




BAB:01

PENGENALAN KEPADA METODOLOGI KEJURUTERAAN SISTEM APLIKASI

Bab ini merupakan pengenalan dan penerangan ringkas kepada Kitar Hayat

Pembangunan Sistem (SDLC).

Bab ini juga menerangkan Ekosistem Pembangunan Sistem Sektor Awam yang terdiri dari Metodologi Pembangunan Sistem dan aspek-aspek pertimbangan yang mempengaruhi kepada kejayaan pelaksanaan projek pembangunan sistem.



1 PENGENALAN

1.1 Pengenalan Kepada Kitar Hayat Kejuruteraan Pembangunan Sistem Aplikasi

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 seperti digambarkan dalam **Rajah 1**. Setiap fasa mempunyai siri aktiviti melalui penggunaan teknik-teknik tertentu bagi penghasilan dokumentasi serahan.



Rajah 2: Kitar Hayat Pembangunan Sistem Aplikasi

Terdapat 2 elemen utama yang perlu difahami dalam SDLC iaitu.

- Perlu memahami fasa-fasa dalam SDLC, langkah-langkah yang perlu dilakukan dalam projek pembangunan sistem aplikasi dan teknik-teknik yang diperkenalkan dalam penghasilan serahan yang tertentu.
- Perlu memahami bahawa model SDLC melibatkan serahan yang dihasilkan dalam setiap fasa. Serahan dalam setiap fasa akan digunakan sebagai input kepada fasa berikutnya.

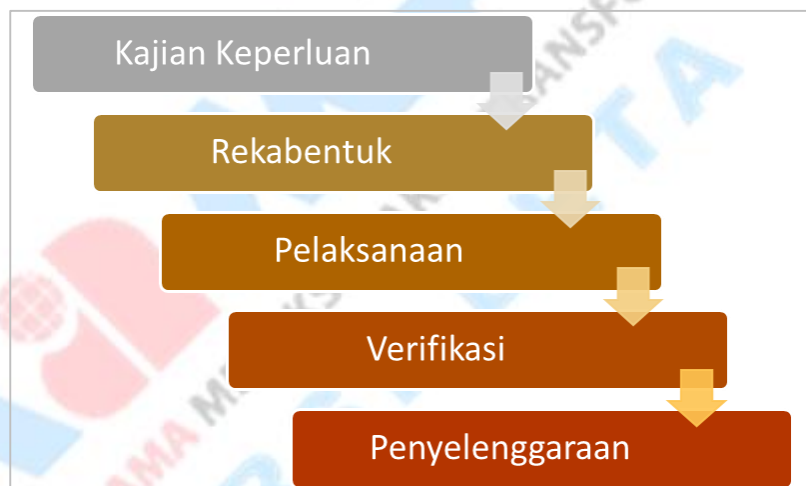
¹ <http://istqbexamcertification.com/what-are-the-software-development-life-cycle-sdlc-phases/>

1.2 Metodologi Pembangunan Sistem Aplikasi

Metodologi Pembangunan Sistem Aplikasi adalah satu rangka kerja yang digunakan untuk menstruktur, merancang dan mengawal proses pembangunan sistem aplikasi. Terdapat pelbagai metodologi pembangunan sistem aplikasi yang telah diperkenalkan. Kesesuaian sesuatu metodologi pembangunan sistem aplikasi adalah bergantung kepada jenis organisasi, keupayaan dan keperluan teknikal, serta jenis dan pasukan projek. Antara model yang sering digunakan adalah:

a) Metodologi *Waterfall*

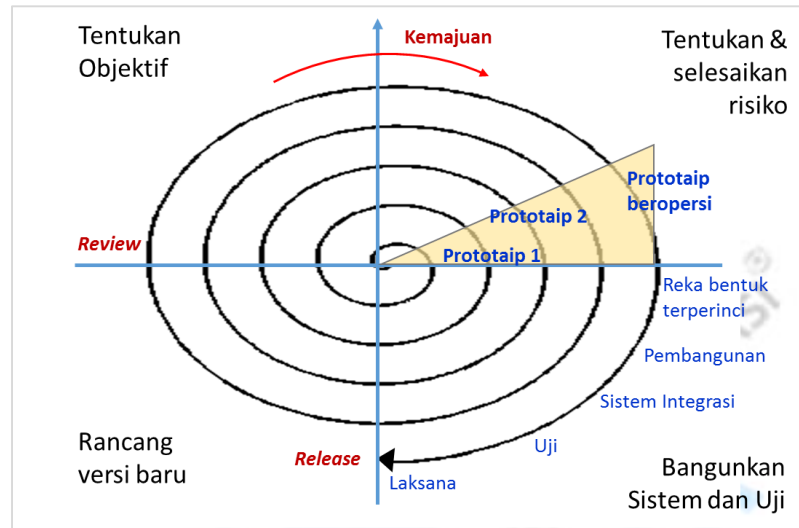
Metodologi *waterfall* juga dikenali Metodologi Jujukan *Linear*. Model ini menerapkan kawalan yang ketat terhadap setiap fasa di dalam SDLC. Semakan dan pengesahan serahan dilakukan secara formal dengan pemegang taruh pada setiap penghujung fasa. Sesuatu fasa seterusnya tidak akan dimulakan sekiranya semakan dan pengesahan serahan bagi fasa semasa tidak disempurnakan.



