



BAB:04

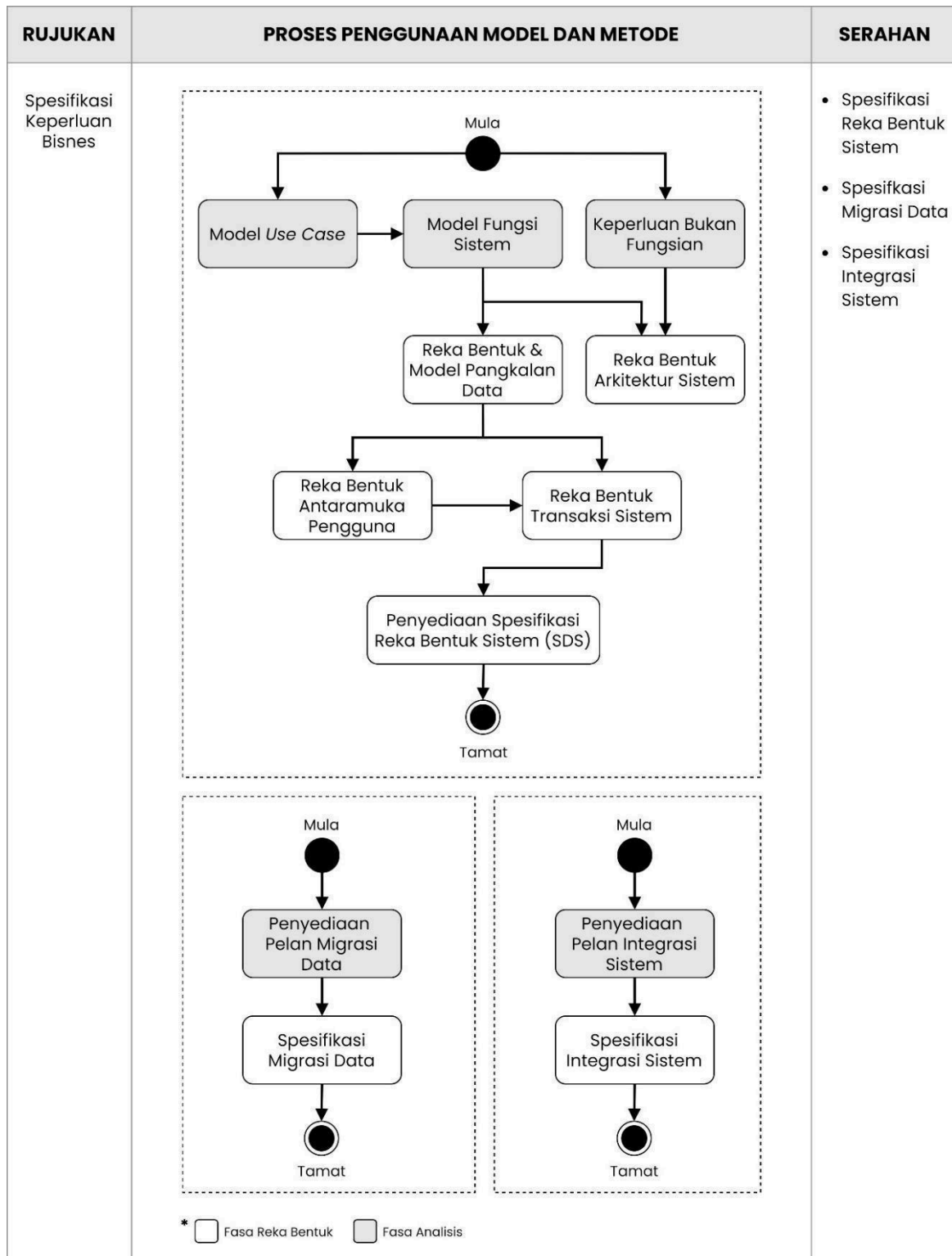
FASA REKABENTUK

Bab ini menerangkan berkenaan dengan aktiviti-aktiviti penyediaan Arkitektur Sistem, Rekabentuk Pangkalan Data, Rekabentuk Antara Muka Pengguna termasuk Pemetaan Data, Rekabentuk Transaksi Sistem, Rekabentuk Migrasi dan Rekabentuk Integrasi berdasarkan kepada maklumat keperluan sistem yang telah dikenalpasti.



4 FASA REKA BENTUK

4.1 GAMBARAN KESELURUHAN



Rajah 4.1 Gambaran Keseluruhan Fasa Reka Bentuk

4.2 PENGENALAN

Fasa reka bentuk adalah fasa bagi mereka bentuk penyelesaian masalah dan ekspektasi yang dinyatakan dalam **D03 Spesifikasi Keperluan Sistem**. Fasa ini adalah langkah permulaan untuk ترجمahkan dari domain masalah kepada domain penyelesaian iaitu peralihan daripada ‘Apa?’ kepada ‘Bagaimana?’. Reka bentuk sistem adalah faktor yang paling kritikal yang akan menjejaskan kualiti perisian dan mempunyai kesan yang besar kepada aktiviti pembangunan atau pengaturcaraan sistem kelak. Fasa reka bentuk sistem menggariskan model dan metode seperti berikut:

- a) Reka Bentuk Seni Bina Sistem
- b) Reka Bentuk dan Model Pangkalan Data
- c) Reka bentuk Antara Muka Pengguna
- d) Reka Bentuk Transaksi Sistem
 - i) Senario *Use Case*; atau
 - ii) Aliran Proses Sistem
- e) Penyediaan Spesifikasi Reka Bentuk Sistem (SDS)
- f) Spesifikasi Migrasi Data
- g) Spesifikasi Integrasi Sistem

Dokumen Rujukan kepada Fasa Reka Bentuk adalah **D03 Spesifikasi Keperluan Sistem**.

Dokumen Serahan kepada Fasa Reka Bentuk adalah seperti berikut:

- a) **D04 Spesifikasi Reka bentuk Sistem**
- b) **D06 Spesifikasi Migrasi Data**
- c) **D08 Spesifikasi Integrasi Sistem**

4.3 PENGLIBATAN PEMEGANG TARUH

Pemegang taruh utama yang perlu terlibat dalam reka bentuk sistem adalah juruanalisa sistem. Juruanalisa sistem memainkan peranan penting dalam memastikan reka bentuk sistem yang dibangunkan adalah selaras dengan keperluan satu-satu organisasi. Juruanalisa sistem juga bertanggungjawab untuk menterjemahkan keperluan bisnes dan sistem yang telah dikenal pasti dalam fasa-fasa sebelumnya kepada reka bentuk logikal yang jelas serta terperinci. Keupayaan ini adalah kritikal bagi memastikan sistem yang dibangunkan menepati

kepada ekspektasi organisasi berkenaan, berkualiti tinggi, serta mudah untuk dibangun, diselenggara dan digunakan pada masa akan datang.

Cadangan penglibatan kategori pemegang taruh adalah seperti berikut:

- a) **Juruanalisis Sistem** yang membangunkan reka bentuk sistem termasuk aspek fungsi, antara muka pengguna (UI/UX), pangkalan data dan arkitektur teknikal.
- b) **Subject Matter Expert (SME)** atau **Pemilik Proses** untuk menyemak reka bentuk fungsian, data dan antara muka sistem yang dibangun pada fasa ini.
- c) **Pengurus Projek** yang memastikan penyelarasan, pemantauan kemajuan dan pematuhan kepada skop serta jadual projek.

4.4 FAKTOR KEJAYAAN

Untuk memastikan aktiviti reka bentuk berjaya dilaksanakan, faktor-faktor kejayaan utama yang perlu dipertimbangkan sebelum dan semasa aktiviti dilaksanakan adalah seperti berikut:

- a) **D03 Spesifikasi Keperluan Sistem** adalah lengkap dan muktamad serta perlu mendapatkan pengesahan pemegang-pemegang taruh yang terlibat, khususnya SME atau/dan Pemilik Proses.
- b) Pasukan Juruanalisis Sistem mempunyai kompetensi dan kemahiran yang secukupnya untuk menterjemahkan keperluan bisnes dan sistem kepada reka bentuk logikal.
- c) **D04 Spesifikasi Reka Bentuk Sistem, D06 Spesifikasi Migrasi Data dan D08 Spesifikasi Integrasi Sistem** disemak dan mendapat pengesahan pemegang-pemegang taruh yang ditetapkan oleh organisasi.

4.5 REKA BENTUK SENI BINA SISTEM

4.5.1 KETERANGAN

Penentuan reka bentuk seni bina sistem adalah langkah penting dalam pembangunan sistem bagi memastikan setiap komponen perisian dirancang dengan teliti untuk memenuhi keperluan fungsian dan bukan fungsian yang telah ditentukan dari fasa-fasa yang sebelum. Melalui penyediaan seni bina sistem yang teratur dan terperinci, sistem dapat dibangun dengan ciri-ciri kualiti yang tinggi, merangkumi prestasi, keteguhan (*robustness*), pengagihan (*distributability*), kebolehsenggaraan dan ciri-ciri lain yang berkaitan.

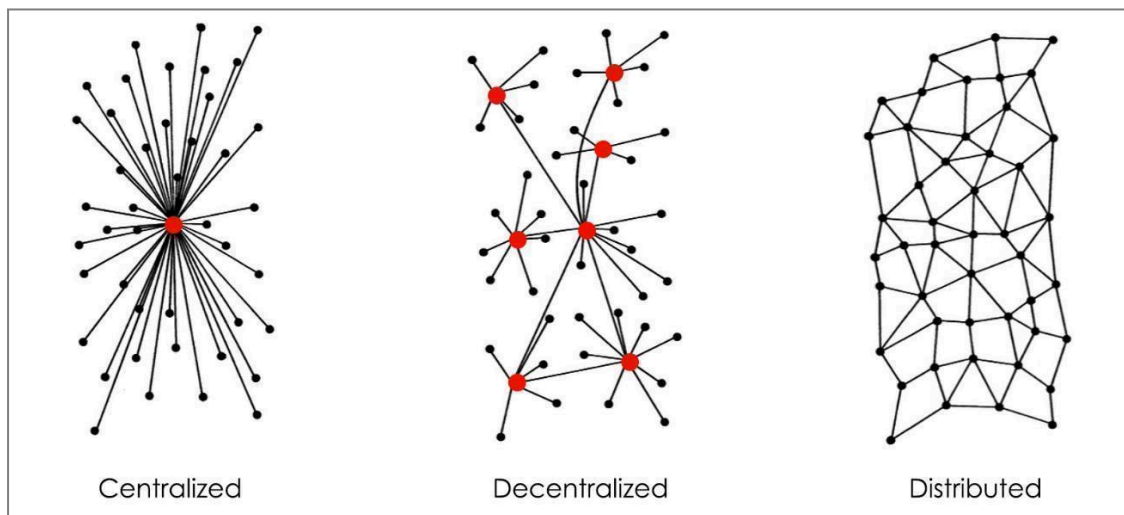
Definisi bagi reka bentuk seni bina sistem adalah penyusunan dan pengaturan struktur dan komponen perisian bagi sistem yang ingin dibangunkan. Reka bentuk bagi seni bina sistem menerangkan bagaimana sesuatu sistem disusun atur sebagai set komponen aplikasi dan pangkalan data yang saling berkomunikasi di antara satu sama lain.

4.5.2 SUSUN ATUR SENI BINA SISTEM

Susun atur seni bina sistem merujuk kepada pendekatan yang digunakan untuk menyusun atur dan menyelaraskan komponen-komponen aplikasi dan data dalam satu-satu sistem bagi memenuhi keperluan dan objektif yang telah ditentukan.

Secara amnya, jenis-jenis susun atur bagi seni bina sistem boleh dibahagikan kepada:

- a) Berpusat (*Centralized*)
- b) Tidak Berpusat (*Decentralized*)
- c) Teragih (*Distributed*)



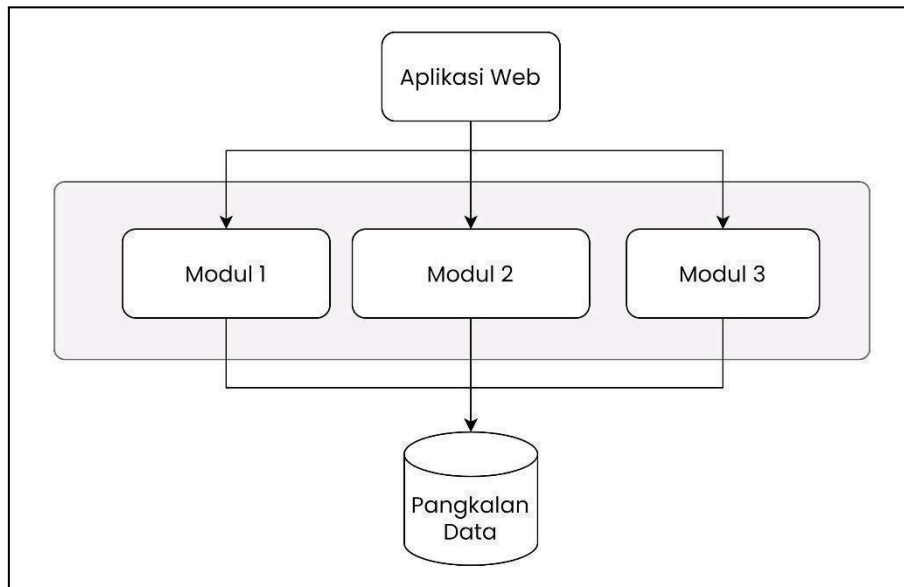
Rajah 4.2 : Susun Atur Seni Bina Sistem

4.5.3 JENIS-JENIS SENI BINA SISTEM

a) **Seni Bina Monolitik**

Seni bina monolitik adalah merujuk kepada pendekatan pembangunan aplikasi sebagai satu unit tunggal. Kebergantungan bagi setiap modul adalah tinggi dan komunikasi antara modul hanya berlaku secara dalaman bagi aplikasi tersebut. Secara amnya, seni bina monolitik hanya mempunyai pangkalan data tunggal

sebagai pangkalan data utama. Penyelenggaraan aplikasi lebih mudah dilaksanakan serta sesuai diadaptasi oleh pasukan pembangun yang kecil, juga pada peringkat awal pembangunan sesebuah aplikasi.



Rajah 4.3 : Seni Bina Monolitik Secara Am

Kelebihan dan kekurangan seni bina monolitik adalah :

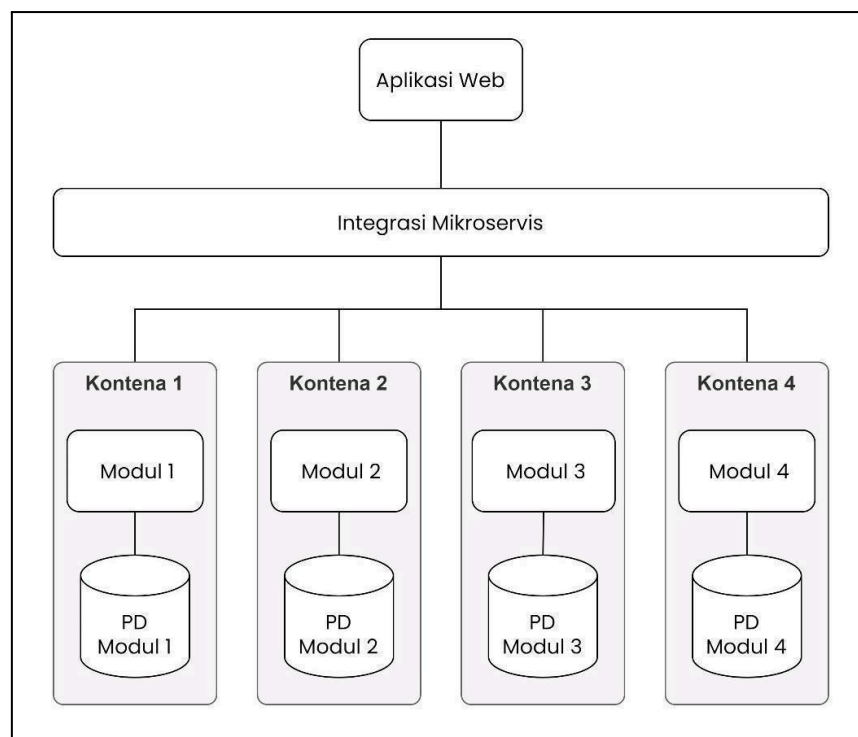
Jadual 4.1 : Jadual Kelebihan dan Kekurangan Seni Bina Monolitik

Aspek	Kelebihan	Kekurangan
Prestasi	Prestasi lebih baik untuk sistem kecil/serhana kerana interaksi komponen adalah minimum dan tidak terpengaruh.	Prestasi boleh merosot apabila sistem menjadi terlalu besar kerana semua komponen berkongsi sumber yang sama.
Pengujian & NyahPepijat	Lebih mudah diuji dan dinyahpepijat kerana semua komponen berada dalam satu kod asas dan tidak teragih.	Proses debugging boleh menjadi sukar apabila kod menjadi terlalu besar dan kompleks.
Kebolehskaaan	Peningkatan kapasiti (vertical scaling) lebih mudah dan ringkas kerana kebergantungan kepada komponen luar adalah rendah.	Pensakalan selari (horizontal scaling) adalah terhad berbanding arkitektur moden seperti mikroservis.
Peningkatan Struktur Sistem	Struktur berpusat memudahkan perubahan kecil dilakukan dengan pantas sebelum sistem menjadi terlalu besar.	Penambahbaikan menjadi rumit apabila saiz, transaksi dan kompleksiti meningkat.

Penyelenggaraan	Penyepaduan komponen yang rapat memudahkan pemahaman sistem pada peringkat awal pembangunan.	Komponen yang tightly coupled menyebabkan perubahan pada satu bahagian boleh menjejaskan bahagian lain; penyelenggaraan menjadi reaktif.
Teknologi	Pembangunan lebih mudah pada awalnya kerana hanya menggunakan satu set teknologi yang seragam.	Sukar memperkenalkan teknologi baharu kerana bergantung kepada satu stack teknologi yang statik.

b) Seni Bina Mikroservis

Seni bina mikroservis pula adalah pendekatan pembangunan aplikasi yang mengasingkan modul atau komponen sesebuah aplikasi kepada mikroservis-mikroservis yang telah dikenalpasti berdasarkan proses, fungsi, atau perkhidmatan yang tertentu. Mikroservis-mikroservis ini mempunyai kebergantungan yang rendah antara satu sama lain yang membolehkan mikroservis ini beroperasi secara sendiri. Secara amnya, setiap mikroservis mempunyai pangkalan data yang tersendiri, dan komunikasi antara mikroservis ini adalah menggunakan API. Ini membolehkan setiap mikroservis ini dibangunkan menggunakan teknologi yang berbeza mengikut keperluan khusus sesuatu perisian.



Rajah 4.4 : Seni Bina Mikroservis Secara Am

Kelebihan utama seni bina mikroservis adalah:

Jadual 4.2 : Jadual Kelebihan dan Kekurangan Seni Bina Mikroservis

Aspek	Kelebihan	Kekurangan
Teknologi	Fungsi sistem boleh dibangunkan menggunakan teknologi berbeza mengikut keperluan. Komponen yang <i>loosely coupled</i> memudahkan fungsi ditambah baik, dipinda atau dikeluarkan tanpa menjejaskan keseluruhan sistem.	Memerlukan kepakaran teknikal pelbagai teknologi, menjadikan pengurusan dan penyelenggaraan lebih kompleks.
Kebolehskalaan	Setiap servis boleh diskalakan secara individu tanpa menjejaskan keseluruhan aplikasi, membolehkan penggunaan sumber lebih efisien.	Penskalaan yang melibatkan banyak servis memerlukan orkestrasi kompleks dan penalaan sistem yang teliti.
Daya Tahan	Tiada <i>single point of failure</i> . Jika satu servis gagal, servis lain kekal beroperasi.	Perlu mekanisme <i>fault tolerance</i> tambahan seperti <i>circuit breaker</i> , <i>retry</i> dan <i>timeout</i> yang meningkatkan kerumitan.
Aktiviti Pembangunan	Pembangunan boleh dibahagi kepada pasukan kecil yang berdikari, mempercepatkan pembangunan, inovasi dan kitaran pelepasan (<i>release cycle</i>).	Koordinasi antara pasukan sukar, memerlukan dokumentasi jelas dan integrasi yang lebih teliti.
Kependaman / Latency	Komunikasi dalaman boleh dioptimumkan menggunakan protokol ringan seperti gRPC yang mengurangkan beban permintaan.	Masa tindak balas meningkat kerana komunikasi rangkaian antara servis yang teragih.
Pengujian & Nyahpejijat	Ujian unit lebih mudah kerana setiap servis berdiri sendiri dan berskala kecil.	Pengujian menyeluruh dan debugging lebih mencabar kerana melibatkan banyak komponen teragih.
Sumber & Kos	Menggunakan sumber secara fleksibel kerana setiap servis boleh menggunakan konfigurasi yang paling efisien untuk fungsinya.	Memerlukan infrastruktur, pemantauan dan automasi lebih banyak, menyebabkan kos naik.

c) Seni Bina *Serverless*

Seni bina *serverless* merupakan pendekatan reka bentuk berasaskan pengkomputeran awan dimana kod sumber sesuatu aplikasi dijalankan pada

persekitaran pengkomputeran awan yang diuruskan oleh pembekal perkhidmatan pengkomputeran awan. Pendekatan ini membolehkan pasukan pembangun memberi tumpuan sepenuhnya kepada pembangunan kod pengaturcaraan, tanpa perlu menguruskan pelayan dan infrastruktur yang berkaitan.

Kelebihan dan kekurangan seni bina *serverless* adalah:

Jadual 4.3 : Jadual Kelebihan dan Kekurangan Seni Bina *Serverless*

Bil.	Aspek	Kelebihan	Kekurangan
1.	Pengurusan Infrastruktur	Tidak perlu mengurus pelayan secara manual kerana semuanya dikendalikan penyedia awan.	Kawalan terhadap infrastruktur adalah terhad. Sukar melakukan konfigurasi khusus pada pelayan atau OS.
2.	Penskalaan (<i>Scalability</i>)	Autoscaling berlaku secara automatik mengikut beban permintaan dan tidak perlu merancang kapasiti pelayan.	Penskalaan mendadak boleh meningkatkan kos tanpa disedari. Boleh berlaku " <i>cold start</i> " yang melambatkan tindak balas pada fungsi tertentu.
3.	Kos	Kos berasaskan penggunaan sebenar (<i>pay-as-you-go</i>) dan tiada kos tetap untuk pelayan idle.	Sukar menganggar kos dalam trafik yang tidak konsisten. Boleh menjadi lebih mahal berbanding pelayan biasa jika panggilan fungsi terlalu banyak.
4.	Reka Bentuk & Corak Arkitektur	Boleh digabung dengan corak mikroservis atau monolitik. Fleksibel mengikut keperluan sistem.	Tidak semua corak arkitektur sesuai dengan fungsi <i>Serverless</i> (FaaS) yang bersifat stateless.
5.	Prestasi	Berprestasi tinggi untuk aplikasi event-driven dan beban berselang.	Risiko latency " <i>cold start</i> " bergantung pada platform. Had masa eksekusi tertentu bagi fungsi (bergantung penyedia awan).
6.	Keselamatan	Kebanyakan aspek keselamatan infrastruktur dikendalikan penyedia awan.	Kawalan keselamatan terhadap pada lapisan aplikasi sahaja. Risiko vendor <i>lock-in</i> kerana bergantung kepada polisi keselamatan penyedia awan.

7.	Integrasi & Ekosistem	Integrasi mudah dengan servis lain dalam platform awan (contoh: <i>queue</i> , <i>database</i> , <i>storage</i>).	Integrasi antara platform berbeza boleh menjadi lebih kompleks. Penggunaan teknologi mungkin terhad kepada apa yang disokong penyedia awan.
----	----------------------------------	--	--

d) Seni Bina *Cloud-Native*

Seni bina *cloud-native* merujuk kepada pendekatan reka bentuk seni bina aplikasi yang memanfaatkan kelebihan yang ditawarkan oleh teknologi pengkomputeran awan. Bagi mencapai tujuan ini, sesuatu aplikasi perlulah dibangunkan dengan mengadaptasi seni bina mikroservis, mengguna pakai teknologi yang menyokong seni bina pengkomputeran awan seperti *container*, di samping melaksanakan kaedah pembangunan aplikasi secara *agile* melalui amalan *DevOps* dan *CI/CD*.

Kelebihan dan kekurangan seni bina *cloud-native* adalah:

Jadual 4.4 : Jadual Kelebihan dan Kekurangan Seni Bina *Cloud-Native*

Bil.	Aspek	Kelebihan	Kekurangan
1.	Skalabiliti	Skalabiliti automatik dan elastik mengikut beban penggunaan.	Kos boleh meningkat jika auto-scaling tidak dikawal dengan baik.
2.	Ketahanan (<i>Resilience</i>)	Reka bentuk mikroservis dan kontena memudahkan sistem pulih daripada kegagalan.	Memerlukan pemantauan yang lebih kompleks untuk memastikan setiap komponen berjalan baik.
3.	Kelajuan Pembangunan	Penyebaran (<i>deployment</i>) lebih pantas melalui <i>CI/CD</i> dan automasi awan.	Memerlukan perubahan budaya pembangunan serta kemahiran <i>DevOps</i> yang lebih tinggi.
4.	Fleksibiliti Teknologi	Setiap komponen boleh menggunakan teknologi atau bahasa yang paling sesuai.	Kepelbagaian teknologi sukar diurus jika tiada standardisasi dalam pasukan.

5.	Penggunaan Sumber	Penggunaan sumber lebih efisien melalui orkestrasi (contoh: Kubernetes).	Konfigurasi orkestrasi adalah kompleks dan memerlukan kepakaran.
6.	Kos Infrastruktur	Tidak perlu membeli atau mengurus perkakasan fizikal.	Bergantung sepenuhnya kepada penyedia awan, risiko vendor <i>lock-in</i> meningkat.
7.	Kebolehskalaan Pasukan	Mikroservis membenarkan pasukan bekerja secara selari pada komponen berbeza.	Penyelarasannya menjadi sukar jika terlalu banyak komponen kecil.
8.	Pemerhatian (<i>Observability</i>)	Mudah dilaksanakan menggunakan alat <i>cloud-native</i> seperti logging, metrics & tracing.	Data log dan metrics yang banyak boleh menaikkan kos storan dan analisis.

e) Model Seni Bina Lapisan (*Layered / N-Tier*) sebagai Struktur Asas

Seni bina lapisan merupakan satu reka bentuk seni bina yang sering digunakan dalam dunia pembangunan sistem aplikasi disebabkan oleh strukturnya yang ringkas dan mudah untuk difahami. Model seni bina lapisan merupakan struktur asas yang digunakan dalam penyediaan seni bina sistem, sama ada seni bina tersebut mengadaptasikan teknologi monolitik ataupun mikroservis.

Secara amnya, seni bina ini mengandungi beberapa lapisan yang berbeza, tetapi bilangan lapisannya boleh ditambah atau dikurangkan mengikut kesesuaian, kompleksiti dan perincian diagram seni bina yang ingin dibangunkan. Contoh lapisan-lapisan yang terlibat adalah seperti berikut:

- i) Lapisan Persembahan
- ii) Lapisan Pertengahan
- iii) Lapisan Data
- iv) Lapisan Integrasi

4.5.4 OBJEKTIF

Menyediakan seni bina sistem aplikasi yang terdiri daripada komponen-komponen perisian, iaitu aplikasi dan pangkalan data, berpandukan kepada keperluan-keperluan yang telah diperolehi dalam fasa-fasa sebelum.

4.5.5 LANGKAH-LANGKAH MEREKA BENTUK SENI BINA SISTEM

Langkah 1 : Kenal Pasti Keperluan Sistem

Rujuk dokumen-dokumen **D03 Spesifikasi Keperluan Sistem** bagi mendapatkan keperluan fungsian dan bukan fungsian yang telah diperolehi daripada pemegang-pemegang taruh.

Langkah 2 : Laksanakan Pertimbangan Reka bentuk

Pertimbangan reka bentuk perlu dilakukan berdasarkan keperluan yang telah didapati semasa kajian keperluan sistem. Penilaian ke atas pertimbangan reka bentuk perlu mengambil kira perkara-perkara seperti berikut:

a) Penentuan Jenis Aplikasi

Pemilihan jenis aplikasi yang sesuai adalah penting bagi proses reka bentuk aplikasi. Pemilihan adalah bergantung kepada keperluan keupayaan infrastruktur dan kemampuan teknologi sedia ada yang telah dikenal pasti. Contoh jenis-jenis aplikasi adalah seperti:

- i) aplikasi mudah alih;
- ii) client-server application.
- iii) portal.
- iv) aplikasi web tradisional; atau
- v) aplikasi web moden (*Rich Internet Application*).

b) Penentuan Strategi Pelaksanaan

Penentuan strategi pelaksanaan melibatkan langkah-langkah untuk membuat perancangan bagaimana sistem akan dilaksanakan dengan mengambil kira spesifikasi persekitaran yang diperlukan. Setiap persekitaran pelaksanaan sistem mempunyai kekangan tersendiri, seperti keperluan untuk mengasingkan komponen perisian secara fizikal, pelaksanaan pada pelayan yang berbeza, dan kekangan konfigurasi pada peranti-peranti tertentu. Strategi pelaksanaan perlu memastikan keseimbangan antara keperluan sistem dan kemampuan perkakasan yang sedia ada di pasaran.

c) **Penentuan Ciri-ciri Kualiti**

Ciri-ciri kualiti atau keperluan bukan fungsian seperti prestasi, modulariti, kebolehskalaan, keselamatan, capaian, ketersediaan dan kebolehgunaan merupakan faktor-faktor yang perlu dipertimbangkan dalam penyediaan reka bentuk seni bina. Pemilihan ciri-ciri kualiti yang ingin diutamakan adalah bergantung kepada keperluan yang telah diperolehi dan strategi pelaksanaan sistem yang akan dijalankan kelak. Pemilihan keutamaan ini perlu dilakukan terlebih dahulu sebelum sebarang reka bentuk seni bina dibangunkan bagi mengelakkan percanggahan di antara ciri-ciri kualiti yang perlu diutamakan dalam satu-satu sistem. Sebagai contoh, keutamaan yang berlebihan kepada ciri-ciri keselamatan berkemungkinan besar boleh membebankan prestasi dan menyukarkan pengguna membuat capaian kepada sistem.

Langkah 3 : Kenal Pasti Susun Atur dan Jenis Seni Bina Aplikasi

Penjelasan ringkas berkenaan dengan susun atur dan jenis seni bina aplikasi adalah seperti berikut:

d) **Susun Atur Seni Bina Sistem**

Penentuan adalah berdasarkan kepada strategi pelaksanaan sistem sama ada susun atur tersebut lebih sesuai disediakan secara berpusat, tidak berpusat, dan teragih.

e) **Jenis Seni Bina Aplikasi**

Pemilihan gaya dan corak adalah bergantung kepada keperluan bukan fungsian sistem yang telah ditetapkan, sama ada memerlukan pendekatan seni bina monolitik, mikroservis atau mana-mana seni bina yang bersesuaian.

Langkah 4 : Tentukan Struktur Asas Seni Bina

Model lapisan atau *N-Tier/Layered* sering digunakan sebagai struktur asas dalam membangunkan seni bina sistem. Melalui pendekatan ini, komponen-komponen perisian dikumpulkan ke dalam lapisan-lapisan khusus seperti lapisan persembahan, integrasi, pertengahan dan data. Pengumpulan komponen-komponen ini dilakukan berdasarkan kepada fungsi serta peranan setiap komponen yang terlibat, dengan mengambil kira jenis, gaya, dan corak seni bina yang telah ditentukan.

