Verifikasi adalah proses semakan dokumentasi dan rekabentuk sistem (*static testing*) untuk memastikan produk yang telah dibangunkan mematuhi keperluan yang ditetapkan (*system was built right*) dalam setiap fasa pembangunan sistem. Antara tujuan verifikasi adalah untuk menghasilkan keperluan yang betul, tepat, lengkap dan konsisten. Validasi adalah aktiviti bagi memastikan produk yang dihasilkan memenuhi spesifikasi keperluan (*right system built*) melalui proses pengujian dinamik. Rujuk rajah di bawah.



Rajah 1.11: : Proses Verifikasi dan Validasi

Pengujian Verifikasi dan Validasi (V&V) hendaklah dilaksanakan oleh pasukan pembangunan sistem itu sendiri, pada setiap fasa kitaran hayat pembangunan sistem. Rajah di bawah menunjukkan 4 jenis pengujian yang perlu dilaksanakan iaitu:



Rajah 1.12 : Jenis-Jenis Pengujian V&V

Jadual 1.4 : Jenis-Jenis Pengujian IV&V

Jenis Ujian	Keterangan
Pengujian Awal (Early Testing)	Jenis pengujian adalah Ujian Statik dalam fasa permulaan projek. Pendekatan ujian adalah secara <i>review</i> atau/dan <i>walkthrough</i> ke atas keperluan fungsian (<i>functional</i>) dan bukan fungsian (<i>non-functional</i>) seperti yang dinyatakan dalam SRS. Artifak yang terlibat adalah seperti: • Dokumen tender • Cadangan tender daripada pembekal yang dilantik. • <i>Software Requirement Specification (SRS)</i>.
Ujian Unit (Unit Testing)	Aktiviti pengujian bagi menilai unit terkecil (lowest level) di dalam sistem seperti kod program, <i>function</i> dan sebagainya
Ujian Integrasi (Integration Testing)	Aktiviti pengujian dengan sistem-sistem dalaman dan sistem-sistem luaran yang terlibat secara terus dengan aplikasi yang sedang dibangunkan.
Ujian Sistem (System Testing)	Aktiviti pengujian yang terlibat adalah Pengujian Functional dan Non-Functional untuk keseluruhan modul yang dibangunkan
Ujian Penerimaan (Acceptance Testing)	Ujian Penerimaan Pengguna terbahagi kepada tiga (3) seperti turutan berikut: • Ujian Penerimaan Pengguna (UAT) • Ujian Penerimaan Sementara (PAT) • Ujian Penerimaan Akhir (FAT)

Penggunaan SQA dalam proses pembangunan sistem akan membawa kepada faedah-faedah berikut:

- i) Memastikan standard dan prosedur dipatuhi.
- ii) Mewujudkan dokumentasi system yang seragam dan mudah difahami.
- iii) Memastikan masalah dapat ditemui awal dan diuruskan dengan sewajarnya.
- iv) Memantau dan menambah baik proses pembangunan perisian.

1.6.5 PENGURUSAN KAWALAN PINDAAN

Permintaan pindaan adalah permohonan untuk membuat pelarasan semula skop atau keperluan sistem yang telah dipersetujui. Permintaan perubahan ini amat penting diuruskan bagi memastikan tiada kesan kelewatan dalam pembangunan sistem aplikasi yang sedang berlaku. Permintaan pindaan keperluan biasanya berpunca dari sebab-sebab berikut:

- a) Masalah yang dilaporkan dikenalpasti sebagai ralat yang mesti diperbetulkan.
- b) Peningkatan sistem yang diminta dari pihak pengguna.
- c) Pertukaran standard, polisi atau akta berkaitan bisnes yang membawa kepada perlunya sistem diubahsuai.
- d) Permintaan daripada pengurusan kanan yang memerlukan pertambahan atau pengubahsuaian fungsi sistem.