Rajah 3 : Metodologi *Waterfall*

b) Metodologi *Spiral*

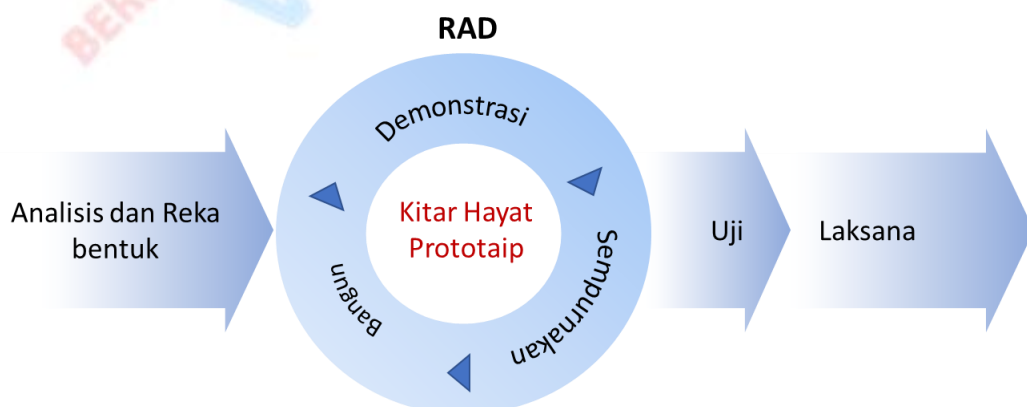
Metodologi ini adalah gabungan konsep berulang di dalam Metodologi *Prototype* dan aspek kawalan yang sistematik seperti yang terdapat pada metodologi Jujukan *Linear*. Metodologi ini memberi fokus kepada penilaian produk dan meminimumkan risiko dalam pembangunan sistem. Model ini memecahkan skop projek kepada segmen-segmen yang kecil bagi memudahkan semakan dan perubahan dilakukan.



Rajah 4 : Metodologi *Spiral*

c) Metodologi *Rapid Application Development (RAD)*

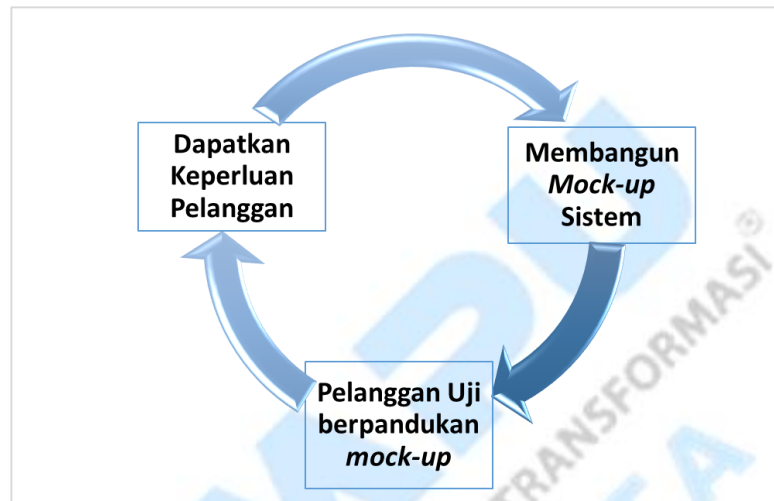
Putaran *RAD* merangkumi 4 fasa utama iaitu perancangan keperluan, reka bentuk, pembangunan dan pelaksanaan. Fasa ini dilaksanakan oleh sekumpulan pembangun aplikasi yang mahir yang bekerjasama rapat dengan pengguna sepanjang tempoh pembangunan. Teknik dan *tools* yang digunakan merupakan faktor utama kejayaan *RAD*. Matlamat utama metodologi ini adalah menghasilkan sistem yang berkualiti tinggi secara cepat dengan memberikan penekanan terhadap keperluan pengguna.



Rajah 5 : Metodologi *RAD*

d) Metodologi *Prototype*

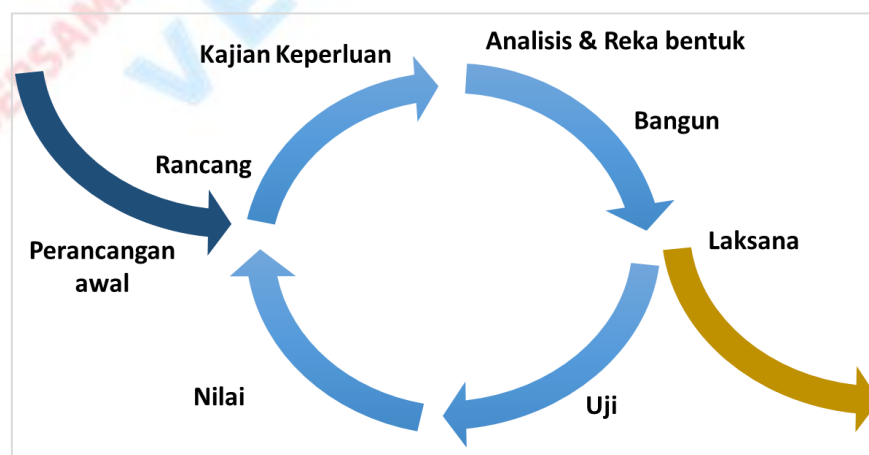
Metodologi ini terdapat sesuai digunakan jika pengguna sistem sukar mengenalpasti keperluan sistem. Terdapat 3 fasa penting dalam metodologi *Prototype* iaitu fasa keperluan pengguna, fasa pembangunan dan fasa pengujian. Proses ini diulang sehinggalah sistem lengkap dibangunkan. Metodologi ini mengaplikasikan proses perubahan lelaran (*iterative modification process*) sehingga sistem prototaip berkembang dan memenuhi kehendak pengguna.