Langkah 5 : Sediakan Seni Bina Sistem

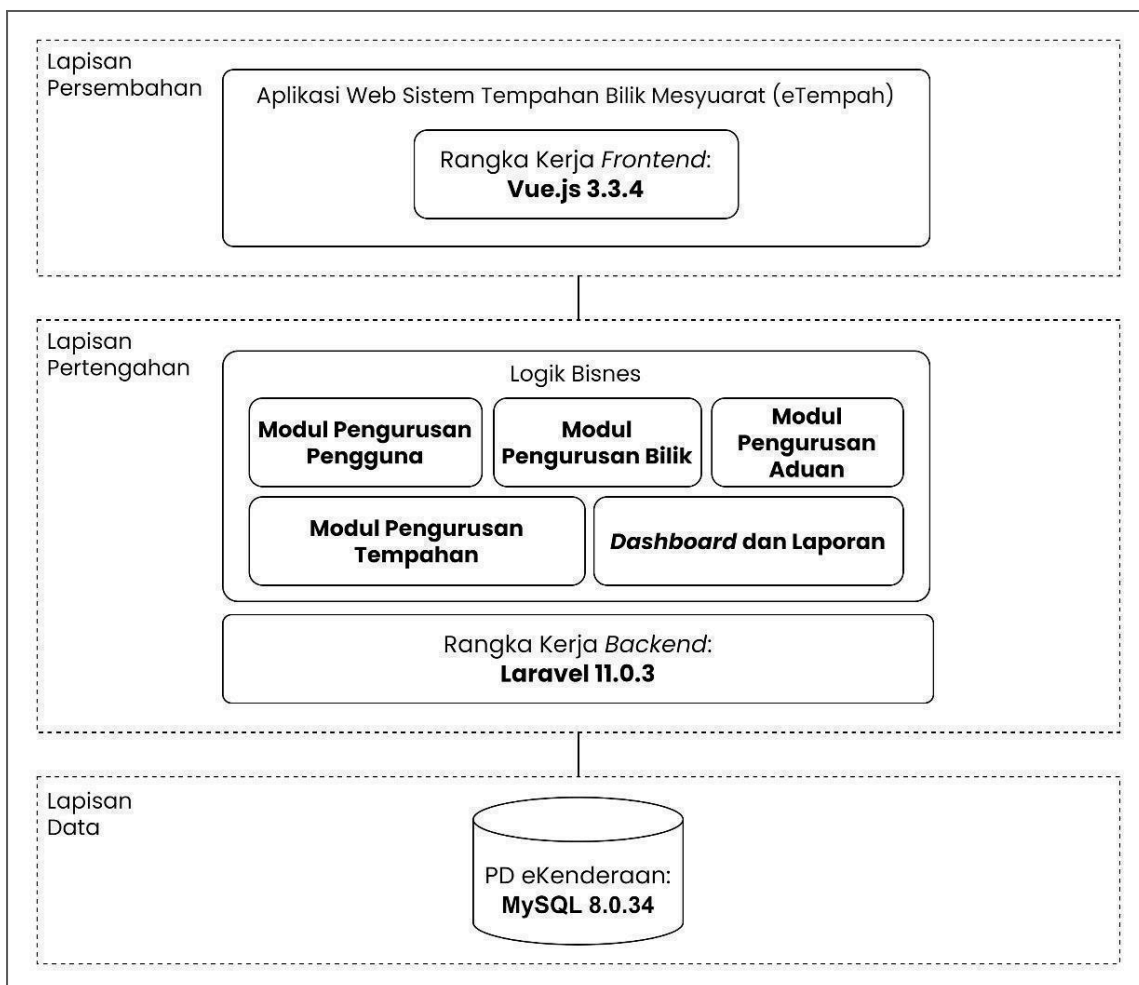
Seni bina sistem merupakan struktur logikal komponen-komponen perisian yang menyeluruh, termasuk aplikasi dan pangkalan data, serta hubungannya di antara satu komponen dengan komponen-komponen yang lain. Penyediaan seni bina ini adalah

berdasarkan kepada gabungan susun atur dan jenis seni bina aplikasi seperti yang telah dinyatakan dalam penerangan-penerangan sebelum.

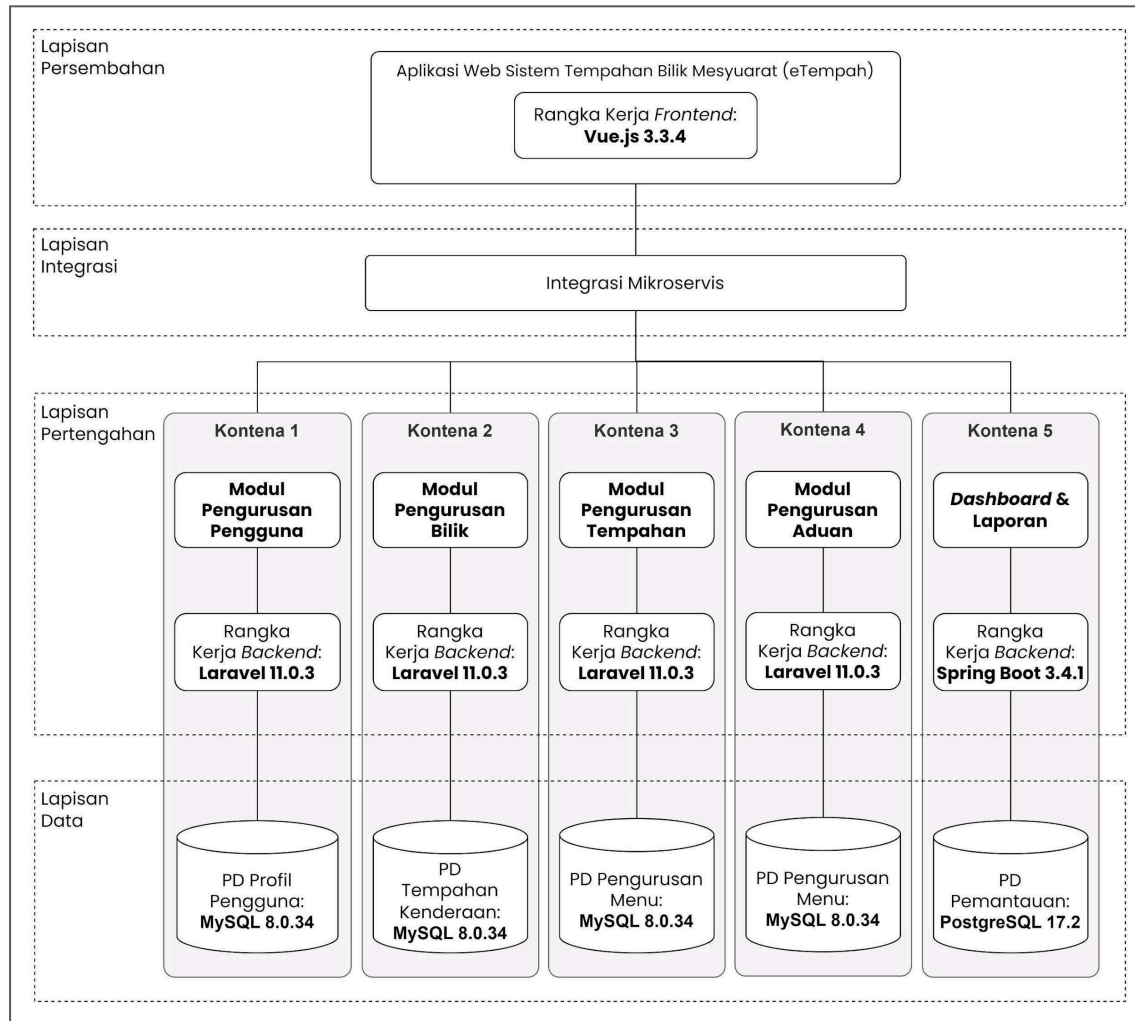
Perincian atau granulariti seni bina ini adalah tertakluk kepada kompleksiti sistem yang ingin dibangunkan, teknologi, dan produk yang akan diguna pakai berpandukan kepada keperluan projek masing-masing. Penyediaan seni bina sistem boleh dilaksanakan melalui perincian seperti berikut:

a) Seni Bina Sistem Secara Umum (*High Level*)

Seni bina sistem secara umum memberikan gambaran abstrak dan generik terhadap komponen-komponen perisian di mana ia hanya memfokuskan kepada komponen-komponen utama (termasuk produk yang diguna pakai), seperti rangka kerja aplikasi dan sistem pengurusan pangkalan data, serta interaksinya di antara satu sama lain tanpa memerlukan perincian yang lanjut. Berikut adalah contoh-contoh penyediaan seni bina sistem secara umum (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) bagi sistem aplikasi yang berteraskan kepada teknologi monolitik dan mikroservis.



Rajah 4.6: Seni Bina Sistem (Monolitik) Secara Am Bagi Sistem eTempah



Rajah 4.7: Seni Bina Sistem (Mikroservis) Secara Am Bagi Sistem eTempah

b) **Arkitektur Sistem Terperinci (Low Level)**

Arkitektur sistem terperinci memberikan gambaran komponen-komponen perisian (termasuk produk yang diguna pakai) secara komprehensif dan lebih mendalam. Setiap lapisan diperincikan dengan komponen-komponen yang melibatkan dan menyokong kepada pengoperasian logik aplikasi dan pangkalan data. Perincian komponen-komponen yang terkandung bagi setiap lapisan adalah seperti berikut:

i) **Komponen Terperinci Lapisan Persembahan**

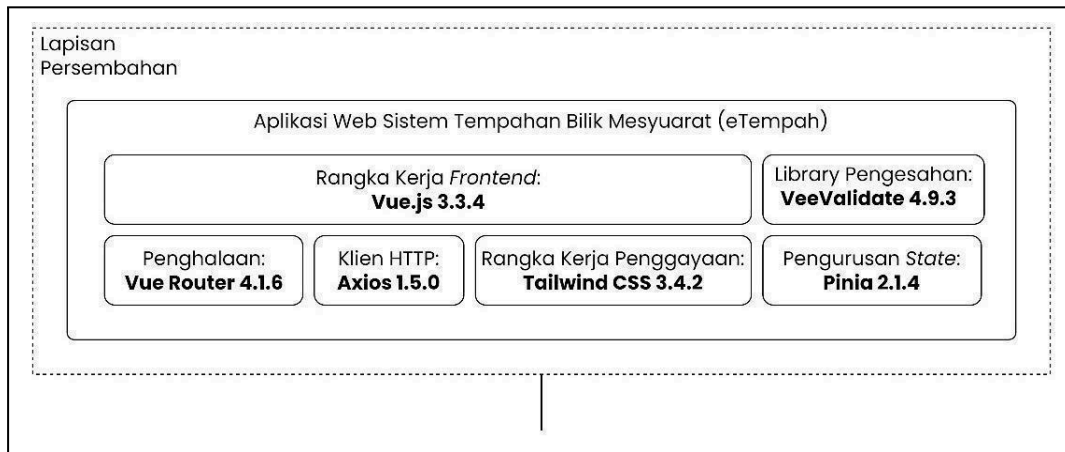
Komponen-komponen perisian di dalam lapisan persembahan bagi arkitektur monolitik dan mikroservis secara lazimnya adalah sama oleh kerana kedua-duanya bertujuan untuk menyediakan UI yang konsisten, serta berfungsi sebagai penghubung antara pengguna dengan logik aplikasi tanpa mengira bagaimana pemprosesan sistem diuruskan di latar belakang (backend) sistem. Komponen-komponen perisian yang terkandung di dalam lapisan persembahan

secara lazimnya (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) adalah seperti berikut:

Jadual 4.5: Komponen-komponen Monolitik Lapisan Persembahan

Bil.	Komponen	Penerangan
1	Rangka Kerja <i>Frontend</i>	Rangka kerja untuk membina antara muka pengguna (UI) aplikasi yang dinamik. Ia berfungsi untuk: i) menerima tindakan dan interaksi dari pengguna ke atas antaramuka pengguna, seperti mengklik butang simpan. ii) mengemaskini paparan UI secara dinamik sekiranya terdapat perubahan data.
2	Penghalaan (Routing)	Mengurus navigasi antara halaman di dalam aplikasi berdasarkan alamat URL fail aplikasi yang terlibat.
3	Pengurusan State (State Management)	Menyimpan, mengemaskini dan mengurus data yang disimpan dalam storan <i>cache</i> untuk digunakan dalam satu-satu antaramuka aplikasi.
4	Klien HTTP	Menerima atau menghantar data yang diterima dari UI ke pelayan pangkalan data.
5	<i>Library</i> Pengesahan	Mengendalikan pengesahan input dalam borang web yang telah dibangunkan di dalam UI sistem.
6	Rangka Kerja Penggayaan (Styling Framework)	Menyediakan kelas CSS sedia ada untuk menggayakan elemen UI dengan cepat dan mudah.

Berikut adalah contoh penyusunan komponen-komponen aplikasi di lapisan persembahan bagi arkitektur monolitik dan mikroservis (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) bagi sistem eTempah:



Rajah 4.8: Komponen Lapisan Persembahan Terperinci (Monolitik dan Mikroservis) Bagi Sistem eTempah

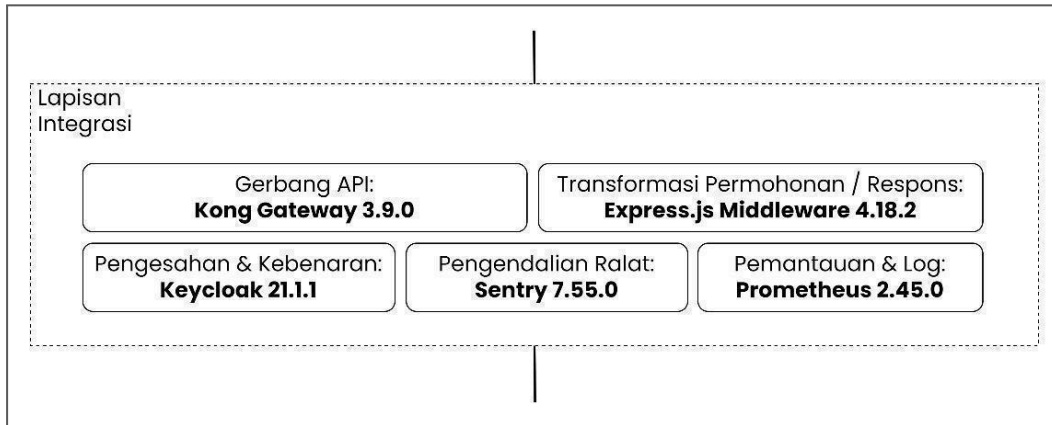
ii) Komponen Terperinci Lapisan Integrasi

Berbeza dengan arkitektur monolitik, arkitektur mikroservis merangkumi lapisan integrasi yang merupakan lapisan tambahan bagi menghubungkan komponen yang berada kontena-kontena perisian yang berbeza. Komponen-komponen yang terkandung di dalam lapisan secara lazimnya (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) adalah seperti berikut:

Jadual 4.6: Komponen-komponen Mikroservis Lapisan Integrasi

Bil.	Komponen	Penerangan
1	Gerbang API	Mengurus penghalauan API, pengagregatan, dan komunikasi antara lapisan.
2.	Pengesahan dan Kebenaran	Melindungi identiti pengguna dan menguruskan kawalan akses.
3.	Transformasi Permohonan dan Respons	Mengubah suai permohonan masuk atau respon keluar semasa maklumat tersebut melalui gerbang API.
4.	Pengendalian Ralat	Memantau, mengenal pasti, mengurus dan memitigasi isu komunikasi di antara komponen-komponen mikroservis.
5.	Pemantauan dan Log	Memantau prestasi sistem serta merekodkan transaksi atau ralat berlaku.

Berikut adalah contoh penyusunan komponen-komponen aplikasi di lapisan integrasi mikroservis (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) bagi sistem eTempah:



Rajah 4.9: Komponen Lapisan Integrasi Terperinci (Mikroservis) Bagi Sistem eTempah

iii) Komponen Terperinci Lapisan Pertengahan

Komponen-komponen perisian yang terkandung di dalam lapisan pertengahan secara lazimnya (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) adalah seperti berikut:

Jadual 4.7: Komponen-komponen Monolitik Lapisan Pertengahan

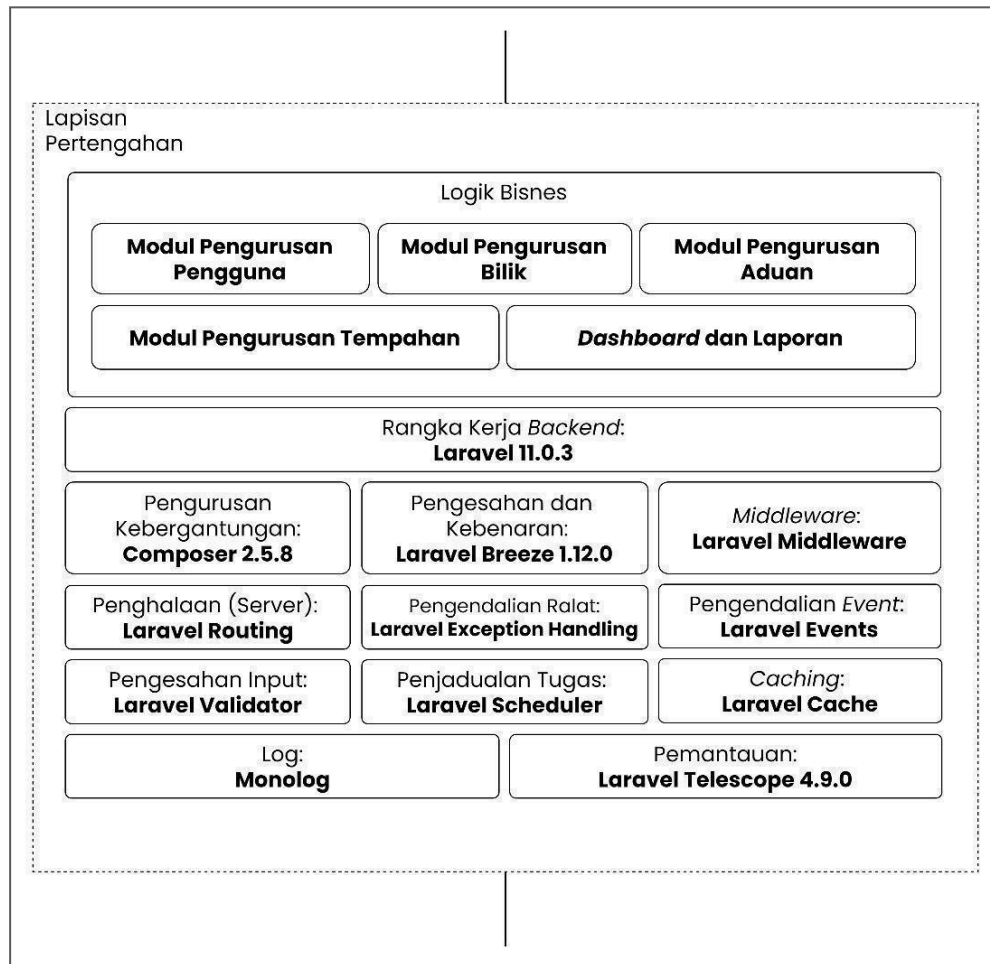
Bil.	Komponen	Penerangan
1	Logik Bisnes	Fungsi-fungsi utama sistem yang akan dirangkumkan sama ada terdiri daripada modul atau/dan submodul.
2	Rangka Kerja <i>Backend</i>	Menyediakan kerangka kerja yang cekap dan berstruktur untuk mengurus logik aplikasi, pangkalan data, pengesahan, dan komunikasi di antara komponen perisian di lapisan pertengahan dengan komponen-komponen di lapisan persembahan, integrasi dan data.
3	Pengurusan Kebergantungan	Menguruskan kebergantungan aplikasi dengan pakej perisian dan <i>library</i> pihak ketiga.

4	Pengesahan dan Kebenaran	Menyediakan rangka kerja pengesahan dan kebenaran pengguna yang ringkas dan mudah menggunakan templat-templat sedia ada.
5	<i>Middleware</i>	Mekanisme perantara yang memproses permintaan dan respons HTTP untuk melaksanakan tugas seperti pengesahan, penentuan kebenaran capaian atau pengubahsuaian data sebelum dipanjangkan untuk logik aplikasi selanjutnya.
6	Penghalaan (Server)	Menghubungkan alamat URL kepada fail-fail yang mengandungi logik aplikasi yang diminta oleh pengguna.
7	Pengendalian Ralat	Menguruskan ralat dalam aplikasi dengan mengenal pasti, merekod dan memberikan respons yang sesuai kepada pengguna.
8	Pengendalian <i>Event</i>	Mengenal pasti dan bertindak balas ke atas satu-satu kejadian (event) yang berlaku seperti menghantar emel setelah pengguna selesai mendaftar akaun.
9	Pengesahan Input	Memastikan input yang diterima adalah sah mengikut peraturan dan format bisnes yang telah ditetapkan sebelum mula untuk diproses.
10	Penjadualan Tugas	Mengautomasi penjadualan satu-satu tugas berkala seperti pengiriman e-mel secara pukal kepada pengguna atau pembersihan data secara harian.
11	<i>Caching</i>	Meningkatkan prestasi aplikasi melalui penyimpanan data secara sementara.
12	Log	Merekodkan acara dan ralat aplikasi untuk tujuan penyahpejitan (debugging) dan audit.
13	Pemantauan	Memantau sistem secara keseluruhan untuk mengenal pasti ralat, mengoptimumkan prestasi, dan memastikan aplikasi berjalan dengan lancar.

- **Komponen Terperinci Lapisan Pertengahan Monolitik**

Komponen-komponen lapisan pertengahan bagi arkitektur monolitik pada kebiasaannya berfungsi secara berpusat untuk mengendalikan logik bisnes dan pengoperasian di latar belakang sistem. Di samping itu, terdapat beberapa komponen tambahan di lapisan ini berbanding dengan lapisan pertengahan arkitektur mikroservis oleh kerana tiada komponen-komponen perisian yang diasingkan dan terletak di lapisan integrasi. Berikut adalah

contoh penyusunan komponen-komponen di lapisan pertengahan monolitik (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) bagi sistem eTempah:



Rajah 4.10: Komponen Lapisan Pertengahan Terperinci (Monolitik) Bagi Sistem eTempah

- **Komponen Terperinci Lapisan Pertengahan Mikroservis**

Komponen-komponen aplikasi di lapisan pertengahan mikroservis perlu ditunjukkan secara berasingan (logikal) untuk setiap kontena yang terlibat. Oleh kerana sebahagian daripada komponen telah dirangkumkan di lapisan integrasi bagi tujuan penyelerasan, jumlah komponen dalam lapisan pertengahan arkitektur ini berbeza dengan komponen-komponen di arkitektur monolitik. Berikut adalah contoh penyusunan komponen-komponen di lapisan pertengahan mikroservis (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) bagi sistem eTempah:



Rajah 4.11: Komponen Lapisan Pertengahan Terperinci (Mikroservis) Bagi Sistem eTempah

iv) Komponen Terperinci Lapisan Data

Komponen-komponen di dalam lapisan data boleh dibahagikan kepada dua (2) bahagian, iaitu:

- **Pangkalan Data**

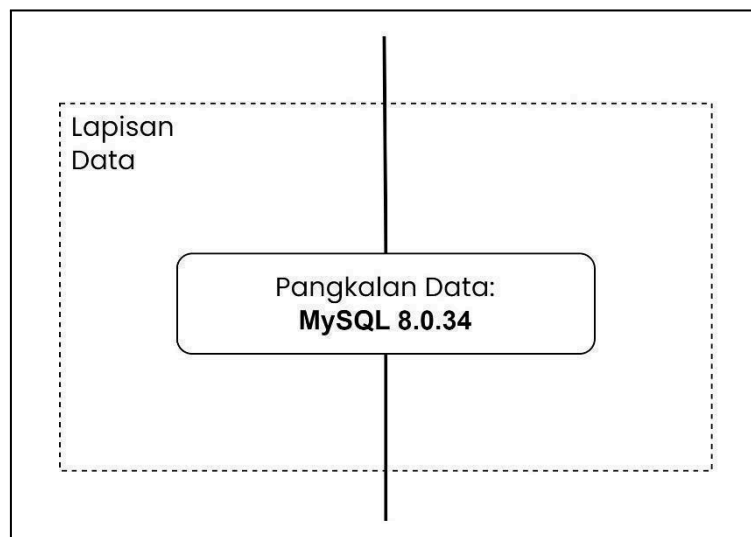
Komponen pangkalan data boleh direka bentuk sama ada berfungsi secara berpusat (centralized) ataupun teragih (distributed) bagi arkitektur monolitik mahupun mikroservis. Berikut merupakan perbezaan di antara kedua-dua jenis pangkalan data tersebut:

Jadual 4.8: Perbezaan Pangkalan Data Berpusat Dan Teragih

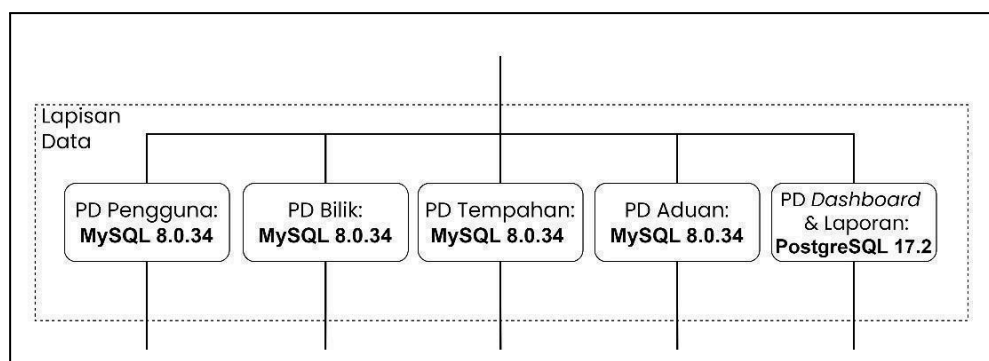
Perkara	Berpusat	Teragih
Definisi	Merupakan pangkalan data di mana fail-failnya disimpan dan diselenggara di satu lokasi storan.	Merupakan pangkalan data di mana sebahagian atau keseluruhan fail-failnya disimpan dan diselenggara di lokasi-lokasi berasingan.
Kekuatan	• Mudah untuk diurus dan diselenggara. • Konsistensi data yang tinggi. • Penguatkuasaan kawalan yang lebih mudah. • Kos lebih rendah berbanding pangkalan data teragih.	• Kebolehskaan dan ketersediaan yang lebih fleksibel dan tinggi. • Akses (prestasi) yang lebih pantas. • Toleransi kegagalan lebih baik kerana lokasinya yang teragih.

Kelemahan	• Titik kegagalan tunggal. • Potensi berlakunya kesesakan adalah tinggi. • Keupayaan penskalaan adalah terhad.	• Pengurusan dan penyelenggaraan menjadi lebih kompleks. • Risiko ketidakselarasan data boleh berlaku. • Kos yang lebih tinggi kerana memerlukan infrastruktur tambahan.
-----------	--	--

Berikut adalah contoh penyusunan komponen-komponen pangkalan data berpusat dan teragih (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) bagi sistem eTempah:



Rajah 4.12: Komponen Pangkalan Data Berpusat Bagi Sistem eTempah



Rajah 4.13: Komponen Pangkalan Data Teragih Bagi Sistem eTempah

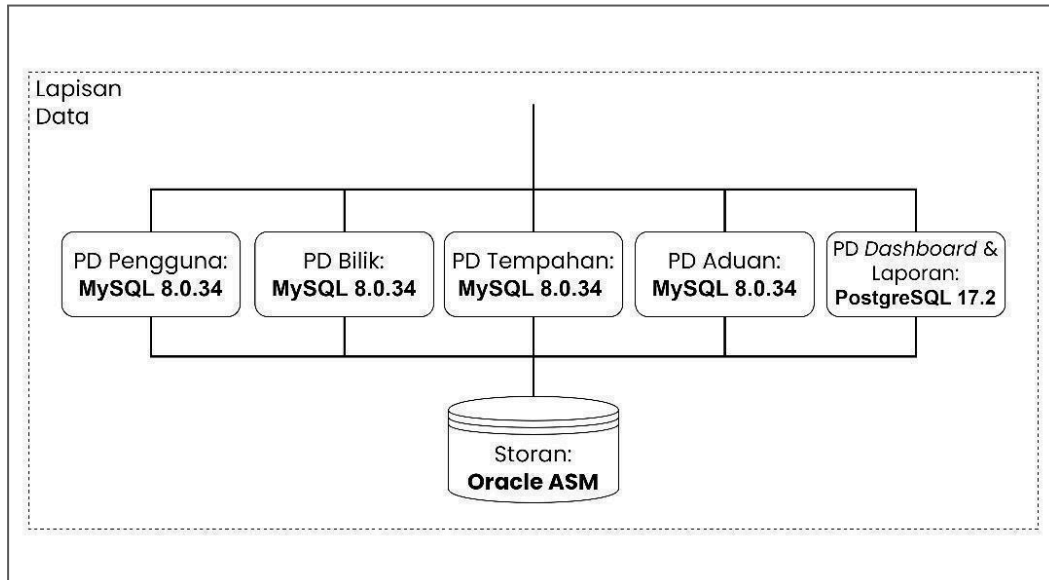
- **Storan**

Komponen storan lazimnya boleh direka bentuk sama ada berfungsi secara shared storage ataupun shared nothing bagi arkitektur monolitik mahupun mikroservis. Berikut merupakan perbezaan di antara kedua-dua jenis storan tersebut:

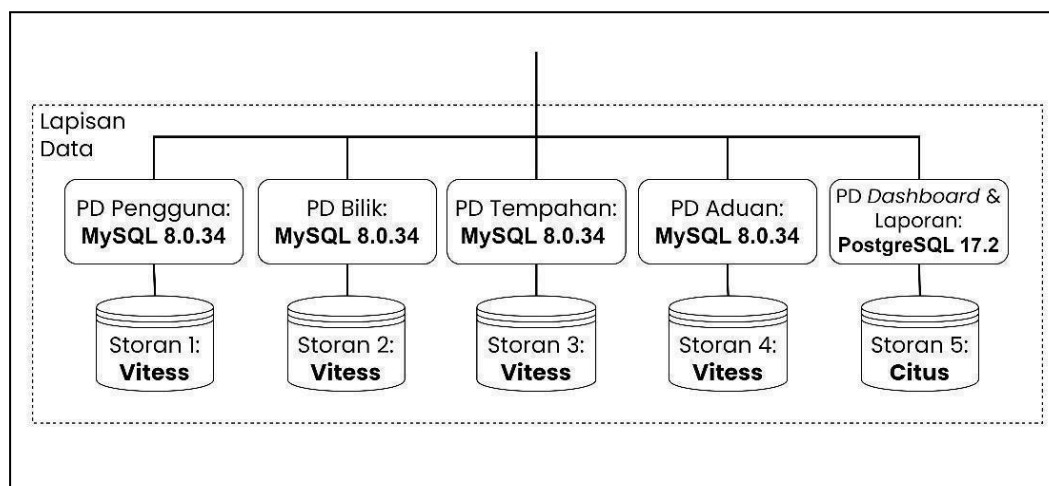
Jadual 4.9: Perbezaan *Shared Storage* Dan *Shared Nothing*

Perkara	<i>Shared Storage</i>	<i>Shared Nothing</i>
Definisi	Semua pelayan pangkalan data berkongsi storan berpusat untuk mengakses data yang sama.	Setiap pelayan pangkalan data mempunyai storan tersendiri tanpa berkongsi dengan pelayan yang lain. Storan boleh mengandungi salinan data yang penuh ataupun separa secara <i>partitioning</i> .
Kekuatan	• Konsistensi data lebih mudah untuk dikawal. • Pengurusan data lebih mudah. • Kos lebih rendah kerana berkongsi storan.	• Kebolehskalaan yang tinggi dan fleksibel • Tiada titik kegagalan tunggal • Prestasi lebih pantas kerana tiada pertindihan.
Kelemahan	• Titik kegagalan tunggal pada storan • Latensi tinggi kerana semua pangkalan data berkongsi storan yang sama • Penskalaan storan yang terhad.	• Pengurusan data yang lebih kompleks • Risiko ketidakselarasan data boleh berlaku. • Memerlukan kaedah replikasi data yang cekap. • Kos yang lebih tinggi kerana memerlukan infrastruktur tambahan.