Buku Metodologi PRrISA Panduan Pengurusan Projek ICT Sektor Awam, yang diterbitkan oleh JDN, menggariskan pendekatan bagi kawalan pindaan di bawah Fasa Pelaksanaan dan Kawalan, menjelaskan perkara berikut:

- a) Borang *Change Request* - borang standard yang perlu diisi oleh pihak pengguna bagi membuat permintaan pindaan dan sokongan daripada pemilik sistem
- b) Penyata Pindaan – keperluan untuk mendaftar permintaan pindaan, kelulusan tindakan, kaedah penyelesaian dan status tindakan
- c) Log Penyelesaian Isu – catatan kelulusan penyelesaian ke atas pindaan dan aktiviti yang telah dilaksanakan
- d) Lembaga kawalan perubahan – Lembaga /peringkat yang membincangkan dan meluluskan permintaan kawalan
- e) Aliran Proses Kawalan Pindaan – menjelaskan aliran kerja permintaan pindaan.

1.6.6 KEPENTINGAN PENGUKURAN SAIZ SISTEM

Pengukuran saiz sistem ialah proses menilai sejauh mana besar, kompleks atau luasnya sesuatu sistem perisian atau sistem maklumat. Ia penting untuk tujuan anggaran kos, perancangan masa, pengagihan sumber manusia, dan penilaian keupayaan sistem.

Umumnya, pengukuran saiz sistem bagi Projek Pembangunan Sistem, boleh menggunakan empat (4) kaedah seperti berikut:

a) **Parametrik**

Kaedah ini menggunakan parameter tertentu untuk menentukan saiz projek. Contoh Teknik yang boleh digunakan untuk mengira kos pembangunan sesuatu aplikasi ialah *Function Point Analysis* (FPA), *Lines of codes*, *Use case point*, *RICEF-W Constructive Cost Model* (COCOMO).

b) **Penilaian Pakar**

Anggaran saiz ditentukan berdasarkan pengetahuan dan pengalaman pakar industri.

c) **Anggaran Berasaskan Analogi**

Menggunakan data sejarah untuk membandingkan projek baharu dengan projek pembangunan serupa yang telah siap dibangunkan.

d) **Kaedah Bawah Ke Atas**

Anggaran saiz projek untuk berdasarkan setiap aktiviti utama .

1.6.7 FAKTOR KESELAMATAN ICT

Keperluan keselamatan dalam pembangunan sistem aplikasi perlu diambil kira dalam proses kitaran hayat pembangunan sistem. Risiko ancaman keselamatan harus diminimakan bagi melindungi data, operasi, atau fungsi sistem aplikasi bagi meningkatkan kepercayaan organisasi dan pengguna. Tindakan pencegahan perlu dilaksanakan dengan mematuhi arahan dan pekeliling keselamatan yang ditetapkan. Aspek yang perlu diambil kira dalam menjamin keselamatan sistem seperti berikut:

- a) Penilaian Risiko Keselamatan Maklumat
- b) Rekabentuk Selamat (Secured Design)
- c) Pembangunan Sistem Aplikasi Selamat (Secured Development)
- d) Pengujian Keselamatan (Security Testing)
- e) Keselamatan Pelaksanaan Sistem Aplikasi (Secured Deployment)
- f) Penyelenggaraan Keselamatan (Secured Maintenance)

Penerangan bagi setiap aspek keselamatan yang merangkumi tujuan, pendekatan dan rujukan pematuhan garis panduan / piawaian seperti di Jadual 1.

Jadual 1.5 :: Keperluan Keselamatan Dalam Pembangunan Sistem Aplikasi

ASPEK KESELAMATAN	KEPERLUAN	GARIS PANDUAN / PIAWAIAN
Penilaian Risiko Keselamatan Maklumat	i. Penilaian Risiko Keselamatan Maklumat mengambil kira kemungkinan ancaman dalam menghasilkan keperluan keselamatan. Mematuhi prinsip ini, organisasi dapat mewujudkan keselamatan yang kukuh, melindungi sistem daripada risiko, dan memastikan data serta pengguna sentiasa selamat. ii. Tindakan kawalan perlindungan yang dilaksanakan hendaklah bersesuaian dengan klasifikasi maklumat yang disimpan serta implikasi kepada kerajaan jika maklumat tersebut terdedah atau diubah oleh pihak yang tidak berkenaan. Bagi maklumat rahsia rasmi kerajaan, tindakan kawalan yang dilaksanakan hendaklah memenuhi keperluan kerajaan seperti Arahan Keselamatan (Semakan dan Pindaan 2017). iii. Prinsip keselamatan keperluan sistem yang boleh dilaksanakan seperti Penyulitan Data (<i>Encryption</i>), Kawalan Akses (<i>Access Control</i>), <i>Secure API Design</i>, <i>Logging</i> dan <i>Monitoring</i>.	• Arahan Keselamatan (Semakan dan Pindaan 2017) • ISO/IEC 27001 Information Security Management System (ISMS) • National Institute of Standards and Technology (NIST) Cybersecurity Framework • Open Worldwide Application Security Project (OWASP) Security Framework. • Kerangka STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege). • Surat Pekeliling Am Bilangan 2 Tahun 2021 Garis Panduan Pengurusan Keselamatan Maklumat Melalui Pengkomputeran Awan (<i>Cloud Computing</i>) Dalam Perkhidmatan Awam