Rajah 6 : Metodologi *Prototype*

e) Metodologi Pembangunan Berperingkat (*Incremental Development*)

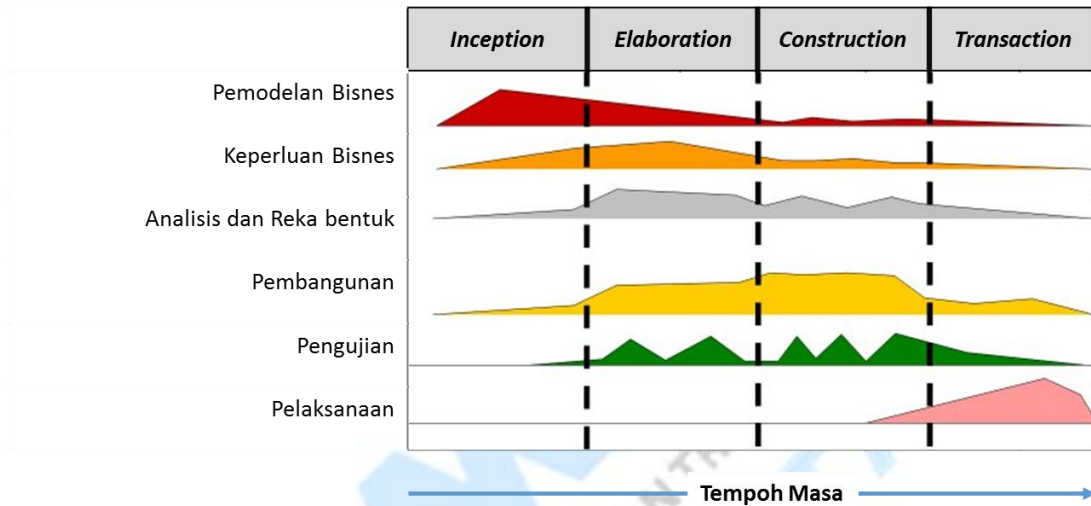
Pendekatan ini mengaplikasikan metodologi *waterfall* dalam pembangunan utama sistem (*system core*) dan diikuti dengan metodologi *Prototype* secara lelaran (*iterative*). Pembangunan secara prototaip sehinggalah prototaip menjadi sistem aplikasi yang lengkap mengikut kehendak pengguna.



Rajah 7 : Metodologi *Incremental Development*

f) Metodologi *Rational Unified Process (RUP)*

Metodologi *RUP* adalah proses pembangunan sistem yang menyediakan satu pendekatan berdisiplin dalam menetapkan tugas dan peranan sesuatu pasukan pembangunan sistem. Matlamat *RUP* adalah untuk memastikan produk yang dihasilkan menepati kehendak pengguna dan di dalam lingkungan peruntukan dan jadual yang dianggarkan. *RUP* menggunakan 4 fasa utama iaitu *Inception*, *Elaboration*, *Construction* dan *Transaction*.



Rajah 8 : Metodologi *RUP*

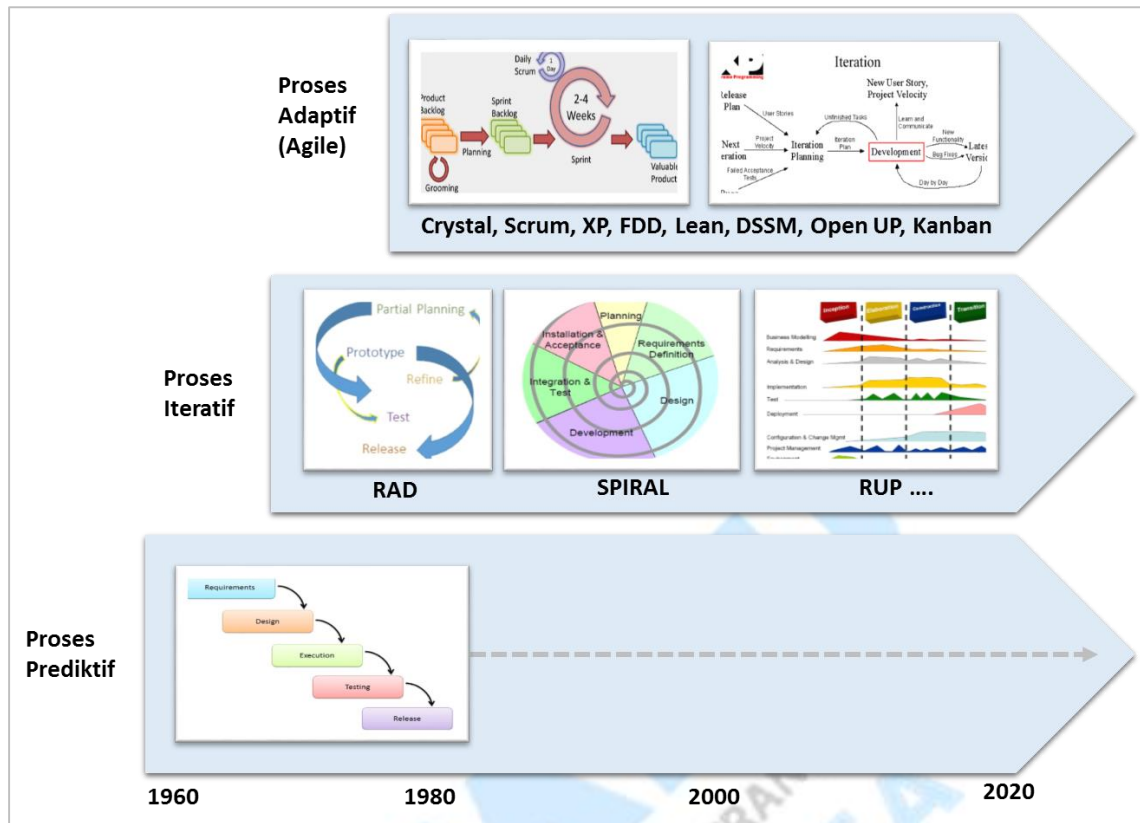
1.3 Klasifikasi Metodologi Pembangunan Sistem Aplikasi

Metodologi-metodologi pembangunan sistem aplikasi yang digunakan boleh diklasifikasikan kepada 3 model utama iaitu model yang berasaskan kepada proses prediktif, model yang berasaskan kepada proses iteratif dan model yang berasaskan kepada proses adaptif. Model-model ini menentukan pendekatan dalam proses pembangunan sistem aplikasi.

Model Prediktif, proses pembangunan sistem aplikasi berlaku dalam satu siri fasa-fasa secara tersusun. Selepas keperluan sistem dipersetujui, reka bentuk sistem dibuat dan diikuti dengan pengaturcaraan. Akhirnya, sistem diuji sebagai pengesahan pematuhan kepada keperluan. Metodologi *Waterfall* termasuk dalam kategori model ini.