Berikut adalah contoh penyusunan komponen-komponen storan shared disk dan shared nothing (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) bagi sistem eTempah:



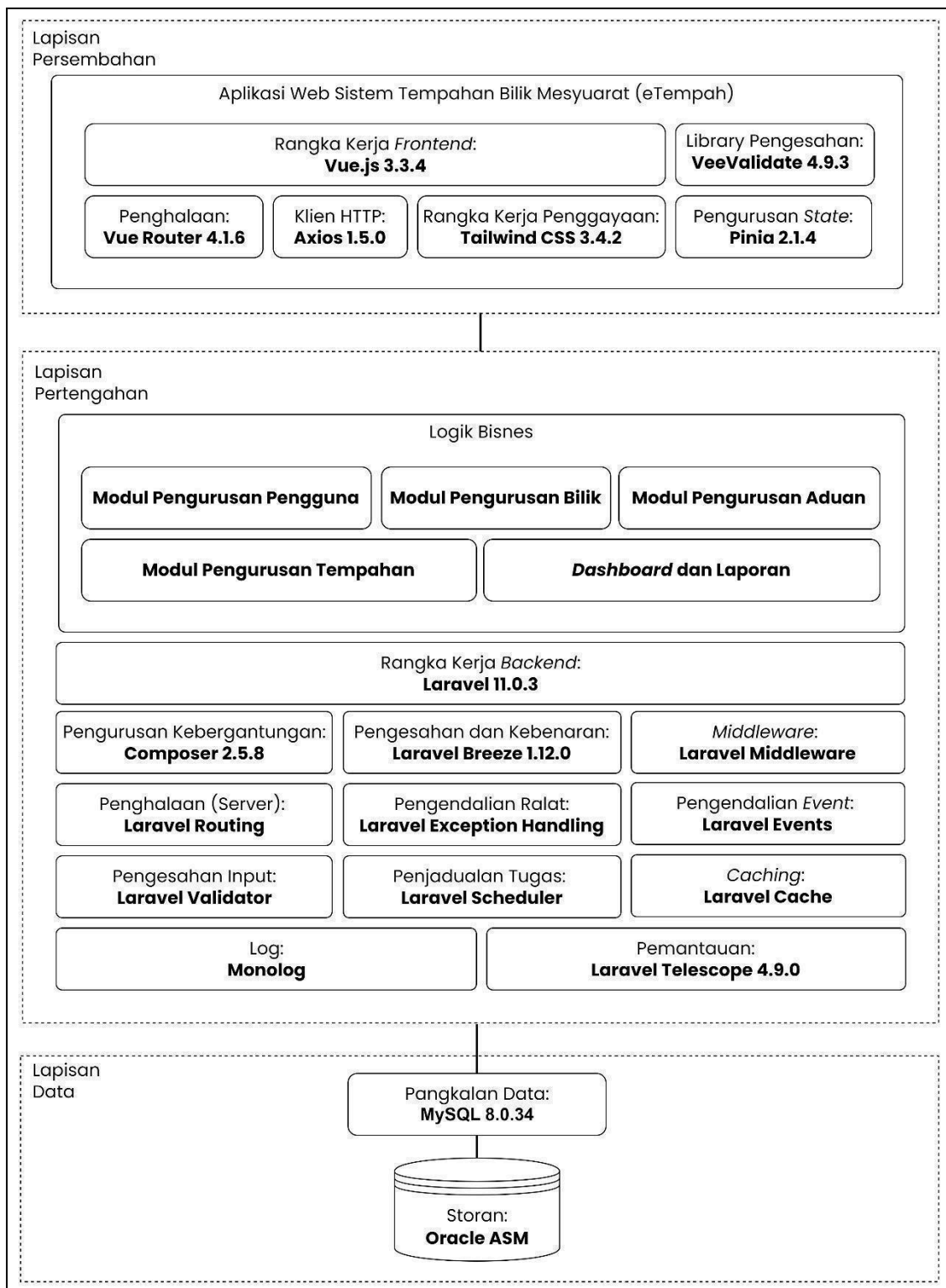
Rajah 4.14: Komponen *Shared Disk* Bagi Sistem eTempah



Rajah 4.15: Komponen *Shared Nothing* Bagi Sistem eTempah

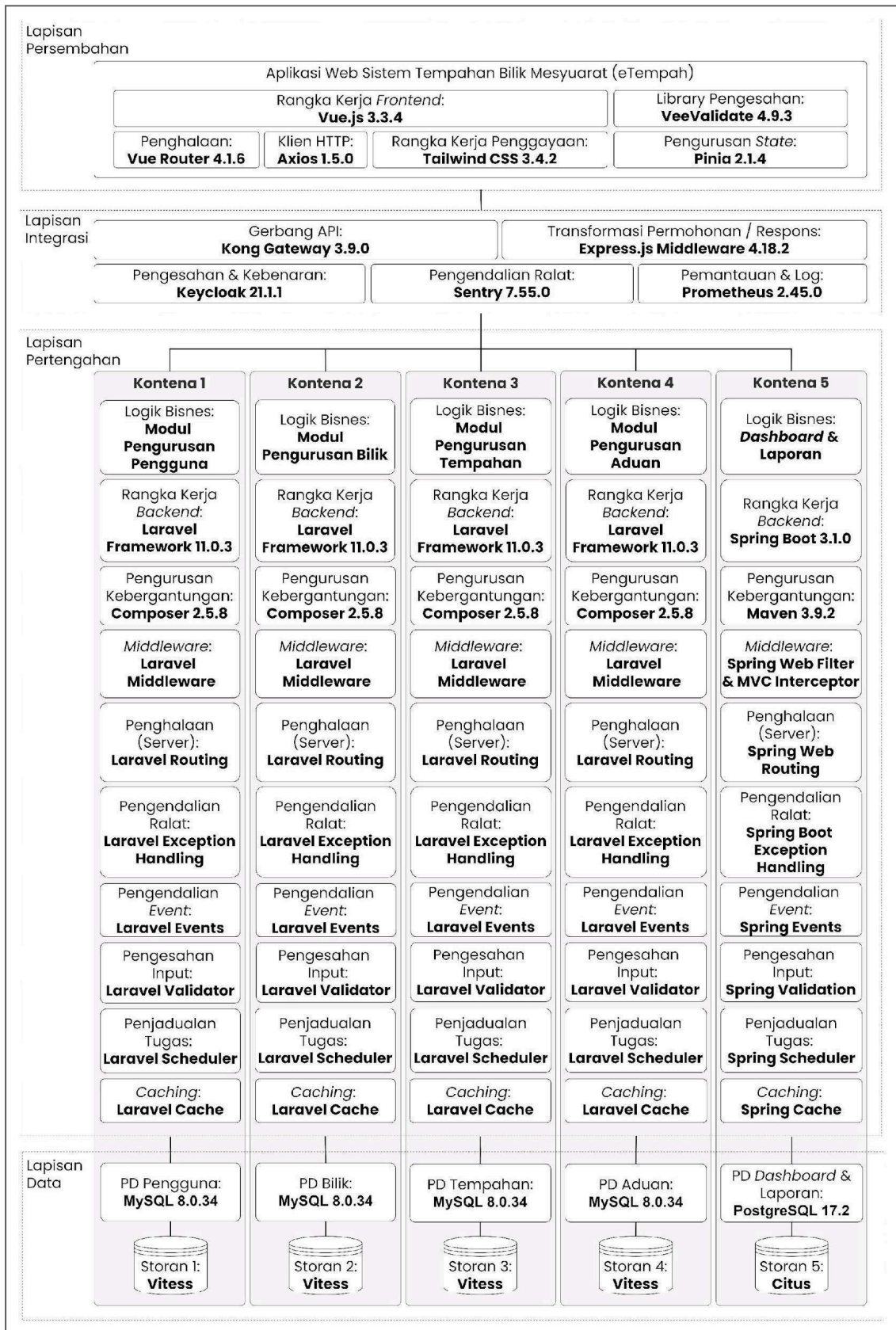
Komponen-komponen yang telah disediakan dalam setiap lapisan ini akan digabungkan untuk mendapatkan satu arkitektur sistem yang lengkap. Berikut merupakan contoh arkitektur monolitik yang merangkumi komponen-komponen terperinci dalam semua lapisan (tertakluk kepada

teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) bagi sistem eTempah:



Rajah 4.16: Arkitektur Monolitik Terperinci Bagi Sistem eTempah

Berikut merupakan contoh arkitektur mikroservis yang merangkumi komponen-komponen terperinci dalam semua lapisan (tertakluk kepada teknologi dan produk yang digunakan serta pengubahsuaian yang diperlukan) bagi sistem eTempah:



Rajah 4.17: Arkitektur Mikro servis Terperinci Bagi Sistem eTempah

Langkah 6 : Sediakan Seni Bina Sistem Secara Iteratif

Penyediaan dan pembangunan seni bina sistem perlu dilaksanakan secara iteratif bagi meningkatkan tahap komprehensif dan memenuhi keperluan-keperluan yang telah diperolehi. Disyorkan supaya iterasi proses penyediaan seni bina sistem dilaksanakan sekurang-kurangnya sebanyak dua kali untuk meminimalkan sebarang kesilapan.

Rujukan

1. Joseph Ingeno, Software Architect's Handbook - Become a Successful Software Architect by Implementing Effective Architecture Concepts (2018)
2. Kirankalyan Kulkarni, Learn Bitcoin and Blockchain: Understanding blockchain and Bitcoin architecture to build decentralized applications (2018).
3. Narayan Prusty, Modern Javascripts Application (2016).
4. Mark Richards, Software Architecture Patterns (2015).
5. Danny Brian, Kirk Knoernschild. Modern Web Architectre (2016).
<http://www.gartner.com/>.
6. Ben Stopford. Shared Nothing v.s. Shared Disk Architectures: An Independent View (2009).
<http://www.benstopford.com/2009/11/24/understanding-the-shared-nothing-architecture/>.
7. What is a distributed system? An overview of distributed systems and microservices architectures. Kev Zettler, Atlassian (2024).
<https://www.atlassian.com/microservices/microservices-architecture/distributed-architecture>
8. An Introduction To Decentralization, district0x Educational Portal (2024).
<https://education.district0x.io/general-topics/what-is-decentralization/introduction/>
9. Centralized vs. Decentralized vs. Distributed Systems, GeeksforGeeks (2025).
<https://www.geeksforgeeks.org/comparison-centralized-decentralized-and-distributed-systems/>

4.6 REKA BENTUK DAN MODEL PANGKALAN DATA [F3.3]

4.6.1 KETERANGAN

Reka bentuk pangkalan data merupakan reka bentuk logikal struktur asas penyusunan data untuk penyimpanan data. Manakala model pangkalan data adalah model data logikal yang menentukan bagaimana data disimpan, disusun dan dikendalikan. Terdapat tiga tahap model maklumat yang biasanya digunakan sebagai perwakilan dalam pembangunan pangkalan data iaitu model konseptual, model logikal dan model fizikal.

- a) **Data model konseptual** : menggambarkan keperluan maklumat dalam struktur dan konsep bisnes secara konseptual dan data model ini bebas daripada sebarang teknologi dan penyelesaian teknikal.
- b) **Data model logikal**: menggambarkan keperluan maklumat bagi sistem yang dibangunkan secara logikal (*tables, collection, field, embended, nested*) dan reka bentuk model ini adalah bergantung kepada teknologi dan penyelesaian teknikal.
- c) **Data model fizikal**: menggambarkan struktur pangkalan data, menentukan bagaimana data disimpan, disusun dan diakses secara fizikal berdasarkan reka bentuk model logikal. Model ini adalah sebagai rujukan untuk membangunkan dan mengoptimumkan prestasi pangkalan data sebenar.

4.6.2 OBJEKTIF

- a) menyediakan model maklumat berdasarkan keperluan.
- b) sebagai rangka kerja/panduan untuk menaik taraf aplikasi sedia ada, atau pembangunan sistem aplikasi yang baharu di organisasi berkenaan.
- c) mengumpul, mendefinisikan dan memahami elemen-elemen maklumat yang diperlukan dan hubungan kait di antaranya.
- d) Mendapatkan saiz sistem aplikasi dari aspek fungsi data (*data function*).

Panduan ini hanya memfokuskan kepada pangkalan data jenis RDBMS dan NoSQL.

4.6.3 REKA BENTUK RELATIONAL DATABASE MANAGEMENT SYSTEM

(RDBMS)

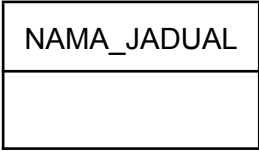
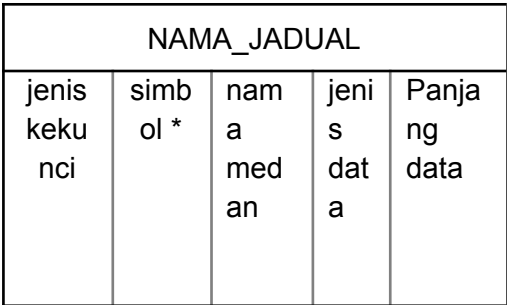
Teknik yang digunakan bagi perwakilan model pangkalan data RDBMS ialah Rajah Hubungan Entiti, atau lebih dikenali sebagai *Entity Relationship Diagram* (ERD). Dalam RDBMS, ERD konseptual menerangkan komponen data yang terdiri daripada entiti, atribut dan hubungan di antaranya. ERD logikal pula menerangkan empat (4) komponen utama iaitu spesifikasi jadual (*table specification*), spesifikasi lajur (*column specification*), spesifikasi

kekunci primer (*primary key specification*) dan spesifikasi kekunci asing (*foreign key specification*). Seterusnya, model logikal ini digunakan untuk mereka bentuk pangkalan data fizikal iaitu pangkalan data yang sebenar.

Notasi

Notasi bagi Model maklumat (ERD) logikal adalah seperti berikut:

Jadual 4.10 : Notasi Model Maklumat Logikal

Elemen	Keterangan
Jadual 	Digunakan bagi mewakili setiap jadual.
Medan 	- Jenis Kekunci Terdapat beberapa jenis kekunci bagi medan yang terdapat dalam jadual. Antaranya ialah: P – Kekunci primer (<i>Primary key</i>) U – Kekunci unik (<i>Unique key</i>) F – Kekunci asing (<i>Foreign key</i>) - Simbol * menunjukkan medan wajib diisi (<i>not null column</i>) manakala sekiranya tiada simbol * menunjukkan medan tidak wajib diisi (<i>null column</i>). - Jenis data bagi setiap medan boleh sama ada dalam format numerik, alfanumerik, aksara, tarikh, masa atau fail (seperti fail imej/audio/video/multimedia). - Panjang (<i>length</i>) data bagi jenis data tersebut dinyatakan dalam tanda kurungan '()' – jika ada.

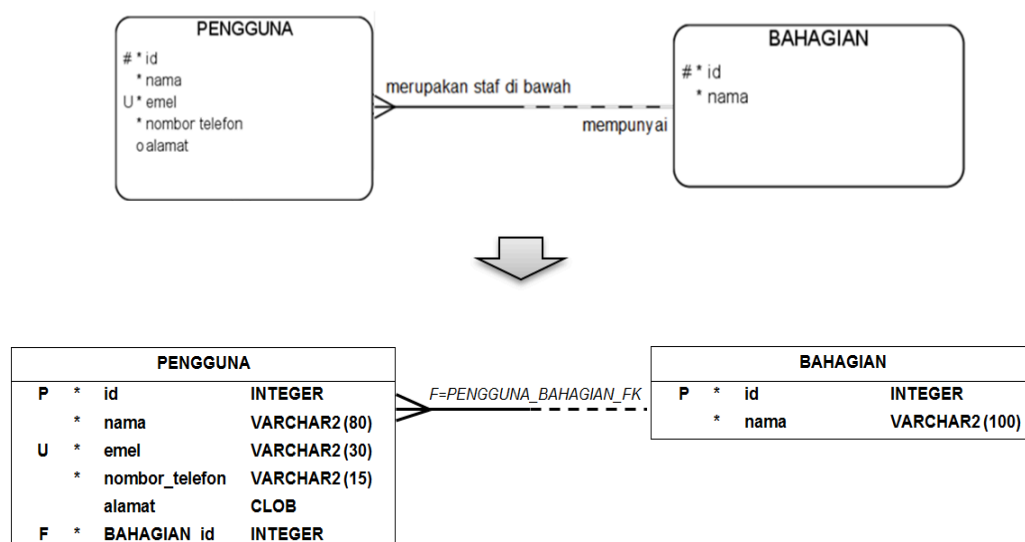
4.6.4 LANGKAH-LANGKAH

Langkah 1: Sediakan Spesifikasi Jadual

- Rujuk notasi seperti di **Jadual 2 Notasi Model Maklumat Logikal** untuk menyediakan spesifikasi jadual.
- Kenal pasti entiti yang mempunyai maklumat yang tidak perlu disimpan dalam pangkalan data terlebih dahulu. Entiti tersebut tidak perlu diterjemahkan ke dalam model maklumat logikal.
- Setiap entiti diterjemahkan terus kepada jadual. Peristilahan untuk penterjemahan adalah seperti berikut.

Model Konseptual		Model Logikal
Nama Entiti		Nama Jadual
PENGGUNA	□	PENGGUNA
ASET	□	ASET
KATEGORI ASET	□	KATEGORI_ASET
TEMPAHAN ASET	□	TEMPAHAN_ASET

Contoh entiti tunggal yang diterjemahkan terus kepada jadual adalah seperti di rajah di bawah.



Rajah 4.18 : ERD – Hubungan antara Entiti

Sekiranya *Intersection Entity* diwujudkan bagi menyelesaikan hubungan banyak-ke-banyak (*many-to-many*) di antara dua entiti, terjemahkan model tersebut kepada model maklumat logikal adalah sama seperti entiti.

- d) Nama jadual mesti dimulakan dengan huruf. Gantikan ruang kosong atau aksara khas yang tidak dibenarkan seperti %, *, –, !, / dan lain-lain kepada aksara garis bawah ‘_’ (*underscore*). Elakkan penggunaan perkataan-perkataan rizab (*reserved words*) yang biasa terdapat dalam bahasa pengaturcaraan seperti *number*, *value*, *type* dan lain-lain.
- e) Nama jadual mestilah unik (nama yang sama tidak boleh digunakan berulang kali) dalam satu skema pangkalan data.

Langkah 2: Sediakan Spesifikasi Medan

- a) Setiap atribut akan diterjemahkan kepada medan (*field*).
- b) Nama atribut akan menjadi nama medan (*field*). Nama medan mesti dimulakan dengan huruf. Gantikan ruang kosong/aksara khas yang tidak dibenarkan kepada aksara garis bawah ‘_’. Elakkan penggunaan perkataan-perkataan rizab, dan beri nama singkatan jika boleh.
- c) Nama medan mestilah unik (nama yang sama tidak boleh digunakan berulang kali) dalam satu jadual.
- d) Atribut Mandatori akan menjadi medan wajib diisi (*not-null column*), manakala Atribut Pilihan menjadi medan tidak wajib diisi (*null column*).
- e) Tentukan jenis data (*data type*) bagi setiap medan. Jenis data boleh sama ada dalam format numerik, alfanumerik, aksara, tarikh, masa atau fail (seperti fail imej/audio/video/multimedia). Nyatakan panjang (*length*) bagi jenis data tersebut dalam tanda kurungan ‘()’ – jika ada.
- f) Sesetengah peraturan bisnes diterjemahkan kepada CHECK Constraint bagi memastikan data yang sah sahaja diterima. Contohnya medan ‘tarikh_tamat_guna’ dalam jadual TEMPAHAN_ASET mestilah lebih besar atau sama dengan medan ‘tarikh_mula_guna’.

tarikh_tamat_guna >= tarikh_mula_guna

- g) Bagi peraturan bisnes yang lebih kompleks, pengkodan/pengaturcaraan tambahan mungkin diperlukan sama ada di komputer pelayan (*server site*), di komputer pelanggan/klien (*client site*), atau kedua-duanya sekali.

Langkah 3: Sediakan Spesifikasi Kekunci Primer

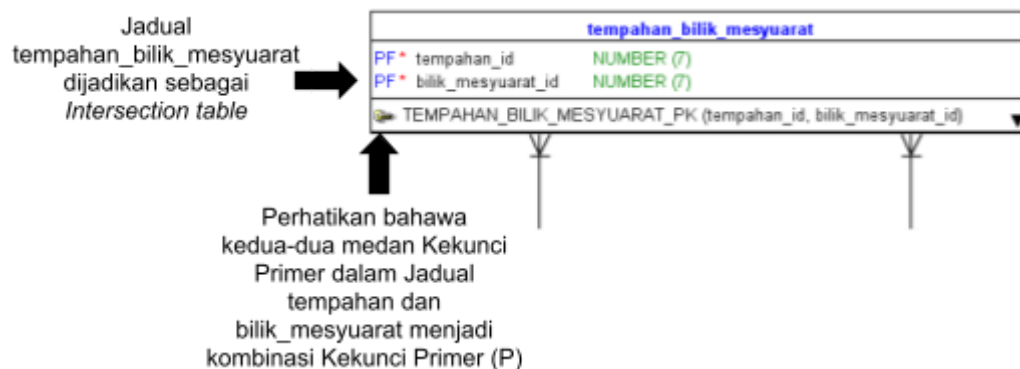
- Terjemahkan UID Primer menjadi Kekunci Primer (*Primary Key*).
- Terjemahkan UID Sekunder akan menjadi Kekunci Unik (*Unique Key*).

Langkah 4: Sediakan Spesifikasi Kekunci Asing

- Hubungan antara dua entiti akan diterjemah menjadi Kekunci Asing (*Foreign Key*).
- Medan Kekunci Asing biasanya dinamakan dengan menggabungkan nama jadual yang dirujuk, dengan nama atribut yang menjadi Kekunci Primer dalam jadual yang dirujuk itu.

Langkah 5: Sediakan Spesifikasi Medan *Intersection Entity*

- Bagi model yang mempunyai *Intersection Entity* yang menyelesaikan hubungan banyak-ke-banyak (many-to-many) di antara dua entiti, UID Primer daripada kedua-dua entiti akan digabungkan untuk menjadi UID Primer bagi *Intersection Entity* tersebut.



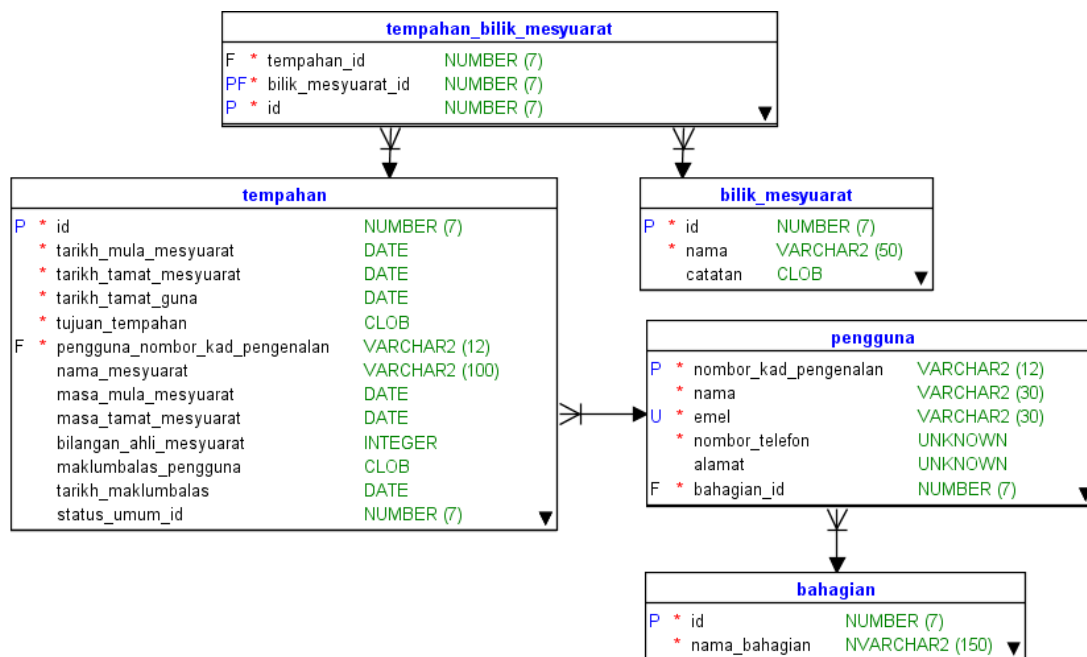
Rajah 4.19 : Kekunci Primer daripada *composite key*

- Dalam model logikal, gabungan UID Primer daripada kedua-dua entiti akan menjadi *Composite Primary key*.
- Merujuk kepada rajah di atas, medan-medan yang terdapat dalam *Intersection Table* 'ITEM_TEMPAHAN', hubungan antara dua entiti diterjemah menjadi Kekunci Asing F_1 dan F_2 , dan kedua-dua Kekunci Asing ini membentuk kombinasi Kekunci Primer (P) (*composite primary key*) dalam jadual berkenaan. Sama ada modaliti hubungan

bersifat mandatori atau sebaliknya, medan Kekunci Asing dalam *Intersection Table* akan sentiasa menjadi medan wajib diisi (*not-null column*).

Langkah 6: Perkemaskan Model Maklumat Logikal

Setelah semua entiti siap diterjemah ke model logikal, lengkapkan dan perkemaskan lagi model maklumat logikal mengikut jenis teknologi yang hendak digunakan. Contoh model maklumat logikal Sistem Tempahan Bilik Mesyuarat (eTempah) adalah seperti rajah berikut.



Rajah 4.20 : Contoh Model Maklumat Logikal Sistem

Langkah 7: Dokumenten Model Maklumat Logikal

Dokumenkan semua output yang dihasilkan sebagai hasil serahan proses reka bentuk pangkalan data logikal ke dalam **D04 Spesifikasi Reka bentuk Sistem**. Dokumentasi mengikut susunan berikut:

- Model Maklumat Logikal
- Rujuk **Apendiks 4 Template Skema Logikal Pangkalan Data (Database Logical Schema)**.

4.6.5 REKA BENTUK NOSQL DBMS

Reka bentuk model pangkalan data NoSQL berbeza dengan reka bentuk pangkalan data RDBMS. Pangkalan Data NoSQL perlu mengambil kira bagaimana data disimpan, diakses dan dimanipulasikan iaitu samada model data adalah *key-value pairs*, *column*, *graphs* atau dokumen (*document-based*). Berikut adalah empat perkara utama yang perlu diambil kira bagi mengenal pasti model data NoSQL yang boleh digunakan.

- Struktur data:** bagaimana data distruktur dan disusun dalam pangkalan data. Sekiranya data adalah distruktur dalam dokumen jenis model data adalah *document-based*. Namun sekiranya struktur data mengandungi hubungan yang kompleks, model data yang sesuai adalah jenis *graphs*.
- Corak pertanyaan (Query pattern):** sekiranya carian dalam sistem aplikasi adalah kebanyakannya berdasarkan kata kunci, jenis model data yang sesuai adalah *key-value*, namun sekiranya pertanyaan kompleks yang melibatkan hubungan yang banyak, model data yang sesuai adalah jenis *graphs*.
- Kadar input ke dalam pangkalan data:** sekiranya sistem aplikasi menerima input ke dalam pangkalan data dengan kadar yang kerap, jenis model data yang sesuai adalah *key-value pairs* atau *column* bagi mengoptimumkan prestasi sistem ketika memasukkan data ke pangkalan data.
- Kebolehskalaan:** perlu mengambil kira saiz pangkalan data berkembang dan penskalaan yang dikehendaki. Penskalaan mendatar adalah sesuai bagi jenis model data dokumen dan *key-value pairs*. Sekiranya menggunakan jenis model data *graphs* berkemungkinan memerlukan infrastruktur yang lebih khusus.

Jadual di bawah menggambarkan model data NoSQL yang utama.

Jadual 4.11: Model Data NoSQL yang utama

Jenis Model data	Penyimpanan	Kegunaan	Contoh Pangkalan Data
<i>key-value pairs</i>	Menyimpan data ringkas yang menghubungkan 2 item iaitu kunci dan nilai data.	•penstoran imej •Sistem berdasarkan kunci •Kadar pengemaskinian rendah	• Memcached • Redis • Riak
<i>column</i>	Menyimpan data dalam bentuk jadual (baris dan lajur)	•Hasil web crawler •Analitis data raya	• HBase • Casandra

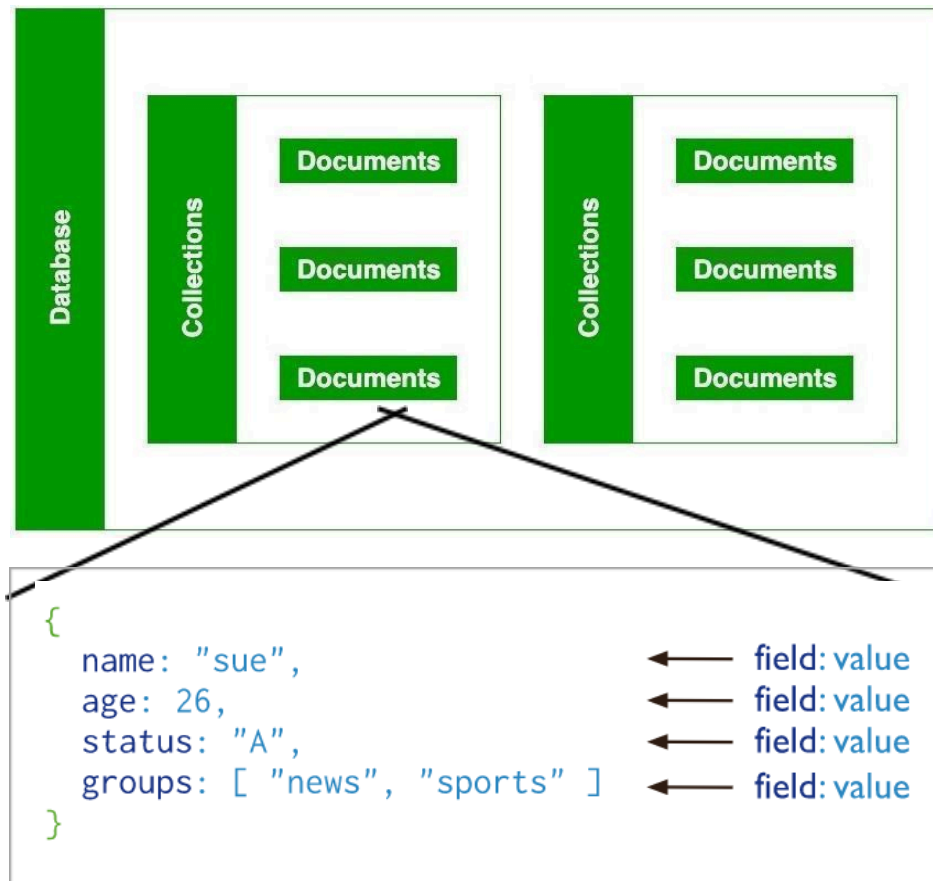
			• Hypertable
Dokumen (<i>Document-based</i>)	Menyimpan data yang berstruktur secara hirarki	• Data kebolehubahan yang tinggi • Carian dokumen • Pengurusan kandungan web • Penerbitan	• MongoDB • CouchDB • CouchBase
<i>Graphs</i>	Menyimpan data yang mempunyai hubungan antaranya yang kompleks	• Media/rangkaian sosial • Pengesanan penipuan • Graf pengetahuan (<i>knowledge graph</i>)	• Neo4j • InfiniteGraph

Panduan ini hanya menyentuh kepada penyediaan reka bentuk pangkalan data dan model data berasaskan dokumen iaitu MongoDB. Perbandingan komponen pemodelan maklumat bagi RDBMS dengan NoSQL dokumen MongoDB adalah seperti berikut.

Jadual 4.12: Perbandingan komponen pemodelan maklumat antara RDBMS dengan NoSQL MongoDB

Model Konseptual	Model Logikal (NoSQL) MongoDB
Pangkalan Data	Pangkalan Data
Entiti	Koleksi (<i>Collection</i>)
Row / Record	Dokumen (<i>Document</i>)
Column	Medan (<i>Field</i>)
Indeks	Indeks (<i>Index</i>)
Hubungan antara entiti	<i>Embedded document, document references, or \$lookup to combine data from different collections</i>

Rajah di bawah menunjukkan struktur pangkalan data dokumen bagi NoSQL MongoDB.



Rajah 4.21 : Struktur pangkalan data dokumen bagi NoSQL MongoDB

4.6.6 MODEL MAKLUMAT DAN REKA BENTUK PANGKALAN DATA MONGODB

Berikut adalah beberapa cara untuk memodelkan maklumat dalam pangkalan data NoSQL dokumen seperti MongoDB iaitu:

a) JSON / BSON Document Representation

MongoDB menggunakan **struktur dokumen format JSON/BSON** sebagai cara utama untuk mewakili data dan hubungannya. Contoh model data boleh ditulis dalam format JSON:

```

Json
{
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "name": "Ali",
  "email": "ali@example.com",
  "order": [
    { "order_id": 101, "jumlah": 250, "tarikh": "2024-02-05" },
    { "order_id": 102, "jumlah": 120, "date": "2024-02-06" }
  ]
}

```

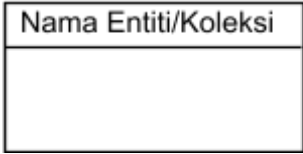
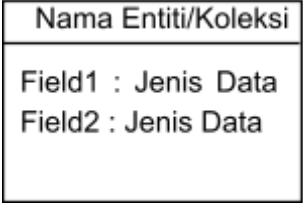
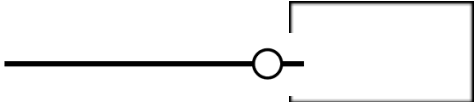
Rajah 4.22 : Struktur dokumen format JSON/BSON

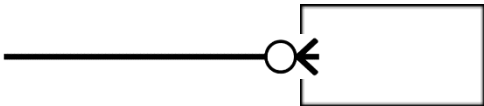
Dalam contoh ini, **hubungan** antara pengguna (*name*) dan pesanan (*orders*) ditunjukkan melalui penyematan (**embedding**).

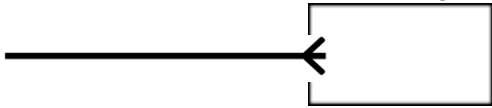
b) Notasi Crow's Foot

Notasi *Crow's Foot* adalah salah satu kaedah visual untuk **memodelkan hubungan antara koleksi** dalam pangkalan data dokumen seperti **MongoDB**. Walaupun MongoDB menggunakan NoSQL, kita masih boleh menggunakan **ERD (Entity-Relationship Diagram) dengan Crow's Foot** untuk memahami **struktur koleksi, hubungan rujukan (references), dan data penyematan (embedded documents)**.

Jadual 4.13: Notasi Crow's Foot

Elemen	Keterangan
Entiti/Koleksi 	• Diwakili sebagai kotak segi empat dengan nama Entiti/Koleksi di bahagian atas.
Medan (Field) 	• Setiap dokumen dalam entiti/koleksi mewakili medan data. • Jenis data ditulis selepas titik dua (:). • Contoh: nama : String, umur : Integer, pesanan : Array<Pesanan>
Kardinaliti: tahap hubungan 'satu' Modaliti: pilihan (zero or one) 	• Menunjukkan tahap hubungan (degree of relationship) satu dengan entiti/koleksi lain dan hubungan tersebut adalah pilihan.