ASPEK KESELAMATAN	KEPERLUAN	GARIS PANDUAN / PIAWAIAN
Rekabentuk Selamat (<i>Secured Design</i>)	i. Rekabentuk Selamat adalah pendekatan dalam pembangunan sistem atau aplikasi yang memastikan aspek keselamatan dimasukkan secara menyeluruh pada peringkat reka bentuk. Pendekatan ini penting untuk melindungi sistem daripada ancaman dan kelemahan yang mungkin timbul, sama ada semasa pembangunan atau setelah sistem digunakan. ii. Pendekatan Rekabentuk Selamat seperti Least Privilege (Keistimewaan Minimum) dan kawalan akses mengikut peranan pengguna, melaksanakan 'Secure by Design', Defense in Depth (Pertahanan Berlapis), Fail Securely (Kegagalan Selamat), Validate Inputs (Pengesahan Input) dan Secure Defaults (tetapan selamat secara ketat)	- National Institute of Standards and Technology (NIST) Cybersecurity Framework - Open Worldwide Application Security Project (OWASP) Security Framework
Pembangunan Sistem Aplikasi Selamat (<i>Secured Development</i>)	i. Pembangunan Sistem Aplikasi Selamat adalah amalan pengkodan selamat, menguji kod secara berkala untuk menangani kelemahan keselamatan dan penggunaan library / kod sumber yang selamat dan dikemaskini. ii. Pendekatan keselamatan bagi memastikan pembangunan sistem aplikasi selamat seperti menggunakan alat seperti SonarQube untuk mengimbas kelemahan dalam kod dan Pengurusan Tampalan (<i>Patch Management</i>) bagi mengelakkan eksploitasi kelemahan yang dikenalpasti.	- OWASP Secure Coding Practices. CWE (Common Weakness Enumeration). - OWASP ZAP
Pengujian Keselamatan (<i>Security Testing</i>)	i. Pengujian Keselamatan adalah proses menguji aplikasi, sistem, atau rangkaian untuk mengenal pasti kerentanan, kelemahan, atau ancaman keselamatan. Tujuannya adalah untuk memastikan bahawa sistem tersebut mampu melindungi data dan menjaga fungsinya walaupun berlaku serangan atau eksploitasi.	- ISO/IEC 27002. - Pelaksanaan ujian seperti Penetration Testing, Static Code Analysis, dan Dynamic Code Analysis, denial of service, security posture assessment (SPA)