Model Iteratif merupakan model yang berasaskan penemuan (*discovery-based*) di mana pada peringkat awal dokumen keperluan sistem lebih ringkas berbanding dengan model prediktif. Bermula dengan dokumen asas yang menerangkan apa yang hendak dibangunkan, keperluan sebenar sistem akan dikenalpasti dan ditemui semasa proses iteratif. Metodologi *Spiral*, *RAD* dan *RUP* termasuk dalam kategori model ini.

Model Adaptif merupakan evolusi daripada model prediktif dan model iteratif. Model ini juga dikenali metodologi *Agile*. Model ini membolehkan organisasi menyelesaikan masalahnya secara holistik melalui persekitaran pembangunan sistem yang transparan dan adaptif. Persekitaran ini diwujudkan melalui pembentukan pasukan pembangunan yang menekankan kolaboratif antara pihak bisnes (SME) dan pihak ICT untuk menghasilkan sistem yang selari dengan strategi organisasi dengan cepat. Tanpa melengkapkan dokumen keperluan lebih awal, aktiviti pengaturcaraan akan dilaksanakan secepat mungkin dan akan terus dinilai oleh SME. Kelemahan atau kekurangan yang ditemui kemudiannya akan terus dibetulkan oleh pihak ICT. Ini bermakna perubahan kepada keperluan sistem boleh berlaku di semua fasa pembangunan sistem dan dokumen-dokumen sistem dimuktamadkan selepas pihak SME berpuas hati dengan produk yang dibangunkan. Metodologi *Scrum*, *Extreme Programming (XP)* dan *Lean* termasuk dalam kategori model ini. **Rajah 9** merupakan model-model Metodologi Pembangunan Sistem Aplikasi.



Rajah 9 : Model Klasifikasi Metodologi Pembangunan Sistem Aplikasi

Sumber : Trail Ridge Consulting, LLC

Rujukan

1. Capers Jones (2010). Software Engineering Best Practices. McGraw-Hill Companies
2. Uml-diagrams.org (2009-2017). Activity Diagrams. <https://www.uml-diagrams.org/activity-diagrams.html>.
3. Leffingwell, Dean (2011), Agile Software Requirements : Lean Requirements Practices For Teams, Programs, And The Enterprise.

1.4 Ekosistem Pembangunan Sistem Sektor Awam

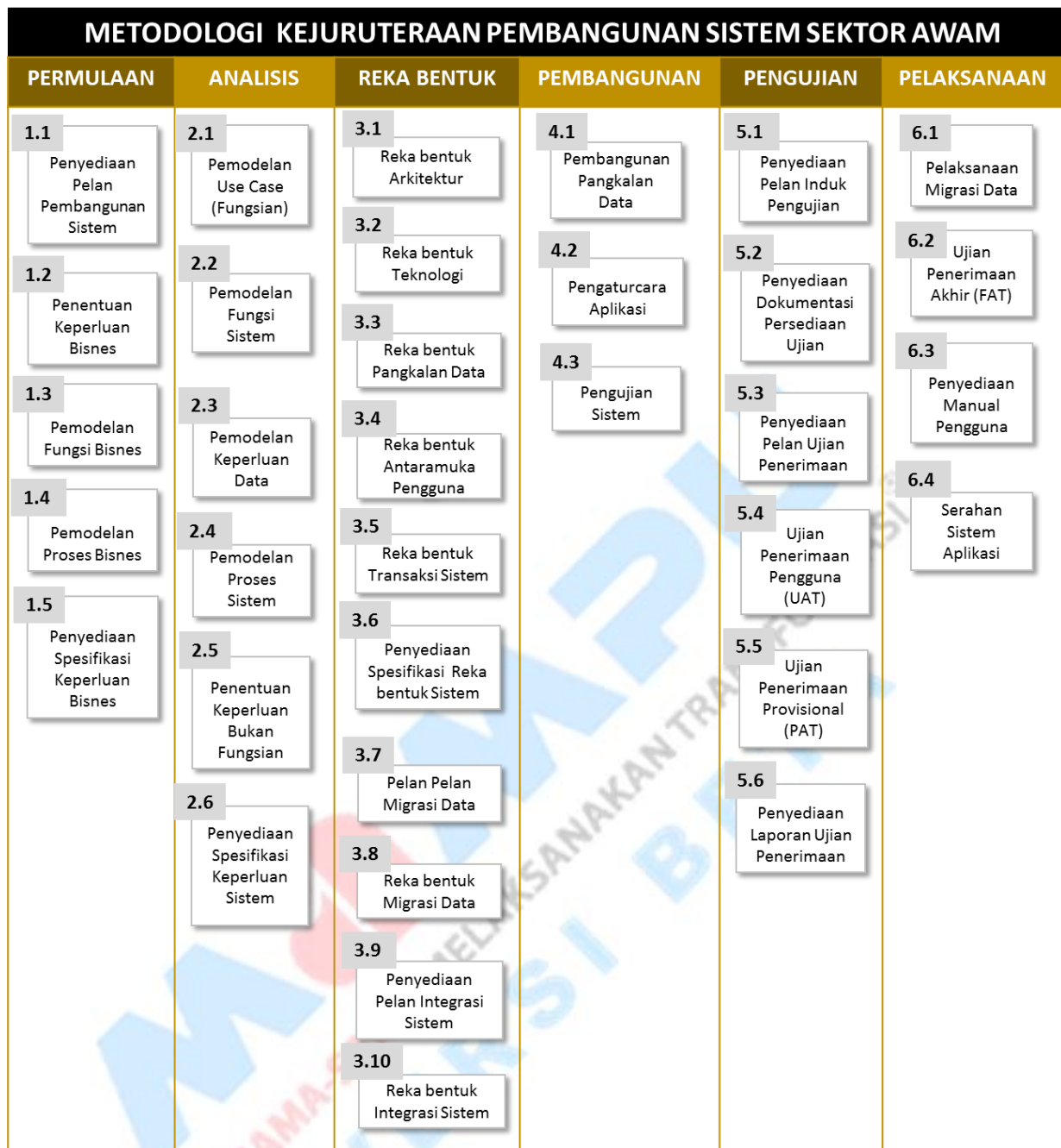
Ekosistem Pembangunan Sistem Aplikasi Sektor Awam mengambil kira elemen-elemen lain dalam persekitaran ICT dan hubungannya dengan metodologi dan pendekatan pembangunan sistem aplikasi di agensi-agensi Sektor Awam. Ekosistem tersebut digambarkan dalam **Rajah 10**. Aspek pertimbangan utama dalam pembangunan sistem aplikasi adalah Anggaran Keperluan Sumber, Pengurusan Kawalan Pindaan, Penglibatan Pemegang Taruh, Jaminan Kualiti Perisian (SQA), Keselamatan Aplikasi dan Tadbir Urus. Aspek-aspek ini perlu dipacu bersama-sama metodologi pembangunan sistem aplikasi ke arah mencapai Pelan Strategik ICT dan Pelan Strategik Bisnes yang dirancang.