	• Menunjukkan hubungan antara entiti adalah pilihan (<i>optional</i>) • Simbol yang digunakan: simbol bulatan (O) Pilihan
Kardinaliti: tahap hubungan 'banyak' Modaliti: pilihan (zero or many) 	• Menunjukkan tahap hubungan (degree of relationship) adalah kosong atau banyak dengan entiti/koleksi lain dan hubungan tersebut adalah pilihan. • Menunjukkan hubungan antara entiti adalah pilihan (<i>optional</i>) • Simbol yang digunakan: simbol bulatan (O) Pilihan
Kardinaliti: tahap hubungan 'satu' Modaliti: Mandatori (one and only one) 	• Menunjukkan tahap hubungan (degree of relationship) adalah satu dengan entiti/koleksi lain dan hubungan tersebut adalah mandatori (satu dan hanya satu) • Menunjukkan hubungan antara entiti/koleksi adalah mandatori (<i>mandatory</i>) • Simbol yang digunakan: Garis menegak () Wajib

Kardinaliti: tahap hubungan 'banyak' Modaliti: Mandatori (one or many) 	• Menunjukkan tahap hubungan (degree of relationship) adalah satu atau banyak dengan entiti lain dan hubungan tersebut adalah mandatori • Menunjukkan hubungan antara entiti/koleksi adalah mandatori (mandatory) • Simbol yang digunakan: Garis menegak () Wajib
--	--

c) Visualisasi dengan *MongoDB Compass* atau *Tools Lain*

- MongoDB Compass:** Boleh menggambarkan struktur dokumen secara automatik.
- DB Diagram:** Alat pihak ketiga seperti **dbdiagram.io** atau **Hackolade** boleh digunakan untuk membuat model NoSQL.
- Arangodb Modeler / Draw.io:** Alat yang boleh digunakan untuk visualisasi skema NoSQL.

4.6.7 TEKNIK PEMODELAN DATA

a) Penyematan (Denormalisasi) / *Embedding*

Penyematan adalah teknik pemodelan data dalam pangkalan data dokumen di mana data berkaitan disimpan dalam satu dokumen yang sama. Ini bermakna, teknik penyematan adalah menyimpan semua maklumat yang berkaitan dalam satu dokumen untuk meningkatkan prestasi bacaan (***read performance***).

Sebagai contoh, jika kita mempunyai sistem yang menyimpan maklumat pengguna dengan alamatnya, kita boleh menyematkan maklumat alamat terus ke dalam dokumen pengguna. Contoh seperti dibawah di dalam format JSON dan notasi Crow's Foot, dua (2) alamat pengguna disematkan dalam dokumen pengguna. Dalam contoh ini, maklumat alamat tidak disimpan dalam koleksi berasingan.

```
JSON

{
  "_id": ObjectId("61234abc"),
  "nama": "Mohd Saiful",
```

Penyematan /
Embedded
Documents

```

"email": "msaiful@contoh.com",
"alamat": [
  {
    "jalan": "123 Jalan Ampang",
    "bandar": "Kuala Lumpur",
    "poskod": "50000"
  },
  {
    "jalan": "45 Jalan Tun Razak",
    "bandar": "Kuala Lumpur",
    "poskod": "50400"
  }
]
}

```

Rajah 4.23 : Contoh penyematan/embedded dalam data JSON

PENGGUNA	
_id	: objectID
nama	: string
email	: string
alamat	: array
[array 0...]: Doc	
jalan	: string
bandar	: string
poskod	: string

Rajah 4.24: Contoh penyematan/embedded dalam notasi Crow's Foot

b) Rujukan (Normalisasi) / Referencing

Dalam pangkalan data dokumen, **rujukan (normalisasi)** bermaksud menyimpan data dalam dokumen yang berasingan dan menggunakan **ObjectID** untuk merujuk kepada dokumen lain. Ini berbeza dengan **penyematan (embedding)**, di mana data berkaitan disimpan dalam satu dokumen. Teknik ini mengekalkan data integriti iaitu menyimpan data dalam lokasi yang konsisten.

Contoh Penggunaan Rujukan:

Jika kita mempunyai sistem yang menyimpan maklumat pengguna dengan pesanan (order) mereka, kita boleh menyimpan maklumat tersebut dalam dua koleksi berasingan.

Dalam contoh ini, dokumen Pengguna dan Pesanan disimpan secara berasingan. ID Pesanan dimasukkan dalam dokumen Pengguna, dan setiap pesanan mempunyai ID Pengguna untuk merujuk kembali kepada pemiliknya.

Koleksi Pengguna

JSON

```
{
  "_id": ObjectId("61234abc"),
  "nama": "John Doe",
  "email": "john@example.com",
  "pesanan": [
    ObjectId("789xyz"),
    ObjectId("456pqr")
  ]
}
```

Rujukan /
Referencing
Documents

Koleksi Pesanan

JSON

```
{
  "_id": ObjectId("789xyz"),
  "id_pesanan": "ORD123",
  "jumlah": 250.00,
}
```

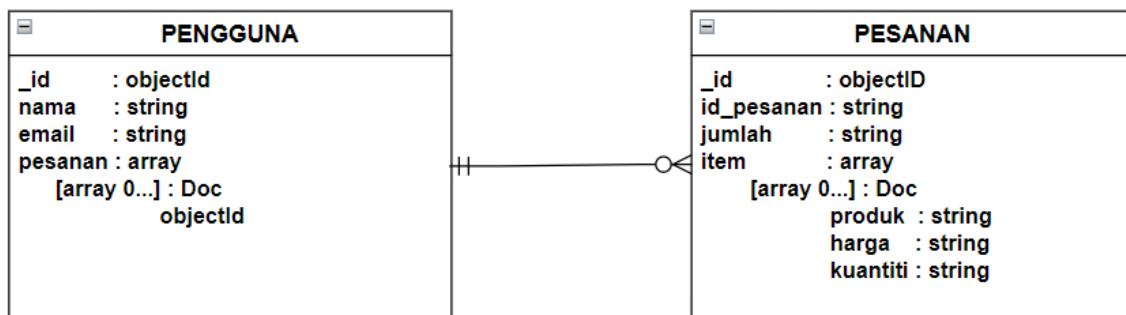
```

"item": [
  { "produk": "Laptop", "harga": 1000,
    "kuantiti": 1 },
  { "produk": "Mouse", "harga": 50,
    "kuantiti": 1 }
],

"id_pengguna": ObjectId("61234abc")
}

```

Rajah 4.25 : Contoh rujukan / referencing dalam data JSON



Rajah 4.26: Contoh rujukan / referencing dalam notasi Crow's Foot

c) Pertimbangan Pemilihan Penyematan (*Embedding*) vs. Rujukan (*Referencing*)

Pertimbangan untuk memilih penyematan (*embed*) data atau rujukan (*reference*) data berdasarkan jenis hubungan antara entiti dan pertimbangan ke atas corak kuiry yang kerap dilakukan. **Jadual 7** adalah amalan terbaik bagi pilihan tersebut berdasarkan jenis hubungan entiti manakala **Jadual 8** adalah kriteria utama yang lain yang perlu dipertimbangkan.

Jadual 4.14: Pemilihan penyematan atau rujukan data mengikut jenis hubungan

Jenis Hubungan	Amalan Terbaik
Satu-ke-satu	Penyematan : Gabungkan data yang mempunyai hubungan ini bagi mengurangkan bilangan pengoperasian <i>read</i> data. Struktur skema data yang baik adalah membolehkan aplikasi melaksanakan kuiry yang diperlukan dalam satu bacaan sahaja.

	Contoh: Maklumat individu dan gambar boleh digabungkan dalam satu <i>document</i>.
Satu-ke-banyak	Penyematan : Gabungkan data yang mempunyai hubungan ini bagi mengurangkan bilangan pengoperasian <i>read</i> data. Struktur skema data yang baik adalah membolehkan aplikasi melaksanakan kuiri yang diperlukan dalam satu bacaan sahaja. Contoh: Maklumat pengguna dan maklumat tempahan boleh digabungkan dalam satu <i>document</i>.
Banyak-ke-banyak	Rujukan: Data yang mempunyai hubungan ini perlu dibuat secara rujukan antara <i>document</i>. Contoh: Hubungan antara penerbit dan buku yang diterbitkan adalah satu penerbit menerbitkan banyak buku. Bagi mengelakkan duplikasi maklumat penerbit dalam satu dokumen, amalan terbaik adalah supaya hubungan dibuat secara rujukan.

Jadual 4.15: Kriteria dalam pertimbangan pemilihan penyematan atau rujukan data

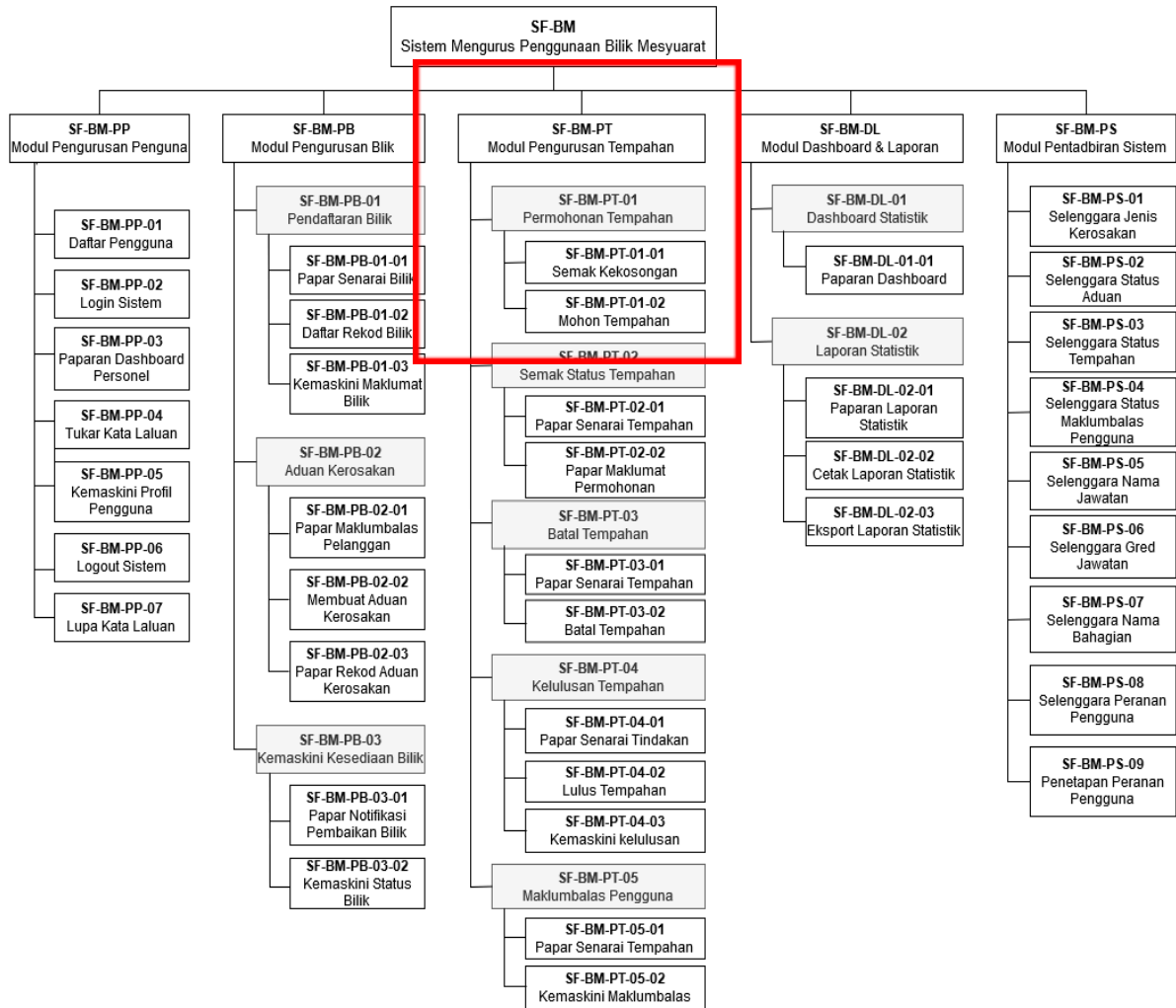
Kriteria	Penyematan (<i>Embedding</i>)	Rujukan (<i>Referencing</i>)
Saiz Data	Sesuai untuk data yang kecil atau sederhana dalam satu dokumen.	Sesuai untuk data berskala besar yang berkembang tanpa had.
Corak Pertanyaan (<i>Query Patterns</i>)	Sesuai jika data sering diambil (<i>retrieved</i>) bersama dalam satu pertanyaan (<i>query</i>).	Sesuai jika data sering diakses secara berasingan atau dari pelbagai konteks.
Kekerapan Bacaan (<i>Read</i>)	Bacaan lebih pantas kerana semua data dalam satu dokumen.	Bacaan mungkin lebih perlahan kerana perlu pertanyaan tambahan (<i>join</i>).
Kekerapan Kemas Kini	Tidak sesuai jika data dalam dokumen sering berubah, kerana keseluruhan dokumen perlu dikemas kini.	Sesuai untuk data yang kerap dikemas kini kerana hanya dokumen berkaitan yang berubah.

Hubungan Antara Data	Sesuai untuk hubungan one-to-many (1:N) (contoh koleksi Pengguna dengan Alamat).	Sesuai untuk hubungan one-to-many (1:N) dan many-to-many (M:N) (contoh koleksi Pengguna dengan Pesanan).
Had Saiz Dokumen (16MB untuk MongoDB)	Tidak sesuai jika dokumen boleh berkembang besar dan melebihi had.	Sesuai kerana data dipecahkan ke dalam dokumen berasingan.
Penggunaan Semula Data	Tidak sesuai jika data perlu dikongsi dengan koleksi lain, kerana ia disemat (<i>embedded</i>) dalam dokumen utama.	Sesuai jika data perlu dikaitkan dengan banyak koleksi.
Prestasi & Skalabiliti	Prestasi tinggi untuk bacaan (<i>read</i>) kerana kurang pertanyaan (<i>query</i>) tambahan.	Lebih fleksibel untuk skala besar kerana boleh dipecah mengikut keperluan.
Kes Kompleksiti Data	Sesuai untuk data yang ringkas dan tidak memerlukan normalisasi tinggi.	Sesuai untuk sistem yang kompleks yang memerlukan struktur lebih fleksibel.

4.6.8 LANGKAH-LANGKAH

Langkah 1: Kenal Pasti Fungsi Sistem yang Kerap Digunakan

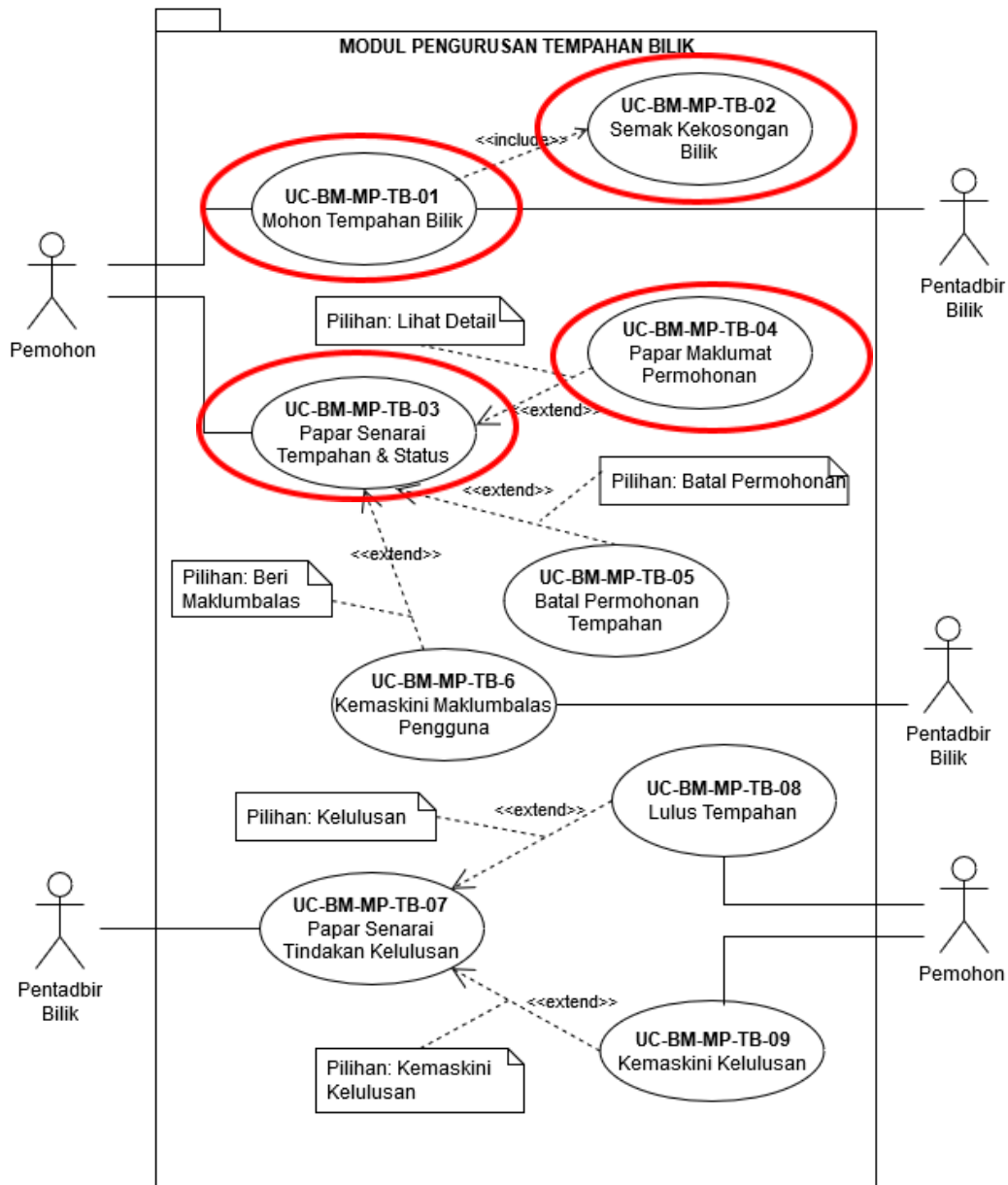
- Berdasarkan kepada proses pemodelan fungsi sistem (rujuk Pemodelan Fungsi Sistem), kenal pasti modul fungsi sistem yang kerap digunakan.
- Berdasarkan **Rajah 4.27**, fungsi Modul Mengurus Tempahan adalah yang kerap digunakan berbanding dengan fungsi-fungsi yang lain.



Rajah 4.27: Contoh Pemodelan Fungsi Sistem

Langkah 2: Kenal Pasti Data Yang Diperlukan

- Kenal pasti data yang diperlukan dalam proses bagi melaksanakan fungsi sistem berdasarkan langkah 1 di atas.
- Sediakan *workload table* bagi mengenal pasti data yang berpotensi kerap dikuiiri. Jadual tersebut mengandungi perkara seperti Aktiviti, Jenis Kuiri (CRUD), Data, Kekerapan dan Prioriti.
 - Berdasarkan pemodelan fungsi sistem (Modul Pengurusan Tempahan) di *Rajah 17*, rujuk rajah *use case* (Modul Pengurusan Tempahan) dan keterangan *use case* bagi menyediakan jadual ini.



Rajah 4.28 : Contoh Use Case

- ii) Berdasarkan rajah use case di atas, terdapat empat (4) *use case* yang dipilih untuk dijadikan contoh dan pengisian *workload table* adalah seperti Jadual 10. Perlu mendapatkan maklumat kekerapan dan keutamaan bagi satu-satu aktiviti daripada pemilik proses. Manakala data boleh merujuk kepada penggunaan maklumat di Definisi Aktiviti Fungsi Bisnes (BRS).

Jadual 4.16: Contoh *workload table* yang menerangkan data yang digunakan bagi aktiviti tempahan bilik mesyuarat

Aktiviti / Kuiri	Jenis Kuiri (CRUD)	Data	Kekerapan (sehari)	Keutamaan

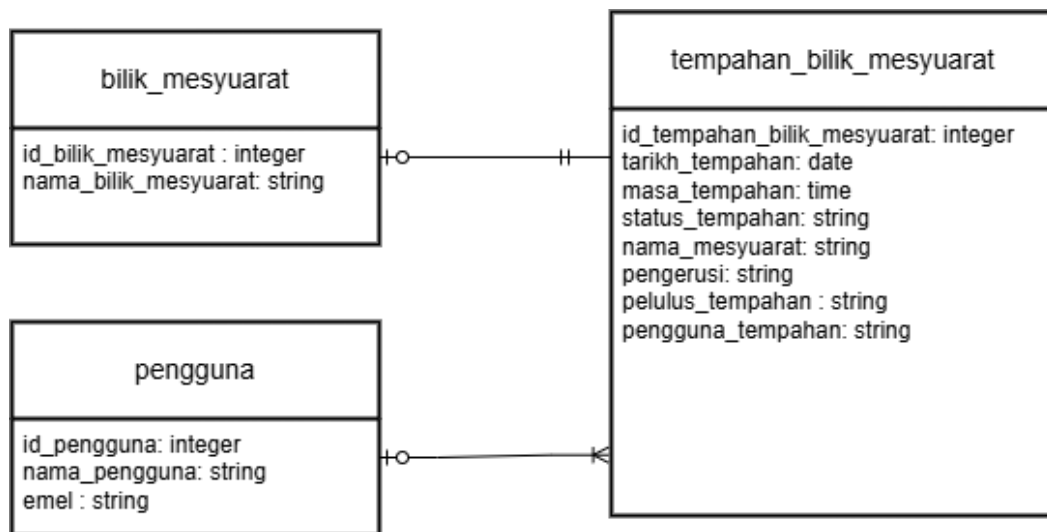
				(Tinggi, sederhana, rendah)
Semak kekosongan bilik	Read	tarikh_tempahan, masa_tempahan, id_bilik_mesyuarat , status_tempahan	30 kali	Tinggi
Mohon tempahan	Create	pengguna, tarikh_tempahan, masa_tempahan, id_bilik_mesyuarat , status_tempahan, nama_mesyuarat, pengerusi	30 kali	Tinggi
Papar senarai tempahan dan status	Read	tarikh_tempahan, masa_tempahan, id_bilik_mesyuarat , status_tempahan	20 kali	Sederhana
Papar maklumat permohonan	Read	pengguna, tarikh_tempahan, masa_tempahan, id_bilik_mesyuarat , status_tempahan, nama_mesyuarat, pengerusi	20 kali	Sederhana

Langkah 3: Kenal Pasti Hubungan Antara Data

- a) Model maklumat pangkalan data dokumen perlu mengambil kira keperluan sistem aplikasi dalam melaksanakan kuiri dan penghasilan jawapan kuiri data yang berkaitan. Amalan terbaik adalah data yang berkaitan perlu berada dalam satu dokumen atau dalam sub-dokumen bagi membolehkan pengoperasian kuiri dengan bacaan tunggal

dan mengelakkan pengoperasian rujukan (lookup) yang lambat. Adakalanya rujukan boleh digunakan bagi merujuk data yang berada di *collection* yang berasingan.

- b) Kenal pasti data berkaitan dalam dalam kuiri dan hubungan antara entiti/data tersebut. Berdasarkan *workload table*, data yang kerap dalam kuiri adalah 'pengguna', 'tarikh_tempahan', 'masa_tempahan', 'id_bilik_mesyuarat', 'status_tempahan',
- c) Kenal pasti hubungan antara data (schema map) yang berkaitan iaitu hubungan satu-ke-satu, satu-ke-banyak dan banyak-ke-banyak.
- d) Dengan merujuk *workload table* di **Jadual 10**, model data yang dihasilkan adalah seperti **Rajah 14** dibawah.



Rajah 4.29: Contoh model data

Hubungan antara bilik_mesyuarat dengan tempahan_bilik_mesyuarat:

- i) Kardinaliti hubungan menunjukkan satu bilik mesyuarat hanya boleh dibuat satu tempahan pada satu masa sahaja. Ini bermaksud hanya satu bilik mesyuarat boleh ditempah pada satu-satu masa sahaja.
- ii) Modaliti bilik_mesyuarat adalah pilihan (optional) iaitu tidak semua bilik_mesyuarat ditempah.
- iii) Modaliti tempahan_bilik_mesyuarat adalah mandatori (mesti) iaitu rekod tempahan_bilik_mesyuarat tidak boleh diwujudkan tanpa wujudnya rekod bilik_mesyuarat.

Hubungan antara pengguna dengan tempahan_bilik_mesyuarat:

- i) Kardinaliti hubungan menunjukkan setiap pengguna boleh tidak membuat tempahan (kosong) atau membuat lebih daripada satu tempahan_bilik_mesyuarat.
- ii) Modaliti hubungan pengguna adalah pilihan iaitu pengguna boleh tidak membuat tempahan.
- iii) Modaliti tempahan_bilik_mesyuarat adalah mandatori (mesti) iaitu rekod tempahan_bilik_mesyuarat tidak boleh diwujudkan tanpa wujudnya rekod pengguna.

Langkah 4: Melaksanakan Pilihan Antara Penyematan (Embedding) atau Rujukan (Reference) Data

- a) Berdasarkan *workload table* tersebut, kenal pasti kaedah yang sesuai untuk memilih penyematan (*embedding*) data atau rujukan (*reference*) data dengan menjawab soalan berikut.
 - i) Apakah fungsi/kuiri utama yang kerap dilaksanakan?
 - ii) Apakah kata kunci yang digunakan untuk mencapai maklumat yang diingini?

Jadual 4.17: Hasil pertanyaan dalam pemilihan penyematan data atau rujukan data

Aktiviti / Kuiri	Kata kunci carian	Maklumat yang ingin dicapai
Semak kekosongan bilik	tarikh_tempahan, masa_tempahan, status_tempahan=null	tarikh_tempahan, masa_tempahan, nama_bilik_mesyuarat , status_tempahan
Mohon tempahan		tempahan_id , nama_pengguna , tarikh_tempah, tarikh_tempahanmula, tarikh_tempahantamat, masa_tempahanmula, masa_tempahantamat, nama_bilik_mesyuarat , status_tempahan, nama_mesyuarat, nama_pengerusi
Papar senarai tempahan dan status	status_tempahan=not null pengguna_id, tempahan_id, id_bilik_mesyuarat	nama_pengguna , tarikh_tempahan, masa_tempahan, tempahan_id , nama_bilik_mesyuarat , status_tempahan

Papar maklumat permohonan	pengguna_id, tempahan_id,	nama_pengguna, tempahan_id, tarikh_tempah, tarikh_tempahanmula, tarikh_tempahantamat, masa_tempahanmula, masa_tempahantamat, nama_bilik_mesyuarat, status_tempahan, nama_mesyuarat, nama_pengerusi
---------------------------	------------------------------	---

- b) Amalan terbaik untuk mengoptimumkan skema NoSQL, aplikasi akan memaparkan maklumat yang dikuiri dalam satu operasi tunggal.
- c) Berdasarkan **Jadual 4.17** dan model maklumat yang disediakan, untuk mengoptimumkan skema, pilihan penyematanan data dilakukan iaitu entiti pengguna, tempahan bilik mesyuarat dan bilik mesyuarat disematkan dalam satu dokumen sahaja.

Rajal

```
{
  "_id_bm": ObjectId("507f1f77bcf86cd799439011"),
  "nama_bm": "Bilik Mesyuarat Cyber 1",
  "tempahan": [
    { "tempahan_id": 101,
      "nama_pengguna": "Siti",
      "emel_pengguna": "siti@jdn.gov.my",
      "tarikh_tempah": "2025-02-13",
      "tarikh_tempahan_mula": "2024-02-14",
      "masa_tempahan_mula": "08:00",
      "tarikh_tempahan_tamat": "2024-02-14",
      "masa_tempahan_tamat": "13:00",
      "nama_mesyuarat": "Mesyuarat JK Digital",
      "nama_pengerusi": "Ts. Ahmad",
      "pelulus_tempahan": "Alia",
      "status_tempahan": "baru"}
    { "tempahan_id": 102,
      "nama_pengguna": "Nazmin",
      "emel_pengguna": "nazmin@jdn.gov.my",
      "tarikh_tempah": "2025-02-13",
      "tarikh_tempahan_mula": "2024-02-17",
      "masa_tempahan_mula": "08:00",
      "tarikh_tempahan_tamat": "2024-02-18",
      "masa_tempahan_tamat": "17:00",
      "nama_mesyuarat": "Perbincangan Digital",
      "nama_pengerusi": "Ts. Nazrul",
      "pelulus_tempahan": "Alia",
      "status_tempahan": "lulus"}
  ]
}
{
  "_id_bm": ObjectId("507f1f77bcf86cd799439022"),
  "nama_bm": "Bilik Mesyuarat Cyber 2",
}
```

Langkah 5 : Dokumenten Model Maklumat Logikal Bagi NoSQL DBMS

Dokumenkan semua output yang dihasilkan sebagai hasil serahan proses reka bentuk pangkalan data logikal ke dalam **D04 Spesifikasi Reka bentuk Sistem**. Dokumentasi mengikut susunan berikut:

1. Model Maklumat Logikal
2. Rujuk **Apendiks 4 Template Skema Logikal Pangkalan Data NoSQL** (*Database Logical Schema*).

Rujukan

1. https://en.wikipedia.org/wiki/Data_modeling#Conceptual.2C_logical_and_physical_schemas.
2. Jan Speelpenning, Patrice Daux and Jeff Gallus; Data Modeling and Relational Database Design (2001).
3. Robert T. Mason, College of Computer & Information Sciences, Regis University, Denver, CO, USANoSQL Databases and Data Modeling Techniques for a Document-oriented NoSQL Database
4. Redis e-book: Data Modeling Pattern
5. <https://www.mongodb.com/>
6. Pengujian prestasi bagi MongoDB di <https://jmeter-plugins.org>
7. <https://www.instaclustr.com/blog/cassandra-data-modeling/>
8. <https://cassandra.apache.org/doc/latest/cassandra/managing/tools/nodetool/status.html>
9. https://cassandra.apache.org/doc/3.11.16/cassandra/tools/cassandra_stress.html

4.7 REKA BENTUK ANTARA MUKA PENGGUNA [F3.4]

4.7.1 KETERANGAN

Penyediaan Reka Bentuk Antara Muka Pengguna (UI) adalah proses untuk menentukan kaedah interaksi di antara pengguna dengan sistem yang akan dibangunkan. Memberi keutamaan kepada reka bentuk UI, khususnya kepada peningkatan pengalaman pengguna (UX), dapat menjadikan sesuatu aplikasi mudah untuk dilayari, efektif dan selesa untuk digunakan.

Di samping itu, UI yang telah dibangunkan perlu dipadankan dengan medan-medan data di dalam jadual (*table*) pangkalan data. Pemetaan data ini bertujuan untuk memudahkan seseorang pembangun sistem mengetahui senarai medan data yang diperlukan bagi satu-satu UI yang telah dibangunkan.

Dalam reka bentuk UI, terdapat pelbagai aspek yang perlu difahami dan dipertimbangkan bagi menghasilkan produk yang efektif serta mesra pengguna. Setiap elemen, daripada definisi asas UI/UX, rangka kerja UX, elemen asas UI, hinggalah kepada penyediaan rangka antara muka (*wireframe*), memainkan peranan penting dalam memastikan kepuasan pengguna yang optimum. Pengenalan kepada konsep-konsep ini adalah langkah awal untuk memahami bagaimana reka bentuk yang dibangunkan dapat memenuhi keperluan dan menepati ekspektasi pengguna melalui pendekatan yang lebih berstruktur dan teratur.

a) Definisi UI/UX

- i) UI merupakan komponen perisian yang membekalkan data dan memberi kawalan kepada pengguna untuk melaksanakan tugas-tugas tertentu, serta membolehkan pertukaran maklumat dengan sistem yang digunakan secara interaktif.
- ii) UX merujuk kepada persepsi dan respons seseorang individu hasil daripada penggunaan kepada satu-satu produk, sistem, atau perkhidmatan yang dipengaruhi oleh faktor-faktor seperti emosi, kepercayaan, budaya, kecenderungan, persepsi, tindak balas fizikal dan psikologi.

b) Rangka Kerja Ciri-ciri UX

Rangka Kerja Ciri-ciri UX merupakan panduan asas kepada pereka dan pembangun sistem dalam menyediakan reka bentuk UI yang efektif serta memiliki tahap kebolegunaan yang tinggi.



Rajah 4.31 : Rangka Kerja Ciri-ciri UX

i) Daya Tarikan

Daya tarikan adalah merupakan keupayaan untuk menarik perhatian dan membangkitkan minat pengguna untuk melayari aplikasi yang dibangunkan. Ciri-cirinya merangkumi:

- **Keterampilan (*Presentation*)**

Keterampilan visual mempengaruhi tanggapan dan ekspektasi pengguna terhadap satu-satu aplikasi. Penggunaan warna, font, imej, dan susun atur yang kemas serta konsisten dapat meningkatkan keselesaan semasa penggunaan sistem. Trend terkini seperti reka bentuk Flat 2.0, Material Design oleh Google, dan Metro UI Windows boleh dijadikan rujukan untuk memperbaiki keterampilan visual aplikasi yang ingin dibangunkan.

- **Kebergunaan (*Usability*)**

Kebergunaan UI yang dibangunkan akan menentukan sama ada aplikasi akan terus dan kerap digunakan oleh pengguna. Aplikasi yang dibangunkan perlu sentiasa menitikberatkan nilai pragmatik serta dapat membantu pengguna melaksanakan tugas dengan berkesan.