ASPEK KESELAMATAN	KEPERLUAN	GARIS PANDUAN / PIAWAIAN
	ii. Antara contoh pendekatan pengujian yang boleh dilaksanakan seperti ujian penembusan, ujian kerentanan, ujian kawalan akses, ujian keutuhan data dan ujian ketahanan seperti <i>application denial of service</i> .	• Surat Pekeliling Am Bilangan 4 Tahun 2024 Garis Panduan Penilaian Tahap Keselamatan Rangkaian dan Sistem ICT Sektor Awam
Keselamatan Pelaksanaan Sistem Aplikasi (<i>Secured Deployment</i>)	i. Keselamatan Pelaksanaan Sistem Aplikasi merujuk kepada langkah-langkah keselamatan yang diambil semasa fasa pelaksanaan sistem aplikasi. Langkah ini bertujuan memastikan aplikasi yang telah dibangunkan berfungsi dengan selamat dalam persekitaran operasi dan tidak terdedah kepada ancaman keselamatan. Pendekatan ini memfokuskan kepada proses pelaksanaan, konfigurasi persekitaran, dan pemantauan sistem, memastikan aplikasi kekal selamat selepas ia dihoskan atau diakses oleh pengguna sebenar. ii. Komponen penting bagi pelaksanaan Keselamatan Pelaksanaan Sistem Aplikasi seperti menyediakan penyulitan data semasa penyimpanan, konfigurasi firewall untuk menyekat akses tidak sah, pemantauan sistem secara aktif, pengurusan tampalan secara berkala, audit keselamatan berkala.	• CIS Controls (Center for Internet Security) • AES-256 • NIST SP 800-53.
Penyelenggaraan Keselamatan (<i>Secured Maintenance</i>)	i. Penyelenggaraan Keselamatan atau Secured Maintenance merujuk kepada usaha berterusan untuk memastikan keselamatan, kebolehpercayaan, dan ketersediaan sistem aplikasi dan infrastruktur walaupun berlaku perubahan teknologi, serangan baru, atau ancaman yang tidak dijangka. ii. Komponen penting bagi penyelenggaraan keselamatan adalah pemantauan sistem berterusan, kemaskini dan tampalan, ujian keselamatan secara berkala, mengurus perubahan sistem dengan	• Akta Keselamatan Siber 2024 (Akta 854) • ITIL Security Management. • ISO/IEC 27005 (Risk Management). • Surat Pekeliling Am Bilangan 1 Tahun 2025 Garis Panduan Pengurusan Kesenambungan Perkhidmatan Dalam

ASPEK KESELAMATAN	KEPERLUAN	GARIS PANDUAN / PIAWAIAN
	memastikan sebarang penambahan atau pengubahsuaian tidak menjejaskan keselamatan dan respons insiden. iii. Memastikan keselamatan data dan business resilience melalui pelaksanaan dan pengujian sandaran data serta membangun, menguji dan menyelenggara pelan pemulihan bencana (DRP) secara berkala (sekurang-kurangnya sekali setahun)	Perkhidmatan Awam <i>Business Continuity Management</i> (BCM) • Surat Pekeliling Am Bilangan 3 Tahun 2024 Garis Panduan Pengurusan Risiko Keselamatan Maklumat Sektor Awam • Surat Pekeliling Am Bilangan 4 Tahun 2024 Garis Panduan Penilaian Tahap Keselamatan Rangkaian dan Sistem ICT Sektor Awam

Nota: Tertakluk kepada garis panduan/ piawaian semasa yang berkuatkuasa.

1.6.8 KEPATUHAN KEPADA REGULASI

Setiap pelaksanaan pembangunan sistem aplikasi sektor awam hendaklah mematuhi akta, peraturan, dasar, arahan, dan pekeliling Kerajaan yang berkuat kuasa. Kepatuhan ini penting bagi memastikan sistem yang dibangunkan bukan sahaja memenuhi keperluan fungsian dan teknikal, tetapi juga mematuhi prinsip perundangan, keselamatan maklumat, dan tadbir urus yang ditetapkan oleh Kerajaan Malaysia.

Agensi bertanggungjawab untuk memastikan semua aktiviti pembangunan, pelaksanaan dan penyelenggaraan sistem dilaksanakan selaras dengan kerangka peraturan sedia ada, termasuk keperluan berkaitan keselamatan siber, perlindungan data peribadi, pengurusan rekod elektronik, serta pematuhan kepada piawaian teknikal dan operasi sektor awam.

Pemantauan kepatuhan perlu dilaksanakan secara berterusan melalui semakan dalaman, audit ICT serta pengesahan teknikal oleh pihak yang berwibawa. Pendekatan ini membantu mengurangkan risiko ketidakpatuhan, meningkatkan akauntabiliti, dan menjamin ketelusan dalam pelaksanaan projek ICT kerajaan.

1.6.9 KEPENTINGAN PENGUKURAN SAIZ SISTEM

Dalam perancangan pembangunan projek yang melibatkan pembangunan sistem aplikasi, langkah pertama yang perlu dilakukan ialah membuat anggaran 4 parameter utama iaitu saiz sistem, usaha pembangunan (development effort), masa dan kos. Kesemua parameter ini merupakan input utama bagi pelan keseluruhan projek.