Rajah 10 : Ekosistem Kejuruteraan Pembangunan Sistem Aplikasi Sektor Awam

1.5 Metodologi Kejuruteraan Sistem Aplikasi Sektor Awam

Metodologi Kejuruteraan Sistem Aplikasi Sektor Awam dibangunkan dengan mengambilkira tinjauan dan pengalaman ke atas metodologi pembangunan sistem yang telah diamalkan dalam industri dan sektor awam. Metodologi ini merangkumi 6 fasa utama iaitu: fasa permulaan projek (*initiation*), fasa analisa (*analysis*), fasa reka bentuk (*design*), fasa pembangunan (*construction*), fasa Pengujian (*testing*) dan fasa pelaksanaan (*implementation*). **Rajah 11** merupakan Metodologi Pembangunan Sistem Aplikasi Sektor Awam yang menunjukkan hubungan antara fasa dan aktiviti dalam setiap fasa.



Rajah 11 : Metodologi Kejuruteraan Pembangunan Sistem Aplikasi Sektor Awam

Fasa 1 : Permulaan Projek

Fasa permulaan projek memberikan penekanan kepada komunikasi dan komitmen Pemegang Taruh dan organisasi yang bertanggungjawab membangunkan sesuatu projek selepas sesuatu permintaan diterima. Fasa ini dibahagikan kepada 2 aktiviti utama iaitu:

a) Penyediaan Pelan Pembangunan Sistem / Piagam Pelanggan

Penyediaan Pelan Pembangunan Sistem (PPS) adalah penting sebagai komitmen pasukan projek dengan pemilik sistem dan pemegang taruh. Butiran komitmen merangkumi skop, tujuan, keperluan sumber, tempoh projek dan faedah/impak sistem kepada bisnes organisasi. Pelan Pembangunan Sistem perlu mendapat persetujuan bersama antara pasukan pembangun dan pemilik sistem.

b) Kajian Keperluan Bisnes

Penentuan Keperluan Bisnes memberikan tumpuan ke atas aktiviti mengumpul keperluan bisnes sesuatu organisasi (organisasi pemegang taruh). Ini untuk memastikan sistem yang dibangunkan menepati keperluan organisasi *Pemegang Taruh* secara menyeluruh (contoh: menepati visi, misi dan objektif agensi) dan menepati keperluan organisasi secara spesifik. Proses-proses yang berlaku di dalam aktiviti ini adalah mendefinisikan unit Bisnes dan Peranan, Pemodelan Fungsi dan Proses Bisnes. Hasil daripada aktiviti ini ialah D02 Spesifikasi Keperluan Bisnes.

Fasa 2 : Analisis

Matlamat utama fasa ini adalah melaksana analisis ke atas keperluan secara terperinci untuk menghasilkan **D02 Spesifikasi Keperluan Bisnes** Output bagi aktiviti ini adalah **D03 Spesifikasi Keperluan Sistem** yang menyatakan keperluan bagi sistem dari perspektif pembangun sistem. Ia menyatakan perkara-perkara atau item-item yang perlu ada didalam sesuatu sistem bagi merealisasikan keperluan bisnes atau pemegang taruh. Proses-proses yang berlaku di dalam aktiviti ini adalah pemodelan use case (fungsian), pemodelan keperluan data dan pemodelan proses sistem.

Fasa 3 : Reka bentuk

Berdasarkan keperluan sistem yang diperolehi di dalam Fasa Analisis, arkitektur keseluruhan sistem akan dihasilkan. Arkitektur sistem ini mendefinisikan komponen, perlakuan dan antaramuka komunikasi bagi sesuatu sistem. Fasa ini juga menerangkan tentang bagaimana sistem ini akan dihasilkan. Ia merangkumi aktiviti-aktiviti seperti reka bentuk arkitektur; reka bentuk sistem; reka bentuk pangkalan data, serta penentuan teknologi yang akan digunakan. Output kepada fasa ini adalah **D04 Spesifikasi Reka bentuk Sistem**.

Fasa 4 : Pembangunan

Aktiviti-aktiviti yang dilaksanakan di dalam fasa pembangunan adalah berkaitan proses penghasilan dan pengujian sistem oleh pasukan pembangun. Fasa pembangunan merealisasikan SDS yang dihasilkan dalam fasa reka bentuk. Komponen dan fungsi sistem dibangunkan melalui proses pengaturcaraan dan diintegrasikan untuk menghasilkan sesuatu produk atau sistem. Aktiviti-aktiviti yang dilaksanakan di dalam fasa ini adalah pembangunan pangkalan data, pengaturcaraan sistem dan pengujian sistem. Di dalam fasa ini, proses penghasilan sistem aplikasi adalah matlamat utama.