- **Kejutan (*Wow Factor*)**

Inovasi di dalam reka bentuk UI yang melangkaui ekspektasi pengguna mampu untuk meningkatkan daya tarikan satu-satu aplikasi. Ciri kejutan ini, atau lebih dikenali sebagai wow factor, dapat memberi impak yang positif kepada pengguna, sekaligus meningkatkan kepuasan semasa melayari aplikasi berkenaan.

ii) **Mesra Pengguna**

Mesra pengguna merupakan keupayaan pengguna untuk mempelajari dan memahami penggunaan sesuatu aplikasi. Ciri-cirinya merangkumi:

- **Mudah Dibaca**

Juruanalisa sistem perlulah terdahulu mengenal pasti kategori dan jenis pengguna aplikasi sebelum sesuatu reka bentuk UI dibangunkan. Pengguna-pengguna aplikasi boleh terdiri sama ada daripada orang awam, kakitangan kerajaan, kumpulan profesional, golongan belia ataupun muda. Oleh yang demikian, seseorang pereka bentuk perlulah bijak dalam menentukan saiz dan warna font yang sesuai serta cerdik dalam menyusun atur kandungan aplikasi dengan kemas dan tidak mengelirukan.

- **Mudah Dicari**

Kebolehcarian sesuatu aplikasi diukur melalui bagaimana mudahnya seseorang pengguna melakukan carian kepada sesuatu maklumat yang tertentu. Contohnya, UI yang dibangunkan perlulah dilengkapi dengan fungsi carian yang pintar, serta susun atur kandungan dan menu perlu diletakkan di ruangan yang strategik.

- **Mudah Dipelajari**

Reka bentuk UI yang efisien perlu mempunyai konsep dan aliran kerja yang konsisten bagi membolehkan aplikasi mudah untuk difahami dan dipelajari oleh pengguna. Penyediaan manual pengguna atau fungsi bantuan berbentuk interaktif bukan sahaja dapat membantu, malah dapat mempercepatkan lagi proses pembelajaran bagi seseorang pengguna.

iii) **Boleh Dipercayai**

Boleh dipercayai merupakan keupayaan sesuatu aplikasi untuk mendapatkan kepercayaan daripada penggunanya. Ciri-cirinya merangkumi:

- **Berkredibiliti**

Kualiti dan pemilihan visual yang bersesuaian bagi UI sesuatu aplikasi, contohnya dari segi gambar yang dipaparkan, ikon yang digunakan serta susun atur yang kemas, mampu untuk meningkatkan imej kredibiliti sesuatu organisasi dari perspektif pengguna.

- **Konsisten**

Elemen-elemen antara muka pengguna seperti warna, font, susun atur dan label-label perlu diselaraskan bagi mencapai aplikasi yang konsisten.

Penyelarasan UI juga dapat membantu pengguna untuk memahami dan mempelajari penggunaan aplikasi dengan lebih pantas.

- **Responsif**

Kelajuan maklumbalas dan prestasi sesuatu aplikasi perlu dititik beratkan di dalam reka bentuk UI. Saiz imej yang dipaparkan serta teknologi UI yang dipilih perlu bersesuaian dengan kebolehan infrastruktur sistem dan selari dengan keperluan yang diperolehi.

c) **Komponen Reka Bentuk UI**

Komponen Reka Bentuk UI secara amnya boleh dikategorikan seperti berikut (senarai ini bukanlah muktamad, dan komponen UI lain yang berkaitan boleh juga diguna pakai).



Rajah 4.32 : Komponen Reka Bentuk UI

i) **Reka Bentuk Visual**

- **Garisan:** Elemen yang digunakan untuk memisahkan atau menonjolkan bahagian dalam antara ruang atau kandungan.

- **Warna:** Pemilihan warna yang membantu mencipta hierarki visual, menarik perhatian, dan menyampaikan emosi atau maklumat.
- **Tipografi:** Gaya, saiz, dan jenis huruf yang digunakan untuk teks bagi meningkatkan kebolehbacaan dan keselarasan visual.
- **Ruang Negatif (Negative/White Space):** Ruang kosong yang dibiarkan tanpa elemen untuk memberikan keseimbangan dan mengelakkan kekeliruan visual.
- **Imej:** Gambar yang digunakan untuk menyampaikan maklumat secara visual dan menambah daya tarikan estetika.

ii) Navigasi dan Organisasi

- **Menu Navigasi:** Struktur pautan yang membolehkan pengguna bergerak antara halaman dalam aplikasi atau laman web.
- **Label:** Komponen UI digunakan untuk memberikan keterangan atau konteks kepada elemen interaktif, seperti tajuk atau arahan, bagi membantu pengguna memahami fungsi elemen tersebut dengan jelas.
- **Breadcrumbs:** Elemen navigasi yang menunjukkan hierarki halaman web yang dilayari, membolehkan pengguna kembali ke muka web sebelumnya dengan mudah.
- **Pagination:** Elemen yang membahagikan kandungan kepada halaman yang lebih kecil, memudahkan pengguna untuk menavigasi data-data yang banyak dengan lebih teratur.
- **Tab:** Elemen navigasi yang membolehkan pengguna membahagi dan mengasingkan kandungan web kepada beberapa bahagian.
- **Akordian:** Elemen yang membolehkan pengguna memapar (expand) atau menyembunyikan (collapse) kandungan web yang dilayari.
- **Carousel:** Elemen interaktif yang memaparkan kandungan secara bergilir-gilir, seperti gambar atau slaid, dalam satu ruang bertujuan untuk menyampaikan maklumat visual secara teratur.

iii) Komponen Interaktif

- **Butang:** Elemen interaktif yang boleh diklik untuk melakukan tindakan ke atas satu-satu tugas, seperti menghantar borang, membuka menu, atau memulakan proses dalam aplikasi.
- **Ikun:** Simbol atau gambar kecil yang digunakan untuk mewakili fungsi, tindakan (tambah, kemaskini atau hapus) atau maklumat tertentu dengan cara yang mudah untuk difahami. Boleh juga diguna sebagai komponen navigasi dan pautan ke menu, portal atau sistem aplikasi yang lain.

- **Switch atau Toggle:** Elemen interaktif yang membolehkan pengguna menukar antara dua pilihan atau status, seperti diaktifkan atau dinyahaktifkan, dengan mudah dan pantas.
- **Checkbox:** Elemen yang membolehkan pengguna menentukan lebih daripada satu pilihan dalam borang web yang dilayari.
- **Butang Radio:** Elemen yang membolehkan pengguna menentukan hanya satu pilihan daripada beberapa pilihan yang ditawarkan.
- **Bar Carian:** Komponen untuk mencari maklumat atau kandungan dalam aplikasi atau laman web.
- **Kotak Teks (Text Box):** Ruang untuk memasukkan teks dalam borang atau aplikasi.
- **Kawasan Teks (Textarea):** Kotak teks yang lebih besar untuk memasukkan maklumat dalam jumlah yang lebih banyak.
- **Slider:** Kawalan interaktif yang membolehkan pengguna memilih nilai dalam julat tertentu, seperti menyesuaikan kecerahan skrin atau tahap bunyi audio.
- **Drop-down Menu:** Senarai pilihan yang boleh dilihat apabila pengguna mengklik atau hover pada menu, digunakan untuk memilih satu daripada beberapa pilihan.

iv) Maklumat dan Status

- **Tooltip:** Mesej ringkas yang muncul apabila pengguna mengapungkan (hover) peranti tetikus di atas satu-satu elemen web.
- **Pop-Up / Modal:** Jendela kecil yang terpapar di atas kandungan satu-satu laman web bertujuan untuk menarik perhatian, menyampaikan maklumat, atau meminta tindak balas pengguna.
- **Progress Bar:** Elemen visual yang menunjukkan kemajuan sesuatu tugasan, dan kebiasaanya dalam bentuk peratusan.
- **Loading Indicator:** Elemen visual yang menunjukkan sesuatu proses sedang berlangsung, biasanya digunakan untuk memberi maklum balas kepada pengguna semasa tempoh menunggu.

d) Wireframe

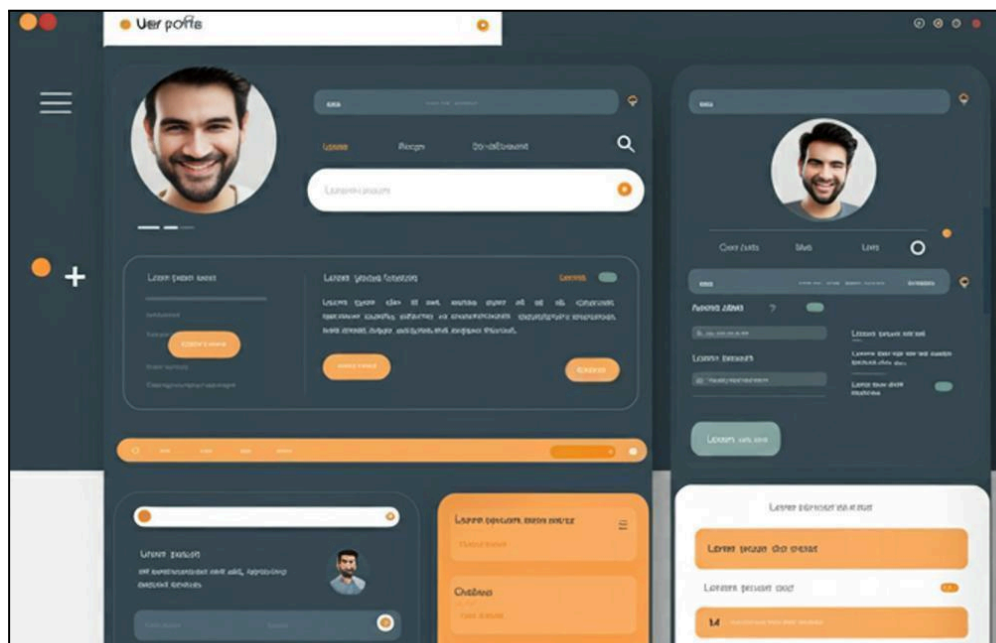
- Wireframe* adalah representasi visual menunjukkan susunan dan struktur elemen-elemen utama dalam sesebuah laman web atau aplikasi. Ia berfungsi sebagai lakaran atau pelan untuk menggambarkan susun atur (layout) satu-satu kandungan muka web aplikasi.
- Terdapat dua jenis *wireframe* yang boleh dibangunkan, iaitu:

- **Low Fidelity Wireframe:** Rangka antara muka yang asas dan ringkas, menggunakan elemen UI yang minima untuk menunjukkan struktur dan susun atur tanpa perincian visual yang kompleks. Output yang dihasilkan adalah dalam bentuk imej statik (format jpeg, png atau gif) bagi menggambarkan secara am reka bentuk UI yang ingin dibangunkan.



Rajah 4.33: Contoh-contoh **Low Fidelity Wireframe** bagi Fungsi Profil Pengguna

- **High Fidelity Wireframe:** Rangka antaramuka yang terdiri daripada tahap visual yang terperinci seakan-akan UI sebenar bagi sistem yang ingin dibangunkan kelak. Output yang dihasilkan sama ada dalam bentuk imej statik, atau membangunkan UI *mockup* yang bersifat interaktif.



Rajah 4.34: Contoh **High Fidelity Wireframe** bagi Fungsi Profil Pengguna

- iii) Penyediaan *wireframe* UI adalah penting bagi:
- **Perancangan Struktur:** Membantu merancang susunan elemen-elemen penting dalam UI bagi fungsi-fungsi sistem.
 - **Penyelarasan Antaramuka:** Memastikan reka bentuk UI yang standard dan selaras bagi semua fungsi-fungsi sistem yang ingin dibangunkan.
 - **Panduan yang Jelas:** Memberikan gambaran reka bentuk UI yang jelas dan terperinci kepada pasukan pembangun dalam aktiviti pengaturcaraan pada fasa selanjutnya.
 - **Pengesanan Awal Kesilapan:** Membantu mengesan dan menyelesaikan kesilapan reka bentuk lebih awal sebelum aktiviti pengaturcaraan bermula. Pada masa yang sama, penyediaan *wireframe* ini dapat mengurangkan risiko jangkamasa dan kos projek berbanding sekiranya kesilapan tersebut ditemui pada fasa-fasa berikutnya, khususnya selepas sistem selesai dibangunkan.

4.7.2 OBJEKTIF

- a) Membangunkan *wireframe* UI, sekurang-kurangnya dalam bentuk imej statik, berpandukan kepada Rangka Kerja Ciri-ciri UX dan Komponen Reka Bentuk UI bagi memenuhi keperluan fungsian dan bukan fungsian sistem aplikasi yang ingin dibangunkan.
- b) Menyediakan Jadual Pemetaan Antara muka Pengguna yang memadankan data-data dengan UI yang telah disediakan.

4.7.3 LANGKAH-LANGKAH

Langkah 1 : Rujuk Rajah Hierarki Fungsi Sistem dan *Use Case*

- a) Lihat kepada rajah hierarki fungsi sistem yang telah dibangunkan dalam **Pemodelan Fungsi Sistem [F2.2]** untuk dijadikan sebagai asas kepada struktur menu antaramuka sistem yang akan dibangunkan.
- b) Rujuk kepada rajah *Use Case* yang telah disediakan di dalam **Pemodelan Use Case (Fungsian) [F2.1]**. Kenal pasti setiap *use case* yang terlibat bagi penyediaan reka bentuk UI. Pada kebiasaannya, reka bentuk UI yang perlu dibangunkan adalah hanya berkaitan dengan *use case* sokongan sahaja. Namun begitu, penyediaan reka bentuk UI bergantung juga kepada bagaimana tahap penguraian (decomposition) *use case*, iaitu pengaturan *use case* utama dan sokongan yang telah dilakukan sebelum ini.

Langkah 2 : Kenalpasti Data-data Terlibat

- a) Rujuk kepada rajah hubungan entiti (ERD) dan logikal pangkalan data yang telah disediakan di dalam **Reka bentuk Pangkalan Data [F3.4]**.
- b) Kenalpasti elemen-elemen data di dalam entiti/jadual yang akan diguna pakai di dalam ruangan-ruangan reka bentuk UI.

Langkah 3 : Pertimbangkan Keperluan Bukan Fungsian

- a) Pertimbangan kepada keperluan bukan fungsian yang diterima pada fasa-fasa sebelumnya adalah bertujuan bagi memastikan kualiti sistem secara keseluruhan diutamakan dan dititikberatkan.
- b) Pertimbangan keperluan bukan fungsian ini akan melibatkan penekanan kepada dua elemen utama: reka bentuk UI dan data.
- c) Rujuk kepada Rangka Kerja Ciri-ciri UX bagi mempertingkatkan kualiti reka bentuk UI yang ingin dibangunkan supaya mencapai tahap kepuasan yang ingin dicapai.
- d) Bagi yang melibatkan data pula, mulakan dengan mengenal pasti jenis dan klasifikasi data yang akan digunakan dalam satu-satu UI fungsi sistem yang terlibat.
- e) Perhatian khusus perlu juga diberikan kepada data terperingkat seperti nombor kad kredit, nombor pengenalan, atau maklumat kesihatan, oleh kerana setiap jenis data tersebut memerlukan kaedah pengendalian yang berbeza.
- f) Sebagai contoh, maklumat harga harus dipaparkan dalam format mata wang yang ditetapkan dalam keperluan bisnes, imej perlu disesuaikan dengan saiz yang bersesuaian, dan data sensitif mesti dienkrpsi serta dilindungi dengan sewajarnya.

Langkah 4 : Gunakan Tools yang Sesuai Mengikut Jenis Wireframe

- a) Pilih *tool* reka bentuk UI yang sesuai berdasarkan jenis dan juga tahap perincian *wireframe* yang ingin disediakan mengikut keperluan projek.
- b) Bagi membangunkan *low fidelity wireframe* ataupun reka bentuk UI dalam bentuk imej statik, contoh *sketching tools* yang sering digunakan (dan tidak terhad kepada mana-mana produk) adalah seperti Diagrams.net (draw.io), Pencil Project, Wireframe.cc, atau mana-mana perisian yang boleh menghasilkan serta mengubah suai imej-imej grafik.
- c) Pembangunan *high fidelity wireframe* yang memerlukan reka bentuk yang lebih interaktif, contoh *prototyping tools* yang sering digunakan (dan tidak terhad kepada mana-mana produk) adalah seperti Figma, Adobe XD, JustInMind, InVision atau mana-mana perisian yang menyediakan fungsi-fungsi pembangunan reka bentuk secara interaktif yang berkaitan.

- d) Perlu diingatkan bahawa kesemua *tools* di pasaran sentiasa berkembang dan diperbaharui dari semasa ke semasa. Kebanyakan *tools*, termasuk seperti yang telah disenaraikan di atas, telah pun sedia ada dengan fungsi-fungsi yang dinamik dan fleksibel serta sesuai untuk pembangunan bagi kedua-dua jenis *wireframe* tersebut.

Langkah 5 : Tentukan Struktur dan Susun Atur UI

- Mulakan pembangunan *wireframe* dengan menentukan struktur dan susun atur (*layout*) UI bagi satu-satu fungsi sistem berdasarkan *use case* yang terlibat.
- Lukis struktur dan susun atur tersebut dengan komponen utama seperti *header*, *footer*, borang, menu, ruang kandungan, atau/dan mana komponen lain yang berkaitan. Langkah ini merupakan asas bagi pembangunan reka bentuk yang selanjutnya.
- Penentuan struktur ini tidak semestinya dilakukan dalam *wireframe tool*, tetapi boleh juga dibuat menggunakan perisian grafik lain ataupun hanya dengan melakarkannya di atas kertas sahaja.
- Contoh penentuan dan lakaran struktur dan susun atur bagi *use case* “UC-BM-AW-MT-02 Berikan Kelulusan” adalah seperti rajah di bawah.



Rajah 4.35: Contoh Penentuan Struktur dan Susun Atur UI

Langkah 6 : Sertakan Komponen Reka Bentuk Visual

- Di dalam *wireframe tool* yang telah dipilih, masukkan komponen reka bentuk visual seperti garisan, warna, tipografi, ruang negatif, dan imej yang sesuai untuk memberi gambaran awal estetika reka bentuk UI yang ingin dibangunkan.
- Contoh kemasukan komponen reka bentuk visual ini adalah seperti rajah di bawah.



Rajah 4.36: Contoh Kemasukan Komponen Reka Bentuk Visual

Langkah 7 : Masukkan Komponen Navigasi dan Organisasi

- Teruskan penyediaan reka bentuk UI dengan memasukkan komponen navigasi dan organisasi seperti menu, label, *breadcrumbs*, *pagination*, tab, akordian, *carousel* atau/dan mana-mana komponen lain yang berkaitan.
- Kemasukkan komponen ini membantu dalam menentukan bagaimana pengguna akan bergerak di dalam satu halaman atau dengan halaman-halaman fungsi sistem yang lain.
- Contoh kemasukan komponen navigasi dan susun atur ini adalah seperti rajah di bawah.



Rajah 4.37: Contoh Pengisian Komponen Navigasi dan Susun Atur Organisasi

Langkah 8 : Perincian dengan Komponen Interaktif

- Penyediaan UI fungsi sistem diteruskan dengan pengisian komponen interaktif UI seperti butang, ikon, *slider*, kotak teks, butang radio, *checkbox* atau/dan mana-mana komponen lain yang berkaitan bagi membolehkan pengguna berinteraksi dengan fungsi sistem yang ingin dibangunkan.
- Masukkan elemen-elemen data yang telah dikenalpasti ke dalam ruang teks sama ada melalui komponen *text box*, *text area*, *radio button*, *drop-down menu* atau/dan mana-mana komponen lain yang berkaitan dengan kemasukan, pengemaskinian, paparan, pilihan dan penghapusan data.
- Contoh kemasukan komponen interaktif ini adalah seperti rajah di bawah.

The screenshot shows a web application titled 'ETEMPAH SISTEM TEMPAHAN BILIK MESYUARAT'. The interface includes a sidebar menu with options like 'Profil Pengguna', 'Bilik Mesyuarat', 'Tempahan', 'Aduan Kerosakan', 'Laporan', and 'Pentadbiran Sistem'. The main content area displays a form for approving a room booking request. The form contains the following fields and values:

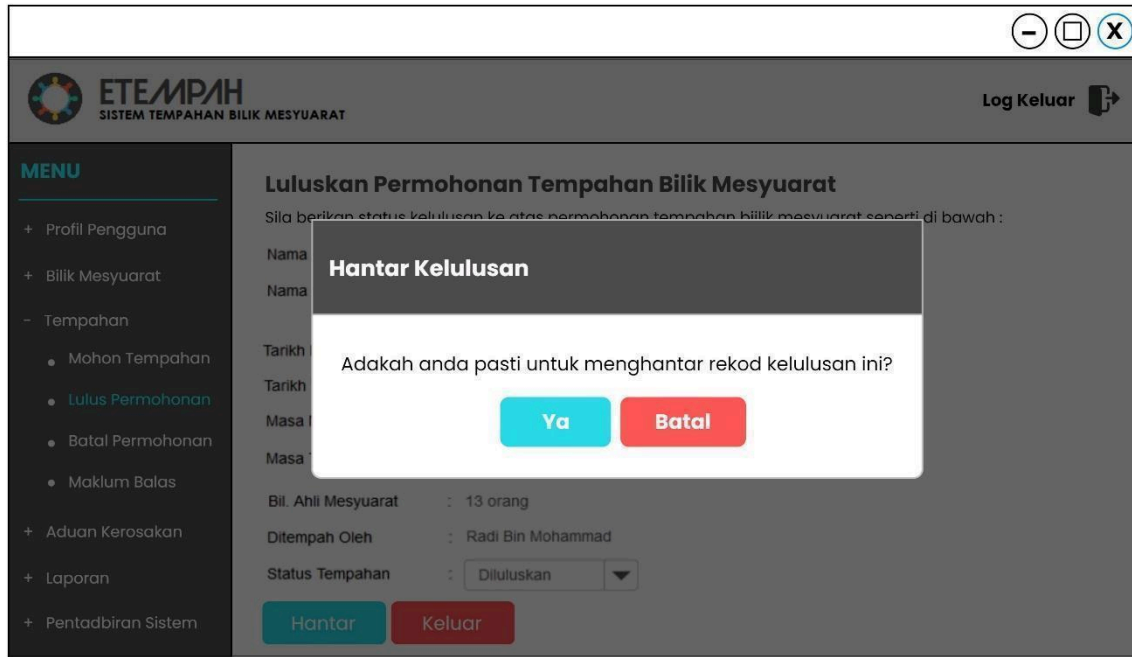
Nama Mesyuarat	: Mesyuarat Kerja Projek MyPerolehan
Nama Bilik Mesyuarat	: Bilik Mesyuarat Melur Bilik Mesyuarat Orkid
Tarikh Mula Mesyuarat	: 12/10/2019
Tarikh Tamat Mesyuarat	: 14/10/2019
Masa Mula Mesyuarat	: 9.00 am
Masa Tamat Mesyuarat	: 5.00 pm
Bil. Ahli Mesyuarat	: 13 orang
Ditempah Oleh	: Radi Bin Mohammad
Status Tempahan	: Diluluskan

At the bottom of the form are two buttons: 'Hantar' (blue) and 'Keluar' (red). The top right corner of the application window has standard window controls (minimize, maximize, close) and a 'Log Keluar' button with a user icon.

Rajah 4.38: Contoh Kemasukan Komponen Interaktif

Langkah 9 : Tambahkan Komponen Maklumat dan Status

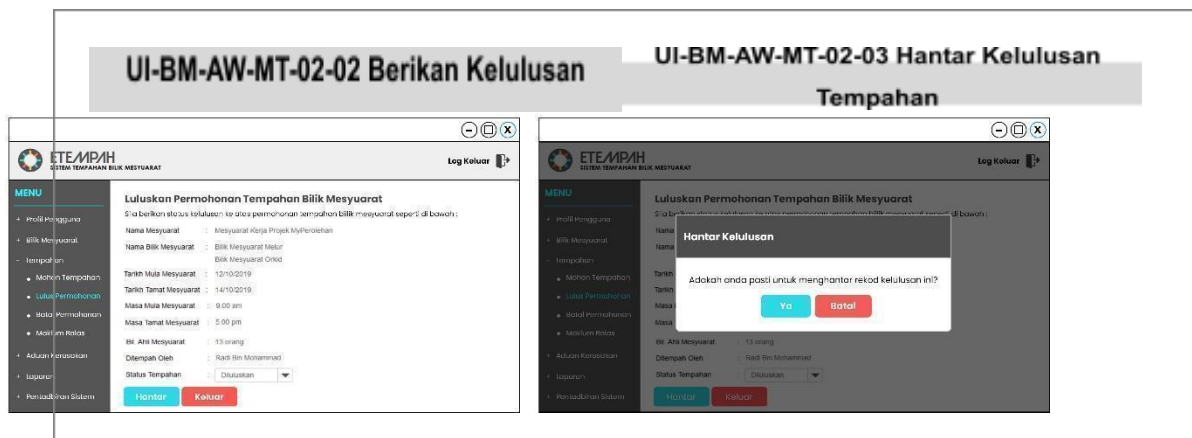
- Masukkan elemen seperti *pop-up/modal*, *progress bar* dan *tooltips* untuk memberi maklum balas kepada pengguna tentang status proses atau interaksi mereka.
- Contoh kemasukan komponen maklumat dan status adalah seperti rajah di bawah.



Rajah 4.39: Contoh Pengisian Komponen Maklumat dan Status

Langkah 10 : Labelkan Reka Bentuk UI

- Labelkan setiap UI yang telah dibangunkan dengan ID dan nama dengan merujuk kepada format yang telah ditetapkan dalam Pemodelan Use Case [F2.1] dan Pemodelan Fungsi Sistem [F2.2].
- Ubahkan nama ringkasan teknik ID tersebut kepada ringkasan teknik reka bentuk antara muka, contohnya dari **UC-BM-AW-MT-02-02 Berikan Kelulusan** kepada **UI-BM-AW-MT-02-02 Berikan Kelulusan**.



Rajah 4.40: Contoh-contoh Penglabelan Reka Bentuk UI

Langkah 11 : Kemaskini Komponen Reka Bentuk UI Secara Iteratif

- Penyediaan *wireframe* dalam reka bentuk UI secara amnya merupakan satu proses yang renyah dan memerlukan usaha fokus yang terperinci.
- Tambahan pula, hasil reka bentuk yang telah dibangunkan sebelum ini besar kemungkinannya perlu dikemas kini berulang kali untuk memenuhi keperluan fungsian dan bukan fungsian yang telah ditetapkan.
- Oleh yang demikian, amalkan sikap terbuka dan sering mendapatkan pandangan dan maklumbalas dari ahli pasukan pembangun lain berhubung dengan kaedah dan teknologi terkini dalam pembangunan *front-end* sistem aplikasi untuk memilih penggunaan komponen grafik UI yang bersesuaian.
- Dapatkan juga maklum balas daripada pemegang-pemegang taruh yang lain supaya reka bentuk UI dapat dibangunkan secara lebih inklusif.

Langkah 12 : Padankan Reka Bentuk UI dengan Elemen Data

- Padankan reka bentuk UI dan medan data yang berkaitan dengan melengkapkan **Apendiks 6 Templat Jadual Pemetaan Data Antaramuka**.
- Selain daripada medan data, masukkan juga objek-objek lain, seperti butang, ikon, label atau pautan, di dalam Jadual Pemetaan Data. Gunakan notasi '<Nama Objek>' apabila merekodkan objek berkenaan. Contoh penulisan menggunakan notasi objek butang adalah seperti <Hantar>, <Simpan> dan <Keluar>.
- Isikan maklumat-maklumat berikut:

Jadual 4.18 : Keterangan Medan-Medan Templat Jadual Pemetaan Data

Medan	Keterangan
Nama Label	Nama label bagi elemen-elemen seperti <i>textbox</i> , <i>textarea</i> atau <i>dropdown menu</i> di dalam reka bentuk UI.
Nama Jadual	Nama jadual (<i>table</i>) di mana medan data disimpan bagi elemen di dalam reka bentuk UI.
Nama Medan Data	Nama medan data yang terlibat bagi elemen di dalam reka bentuk UI.

Medan	Keterangan
Create, Update, Read, Delete (CRUD)	CRUD merujuk kepada empat fungsi utama yang dilaksanakan dalam pangkalan data. Isikan jenis fungsi utama berdasarkan keperluan proses dan reka bentuk UI yang telah dibangunkan. Penerangan lanjut berkenaan fungsi-fungsi CRUD adalah seperti berikut : i) <i>Create</i> (C) adalah merujuk kepada data yang baru diwujudkan atau direkodkan ii) <i>Read</i> (R) adalah merujuk kepada data yang dikeluarkan untuk paparan iii) <i>Update</i> (U) adalah merujuk kepada data yang telah wujud sebelum ini dan hanya perlu dikemaskini. iv) <i>Delete</i> (D) adalah merujuk kepada data yang ingin dihapuskan
Catatan	Ruangan catatan adalah bertujuan untuk memasukkan keterangan tambahan seperti peraturan dan logik bisnes, formula matematik, skrip pangkalan data ataupun apa-apa penerangan lain yang berkaitan bagi membantu pasukan pembangun mengenalpasti penggunaan medan-medan data dalam UI terlibat. Ruangan ini adalah bukanlah mandatori dan boleh digabungkan dengan medan-medan data yang lain sekiranya catatan yang dimasukkan mempunyai ciri-ciri yang sama.

- d) Rujuk jadual di bawah bagi contoh pengisian Templat Jadual Pemetaan Data Bagi Menu Luluskan Tempahan Bilik Mesyuarat, Sistem Tempahan Bilik Mesyuarat (eTempah) berdasarkan kepada contoh UI yang telah disediakan pada langkah 5 sehingga 11.

Jadual 4.19: Contoh Pengisian Templat Jadual Pemetaan Data

Nama Label	Jenis Objek	Nama Jadual	Nama Medan Data	CRU D	Catatan
Nama Mesyuarat	Data	tempahan	nama_mesyuarat	R	

Nama Label	Jenis Objek	Nama Jadual	Nama Medan Data	CRUD	Catatan
Nama Bilik Mesyuarat	Data	tempahan tempahan_bilik_mesyuarat bilik_mesyuarat	id tempahan_id bilik_mesyuarat_id id nama	R	Paparkan semua senarai bilik dalam satu-satu tempahan
Tarikh Mula Mesyuarat	Data	tempahan	tarikh_mula_mesyuarat	R	Format tarikh adalah dalam bentuk 'DD/MM/YYYY'.
Tarikh Tamat Mesyuarat	Data	tempahan	tarikh_tamat_mesyuarat	R	Format tarikh adalah dalam bentuk 'DD/MM/YYYY'.
Masa Mula Mesyuarat	Data	tempahan	masa_mula_mesyuarat	R	Format tarikh adalah dalam bentuk 'DD/MM/YYYY'.
Masa Tamat Mesyuarat	Data	tempahan	masa_tamat_mesyuarat	R	Paparan masa adalah dalam format 12-jam.
Bil. Ahli Mesyuarat	Data	tempahan	bil_ahli_mesyuarat	R	
Ditempah Oleh	Data	tempahan pengguna	no_kad_pengenalan no_kad_pengenalan nama	R	
Status Tempahan	Data	tempahan status_umum	status_umum_id id nama	C/R/U	
<Simpan>	Butang			C/U	Rekod atau pinda maklumat di dalam jadual yang berkaitan setelah butang Simpan diklik. Keluarkan notifikasi <i>pop-up</i> bagi mengesahkan pengguna ingin melakukan operasi simpan.

Nama Label	Jenis Objek	Nama Jadual	Nama Medan Data	CRU D	Catatan
<Keluar>	Butang				Batalan segala operasi dalam menu. Pengguna akan dikembalikan ke muka web/borang yang sebelumnya.

Langkah 13 : Dokumentasikan UI dan Jadual Pemetaan Data

Dokumentasikan semua rajah UI dan Jadual Pemetaan Data yang telah disediakan ke dalam **D04 Spesifikasi Reka bentuk Sistem**.

Rujukan

1. ISO 9241-210:2019 - Ergonomics of human-system interaction, Part 210: Human-centred design for interactive systems
2. IEEE 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology
3. Danny Brian. Attributes of a Great Web User Experience (2013). <https://www.gartner.com/document/2447715>
4. Danny Brian. Building a Great User Experience (2012). <https://www.gartner.com/document/2251816>
5. Ashley Gainer. Tips For Adding White Space To Your Website (2015).
6. <https://getflywheel.com/layout/tips-for-adding-white-space-to-your-website/>
7. Maryam Taheri. 10 Basic Elements of Design (2018).
8. <https://creativemarket.com/blog/10-basic-elements-of-design>
9. Mary Stribley. Design Elements & Principles.
10. <https://www.canva.com/learn/design-elements-principles/>

4.8 REKA BENTUK TRANSAKSI SISTEM – SENARIO USE CASE

Keterangan

4.8.1 KETERANGAN

Reka bentuk Transaksi Sistem – Senario *Use Case* merupakan spesifikasi terperinci sesuatu *Use Case*. Senario *Use Case* terdiri daripada langkah-langkah aktiviti yang akan dilakukan dalam sesuatu *Use Case* dengan disokong oleh reka bentuk antaramuka pengguna dan syarat-syarat/kekangan bagi setiap langkah yang dinyatakan. Ia akan dijadikan sebagai panduan kepada pasukan pembangunan untuk membangunkan kod pengaturcaraan yang lebih tersusun bagi mencapai hasil yang diperlukan.