Apabila saiz sistem tidak diukur dengan tepat, maka kebiasaannya anggaran usaha pembangunan, masa dan kos adalah hanya berdasarkan kepada pengalaman dan rekod-rekod sejarah yang lepas. Pengalaman dan rekod-rekod sejarah tersebut pula dikumpulkan berdasarkan kepada cadangan pihak kontraktor di mana kadang-kadang kos yang ditawarkan boleh dipersoalkan sama ada berpatutan atau sebaliknya. Oleh yang demikian, adalah sangat penting saiz sistem diukur dengan lebih tepat dan secara saintifik serta mematuhi piawaian antarabangsa.

Apabila membuat perancangan projek-projek ICT yang melibatkan pembangunan sistem baru atau peningkatan sistem sedia ada, pihak pengurusan memerlukan jawapan kepada tiga persoalan asas berikut:

- a) Berapakah jumlah kos yang terlibat?
- b) Berapa lama masa diambil untuk menyiapkan projek?
- c) Siapa akan melaksanakannya dan adakah pilihan yang dibuat akan mengurangkan kos atau mempercepatkan pelaksanaan projek?

Jawapan kepada persoalan bisnes ini dapat diberikan sekiranya saiz sistem yang hendak dibangunkan diketahui. Pengiraan saiz sistem merupakan salah satu aktiviti dalam kejuruteraan sistem untuk membuat anggaran saiz sistem aplikasi yang hendak dibangunkan. Terdapat beberapa kaedah pengiraan saiz perisian, antaranya ialah COSMIC Function Points, Mk II Function Points, Nesma Function Points, FiSMA Function Points dan

kaedah IFPUG. Kaedah yang paling terkenal ialah kaedah IFPUG yang juga dikenali dengan Function Point Analysis (FPA). FPA dimiliki dan diselenggara oleh IFPUG (International Function Point Users Group) iaitu sebuah organisasi bukan berasaskan keuntungan yang telah ditubuhkan pada 1986. FPA ini mematuhi standard pengukuran fungsian sistem aplikasi ISO/IEC 14143-1:2007. Metrik pengukuran function points (FP) telah menjadi standard de facto untuk analisis ekonomi sistem aplikasi, sebagai penanda aras kepada produktiviti dan kualiti sistem aplikasi dan lain-lain pengukuran yang berkaitan.

FP sudah dan sedang diterima secara meluas sebagai metrik standards untuk mengukur saiz perisian. Kefahaman mengenai saiz perisian merupakan kunci kepada kefahaman produktiviti dan kualiti dalam proses pembangunan sistem. Tanpa metrik pengukuran saiz perisian yang saintifik dan boleh dipercayai, produktiviti (FPs sebulan kerja) atau kualiti (kecacatan per FP) tidak akan dapat dikira. Dengan FP, pemantauan kemajuan dalam produktiviti dan kualiti pembangunan sistem di setiap fasa pembangunan dapat ditambah baik. Apabila perubahan kepada produktiviti dan kualiti dikira dan diplot mengikut masa, organisasi boleh fokus kepada kekuatan dan kelemahan. Fokus kepada kekuatan dan pembetulan serta penambahbaikan kepada kelemahan akan menjadikan proses pembangunan sistem lebih efektif.

Penggunaan pengukuran saiz fungsian sistem aplikasi yang dibangunkan akan menyediakan maklumat-maklumat tambahan yang akan membantu dalam kejayaan dalam pengurusan projek pembangunan sistem aplikasi. Antara maklumat-maklumat tersebut adalah seperti berikut:

a) Produktiviti

- i) Jam per *function point*, membolehkan organisasi membandingkan produktiviti antara projek-projek atau pun membandingkan dengan *standards* industri.
- ii) Produktiviti Keseluruhan, merujuk kepada *function point* per bilangan sumber manusia terlibat (*total work effort*) yang membolehkan organisasi mentadbir dan memantau pelaksanaan projek secara efektif.
- iii) Kadar pengeluaran/serahan (*rate of delivery*), membolehkan pihak pengurusan membuat perancangan dan analisis masa untuk promosi, atau untuk pelaksanaan projek.