Fasa 5 : Pengujian

Aktiviti-aktiviti yang dilaksanakan di dalam fasa pengujian adalah berkaitan dengan penyediaan pelan ujian dan dokumentasi berkaitan ujian, serta pelaksanaan pengujian penerimaan oleh pengguna ke atas sistem. Pengujian yang dimaksudkan adalah Ujian Penerimaan Pengguna (UAT) dan Ujian Penerimaan Sementara (PAT). Ujian ini dilaksanakan sebagai validasi ke atas sistem aplikasi yang dibangunkan berdasarkan keperluan pengguna dan keperluan sistem bagi memastikan keperluan tersebut dipenuhi sebelum sistem aplikasi dilaksanakan.

Fasa 6 : Pelaksanaan

Aktiviti utama di dalam fasa pelaksanaan adalah melaksanakan aktiviti ke arah persediaan pelaksanaan sistem. Aktiviti-aktiviti yang dilaksanakan di dalam fasa ini adalah migrasi data, ujian penerimaan akhir, persediaan manual pengguna dan laporan serahan sistem.

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 dititikberatkan dalam melaksana metodologi kitar hayat pembangunan sistem ke arah mencapai matlamat bisnes yang ditetapkan.

1.6.1 Tadbir Urus Pengurusan Projek ICT

Merujuk kepada Buku Metodologi PRrISA Panduan Pengurusan Projek ICT Sektor Awam, 2016 yang diterbitkan oleh MAMPU, tadbir urus projek adalah penting bagi memastikan kejayaan projek ICT. Ianya meliputi penubuhan struktur pasukan projek pembangunan dan jawatankuasa pemantauan projek.

Organisasi Pasukan Projek Pembangunan diketuai oleh Pengurus Projek dengan bantuan Pejabat Pengurusan Projek (PMO) yang berfungsi sebagai penyelarasan kepada 2 pasukan utama projek seperti berikut:

a) Pasukan Pembangunan

Pasukan pembangunan bertanggungjawab dalam melaksana aktiviti mengkaji, menganalisa, membangun sistem aplikasi dan melaksana pengujian sistem

b) Pasukan SME

Pasukan SME bertanggungjawab dalam menyediakan input D03 Spesifikasi Keperluan Sistem dan melaksana pengujian penerimaan ke atas sistem

Jawatankuasa pemantauan projek adalah terdiri 2 jawatankuasa utama iaitu:

a) Jawatankuasa Pemandu Projek

Jawatankuasa Pemandu Projek bertanggungjawab memutuskan strategi dan halatuju projek pembangunan yang terdiri daripada keperluan sumber manusia, kewangan dan proses. Jawatankuasa ini juga memantau pelaksanaan projek secara keseluruhan.

b) Jawatankuasa Teknikal Projek

Jawatankuasa Teknikal Projek bertanggungjawab memperakui halatuju, serahan dan strategik, menyelesaikan isu-isu teknikal, menyelaras dan memantau pelaksanaan mengikut skop projek yang dipersetujui.

1.6.2 Penglibatan Pemegang Taruh

Penglibatan pemegang taruh sepanjang aktiviti pembangunan sistem aplikasi dikira sebagai faktor utama bagi memastikan kejayaan dalam pembangunan dan pelaksanaan sistem aplikasi. Di samping menyumbang kepada kualiti sistem yang dibangunkan, penglibatan pemegang taruh juga dapat menyelesaikan strategi pengurusan perubahan ke arah penggunaan sistem aplikasi yang dibangunkan.

Faedah Yang Diperolehi

- a) Meningkatkan kualiti sistem dan ketetapan kepada kehendak dan keperluan pengguna dalam menyokong peranan dan bisnes agensi
- b) Meningkatkan kefahaman dan penerimaan sistem aplikasi yang dibangunkan
- c) Dengan penglibatan SME dalam mengenalpasti keperluan sistem dalam konteks fungsi organisasi, ini dapat mengurangkan masa pembangunan
- d) 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.

Kategori Pemegang Taruh

Dalam pelaksanaan pembangunan sistem agensi sektor awam, terdapat 5 kategori pemegang taruh iaitu:

a) Penganjur Projek

Penganjur projek adalah Ketua Jabatan/Agensi yang melulus dan memberi peruntukan bagi pembangunan sistem aplikasi. Sokongan dari pengurusan atasan agensi adalah faktor utama yang membawa kepada kejayaan pembangunan dan pelaksanaan sistem aplikasi.

b) Pemilik Projek

Pemilik projek adalah Ketua Bahagian yang memiliki fungsi dan mempunyai kuasa terhadap dasar dan prosidur sistem aplikasi yang sedang dibangunkan. Mereka bertanggungjawab dalam memastikan sistem yang dibangunkan memenuhi keperluan jabatan.

c) *Subject Matter Expert (SME)*

SME adalah kumpulan pegawai rujukan kepada fungsi bisnes agensi berkaitan dengan sistem yang akan dibangunkan.

d) Ahli Pasukan Projek

Pasukan pengguna yang terlibat dalam menyatakan keperluan sistem, terlibat dalam reka bentuk sistem dan pengujian sistem. Mereka akan menjadi pegawai penghubung antara pengguna akhir sistem dan pasukan pembangun.

e) Penasihat Projek

Pengurusan atasan agensi yang memberi nasihat dalam keperluan pengguna dan sepanjang pentadbiran dan pengurusan projek dilaksanakan. Selalunya penasihat projek akan mempengerusi Mesyuarat Jawatankuasa Projek.

1.6.3 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 PPrISA Panduan Pengurusan Projek ICT Sektor Awam, 2016 yang diterbitkan oleh MAMPU, 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.

1.6.4 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 semakan 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**.