4.8.2 OBJEKTIF

Membangun Senario *Use Case* bagi setiap Rajah *Use Case* selari dengan reka bentuk antaramuka pengguna yang telah dibangunkan.

4.8.3 LANGKAH-LANGKAH

Langkah 1 : Berikan Penerangan Ringkas *Use Case*

- a) Berdasarkan Rajah *Use Case* yang telah dibangunkan di dalam Pemodelan *Use Case* [F2.1], pilih salah satu Rajah *Use Case* (Modul) sistem.
- b) Ambil salah satu *Use Case* dan terangkan secara ringkas berkaitan *Use Case*
- c) Kenalpasti aktor yang akan berinteraksi dengan *Use Case* yang dipilih
- d) Nyatakan prasyarat atau aktiviti yang perlu dilaksanakan terlebih dahulu sebelum *Use Case* yang terlibat dilaksanakan.

Contoh pra syarat bagi Mohon Tempahan Bilik Mesyuarat adalah:

“Pengguna yang sah berjaya log masuk sistem”.

Langkah 2 : Tentukan Langkah-langkah Aktiviti *Use Case*

- a) Tentukan langkah-langkah aktiviti *Use Case* yang dipilih.
- b) Bagi setiap langkah aktiviti, kenalpasti apakah input (butiran), antaramuka pengguna dan keperluan/syarat-syarat/ kekangan.

Contoh :

Rajah *Use Case* Modul Tempahan Bilik Mesyuarat terdapat 9 *Use Case*, maka sebanyak 9 Senario *Use Case*. Ambil *Use Case* pertama iaitu **UC-BM-MT-TBM-01 Mohon Tempahan Bilik Mesyuarat** dan senaraikan aktiviti yang terlibat dalam memohon tempahan.

Langkah aktiviti Mohon Tempahan Bilik Mesyuarat:

- i) Kemasukan profil pengguna untuk dihubungi
- ii) Pengguna cari kekosongan bilik berdasarkan Tarikh yang diperlukan
- iii) Sistem akan paparkan senarai bilik mesyuarat dan kemudahan yang disediakan
- iv) Pengguna Pilih bilik mesyuarat yang dikehendaki dan tempoh penggunaan
- v) Pengguna hantar permohonan tempahan

Antaramuka Pengguna yang berkaitan:

UI-TBM-01 : Permohonan Tempahan Bilik Mesyuarat

ETEMPAH
SISTEM TEMPAHAN BILIK MESYUARAT

Log Keluar

MENU

- + Profil Pengguna
- + Bilik Mesyuarat
- Tempahan
 - Mohon Tempahan
 - Maklumbalas Penggunaan Bilik
- + Laporan

Mohon Tempahan Bilik Mesyuarat

Butiran pegawai untuk dihubungi

Nama :

E-mel :

Cari kekosongan bilik :

Tarikh Mula :

Masa Tamat Mesyuarat :

Senarai Bilik Mesyuarat :

	Nama Bilik Mesyuarat	Kapasiti	Status
☒	Bilik Mesyuarat Cyber 1	30	Boleh Ditempah
☐	Bilik Bincang Cyber 3	45	Boleh Ditempah
☐	Bilik Bincang Cyber 5 Utara	15	Boleh Ditempah

Rajah 4.41 : Contoh Antaramuka Pengguna Tempahan Bilik Mesyuarat

Langkah 3 : Kenalpasti Pasca Syarat Use Case

Nyatakan pasca syarat selepas semua langkah aktiviti *Use Case* dilaksanakan. Contoh pasca syarat *Use Case* Permohonan Tempahan Bilik Mesyuarat adalah:

“Permohonan tempahan bilik mesyuarat telah dihantar notifiaksi kepada pelulus bagi kelulusan”

Langkah 4 : Nyatakan Proses Alternatif Use Case

Proses alternatif merupakan proses pilihan kepada sesuatu langkah aktiviti yang tidak dapat dilaksanakan secara normal. Contohnya, sekiranya tempahan dalam talian ~~online~~ tidak dapat dilaksanakan maka pengguna boleh dilakukan secara emel dengan menyatakan tarikh, nama bilik mesyuarat.

Langkah 5 : Sedia Dan Lengkapkan Templat Senario Use Case

- Lengkapkan dengan terperinci **Apendiks 7 Templat Senario Use Case** bagi setiap aktiviti *Use Case* yang telah dikenalpasti.
- Maklumat-maklumat Senario *Use Case* adalah seperti berikut :

Jadual 4.20 : Keterangan Medan-medan Templat Senario Use Case

Medan	Keterangan
Rujukan <i>Use Case</i>	Rujukan bagi setiap aktiviti <i>Use Case</i> berdasarkan konvesyen nama dan nombor yang selaras.
Nama <i>Use Case</i>	Nama bagi aktiviti <i>Use Case</i> yang terlibat berdasarkan Rajah <i>Use Case</i> yang telah dibangunkan.
Keterangan	Keterangan secara ringkas aktiviti <i>Use Case</i> yang terlibat.
Pra Syarat	Syarat atau operasi yang perlu dilaksanakan dahulu sebelum aktiviti yang terlibat dilaksanakan.
Aktor	Aktor yang terlibat dengan aktiviti <i>Use Case</i> berkenaan.
Input	Maklumat atau/dan dokumen yang diperlukan bagi setiap proses di dalam aktiviti <i>Use Case</i> yang terlibat.

Medan	Keterangan
Proses	Langkah untuk menavigasi dan melaksanakan operasi berdasarkan reka bentuk antaramuka pengguna yang berkaitan.
Rujukan Reka bentuk Antaramuka	Nama dan nombor rujukan reka bentuk antaramuka pengguna yang terlibat dengan aktiviti <i>Use Case</i> berkenaan.
Keperluan Keterangan / Syarat / Kekangan	Keterangan lanjut atau syarat tambahan atau kekangan yang dihadapi untuk melaksanakan operasi di dalam reka bentuk antaramuka pengguna yang berkaitan.
Pasca Syarat	Syarat atau operasi yang menyusuli selepas aktiviti <i>Use Case</i> berkenaan selesai dilaksanakan.
Proses Alternatif	Proses alternatif sekiranya aktiviti <i>Use Case</i> berkenaan tidak dapat dilakukan.

c) Sila rujuk jadual 10, contoh pengisian Jadual Senario *Use Case*.

Langkah 6 : Dokumentenkan Senario *Use Case*

Dokumenkan semua senario *Use Case* yang telah disediakan ke dalam **D04 Spesifikasi Reka bentuk Sistem**.

Jadual 4.21 : Pengisian Templat Senario *Use Case*

Rujukan Use Case	UC-BM-TM-TBM-01		
Nama Use Case	Mohon Tempahan Bilik Mesyuarat		
Keterangan	Transaksi bagi memohon tempahan penggunaan bilik Mesyuarat		
Pra Syarat	Pengguna yang sah berjaya log masuk sistem		
Aktor	Pemohon		
Input	Langkah	Rujukan Antara Muka Pengguna	Keperluan Keterangan / Syarat / Kekangan
Nama dan emel pegawai untuk dihubungi	1. Kemasukan profil pengguna untuk dihubungi	UI-TBM-01	Secara <i>default</i> nama dan emel login yang dipaparkan. Nama & emel yang dikemaskini perlu divalidasi dengan senarai warga agensi. Sekiranya nama @ emel bukan warga agensi, mesej perlu dipaparkan
Tarikh Mula dan Tarikh Tamat	2. Pengguna cari kekosongan bilik berdasarkan Tarikh yang diperlukan. Masukkan Tarikh mula dan tamat dan tekan semak	UI-TBM-01	Perlu semak: Tarikh mula $\geq$ Tarikh sistem semasa Tarikh tamat $>$ Tarikh mula
	3. Sistem akan paparkan senarai bilik mesyuarat dan kemudahan yang disediakan	UI-TBM-01	Sistem akan menyemak semua aset bilik yang berstatus kekosongan dan paparkan butiran Nama, kapasiti dan status

	4. Pengguna memilih bilik mesyuarat yang dikehendaki dengan <i>tick</i> pada ruangan yang disediakan	UI-TBM-01	Pengguna boleh memilih lebih daripada satu bilik untuk sesuatu tempahan.
	5. Pengguna hantar permohonan tempahan	UI-TBM-01	Sistem perlu papar mesej sekiranya hantar permohonan tanpa <i>tick</i> pada senarai bilik.
Pasca Syarat	Sistem akan menghantar notifikasi kelulusan permohonan tempahan kepada pelulus.		
Proses Alternatif	Pengguna dapat mengemelkan tarikh dan kapasiti bilik yang diperlukan kepada pentadbir untuk tempahan.		

Rujukan

1. Dokumen Spesifikasi Reka bentuk Sistem (SDS) ePPAx.
2. Slaid Bootcamp Pembangunan Sistem Pasukan Perunding ICT MAMPU.

4.9 REKA BENTUK TRANSAKSI SISTEM – ALIRAN PROSES SISTEM

KETERANGAN

4.9.1 KETERANGAN

Reka bentuk transaksi sistem yang berpandukan kepada aliran sistem memberikan gambaran jelas bagaimana proses-proses beroperasi di dalam sistem secara terperinci dan berturutan (*sequential*). Sama seperti di dalam **Pemodelan Proses Bisnes [F1.4] – Aliran Proses Bisnes**, penyediaan aliran proses sistem ini adalah dengan menggunakan pendekatan diagram aktiviti berdasarkan kepada standard *Unified Modelling Language* (UML). Kaedah ini membantu memvisualisasikan proses-proses di dalam sistem dengan lebih berstruktur dan sistematik bagi memastikan keperluan bisnes dapat diterjemahkan dengan tepat ke dalam sistem yang akan dibangunkan.

Perbezaan di antara kedua-dua aliran proses ini adalah seperti jadual berikut:

Jadual 4.22: Komponen-komponen Monolitik Lapisan Persembahan

	Aliran Proses Bisnes	Aliran Proses Sistem
	Disediakan untuk memahami keperluan bisnes organisasi sebagai asas untuk menentukan fungsi dan prosesi yang perlu dirangkumkan ke dalam sistem.	Disediakan untuk menerang pengendalian dan pengoperasian fungsi-fungsi sistem yang ingin dibangunkan berdasar kepada keperluan bisnes yang telah diperolehi.
	Penglibatan penuh oleh pemegang taruh bisnes, seperti pemilik proses dan <i>subject matter expert</i> (SME), bagi memperolehi aliran	Penglibatan lebih kepada pasukan teknikal, seperti juruanalisa sistem dan pengaturcara, untuk membangunkan aliran proses di dalam sistem dengan efektif serta mampu untuk memenuhi kepada keperluan bisnes yang telah diperolehi.

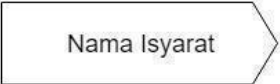
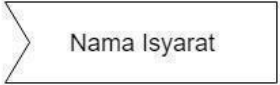
	proses yang komprehensif sejajar dengan visi, misi dan matlamat bisnes organisasi.	
	Penyediaan aliran proses adalah merujuk kepada rajah hierarki bisnes yang telah dibangunkan berdasarkan kepada keperluan bisnes organisasi.	Penyediaan aliran proses adalah merujuk kepada rajah hierarki sistem dan antaramuka pengguna yang telah disediakan.
	Tahap granulariti aliran proses adalah lebih bersifat <i>high level</i> kerana hanya melibatkan gambaran umum proses bisnes, tanpa menerangkan logik sistem dan perincian teknikal yang berkaitan.	Tahap granulariti aliran proses digambarkan dengan lebih terperinci dengan menumpukan kepada pengendalian dan logik sistem dalam satu-satu antaramuka pengguna.

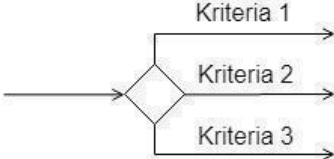
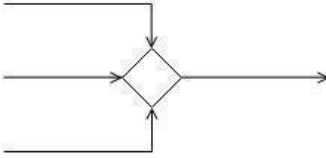
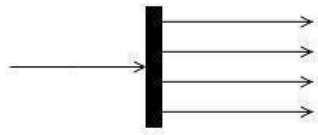
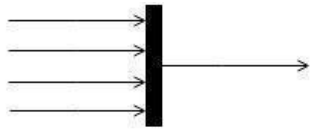
4.9.2 OBJEKTIF

- a) Membangun rajah aliran proses yang menggambarkan pengendalian dan pemprosesan logik yang berlaku di dalam fungsi-fungsi sistem.
- b) Menyediakan Jadual Aliran Proses Sistem yang memerincikan keterangan bagi setiap aksi yang terkandung di dalam rajah aliran proses sistem dan memetakannya dengan fungsi sistem serta antaramuka pengguna yang terbabit.

Notasi

Jadual 4.23 : Notasi Proses Sistem

Elemen	Keterangan
<i>Action</i> 	- Mewakili satu langkah atomik (terkecil) dalam satu-satu aktiviti. - Labelkan ID Aksi Sistem (rujuk notasi di bawah) dan Nama Aksi Sistem di dalam notasi dengan menggunakan ayat yang bermula dengan kata kerja.
<i>Control Flow/Edge</i> 	Digunakan untuk menghubungkan satu elemen notasi kepada notasi-notasi berikutnya.
<i>Swimlane</i> 	- Digunakan untuk mengumpulkan aksi-aksi yang di bawah satu-satu peranan. - Nama peranan dilabelkan berdasarkan kepada peranan-peranan akaun pengguna di dalam sistem.
<i>Initial Node</i> 	- Digunakan untuk menggambarkan permulaan aliran proses. - Perkataan 'Mula' dilabelkan di atas notasi berkenaan.
<i>Activity Final Node</i> 	- Digunakan untuk menamatkan keseluruhan aliran proses. - Perkataan 'Tamat' dilabelkan di atas notasi berkenaan.
<i>Flow Final Node</i> 	- Digunakan untuk memberhentikan aliran proses yang spesifik sahaja. - Perkataan 'Tamat' dilabelkan di atas notasi berkenaan.
<i>Send Signal Action</i> 	- Digunakan untuk penghantaran isyarat (notifikasi). - Nama isyarat yang dihantar perlu dicatatkan di dalam notasi berkenaan.
<i>Accept Event Action</i> 	- Digunakan untuk penerimaan isyarat (notifikasi). - Nama isyarat yang diterima perlu dicatatkan di dalam notasi berkenaan.

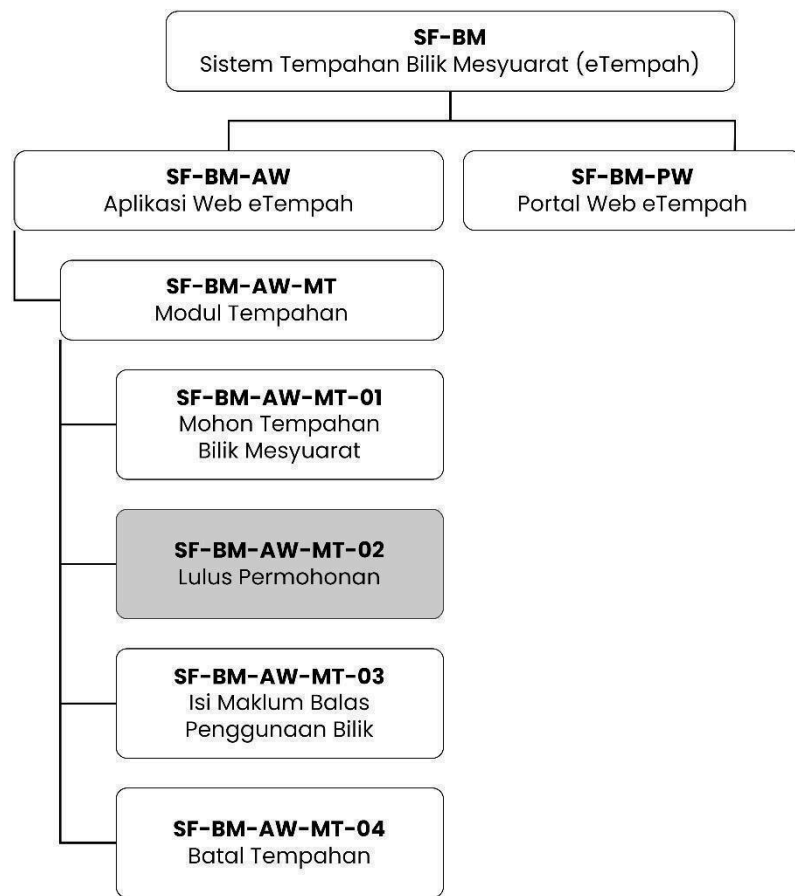
Elemen	Keterangan
<i>Decision</i> 	- Digunakan untuk memisahkan satu aliran proses kepada beberapa aliran proses berbeza berdasarkan salah satu kriteria keputusan yang ingin dipenuhi. - Labelkan kriteria keputusan seperti “Ya” atau “Tidak” bagi setiap cabang aliran proses berkenaan.
<i>Merge</i> 	Digunakan untuk menggabungkan semula aliran-aliran proses yang berlainan supaya hanya satu aliran proses sahaja yang disambungkan kepada aksi seterusnya.
<i>Fork</i> 	Digunakan untuk memisahkan aliran proses kepada beberapa aliran proses berbeza bagi aksi-aksi yang dilaksanakan secara selari.
<i>Join</i> 	Digunakan untuk menggabungkan semula aliran-aliran proses bagi aksi-aksi yang berjalan secara selari supaya proses dapat disambungkan kepada satu aksi seterusnya.
<i>Note</i> 	Digunakan untuk mencatatkan nota ringkas bagi satu-satu aksi. Kandungan nota dicatatkan di dalam ruangan notasi tersebut.
ID Aktiviti Sistem SFD - SS - SBS - MM - SM - 99 - 00	- Digunakan sebagai konvensyen nama dan nombor bagi setiap tajuk rajah aliran proses sistem yang dibangunkan. - Keterangan Kandungan ID: SFD = Ringkasan Nama Teknik SS = Ringkasan Nama Sistem *SBS = Ringkasan Nama Aplikasi *MM = Ringkasan Nama Modul *SM = Ringkasan Nama Submodul 99 = Nombor Rujukan Transaksi

Elemen	Keterangan
	* hanya diperlukan sekiranya tahap hierarki Fungsi Sistem adalah merupakan subsistem, submodul atau subtransaksi.
ID Aksi Sistem SPFD - SS - SBS - MM - SBM - TT - ST - 99	- Digunakan sebagai konvensyen nama dan nombor bagi setiap aksi yang terkandung dalam aliran proses. - Keterangan Kandungan ID: SPFD = Ringkasan Nama Teknik SS = Ringkasan Nama Sistem *SBS = Ringkasan Nama Aplikasi *MM = Ringkasan Nama Modul *SBM = Ringkasan Nama Submodul *TT = Nombor Rujukan Transaksi *ST = Nombor Rujukan Subtransaksi 99 = Nombor Rujukan Aksi

4.9.3 LANGKAH-LANGKAH

Langkah 1 : Kenal pasti Fungsi Sistem dan Antaramuka Pengguna

- Sebelum satu-satu rajah aliran proses sistem dibangunkan, rujuk kepada rajah hierarki sistem bagi menentukan fungsi sistem asas atau terkecil, dan rajah antaramuka pengguna yang berkaitan dengan fungsi berkenaan.
- Pemilihan tahap perincian fungsi sistem perlulah merujuk kepada fungsi yang terkecil sekali, iaitu transaksi, bagi mengurangkan kompleksiti rajah aliran proses yang ingin disediakan. Sekiranya tahap yang lebih tinggi (higher level) digunakan, rajah aliran proses akan menjadi terlalu kompleks dan sukar untuk disediakan, melainkan sistem yang ingin dibangunkan adalah ringkas dan mudah (simple).
- Sebagai contoh, berdasarkan kepada rajah hierarki yang telah dibangunkan dalam **4.x Pemodelan Fungsi Sistem** (seperti di bawah), transaksi SF-BM-AW-MT-02 Lulus Permohonan adalah dipilih untuk menghasilkan rajah aliran proses sistem yang berkaitan.



**Rajah 4.42: Rajah Hierarki Fungsi Sistem eTempah
(Pilihan Transaksi Lulus Permohonan)**

Langkah 2 : Kenal Pasti Peranan dan Sertakan Swimlane

- Kenal pasti peranan-peranan pengguna di dalam sistem yang terlibat bagi fungsi sistem (transaksi) yang telah dipilih.
- Pada kebiasaannya, setiap transaksi hanya mengandungi satu jenis peranan sahaja. Namun begitu, bilangan peranan yang terlibat adalah bergantung kepada tahap perincian fungsi sistem yang dipilih, sama ada fungsi tersebut berada pada tahap transaksi, submodul, modul dan sebagainya.
- Setelah peranan telah ditentukan, lukiskan struktur aliran proses sistem dengan menggunakan notasi *swimlane* berserta label ID aliran, nama fungsi sistem (transaksi) dan nama peranan yang terlibat. Contoh penyediaan *swimlane* adalah seperti berikut:

SFD-BM-AW-MT-02-01 Senarai Kelulusan

Pelulus	
----------------	--

Rajah 4.43: Contoh Swimlane dan Label Dalam Aliran Proses Sistem**Langkah 3 : Mulakan dengan Notasi *Initial Node***

- a) Sama seperti di **Pemodelan Proses Bisnes [F1.4] – Aliran Proses Bisnes**, setiap aliran perlu dimulakan dengan notasi *initial node*.
- b) Labelkan notasi tersebut dengan perkataan “Mula” pada sebelah bahagian atas bagi memudahkan pembaca mengenali notasi berkenaan. Sila rujuk contoh di bawah.

SFD-BM-AW-MT-02-01 Senarai Kelulusan

Pelulus	Mula 
----------------	---

Rajah 4.44: Contoh Penggunaan Notasi *Initial Node* Dalam Aliran Proses Sistem**Langkah 4 : Sertakan Notasi Aksi**

- a) Notasi aksi disertakan bagi melihat tindakan-tindakan yang dilakukan oleh pengguna di dalam antaramuka pengguna di bawah satu-satu fungsi sistem.

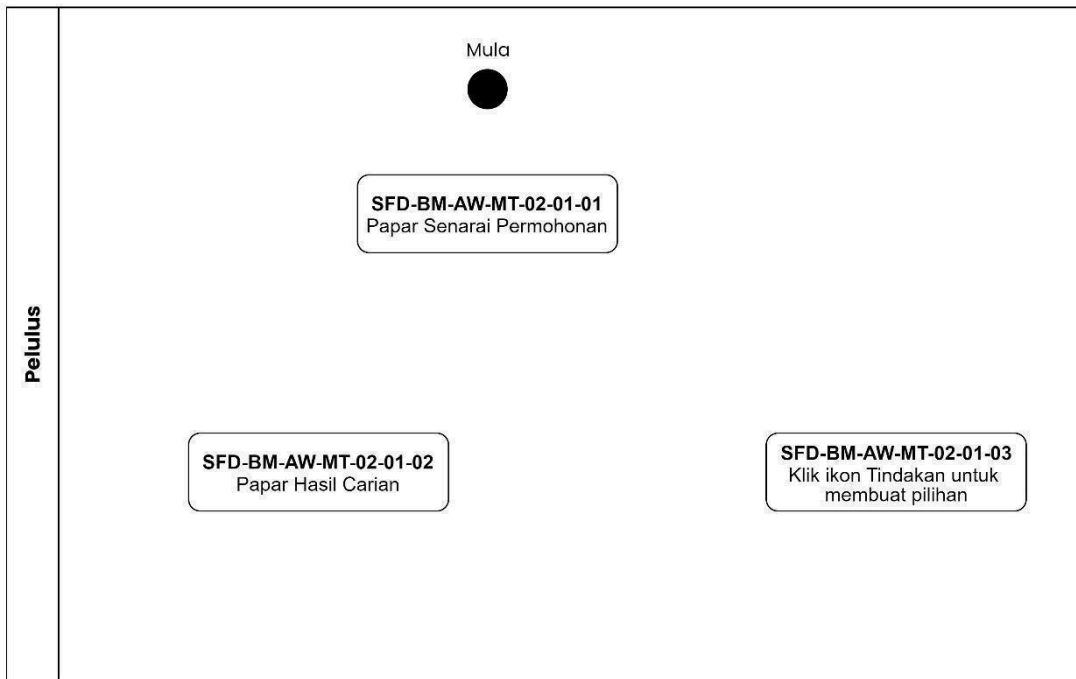
- b) Berbeza dengan notasi aktiviti di dalam **Pemodelan Proses Bisnes [F1.4] – Aliran Proses Bisnes**, notasi aksi merupakan perincian kepada nota aktiviti di mana ia merangkumi tindakan-tindakan seperti klik butang hantar, klik ikon carian, isi maklumat, buat pilihan butang radio, paparan senarai dan sebagainya.
- c) Sebagai contoh, rujuk reka bentuk antaramuka pengguna yang telah dibangunkan iaitu UI-BM-AW-MT-02-01 Senarai Kelulusan, seperti di bawah.

The screenshot displays the ETEMPAH (SISTEM TEMPAHAN BILIK MESYUARAT) web application. It features a sidebar menu with options like 'Profil Pengguna', 'Bilik Mesyuarat', and 'Tempahan'. The main content area is titled 'Senarai Permohonan Tempahan Bilik Mesyuarat' and includes a search bar. Below the search bar is a table listing booking requests with columns for 'Nama Mesyuarat', 'Nama Bilik', 'Nama Pemohon', and 'Status'. Each row also has an edit icon.

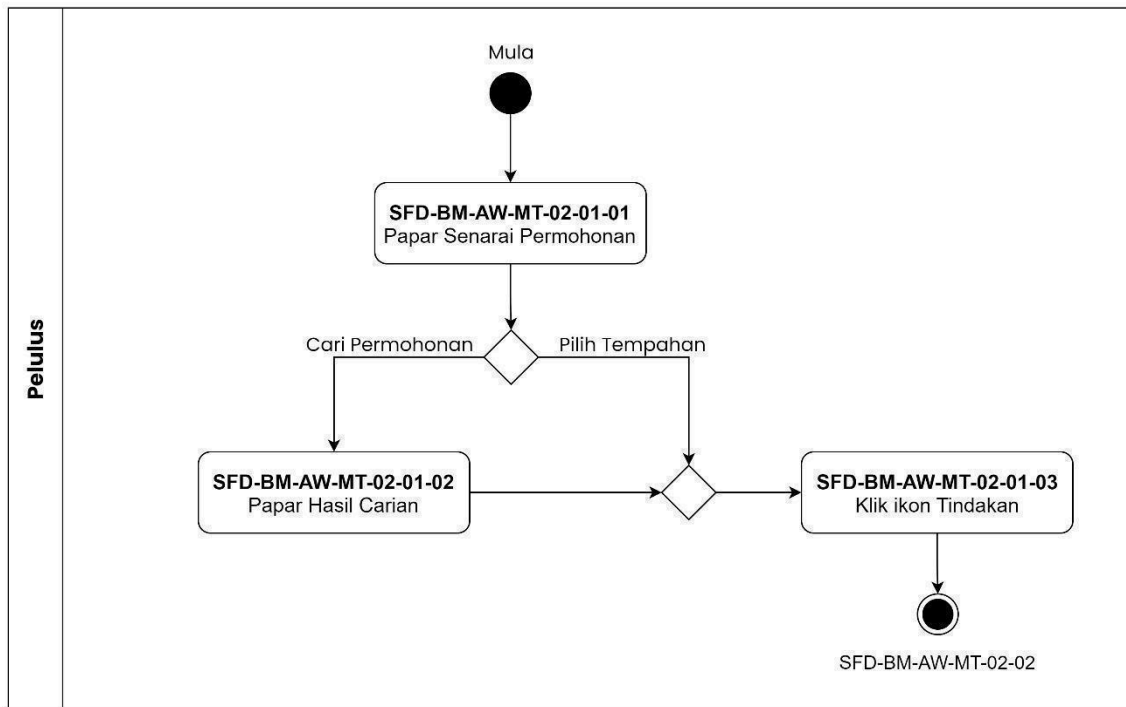
Nama Mesyuarat	Nama Bilik	Nama Pemohon	Status
Mesyuarat Sektor BPM	Suria 1	Ali Bin Ahmad	Baru
Mesyuarat Pengurusan	Bayu 3	Chan Wei Ling	Selesai
Mesyuarat Kelab	Bayu 1	Ravi a/I Kumar	Selesai
Perbincangan Unit	Embun 4	Nadia Binti Nasir	Selesai

Rajah 4.45: Contoh Antaramuka Senarai Kelulusan Tempahan Bilik Mesyuarat bagi Sistem eTempah

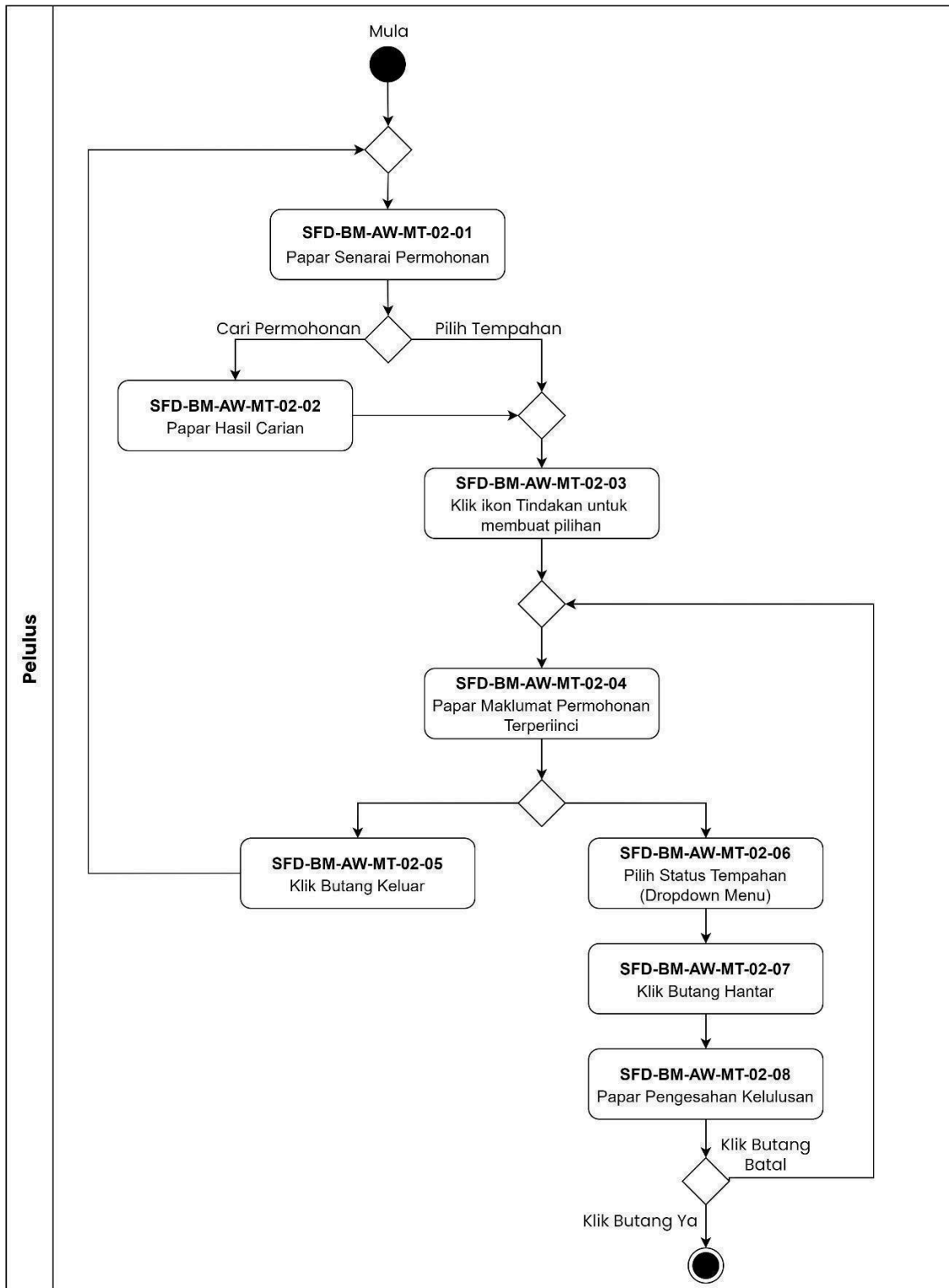
- d) Berdasarkan antaramuka pengguna di atas, sertakan aksi-aksi yang terlibat di dalam *swinlane* yang telah disediakan bersekalikan dengan label ID aksi.