b) Kualiti

- i) Saiz fungsian perisian dalam bentuk function points boleh digunakan sebagai alat untuk membuat anggaran jadual pelan perancangan projek yang lebih tepat.
- ii) Metrik tahap kesempurnaan projek boleh dikira dengan membandingkan bilangan function points berasaskan fungsian yang diminta pengguna berbanding bilangan function points berasaskan fungsian yang diserahkan.
- iii) Metrik kadar perubahan kepada skop projek. Organisasi boleh menggunakan maklumat ini untuk membuat justifikasi kepada permohonan bajet tambahan atau perubahan kepada tarikh serahan projek.
- iv) Kadar kecacatan produk. Ianya boleh digunakan untuk perancangan dan pemantauan di setiap fasa pembangunan perisian bagi memastikan kecacatan yang dikenalpasti dibetulkan sebelum produk dilaksanakan.

c) Kewangan

- i) Kos per *function point* yang boleh digunakan membuat anggaran kos pembangunan aplikasi dan membantu membuat keputusan dalam pemilihan perisian.
- ii) Kos pembetulan perisian per *function points*. Ianya digunakan untuk mengukur kos pembetulan perisian selepas dilaksanakan dalam tempoh tertentu.
- iii) Nilai aset perisian organisasi merupakan satu metrik yang penting untuk menilai aset perisian kepunyaan organisasi.

d) Penyelenggaraan

- i) *Maintainability* merupakan *effort* dalam bentuk kos penyelenggaraan per *function point* – boleh digunakan untuk membuat perancangan dan pemantauan kos penyelenggaraan aplikasi.
- ii) *Reliability* ialah bilangan kegagalan aplikasi berbanding dengan saiz fungsinya, boleh digunakan untuk mengukur tahap kebolehpercayaan aplikasi.
- iii) Bilangan sumber manusia per *function points* untuk membuat penyelenggaraan.

RUJUKAN

- 1 Sommerville, I. (2015). *Software Engineering* (10th ed.). Pearson.
- 2 Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill.
- 3 Brooks, F. P. (1995). *The Mythical Man-Month*. Addison-Wesley.
- 4 Boehm, B. W. (1981). *Software Engineering Economics*. Prentice Hall.
- 5 Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice* (3rd ed.). Addison-Wesley.
- 6 Kerzner, H. (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* (12th ed.). Wiley.
- 7 McGraw, G. (2006). *Software Security: Building Security In*. Addison-Wesley.
- 8 IEEE Std 14764-2006. *IEEE Standard for Software Maintenance*.
- 9 ISO/IEC 27001:2013. *Information Security Management Systems*.
- 10 Kruchten, P. (2004). *The Rational Unified Process: An Introduction* (3rd ed.) Addison-Wesley.

- 11 Madison Technologies. (n.d.). In-House Development vs. IT Outsourcing: Pros, Cons, and Costs. Diperoleh daripada <https://madison-technologies.com/in-house-development-vs-it-outsourcing-pros-cons-and-costs/>
- 12 SmartTek Solutions. (n.d.). In-House vs Outsourced Development: Which One to Choose? Diperoleh daripada <https://smarttek.solutions/blog/in-house-vs-outsourced-development/>
- 13 Argentics. (n.d.). Outsourcing vs Co-Development: Which Strategy Suits Your Game Studio? Diperoleh daripada <https://www.argentics.io/outsourcing-vs-co-development-which-strategy-suits-your-game-studio>
- 14 IEEE Std 12207: Systems and Software Engineering — Software Life Cycle Processes
- 15 IEEE Std 1074-1997: Standard for Developing a Software Project Life Cycle Process
- 16 ISO/IEC/IEEE 24765:2017: Systems and software engineering — Vocabulary
- 17 ISTQB Glossary – <https://glossary.istqb.org>
- 18 Sommerville, I. 2016. Software Engineering (10th Edition). Pearson Education
- 19 Dr. Bruce Powel Douglass. (2021). Agile Model-Based Systems Engineering Cookbook. Packt Publishing Ltd.
- 20 Digital Ocean. (2023). Agile vs. Waterfall. Diperoleh daripada <https://www.digitalocean.com/>
- 21 Nieto-Rodriguez, A. (2023). Harvard Business Review. Diperoleh daripada <https://hbr.org/>

Glosari / Takrifan

DevOps	DevOps merupakan pendekatan atau budaya dalam pembangunan sistem aplikasi yang membolehkan pasukan untuk membangun, menguji dan melepaskan (melaksanakan) sistem aplikasi dengan lebih pantas dan efisien dengan menggabungkan prinsip dan Metodologi <i>Agile</i> .
--------	--