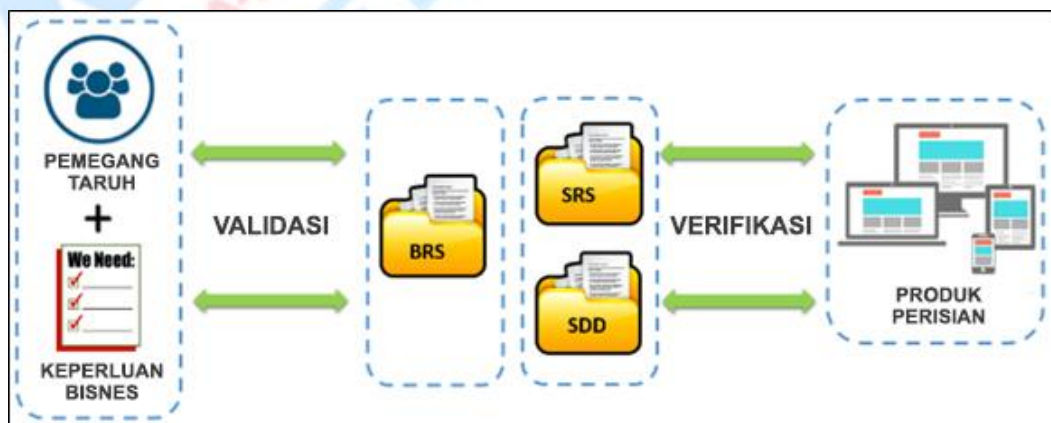
Jadual 3 : Atribut Kualiti Perisian

Atribut	Keterangan	Contoh
Atribut fungsian (Functional attributes)	Input dan output produk perisian	<ul style="list-style-type: none"> i. Sistem dapat memenuhi keperluan pengguna dengan tepat (<i>correctness</i>). ii. Sistem mengawal capaian modul mengikut peranan pengguna (<i>authentication</i>). iii. 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	<ul style="list-style-type: none"> i. Sistem berupaya memberikan tindak balas (<i>latency or response time</i>) dalam masa yang ditetapkan ii. Sistem dapat menampung kapasiti (<i>capacity</i>) pengguna serentak yang ditetapkan iii. Sistem dapat beroperasi walaupun melebihi kapasiti pengguna yang ditetapkan.
Atribut kebolegunaan (Usability attributes)	Sejauh mana produk perisian boleh digunakan dan disesuaikan	<ul style="list-style-type: none"> i. Sistem mudah digunakan (<i>ease of use</i>) oleh pengguna ii. Sistem mudah dipelajari (<i>ease of learn</i>) oleh pengguna

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

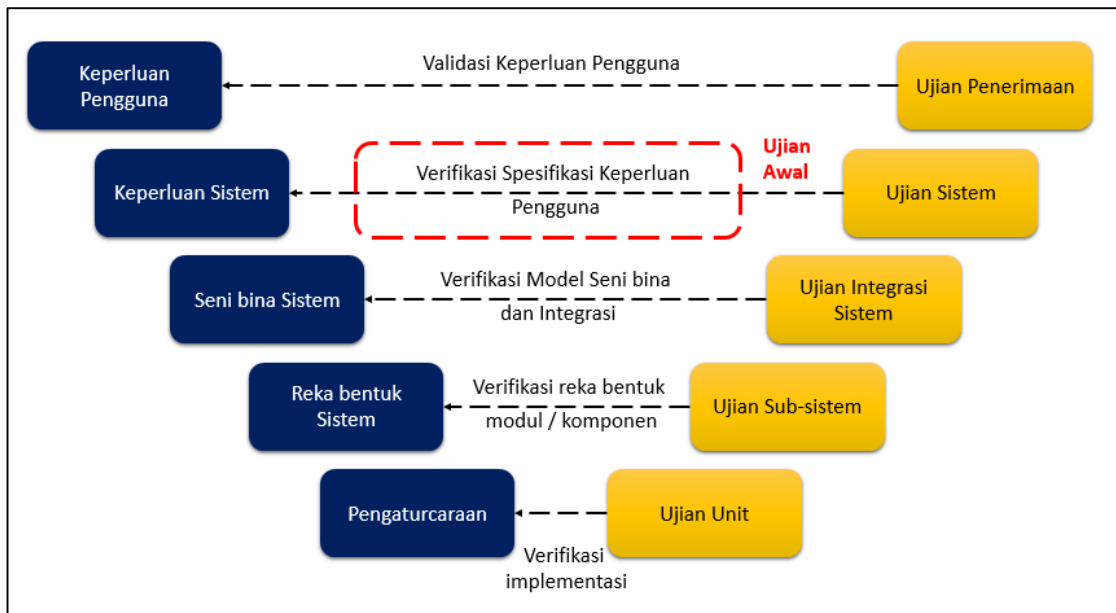
b) Verifikasi dan Validasi

Verifikasi adalah proses semakan dokumentasi dan reka bentuk 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 12 : 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 13 : Jenis-Jenis Pengujian V&V

Jadual 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • Ujian Penerimaan Pengguna (UAT) • Ujian Penerimaan Sementara (PAT) • Ujian Penerimaan Akhir (FAT)

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

- a) Memastikan standard dan prosedur dipatuhi.
- b) Mewujudkan dokumentasi system yang seragam dan mudah difahami.
- c) Memastikan masalah dapat ditemui awal dan diuruskan dengan sewajarnya.
- d) Memantau dan menambah baik proses pembangunan perisian.