SFD-BM-AW-MT-02-01 Senarai Kelulusan**Rajah 4.46: Contoh Penggunaan Notasi Aksi Dalam Aliran Proses Sistem****Langkah 5 : Sertakan Notasi-notasi Lain**

- Sertakan notasi-notasi lain seperti *control flow*, *fork*, *join*, *decision*, *merge* dan mana-mana notasi lain yang terlibat.
- Pastikan setiap aliran ditamatkan dengan notasi *activity* atau *flow final node*.
- Contoh rajah aliran proses yang selesai disediakan berserta notasi-notasi lengkap adalah seperti berikut:

SFD-BM-AW-MT-02-01 Senarai Kelulusan**Rajah 4.47: Contoh Rajah Aliran Proses Sistem Berserta Notasi Lengkap****Langkah 6 : Gabungkan Aliran Proses Sistem (Opsyen)**

- Bagi proses-proses yang tidak terlalu kompleks, rajah-rajah aliran proses sistem boleh digabungkan supaya keseluruhan proses bagi satu-satu fungsi/subfungsi sistem dapat dilihat dalam satu rajah.
- Namun begitu, langkah ini adalah hanya sebagai opsyen dan tidak sesuai bagi proses yang kompleks kerana ia boleh membuatkan rajah sukar untuk difahami dengan mudah.
- Berikut merupakan contoh rajah aliran proses sistem bagi keseluruhan proses yang terkandung di bawah subfungsi Lulus Tempahan bagi sistem eTempah, dengan merujuk kepada antaramuka-antaramuka pengguna yang sebelum. Label-label ID bagi setiap notasi aksi boleh ditentukan dan diubahsuai mengikut kesesuaian.

SFD-BM-AW-MT-02 Lulus Tempahan

Rajah 4.48: Contoh Gabungan Rajah Aliran Proses Sistem bagi Subfungsi Lulus Tempahan

Langkah 7 : Isikan Jadual Aliran Proses Sistem

- a) Isikan maklumat yang dikehendaki di dalam Jadual Aliran Proses Sistem bagi elemen-elemen berikut:

Jadual 4.24: Penerangan Elemen-elemen Jadual Aliran Proses Sistem

	Elemen	Penerangan
	ID Aliran Proses Sistem	Label ID bagi aliran proses sistem yang disediakan.
	ID Aksi	ID bagi notasi aksi yang terkandung di bawah satu-satu aliran proses sistem.
	ID UI - Rujukan ID Antaramuka Pengguna (UI)	ID reka bentuk antaramuka pengguna yang dirujuk bagi fungsi/subfungsi terlibat.
	Penerangan	Penerangan ringkas bagi aksi yang terlibat.
	Logik/Syarat	Logik-logik dan syarat-syarat bisnes atau aplikasi yang berkaitan dengan aksi terlibat sekiranya ada.

- b) Templat Jadual Aliran Proses Sistem adalah seperti berikut:

Jadual 4.25: Templat Jadual Aliran Proses Sistem

ID Aliran Proses Sistem				
	ID Aksi	ID UI	Penerangan	Logik/Syarat

- c) Penyediaan Jadual Aliran Proses Sistem perlu dilakukan untuk setiap Rajah Aliran Proses Sistem yang telah disediakan bagi fungsi/subfungsi sistem.
- d) Contoh pengisian Jadual Aliran Proses Sistem, dengan merujuk kepada rajah SFD-BM-AW-MT-02-01 Senarai Kelulusan, adalah seperti berikut:

Jadual 4.26: Contoh Pengisian Jadual Aliran Proses Sistem

ID Aliran Proses Sistem			SFD-BM-AW-MT-02-01	
	ID Aksi	ID UI	Penerangan	Logik/Syarat
	SFD-BM-AW-MT-02-01-01	UI-BM-AW-MT-02-01	Memaparkan semua senarai permohonan tempahan bilik mesyuarat yang telah dibuat oleh pengguna.	- Paparan disusun mengikut permohonan terkini. - Senarai mengikut kaedah <i>pagination</i>.
	SFD-BM-AW-MT-02-01-02	UI-BM-AW-MT-02-01	Memaparkan hasil carian yang dilakukan oleh pelulus.	Paparkan label “tiada rekod ditemui” bagi maklumat carian yang tiada di dalam rekod sistem.
	SFD-BM-AW-MT-02-01-03	UI-BM-AW-MT-02-01	Klik pada ikon Tindakan () untuk membuat pilihan untuk dipanjangkan ke perincian permohonan tempahan bagi proses kelulusan selanjutnya.	

Langkah 8 : Sediakan Aliran Proses Sistem Secara Iteratif

Penyediaan rajah dan jadual Aliran Proses Sistem perlu dijalankan secara iteratif bagi memastikan proses tersebut sentiasa dikemas kini selari dengan perubahan kepada keperluan bisnes dan sistem, termasuklah juga logik bisnes dan aplikasi, serta reka bentuk antaramuka pengguna (UI) yang terkini.

Langkah 9 : Dokumentkan Rajah dan Aliran Proses Sistem

Dokumentkan semua rajah dan jadual Aliran Proses Sistem yang telah disediakan ke dalam **D04 Spesifikasi Reka bentuk Sistem**.

Rujukan

1. Alan Dennis, Barbara Haley Wixom, David Tegarden (2012). Systems Analysis and Design with UML Version 2.0, Second Edition.
2. Uml-diagrams.org (2009-2017). Activity Diagrams.
<https://www.uml-diagrams.org/activity-diagrams.html>.

4.10 PENYEDIAAN SPESIFIKASI SISTEM

4.10.1 KETERANGAN

Spesifikasi Reka Bentuk Sistem (SDS) adalah dokumen yang menghimpunkan keseluruhan model dan metode yang telah disediakan dalam fasa analisis dan fasa reka bentuk sistem. Dokumen ini merangkumi penerangan teknikal terperinci berkenaan keperluan sistem, reka bentuk arkitektur, pangkalan data, antaramuka dan transaksi sistem, serta aspek-aspek teknikal lain yang berkaitan. SDS disediakan sebagai rujukan utama, khususnya kepada pasukan pembangun, bagi melaksanakan aktiviti-aktiviti pada fasa berikutnya sepanjang kitar hayat pembangunan sistem aplikasi.

4.10.2 OBJEKTIF

- a) Menyediakan model bagi fungsi sistem secara terperinci.
- b) Menyediakan model use case yang lengkap dan jelas.
- c) Menyediakan reka bentuk arkitektur keseluruhan sistem aplikasi, aplikasi dan arkitektur pangkalan data.
- d) Menyediakan reka bentuk transaksi sistem, reka bentuk antaramuka pengguna dan pemetaan data.
- e) Menyediakan reka bentuk serta skema pangkalan data logikal.
- f) Menyenaraikan keperluan bukan fungsian yang telah dikenalpasti.

4.10.3 LANGKAH-LANGKAH

Langkah 1 : Sediakan Pengenalan Kepada Reka bentuk Sistem

Sila sedia dan lengkapkan pengenalan bisnes bagi perkara-perkara berikut:

a) Tujuan Sistem

Terangkan tujuan, objektif dan matlamat yang ingin dicapai untuk sistem yang ingin dibangunkan selaras dengan tujuan bisnes yang telah disasarkan. Contoh penyediaan tujuan sistem adalah seperti berikut:

“ Sistem ini dibangunkan untuk menyelaraskan dan memudahkan pengurusan maklumat serta proses berkaitan dengan tempahan bilik mesyuarat secara berpusat dan sistematik, serta selaras dengan keperluan organisasi. ”

b) Skop Sistem

Nyatakan skop sistem aplikasi berdasarkan kepada keseluruhan keperluan dan reka bentuk sistem yang telah dikenal pasti dalam fasa analisis dan fasa reka bentuk. Skop sistem merangkumi penentuan ruang lingkup model dan metode yang telah disediakan, termasuk model fungsi sistem, *use case*, reka bentuk arkitektur sistem, pangkalan data, antaramuka dan transaksi sistem. Ruang lingkup ini boleh dijelaskan dengan merujuk kepada bilangan modul dan submodul, serta komponen teknikal lain yang akan dibangunkan dalam sistem. Contoh pengisian skop sistem adalah seperti berikut:

“ Sistem ini merangkumi modul-modul dan submodul berikut:

- **Modul Pengurusan Pengguna**
 - **Pendaftaran Pengguna**
 - **Pengurusan Profil Pengguna**
- **Modul Tempahan Bilik Mesyuarat**
 - **Permohonan Tempahan**
 - **Semakan Status Tempahan**
 - **Pembatalan Tempahan**
- **Modul Jadual dan Laporan**
 - **Paparan Jadual Bilik Mesyuarat**
 - **Penjanaan Laporan Tempahan "**

Langkah 2 : Sediakan Model Fungsi Sistem

Pemodelan Fungsi Sistem

a) Penggunaan Notasi

Senaraikan notasi-notasi yang akan digunakan untuk menyediakan Model Fungsi Sistem. Rujuk kepada Pemodelan Fungsi Sistem [F2.2] untuk menyediakan senarai notasi berkenaan.

b) Rajah Hierarki Fungsian Sistem

Sediakan Rajah Hierarki Fungsian Sistem yang terdiri dari komponen-komponen Fungsi Sistem pada tahap sistem, subsistem, modul, submodul dan transaksi. Sila rujuk kepada langkah-langkah di dalam Pemodelan Fungsi Sistem [F2.2] bagi penyediaan rajah serta jadual berkenaan.

Langkah 3 : Dokumentasikan Model Use Case

Pemodelan Use Case

a) Penggunaan Notasi

Senaraikan notasi-notasi yang akan digunakan untuk menyediakan Model Use Case. Sila rujuk kepada Pemodelan Use Case [F2.1] untuk menyediakan senarai notasi berkenaan.

b) Model Use Case

Sediakan Model Use Case yang terdiri daripada Rajah Use Case serta keterangan bagi Use Case yang terlibat. Sila rujuk kepada langkah-langkah di dalam Pemodelan Use Case [F2.1] bagi penyediaan rajah dan keterangan berkenaan.

Langkah 4 : Sediakan Reka Bentuk Arkitektur Sistem

Reka bentuk Arkitektur Sistem

a) Arkitektur Keseluruhan Sistem Aplikasi

Sediakan Arkitektur Keseluruhan Sistem Aplikasi yang merupakan gambaran menyeluruh (*bird's eye view*) bagi komponen-komponen antaramuka sistem, aplikasi dan pangkalan data. Sila rujuk kepada **Reka bentuk Arkitektur [F3.1]** untuk menyediakan arkitektur berkenaan.

b) Arkitektur Aplikasi

Sediakan Arkitektur Aplikasi yang merupakan gambaran komponen-komponen aplikasi terperinci yang terkandung di dalam Arkitektur Keseluruhan Sistem Aplikasi. Sila rujuk kepada **Reka bentuk Arkitektur [F3.1]** untuk menyediakan arkitektur berkenaan.

c) Arkitektur Pangkalan Data

Sediakan Arkitektur Pangkalan Data yang merupakan gambaran komponen-komponen pangkalan data terperinci yang terkandung di dalam Arkitektur Keseluruhan Sistem Aplikasi. Sila rujuk kepada **Reka bentuk Arkitektur [F3.1]** untuk menyediakan arkitektur berkenaan.

Langkah 5 : Dokumentkan Reka Bentuk Fungsian

a) Reka bentuk Antaramuka Pengguna dan Pemetaan Data

Sertakan imej-imej Reka bentuk Antaramuka Pengguna bagi setiap skrin di bawah fungsi, modul, menu dan submenu aplikasi. Reka bentuk Antaramuka Pengguna yang telah disediakan kemudiannya akan dipadankan dengan entiti (jadual) dan atribut yang terkandung di dalam pangkalan data. Sila rujuk kepada langkah-langkah di bawah Reka bentuk Antaramuka [F3.4] bagi penyediaan reka bentuk tersebut dan pemetaan kepada data-data yang terlibat.

b) Reka bentuk Transaksi Sistem

Sertakan Reka bentuk Transaksi Sistem yang terdiri dari jadual-jadual Senario *Use Case*. Sila rujuk kepada langkah-langkah di dalam **Reka bentuk Transaksi Sistem [F3.5]** bagi penyediaan reka bentuk berkenaan.

Langkah 6 : Sediakan Reka bentuk Pangkalan Data

Reka Bentuk Pangkalan Data

Sertakan Skema Pangkalan Data Logikal dengan merujuk kepada langkah-langkah di dalam Pemodelan Keperluan Data [F2.3] dan Reka Bentuk dan Model Pangkalan Data [F3.3] bagi penyediaan reka bentuk berkenaan.

Langkah 7 : Rekodkan Maklumat Keperluan Bukan Fungsian

Maklumat Keperluan Bukan Fungsian yang perlu direkodkan perlu mengambil kira aspek-aspek seperti berikut:

- a. Aspek Sistem
- b. Aspek Organisasi
- c. Aspek Luaran

Lengkap dan sertakan jadual yang mengandungi maklumat keperluan bukan fungsian seperti yang diterangkan di dalam **Penentuan Keperluan Bukan Fungsian [F2.4]**.

Langkah 8 : Sertakan Dokumen Sokongan Sebagai Lampiran

Sertakan dokumen-dokumen sokongan yang perlu dirujuk seperti format borang fizikal, format laporan dan pelbagai lagi dokumen-dokumen lain yang berkaitan.

Langkah 9 : Lakukan Semakan Dan Pengesahan Ke Atas Dokumen SDS

Dokumen SDS perlu dilakukan semakan oleh Ketua Pasukan Analisis dan Reka bentuk, atau pegawai-pegawai yang lain yang bersesuaian. Disarankan supaya juga dokumen ini disemak bersama *Subject Matter Expert* (SME) dan pemilik proses bagi mengelakkan sebarang kesulitan akan datang, terutamanya apabila aktiviti pengaturcaraan sistem telah bermula. Setelah semakan dilakukan, dokumen SDS yang telah disediakan perlu disahkan oleh Pengurus Projek atau/dan Pengarah Bahagian atau/dan pegawai-pegawai yang lain yang bersesuaian.

Rujukan

1. ISO/IEC/IEEE 29148-2011 Systems and software engineering - Life cycle processes - Requirements engineering (2011).
2. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications (1998).
3. IEEE 1233-1998 - IEEE Guide for Developing System Requirements Specifications (1998).
4. IEEE Std 1016-2009 : IEEE Standard for Information Technology - Systems Design -Software Design Descriptions (2009).

4.11 SPESIFIKASI MIGRASI DATA [F3.8]

4.11.1 KETERANGAN

Proses reka bentuk dan pemetaan akan dilakukan selepas selesai pembangunan pelan migrasi. Proses ini boleh dilakukan secara manual atau menggunakan *tools*.

4.11.2 OBJEKTIF

Proses reka bentuk dan pemetaan akan memastikan data yang dipindahkan adalah berintegriti dan dapat dipetakan dari sumber asal ke destinasi baharu.

4.11.3 LANGKAH-LANGKAH

Langkah 1: Kenal pasti Maklumat Persekitaran Sistem

Kenal pasti maklumat persekitaran sistem dan pangkalan data yang terlibat dengan pelaksanaan migrasi. Maklumat yang diperlukan adalah seperti berikut.

Jadual 4.27: Keperluan Maklumat Persekitaran Sistem dan Pangkalan Data bagi Pelaksanaan Migrasi

Perkara	Keterangan
Maklumat Sistem Sumber	Maklumat bagi sistem yang terlibat dalam migrasi data adalah seperti berikut. - Nama Sistem - Fungsi sistem - Bahasa Pengaturcaraan - Jenis sistem aplikasi (stand alone/client-server/web-based) - Nama Server (application server) - Operating system - IP
Maklumat Pangkalan Data Sumber	Maklumat bagi pangkalan data yang terlibat dalam migrasi data adalah seperti berikut. - Nama dan versi pangkalan data - Database Management System (DBMS) (MySQL, Oracle, PostgreSQL, MongoDB) - Volume Data (bilangan rekod keseluruhan) - Senarai jadual dan medan

	e. Arkitektur (Hybrid/centralised/decentralised) f. Kepatuhan Data Dictionary Sektor Awam (DDSA) (ya/tidak)
Maklumat Sistem Destinasi	Maklumat bagi sistem destinasi yang menerima data yang dimigrasi adalah seperti berikut. a. Nama sistem b. Modul terlibat c. Bahasa Pengaturcaraan d. Jenis sistem aplikasi (stand alone/client-server/web-based) e. Nama Server (application server) f. Operating system g. IP
Maklumat Pangkalan Data Destinasi	Maklumat bagi pangkalan data yang terlibat dalam migrasi data adalah seperti berikut. a. Nama pangkalan data b. Database Management System (DBMS) (MySQL, Oracle, PostgreSQL, MongoDB) c. Senarai jadual dan medan d. Arkitektur (Hybrid/centralised/decentralised) e. Kepatuhan Data Dictionary Sektor Awam (DDSA) (ya/tidak)

Langkah 2: Laksanakan data profiling

Pastikan data telah diekstrak daripada pangkalan data sumber ke persekitaran pembangunan atau *staging*.

Langkah pertama ini adalah proses semakan data sumber bagi mengenal pasti isu kualiti data yang timbul.

Kaedah yang digunakan adalah *column profiling* iaitu mengenal pasti dan menganalisis ciri data dalam setiap *column* (medan data) pada setiap jadual dengan. Isu kualiti data yang perlu dilihat antaranya adalah:

- a) Medan data yang tidak lengkap
- b) Medan data yang tidak konsisten
- c) Medan data yang tidak sah seperti alamat emel yang bukan dalam format yang betul
- d) Rekod data yang bertindih

Sebarang penemuan hendaklah dicatatkan untuk tindakan pembersihan data. Catatan hendaklah mengandungi nama jadual, nama medan dan penemuan isu kualiti data. Berdasarkan penemuan ini, senarai jadual dan medan yang tiada data perlu disahkan terlebih dahulu dengan pemilik sistem untuk tidak dimasukkan dalam senarai migrasi. Manakala senarai jadual (table) dan medan (field) yang ada isu kualiti data, perlu sedia untuk proses pembersihan data. Contoh penemuan adalah seperti jadual di bawah.

Jadual 4.28: Contoh Catatan Penemuan bagi Pembersihan Data

NO	JADUAL (TABLE)	MEDAN (FIELD)	ISU KUALITI DATA	CATATAN
1.	trx_tangkapan	semua	Tidak lengkap	Table kosong
2.	individu	emel	Tidak lengkap	<i>Missing value</i> pada medan wajib
3.	individu	tarikh	Format salah	Format tarikh tidak standard
4.	pengguna	id	Data bertindih	Id yang sama bagi pengguna yang berbeza
5.	individu	Alamat1	Data tidak konsisten	'jln' dan 'jalan' tidak selaras

Langkah 3: Kemaskini data yang terlibat

Kemaskini data yang terlibat dalam migrasi dengan merujuk kepada penemuan *data profiling* seperti di **Langkah 2**. Maklumat ini telah didokumenkan dalam Dokumen Keperluan Bisnes dan perlu dikemaskini sekiranya berkaitan. Senarai keperluan migrasi data yang telah dikemaskini boleh dimasukkan sebagai lampiran dalam Dokumen Spesifikasi Migrasi Data.

Jadual 4.29: Data yang terlibat dalam migrasi

Perkara	Keterangan
ID Fungsi Bisnes	ID fungsi bisnes dari rajah Hierarki Fungsi Bisnes dalam Dokumen Keperluan Bisnes
Nama Fungsi Bisnes	Nama fungsi bisnes dari rajah Hierarki Fungsi Bisnes dalam Dokumen Keperluan Bisnes
Kategori Data	Kategori data sama ada data terbuka atau terperingkat daripada sistem legasi. Maklumat ini diperlukan sebagai input kepada strategi dan kaedah migrasi.
Sumber Data (Fail sumber / pangkalan data)	Nama agensi, sistem dan pangkalan data, jadual, medan atau fail sumber untuk migrasi data

Langkah 4: Sediakan peraturan pembersihan / transformasi data

Peraturan pembersihan / transformasi data disediakan bagi menyelesaikan isu kualiti / penyelarasan data yang telah dikenalpasti dalam **Langkah 2**. Berdasarkan penemuan isu tersebut, tetapkan kriteria dan kaedah pembersihan / transformasi data yang perlu dilakukan. Kaedah ini perlu mendapat pengesahan daripada pemilik sistem terlebih dahulu. Senaraikan peraturan yang berkaitan bagi setiap jadual (table), medan (field) yang terlibat, kriteria dan kaedah pembersihan / transformasi data. Contoh senarai peraturan pembersihan / transformasi data adalah seperti jadual dibawah.

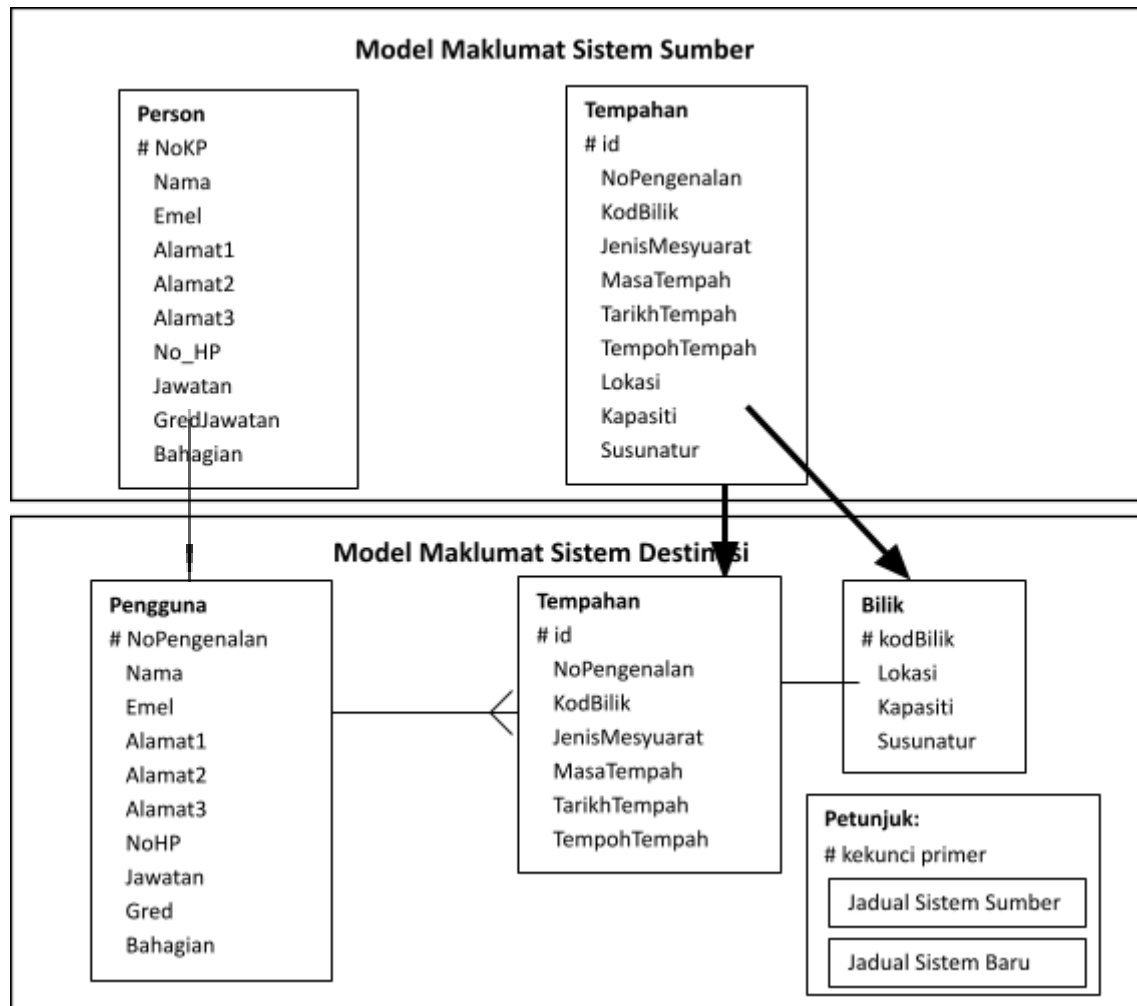
Jadual 4.30: Contoh Senarai Peraturan Pembersihan / Transformasi Data

PANGKALAN DATA		DB_SOURCE		PERATURAN PEMBERSIHAN/TRANSFORMASI DATA		
TABLE (JADUAL)		individu				
NO .	MEDAN	JENIS (LENGTH)	MANDAT ORI (Y/N)	KRITERIA	KAEDAH	CATATAN
i.	emel	varchar (20)	Y	data consistency	<i>Find and replace, trim data</i>	Tukar dalam format emel <u>xx@gov.my</u> , remove white space
ii.	tarikh	Date (8)	Y	format	Tukar ke format yang betul dd/mm/yyyy	Tukar format
iii.	alamat 1	100	N	data consistency	<i>Find and replace, trim data</i>	Tukar ringkasan 'jln' kepada 'jalan', remove white space

Langkah 5: Pemetaan Jadual

Langkah ini akan mengenal pasti bagaimana data daripada pangkalan data sistem **legasi** sumber dipetakan kepada pangkalan data sistem baharu. Proses ini melibatkan mengenalpastian jadual yang terlibat pada kedua-dua pangkalan data sistem sumber dan sistem baharu. Sumber rujukan bagi proses ini adalah Model Maklumat Logikal bagi pangkalan data sistem yang terlibat.

Contoh mengenalpastian dan pemetaan jadual yang dijalankan dalam proses migrasi adalah seperti berikut:



Rajah 4.49 : Pemetaan Jadual antara Pangkalan Data Sistem Sumber dan Sistem Baharu

Langkah 6: Pemetaan medan data

- Kenal pasti elemen data berdasarkan kepada Skema Logikal Pangkalan Data bagi sistem sumber serta sistem baharu.
- Bangunkan peraturan bisnes dalam mengendalikan kesemua proses ini. Sebagai contoh:

- i) Alamat lokasi aset disimpan dalam 3 baris berlainan pada pangkalan data sumber iaitu *Address1*, *Address2* dan *Address3*. Tentukan sama ada alamat lokasi aset di destinasi baharu akan mengikut format sumber atau disimpan dalam 1 baris sahaja pada *field* "Alamat".
 - ii) Tinggi dan lebar aset direkodkan dalam format sentimeter pada pangkalan data sistem baharu. Oleh yang demikian, data dalam format inci pada pangkalan data sistem sumber perlu ditukar kepada sentimeter.
 - iii) Kod aset pada pangkalan data sistem baharu akan diselaraskan menggunakan kod standard yang telah ditetapkan.
- c) Lengkapkan **Apendiks 8a** Templat Pemetaan Data Migrasi.

Contoh pengisian pemetaan data :

Jadual 4.31 : Pemetaan Medan Data antara Sistem Legasi dan Sistem Baharu

Bi l.	MAKLUMAT PANGKALAN DATA SUMBER						MAKLUMAT PANGKALAN DATA DESTINASI							Catat an
	Penyedia (SISTEM SUMBER MANUSIA)						Penerima (SISTEM TEMPAHAN BILIK MESYUARAT)							
	Jadua l	Med an	Jeni s	P/ F	Manda tori /Tidak	Ko d Sta nd ard	Jadual	Meda n	Keteran gan	Je nis	P/ F	Mandat ori /Tidak	Null / Tida k	
1	PERS ON	EP D_ NA MA	Varc har (100)		Manda tori		PENGG UNA	NAMA	Nama pegaw ai	Varc har (100)		Mandat ori	Tidak	
2	PERS ON	EPD _NO KP	Varc har (16)	P	Manda tori	Ya	PENGG UNA	NO_ PEN GEN ALA N	No. Kad Pengen alan	Varc har (16)	P	Mandat ori	Tidak	
3	PERS ON	EM EL	Varc har (50)		Tidak		PENGG UNA	EMEL	Alamat emel pegaw ai	Varc har (50)		Tidak	Tidak	

	TEMPAHAN	KODBIK	Integer		Mandatori		TEMPAHAN	KODBILIK	kod bilik	Integer	F	Mandatori	Tidak	
	TEMPAHAN	KODBIK	Integer		Mandatori		BILIK	KODBILIK	kod bilik	integer	P	Mandatori	Tidak	
	TEMPAHAN	JENISMESYUARAT_	Varchar (100)		Mandatori		TEMPAHAN	JENISMESYUARAT	nama mesyuarat	Varchar (100)		Mandatori	Tidak	
	TEMPAHAN	LOKASI	Varchar (100)		Mandatori		BILIK	LOKASI	lokasi bilik mesyuarat	Varchar (100)		Mandatori	Tidak	

Nota:

- i) P – Kekunci Primer
- ii) F – Kekunci Asing
- iii) Kod Standard – kod standard untuk jadual rujukan yang telah ditetapkan

Langkah 7: Sediakan keperluan pemetaan kod dan rekod disebabkan perubahan dasar

Proses pemetaan kod dijalankan sekiranya ada keperluan untuk menyesuaikan dan menyelaraskan kod (atau ID) bagi atribut daripada sumber data dengan destinasi baharu data. Proses ini biasanya dijalankan ke atas jadual rujukan.

Adakalanya perubahan dasar menyebabkan perubahan ke atas data dalam rekod maklumat dan memerlukan pemetaan data bagi menyesuaikan dan menyelaraskan rekod data.

- a) Contoh bagi keperluan pemetaan rekod (data) disebabkan perubahan dasar adalah seperti berikut.

Perubahan gred di SSM kepada gred baharu di SSPA seperti jadual dibawah.

SKIM PERKHIDMATAN : Pegawai Teknologi Maklumat					
GRED SSM	F41	F44	F48	F52	F54

GRED SSPA	F9	F10	F12	F13	F14
------------------	----	-----	-----	-----	-----

- b) Dalam sistem lama, rekod skim dan gred adalah dalam jadual seperti dibawah.

Jadual 4.32: Jadual Sumber Asal ref_skimJwtn

Kod	Keterangan
F41	Pegawai Teknologi Maklumat gred F41
F44	Pegawai Teknologi Maklumat gred F44
F48	Pegawai Teknologi Maklumat gred F48
F52	Pegawai Teknologi Maklumat gred F52
F54	Pegawai Teknologi Maklumat gred F54

Manakala dalam pangkalan data sistem baru pula, kod bagi gred SSPA adalah seperti jadual dibawah.

Jadual 4.33: Jadual Destinasi Baru ref_skimGredJwtn

Kod	Keterangan
F9	Skim jawatan ICT untuk gred F9
F10	Skim jawatan ICT untuk gred F10
F12	Skim jawatan ICT untuk gred F12
F13	Skim jawatan ICT untuk gred F13
F14	Skim jawatan ICT untuk gred F14

Jadual 'pekerja' dalam pangkalan data sumber adalah seperti berikut.

Jadual 4.34: Jadual Sumber Asal 'pekerja'

idPekerja	nama	gaji	gred
0000001	Ali bin Abu	10,070.00	F54
0000002	Aminah binti Aziz	8,800.00	F52
0000003	Latifah binti Mazlan	6500.00	F48
0000004	Suriya binti Abdullah	3500.00	F44

0000005	Hassan bin Abu	2100.00	F41
---------	----------------	---------	-----

c) Bagi keperluan pemetaan kod dan rekod disebabkan perubahan dasar, berdasarkan pernyataan a) dan b) di atas, berikut adalah langkah-langkah yang perlu diambil bagi melaksanakan pemetaan kod dan rekod (data).

i) Kenal pasti jadual dan medan di pangkalan data sistem lama yang mengandungi data yang perlu dikemaskini.

Pangkalan Data Sumber	
Jadual	Medan
ref_skimJwtn	• kod
pekerja	• Gaji • gred

ii) Sediakan peraturan pemetaan data bagi rekod yang terlibat seperti dibawah.

Data	Peraturan /Nota
Gaji pokok SSPB	$\text{Pelarasan Gaji pokok SSPA} = \text{gaji pokok SSM} + (\text{gaji pokok SSM} \times 8\%);$ atau $\text{Gaji pokok SSPA} = \text{gaji pokok SSM} + \text{RM240};$

d) Pemetaan bagi rekod yang menunjukkan maklumat pelarasan gaji, gaji pokok SSPN, gred lama dan gred baru adalah seperti jadual di bawah.

Jadual 4.35 : Contoh Pemetaan Rekod Yang Menunjukkan Maklumat Perubahan Gred

idPekerja	Pelarasan Gaji GP (8%)			Gred lama	Gred baru
	Gaji Pokok SSN	Pelarasan	Gaji Pokok SSPB		
0000001	10,070.00	805.60	10,875.60	F54	F14

0000002	8,800.00	704.00	9,504.00	F52	F13
0000003	6500.00	520.00	7,020.00	F48	F12
0000004	3500.00	280.00	3,780.00	F44	F10
0000005	2100.00	240.00	2,340.00	F41	F9

- e) Rekod data baharu selepas pelarasan yang dimasukkan dalam jadual 'pesonel' di pangkalan data baharu adalah seperti jadual di bawah.