1.6.5 Faktor Keselamatan ICT

Elemen keselamatan dalam pembangunan sistem aplikasi adalah perlu diambil kira dalam proses kitaran hayat pembangunan sistem aplikasi. Sistem aplikasi sering terdedah kepada serangan kerana wujudnya kelemahan semasa aktiviti reka bentuk dan pengaturcaraan. Kelemahan sistem aplikasi boleh menyebabkan ia boleh diubahsuai oleh pihak yang tidak bertanggungjawab sehingga sistem aplikasi tersebut tidak dapat beroperasi dengan baik. Tindakan pencegahan adalah penting untuk mengurangkan ancaman ke atas sistem aplikasi. Aspek yang perlu diambil kira dalam menjamin keselamatan sistem adalah:

a) Prinsip umum Pembangunan Sistem Terselamat

Keselamatan sistem boleh diancam pada mana-mana aktiviti sepanjang kitaran hayatnya, sama ada secara sengaja atau tidak sengaja oleh "orang dalam" atau oleh "orang luar" yang tidak mempunyai hubungan dengan organisasi. Penambahbaikan kelemahan yang dikesan pada peringkat awal sepanjang kitaran hayat sistem aplikasi adalah lebih kos efektif daripada membangun dan mengeluarkan versi/patch keselamatan baru untuk sistem beroperasi. Untuk dianggap selamat, sistem mestilah mempunyai ciri-ciri:

i) Kebergantungan (*Dependability*)

Sistem yang dibangunkan boleh beroperasi dengan lancar di bawah semua keadaan dan platform, termasuklah sekiranya terdapat serangan atau pelbagai niat jahat.

ii) Dipercayai (*Trustworthiness*)

Sistem boleh dipercayai walaupun terdapat kelemahan yang sengaja dieksploitasi untuk mengganggu pengoperasian sistem. Sistem yang boleh dipercayai mesti mengandungi logik yang boleh menghalang sebarang pencerobohan jahat.

iii) Daya Tahan (*Resilience*)

Sistem yang berdaya tahan adalah sistem yang mampu menentang serangan atau dapat dipulihkan dengan cepat sekiranya ada serangan yang tidak boleh ditolak/ ditahan.

b) Ciri-Ciri Keselamatan Utama Pembangunan Sistem

Terdapat 7 ciri-ciri keselamatan yang perlu diambil kira bagi memastikan sistem aplikasi adalah selamat. Ciri-ciri tersebut adalah seperti berikut:

i) Keselamatan Tahap Sistem

Sistem dapat mengawal akses aplikasi pada peranan setiap pengguna. Biasanya sistem dikawal berasaskan paparan pilihan menu yang berbeza mengikut peranan pengguna.

ii) Keselamatan *Row-Level*

Pengguna hanya dapat melihat data yang dibenarkan mengikut peranan mereka di dalam aplikasi yang sama.

iii) *Single Sign-On (SSO)*

Satu proses pengesahan pengguna yang membolehkan pengguna memasukkan id pengguna dan kata laluan mereka pada satu sistem sahaja tetapi pengesahan tersebut memberi kebenaran kepada pengguna untuk mengakses sistem lain yang berkaitan pada sesi yang sama tanpa perlu log masuk ke sistem tersebut. SSO akan mengurangkan bilangan ID Pengguna/kata laluan yang perlu diingat dan memerlukan login pada setiap aplikasi yang diperlukan.

iv) Parameter Keistimewaaan Pengguna

Parameter keistimewaaan pengguna digunakan untuk memperibadikan ciri dan keselamatan kepada pengguna individu atau peranan pengguna. Parameter keistimewaaan pengguna disimpan ke profil pengguna dan boleh diakses pada keseluruhan sistem. Ianya sangat fleksibel, dapat mengawal, menambah atau menyembunyi pilihan pengguna. Contoh: walaupun semua pengguna boleh mengakses aplikasi yang sama, tetapi hanya pengguna tertentu sahaja yang boleh melihat pilihan untuk mengemaskini data.

v) Pilihan Pengesahan Fleksibel (*Flexible Authentication Options*)

Sistem aplikasi yang dibangunkan sepatutnya menawarkan pelbagai pilihan kaedah pengesahan pengguna, sistem sepatutnya fleksibel untuk menggunakan kaedah pengesahan yang sedia ada tanpa perlu wujud atau mengubah kaedah pengesahan semasa.

vi) Sumber Data Pengguna (*User Specific Data Source*)

Ciri keselamatan ini adalah sama dengan row-level keselamatan, tetapi di peringkat pangkalan data. Aplikasi yang dibangunkan boleh mengakses sumber data yang berbeza bergantung kepada pengguna. Contoh: Apabila berlaku penggabungan 2 syarikat yang menggunakan sistem yang sama. Syarikat A perlu akses pangkalan data *local*, manakala pekerja syarikat B perlu akses data dari pangkalan data berlainan.

vii) Pengauditan Aktiviti Sistem

Pengauditan aktiviti sistem membolehkan pembangun log aktiviti pengguna akhir untuk setiap aktiviti *Sign On / Sign Off*. Ini membolehkan sistem menjejaki dengan cepat dan merekod log masuk dan keluar pengguna, fungsi yang dicapai dan aktiviti yang dilakukan terhadap data.

1.6.6 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 ekonomik sistem aplikasi, sebagai penanda aras kepada produktiviti dan kualiti sistem aplikasi dan lain-lain pengukuran yang berkaitan. Keterangan yang terperinci mengenai *function points* dan kaedah IFPUG FPA akan diterangkan dalam Bab 8.

Persoalan yang selalu ditanya apabila membuat inisiatif berkaitan pembangunan sistem aplikasi ialah berapa besar atau berapa kompleks sistem aplikasi yang hendak dibangunkan. Adakah sistem yang hendak dibangunkan bersaiz kecil, sederhana atau besar? Apabila kaedah FPA diaplikasikan, penentuan kategori saiz sistem dapat ditentukan dengan yakin dan jelas. Capers Jones Pengarang buku *Software Engineering Best Practices - Lessons from Successful Projects in the Top Companies* (2010), menyatakan perisian kategori kecil ialah perisian yang kurang daripada 100FP, kategori sederhana bagi perisian bersaiz antara 100FP hingga 999FP dan kategori

besar bagi perisian bersaiz 1000FP ke atas. Apabila saiz sistem dapat diketahui, maka kos, tempoh masa dan usaha pembangunan dapat ditentukan dengan merujuk kepada jadual standards antarabangsa mengikut negara atau mengikut standards organisasi sekiranya ada.

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. Hassan Gomma (2011). Software Modelling and Design . Cambridge University Press
2. Ali Mili & Fairouz Tahier (2011). Software Testing. John Wiley & Sons. Inc.