Jadual 4.36 : Contoh Jadual Pesonel di Pangkalan Data Baharu

idPesonel	namaPesonel	tarikhMulaKhidmat	gaji	gred
0000001	Ali bin Abu	01 April 1985	10,875.60	F14
0000002	Aminah binti Aziz	15 Januari 1990	9,504.00	F13
0000003	Latifah binti Mazlan	01 Mac 2005	7,020.00	F12
0000004	Suriya binti Abdullah	01 Mei 2009	3,780.00	F10
0000005	Hassan bin Abu	01 Jun 2017	2,340.00	F9

Langkah 8: Dokumentasikan Spesifikasi Migrasi Data

- a) Spesifikasi Migrasi Data perlu dibangunkan dan dijadikan rujukan dalam pelaksanaan migrasi data bersama-sama Pelan Migrasi Data. Spesifikasi ini akan mendokumentasikan langkah 1 hingga 7 yang telah diterangkan di atas iaitu mengandungi perkara seperti berikut:

Jadual 4.37 : Isi Kandungan Spesifikasi Migrasi Data

Tajuk	Isi Kandungan
Tujuan Dokumen	Penerangan tujuan dokumen dihasilkan adalah untuk merekodkan maklumat bagi reka bentuk migrasi data bagi tujuan memindahkan data daripada pangkalan data sumber ke destinasi pangkalan data baharu.
Maklumat Sistem	Maklumat sistem dan pangkalan data yang terlibat bagi tujuan migrasi data daripada sumber asal ke destinasi baharu direkodkan. Ini bertujuan bagi memudahkan penyelarasan dan konfigurasi sistem untuk tujuan migrasi data dijalankan.

Tajuk	Isi Kandungan
Pembersihan Data	Penerangan bagi pembersihan data yang dilakukan ke atas data di sumber asal bagi mendapatkan data yang berkualiti semasa melaksanakan migrasi data.
Pemetaan Jadual	Penerangan dan rajah pemetaan jadual yang terlibat pada kedua-dua pangkalan data sistem legasi dan sistem baharu.
Pemetaan Medan Data	Penerangan dan jadual yang mengandungi perincian data yang perlu dipindahkan perlu dikenal pasti berdasarkan kepada Skema Logikal Pangkalan Data bagi sistem legasi serta sistem baharu.
Peraturan Bisnes	Senarai peraturan bisnes yang perlu dipatuhi dalam proses migrasi data yang telah ditetapkan untuk menyelaraskan format data yang dipindahkan.
Pemetaan Kod	Penerangan dan jadual yang mengandungi perincian kod bagi atribut yang perlu diselaraskan antara pangkalan data sistem legasi dan sistem baharu.

Spesifikasi Migrasi Data perlu dikemaskini dari semasa ke semasa mengikut keperluan yang telah dipersetujui.

- b) Rujuk format dokumen **D06 Spesifikasi Migrasi Data**. Dokumen ini juga boleh dijadikan sebagai lampiran dalam **D05 Pelan Migrasi Data**.
- c) Langkah seterusnya iaitu proses Pelaksanaan Migrasi Data **[F6.1]** yang mana terdiri daripada aktiviti Pelaksanaan, Pengujian dan Penyediaan Laporan Migrasi data.

Langkah 9: Sahkan Spesifikasi Migrasi Data

D06 Spesifikasi Migrasi Data yang didokumenkan perlu dibentang dan mendapat pengesahan pemilik sistem bagi memastikan kesahihan dan spesifikasi yang dihasilkan memenuhi keperluan migrasi data.

Rujukan

1. Oracle White Paper (2011). Successful Data Migration. <http://www.oracle.com/technetwork/middleware/oedq/successful-data-migration-wp-1555708.pdf>
2. Credosoft White Paper. Eight key steps which help ensure a successful data migration project: A white paper for inspection management professionals. <http://credosoft.com/wp/wp-content/uploads/2014/01/Eight-key-steps-which-help-ensure-a-successfu-data-migration-project.pdf>
3. SAGA Group (2012). Methods of Data Migration
4. Astera Whitepaper: Data Profiling: The First Step in the Data Migration Process

4.12 SPESIFIKASI INTEGRASI SISTEM [F3.10]

4.12.1 KETERANGAN

Integrasi sistem dilaksanakan bagi membolehkan sistem-sistem aplikasi yang berasingan bertukar maklumat secara automatik dan *seamless*. Pelaksanaan proses bisnes adakalanya memerlukan interaksi dengan sistem aplikasi atau modul yang lain untuk melaksanakan fungsinya.

4.12.2 OBJEKTIF

Menghasilkan spesifikasi integrasi sistem yang merangkumi pelaksanaan proses perkhidmatan dan reka bentuk integrasi yang merentasi fungsi agensi atau unit dalam agensi.

4.12.3 LANGKAH-LANGKAH

Langkah 1 : Kemaskini keperluan integrasi yang telah dikenal pasti

Kemas kini keperluan integrasi yang telah dikenalpasti dalam Pelan Integrasi Data sekiranya ada.

Langkah 2 : Kenal Pasti Servis Integrasi Yang Diperlukan

- a. Keperluan integrasi yang diperincikan adalah berdasarkan Rajah Aliran Proses Bisnes (PFD) dalam **D02 Spesifikasi Keperluan Bisnes**. Maklumat integrasi tersebut boleh juga merujuk kepada pemodelan *Use Case* dalam **D03 Spesifikasi Keperluan Sistem**.
- b. Maklumat yang perlu ada bagi keperluan servis integrasi adalah seperti di **Apendiks 9**
 - a) Maklumat Servis Integrasi. Jadual di bawah menunjukkan keterangan templat maklumat servis integrasi.

Jadual 4.38: Keterangan Templat Maklumat Servis Integrasi

Nama Label	Keterangan
Id Servis Integrasi	Label Id servis integrasi
Rujukan Fungsi	Label (ID fungsi bisnes) dan nama Rajah Aliran Proses Bisnes yang dirujuk
Rujukan Aktiviti Bisnes	Label (ID aktiviti bisnes) dan nama aktiviti bisnes dari rajah aliran proses yang dirujuk dalam Dokumen Keperluan Bisnes
Rujukan Aktiviti	Label dan nama <i>Use Case</i> yang dirujuk

Nama sistem luaran yang terlibat	Sistem yang membekalkan data (sistem sumber) atau sistem yang menerima data (sistem destinasi)
Inbound / Outbound	Data masuk (Inbound) atau data keluar (outbound)
Pemilik maklumat	Bahagian yang bertanggungjawab atas maklumat tersebut
Keterangan maklumat yang dihantar	Penerangan ringkas mengenai maklumat yang dihantar ke sistem lain

Contoh Kajian Kes:

Rujukan Fungsi **PFD-BM-MP-SB Selenggara Bilik Mesyuarat** menunjukkan bahawa integrasi berlaku semasa menguruskan aduan kerosakan. Perincian aktiviti berkaitan integrasi tersebut boleh dirujuk pada Pemodelan Proses Bisnes (PFD), **PFD-BM-MA** dan Rajah Use Case **UC-BM-PB-AK**. Senaraikan maklumat yang diperlukan seperti jadual berikut:

Jadual 4.39 : Contoh Maklumat Aktiviti yang Memerlukan Integrasi

Bil	ID Servis Integrasi	Rujukan Fungsi	Rujukan Aktiviti Bisnes	Rujukan Aktiviti	Nama Sistem Luar yang terlibat	Inbound / outbound	Pemilik Maklumat	Keterangan Maklumat yang Dihantar
1	SI-BM-PB-AK-01	PFD-BM-MP-SB Selenggara Bilik Mesyuarat	PFD-BM-MP-SB-03 Hantar Aduan Kerosakan	UC-BM-PB-AK-01 Membuat Aduan Kerosakan	Sistem Pengurusan Tempahan Bilik Mesyuarat	outbound	Unit Pentadbiran, Bahagian Khidmat Pengurusan (BKP)	Aduan baru bagi kerosakan bilik mesyuarat
2	SI-BM-MA-01-002	PFD-BM-MP-SB Selenggara Bilik Mesyuarat	PFD-BM-MP-SB-04 Kemas kini Status Bilik Mesyuarat	UC-BM-MA-01-02 Kemaskini Status Kesediaan Bilik Mesyuarat	Sistem Selenggara Aset	inbound	Unit Pentadbiran, Bahagian Khidmat Pengurusan (BKP)	Aduan kerosakan yang telah selesai

Berdasarkan jadual di atas, terdapat dua servis integrasi yang perlu dibangunkan iaitu:

- Sistem Pengurusan Tempahan Bilik Mesyuarat akan menghantar maklumat aduan bagi kerosakan bilik mesyuarat.
- Sistem Selenggara Aset akan menghantar maklumat aduan kerosakan yang telah selesai.

Langkah 3: Kenal pasti pendekatan integrasi data

Kenal pasti pendekatan integrasi data yang dilaksanakan mengikut keperluan berdasarkan data yang diintegrasikan dengan merujuk maklumat di Langkah 2. Antara pendekatan yang boleh digunakan adalah seperti dibawah.

a) Konsolidasi Data

Konsolidasi data adalah untuk mengekstrak data daripada sumber berbeza, membersihkan, dan menyimpan data fizikal di lokasi penyimpanan akhir. Pendekatan ini adalah sebagai penyelesaian kepada silo data dan mengurangkan kos infrastruktur.

b) Replikasi Data

Replikasi data adalah mencipta salinan data dari satu pangkalan data sumber ke pangkalan data yang lain.

c) Virtualisasi Data

Virtualisasi data tidak memindahkan data antara sistem tetapi mencipta view yang menyatukan semua sumber data. Kaedah ini tidak memindahkan data antara pangkalan data semasa virtualisasi data. Sebaliknya, ia memaparkan data dari satu atau pelbagai sumber setelah menerima pertanyaan (query).

d) Federasi Data

Federasi data melibatkan pengwujudan satu pangkalan data maya dari pelbagai sumber data tanpa perlu memindah atau menyalin data ke satu tempat.

Kenal pasti juga teknik yang digunakan dalam melaksanakan integrasi. Antaranya ialah:

a) Extract, Transform and Load (ETL)

ETL bermaksud ekstrak, transformasi, dan muat. Ia dimulai dengan mengekstrak data dari sumber yang berbeza. Seterusnya, ia menukar data (transformasi) mengikut peraturan bisnes dan format yang tertentu sebelum menghantar atau muat ke pangkalan data atau gudang data destinasi. ETL adalah teknik yang sering digunakan untuk integrasi secara berkelompok (batch).

b) Extract, Load and Transform (ELT)

ELT bermaksud ekstrak, muat, dan transformasi. Kaedah ini adalah serupa dengan ETL, kecuali urutan yang berbeza. Ia dimulai dengan mengekstrak data dari sumber yang berbeza. Seterusnya data dimuatkan ke dalam pangkalan data yang tidak berstruktur dan diubah (transformasi) hanya apabila diperlukan. ELT adalah metod yang sering digunakan untuk integrasi secara strim masa nyata (real time streaming) dalam volum yang banyak dan kekerapan yang tinggi.

Kemaskini keperluan integrasi yang telah disenaraikan dalam Langkah 2 dengan melengkapkan pendekatan integrasi data.

Jadual 4.40 : Contoh Maklumat Pendekatan Integrasi Data

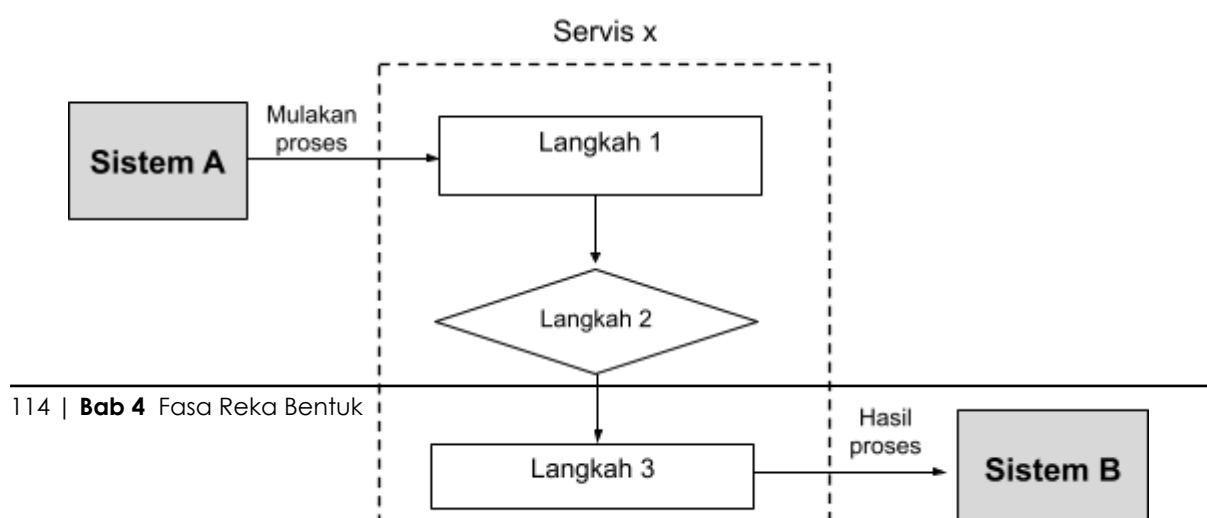
Nama Label	Keterangan
Id Servis Integrasi	Label Id servis integrasi
Nama Servis	
Rujukan Fungsi	Label (ID fungsi bisnes) dan nama Rajah Aliran Proses Bisnes yang dirujuk
Rujukan Aktiviti Bisnes	Label (ID aktiviti bisnes) dan nama aktiviti bisnes dari rajah aliran proses yang dirujuk dalam Dokumen Keperluan Bisnes
Rujukan Aktiviti	Label dan nama <i>Use Case</i> yang dirujuk
Nama sistem luaran yang terlibat	Sistem yang membekalkan data (sistem sumber) atau sistem yang menerima data (sistem destinasi)
Inbound / Outbound	Data masuk (Inbound) atau data keluar (outbound)
Pemilik maklumat	Bahagian yang bertanggungjawab atas maklumat tersebut
Keterangan maklumat yang dihantar	Penerangan ringkas mengenai maklumat yang dihantar ke sistem lain
Pendekatan Integrasi	Sama ada penyatuan data/replikasi data/ virtualisasi data / federasi data
Teknik Integrasi	Sama ada ETL atau ELT

Langkah 4 : Sediakan Aliran Proses Integrasi Sistem

Kenal Pasti aliran proses yang terlibat semasa pertukaran data berlaku. Nyatakan peraturan yang dilaksanakan semasa pertukaran data tersebut seperti logik integrasi dan transformasi data. Terdapat 2 jenis proses integrasi iaitu:

a) Proses Khusus (*Specialized Processes*)

Proses Khusus merupakan pengintegrasian untuk melaksanakan proses yang khusus untuk sistem tertentu. Ini bermakna hanya ada satu output sahaja yang dihasilkan oleh proses tersebut untuk diproses oleh sistem tertentu.

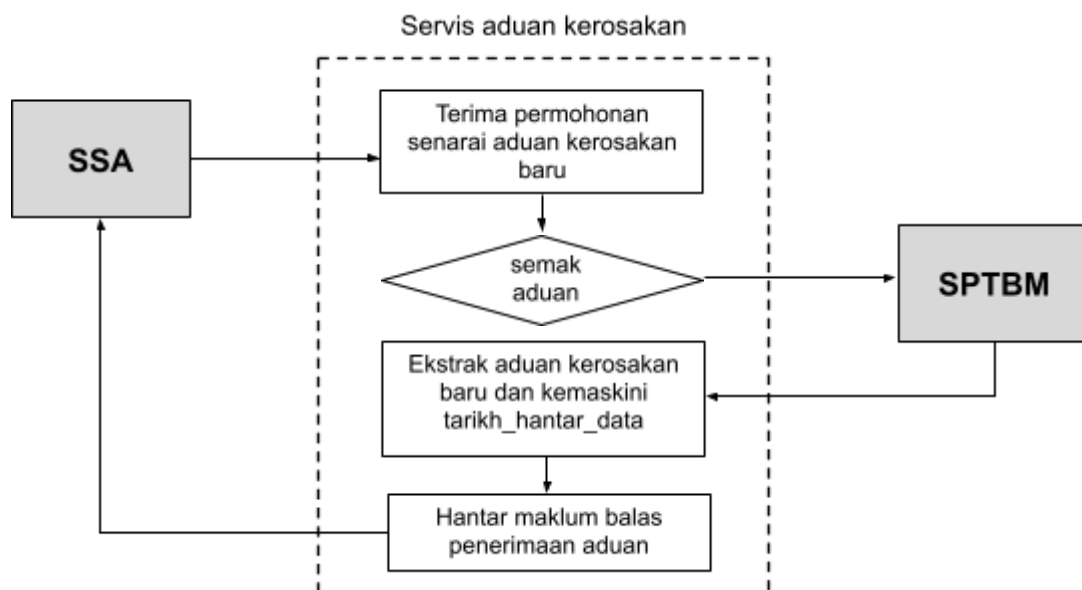


Rajah 4.50 : Proses Khusus

Berdasarkan rajah di atas, Sistem A akan memulakan Proses X yang terdapat pada server integrasi. Output yang dihasilkan oleh Proses X akan dihantar kepada Sistem B untuk diproses. Proses X ini diwujudkan untuk menyokong Sistem B agar sistem lain boleh melaksanakan prosedur tersebut tanpa perlu mengetahui senarai proses yang perlu dilakukan.

Contoh Kajian Kes:

Sistem Selenggara Aset (SSA) akan memohon maklumat aduan kerosakan daripada Sistem Pengurusan Tempahan Bilik Mesyuarat (SPTBM). Sistem Pengurusan Tempahan Bilik Mesyuarat akan mendapatkan maklumat aduan kerosakan baru dan menghantar maklumat tersebut bersama tarikh_hantar_data yang telah dikemaskini. Sistem Selenggara Aset akan menerima maklumat tersebut dan menyimpannya untuk tindakan seterusnya.

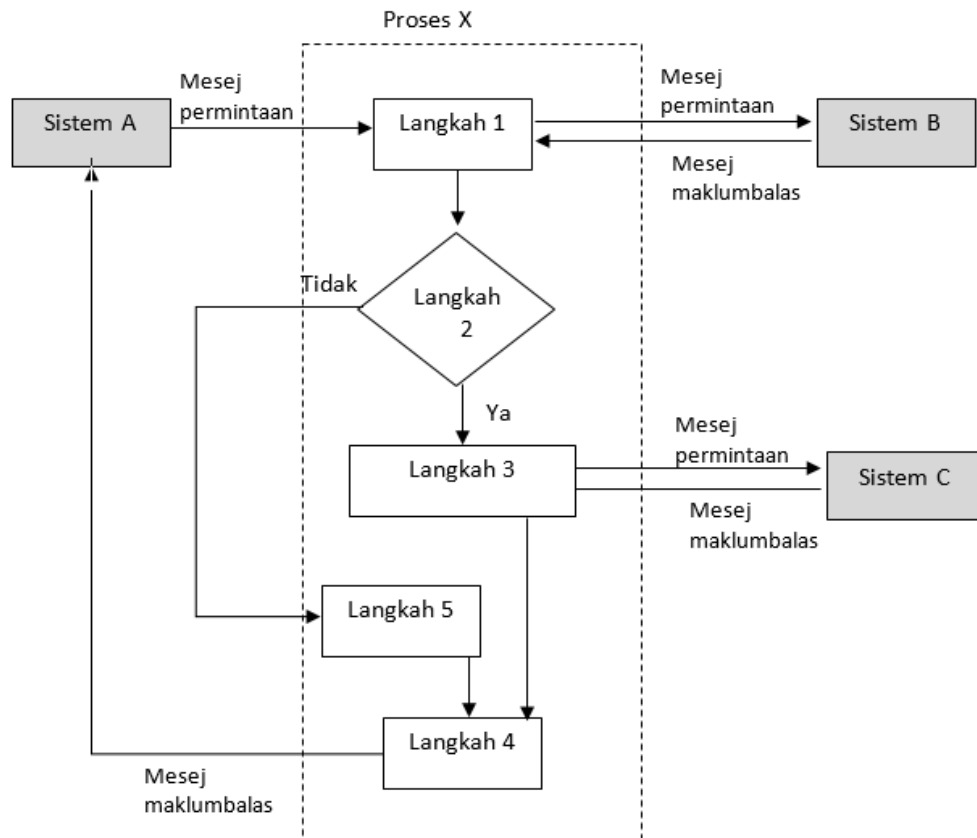


Rajah 4.51 : Contoh Proses Integrasi Servis Aduan Kerosakan

b) Proses Berbilang Langkah (*Multistep Process*)

Proses Berbilang Langkah melibatkan pengintegrasian lebih dari satu sistem untuk melaksanakan proses tertentu. Aliran proses yang diwujudkan dalam server integrasi akan mengendalikan interaksi antara sistem tersebut. Semua transaksi antara proses

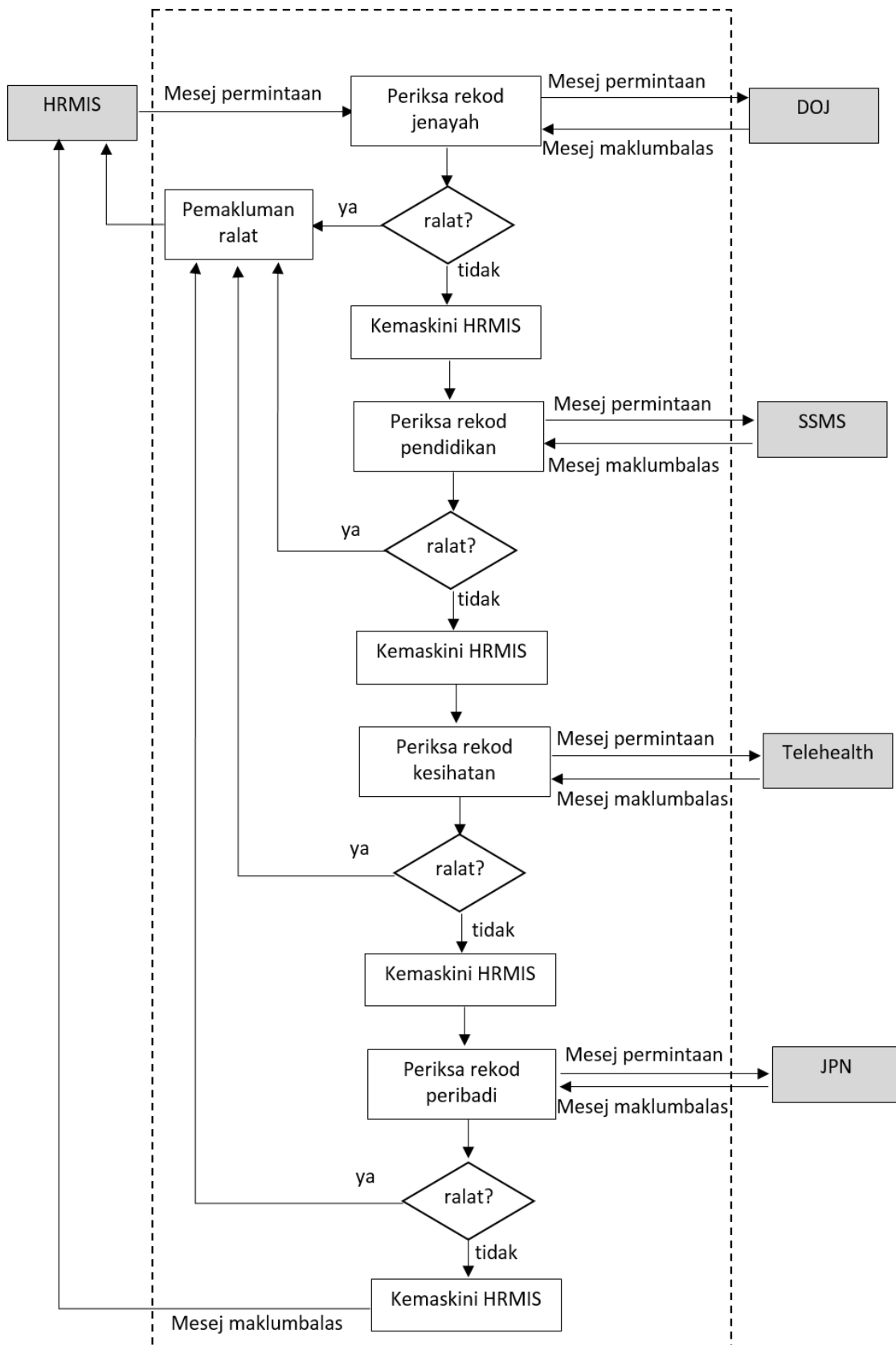
dan sistem tersebut perlu mematuhi kekangan transaksi *atomic*. Ini bermakna sekiranya salah satu transaksi gagal, maka proses tersebut akan gagal. Maka semua perubahan yang telah berlaku kepada data perlu kembali kepada asal seolah-olah proses tersebut tidak berlaku. Mekanisma pengurusan pengecualian (*exception handling mechanism*) perlu diwujudkan untuk melaksanakan pembalikan (*rollback*) tersebut.



Rajah 4.52 : Proses Berbilang Langkah

Rajah di atas menunjukkan Sistem A perlu berintegrasi dengan Sistem B dan Sistem C untuk melaksanakan Proses X. Mesej permintaan digunakan untuk meminta menggunakan servis atau data yang dikongsi oleh sistem lain manakala mesej maklumbalas ialah hasil proses servis atau data yang diminta oleh mesej permintaan tersebut.

Contoh Senario Proses Integrasi:



Rajah 4.53 : Contoh Senario Proses Integrasi

Berdasarkan proses tersebut, Sistem Maklumat Rekod Pekerja (HRMIS) akan menghantar maklumat pekerja kepada proses integrasi untuk memeriksa rekod jenayah daripada Sistem Maklumat Kehakiman (DOJ), rekod pendidikan daripada Sistem Maklumat Pendidikan (SSMS), rekod kesihatan daripada Sistem Maklumat Kesihatan (Telehealth) dan rekod peribadi daripada Sistem Maklumat Pendaftaran (JPN). Jika semua rekod memenuhi kriteria, proses pengemaskinian akan dilakukan kepada sistem Telehealth dan HRMIS. Jika berlaku kegagalan semasa transaksi, mekanisma pengurusan pengecualian (exception handling mechanism) akan memainkan peranan untuk melaksanakan pembalikan (rollback) proses tersebut. Segala perubahan yang telah berlaku kepada sistem yang terlibat akan dibatalkan dan proses tersebut perlu diulang semula.

Langkah 5 : Muktamadkan Format Pertukaran Data (*Data Exchange Format*)

- a) Berdasarkan senarai integrasi yang telah dikenalpasti, berikan nama servis untuk setiap integrasi tersebut dan pendekatan kaedah integrasi yang digunakan. Terangkan format pertukaran data yang akan digunakan. Rujuk **Apendiks 9 b)** Format Pertukaran Data. Keterangan bagi templat Format Pertukaran Data adalah seperti jadual di bawah.

Jadual 4.41 : Keterangan Templat Format Pertukaran Data

Nama Label	Keterangan
ID Servis	Label Id servis integrasi yang ingin dibangunkan
Nama Servis	Nama servis integrasi yang ingin dibangunkan.
Keterangan	Penerangan mengenai servis integrasi yang dibangunkan.
Kaedah Integrasi	Kaedah integrasi yang digunakan dan format pertukaran data yang digunakan.
URL <i>Web Service</i>	Alamat URL servis integrasi tersebut. Nyatakan URL yang digunakan semasa pembangunan, pengujian dan pelaksanaan (jika berkaitan).
Sistem Sumber	Sistem yang membekalkan data
Sistem Destinasi	Sistem yang menerima data
Jenis Transaksi	Kekerapan (semanan/ rekod) dan jenis transaksi integrasi data (rekod tunggal/banyak)
Penyulitan Data	Nyatakan sekiranya data perlu disulitkan (encryption)
Nama Fail	Nama fail bagi servis integrasi yang dibangunkan
<i>Request</i>	Skrip permohonan data

Data yang terlibat (Request)	Penerangan struktur data yang terlibat dalam permohonan (request) iaitu termasuklah <i>header</i> , butiran rekod dan sistem log.
<i>Respond</i>	Skrip maklum balas data yang dipohon.
Data yang terlibat (Respond)	Penerangan struktur data yang terlibat dalam maklum balas iaitu termasuklah <i>header</i> , butiran rekod dan sistem log.

Contoh Kajian Kes:

Berdasarkan contoh kajian kes, jadual di bawah menunjukkan keterangan servis integrasi dan data yang terlibat dalam integrasi tersebut.

Jadual 4.42 : Contoh Penerangan Servis Integrasi dan Data yang Terlibat

Id Servis	SI-BM-PB-AK-01				
Nama Servis	Aduan Kerosakan				
Keterangan	Menghantar maklumat aduan kerosakan baru daripada Sistem Pengurusan Tempahan Bilik Mesyuarat (sistem sumber) ke Sistem Selenggara Aset (sistem destinasi)				
Kaedah Integrasi	RESTful service Format: JSON				
Url Web Service	Testing: testing.company.com.my/sptbm/rest/integration Production: www.sptbm.my/rest/integration				
Sistem Sumber	Sistem Selenggara Aset				
Sistem Destinasi	Sistem Pengurusan Tempahan Bilik Mesyuarat				
Jenis Transaksi	Penghantaran rekod tunggal				
Penyulitan Data	Tiada				
Nama Fail	AduanRosak_ddMMyyyy.json				
Request	<pre>{ "hantar_aduan" : { "aduan" : [{ "id_aduan" : , "id_bilik_mesy" : , "id_jenis_kerosakan": , "keterangan_aduan" : , "no_ic_daftar_aduan" : , "tarikh_daftar_aduan" : }] } }</pre> Arahan: POST				
Data yang terlibat (Request)	Nama	Jenis	Sai z	Nullab le	Rules
	id_aduan	integer		N	

	id_bilik_mesy	integer		N																
	id_jenis_kerosakan	integer		N																
	no_ic_daftar_aduan	varchar2	12	N																
	tarikh_daftar_aduan	date		N	dd/MM/yyyy															
	keterangan_aduan	clob		N																
Respond	{ “result” : { “aduan” : [{ “id_aduan” : , “id_status_aduan” : , “mesej_penerimaan_aduan” }] } }																			
Data terlibat yang	<table><tr><th>Nama</th><th>Jenis</th><th>Sai z</th><th>Nullable</th><th>Rules</th></tr><tr><td>id_aduan</td><td>integer</td><td></td><td>N</td><td></td></tr><tr><td>id_status_aduan</td><td>integer</td><td></td><td>N</td><td></td></tr></table>					Nama	Jenis	Sai z	Nullable	Rules	id_aduan	integer		N		id_status_aduan	integer		N	
Nama	Jenis	Sai z	Nullable	Rules																
id_aduan	integer		N																	
id_status_aduan	integer		N																	

Jadual di atas menerangkan servis integrasi yang akan dibangunkan untuk membolehkan Sistem Pengurusan Tempahan Bilik Mesyuarat menghantar maklumat aduan kerosakan bilik mesyuarat ke sistem luar (Sistem Selenggara Aset).

Langkah 6 : Sediakan Pemetaan Data (*Data Mapping*)

Kenal pasti struktur data yang diperlukan oleh sistem dan struktur data yang dihantar oleh sistem sumber. Petakan struktur data tersebut. **Apendiks 9 c) Pemetaan Struktur Data.** Keterangan bagi templat pemetaan struktur data adalah seperti di jadual di bawah.

Jadual 4.43 : Keterangan Templat Pemetaan Struktur Data

Nama Label	Keterangan
ID Servis	Label Id servis integrasi.
Nama Servis	Nama servis integrasi.

Nama Sistem Penerima	Sistem yang memohon dan menerima data tersebut melalui servis integrasi. Butiran pemetaan struktur data. <table border="1" data-bbox="572 367 1386 672"> <tr> <td>Nama medan</td><td>Nama medan data</td></tr> <tr> <td>Jenis</td><td>Jenis medan data</td></tr> <tr> <td>Saiz</td><td>Saiz medan</td></tr> <tr> <td>Keterangan</td><td>Keterangan lanjut berkaitan medan (format, peraturan atau maklumat berkaitan)</td></tr> </table>	Nama medan	Nama medan data	Jenis	Jenis medan data	Saiz	Saiz medan	Keterangan	Keterangan lanjut berkaitan medan (format, peraturan atau maklumat berkaitan)
Nama medan	Nama medan data								
Jenis	Jenis medan data								
Saiz	Saiz medan								
Keterangan	Keterangan lanjut berkaitan medan (format, peraturan atau maklumat berkaitan)								
Nama Sistem Pemilik	Sistem yang menyedia dan menghantar data tersebut kepada pemohon melalui servis integrasi. Butiran pemetaan struktur data. <table border="1" data-bbox="572 938 1386 1243"> <tr> <td>Nama medan</td><td>Nama medan data</td></tr> <tr> <td>Jenis</td><td>Jenis medan data</td></tr> <tr> <td>Saiz</td><td>Saiz medan</td></tr> <tr> <td>Keterangan</td><td>Keterangan lanjut berkaitan medan (format, peraturan atau maklumat berkaitan)</td></tr> </table>	Nama medan	Nama medan data	Jenis	Jenis medan data	Saiz	Saiz medan	Keterangan	Keterangan lanjut berkaitan medan (format, peraturan atau maklumat berkaitan)
Nama medan	Nama medan data								
Jenis	Jenis medan data								
Saiz	Saiz medan								
Keterangan	Keterangan lanjut berkaitan medan (format, peraturan atau maklumat berkaitan)								

Contoh Kajian Kes:

Berdasarkan contoh kajian kes, jadual di bawah menunjukkan contoh pemetaan struktur data yang terlibat.

Jadual 4.44 : Contoh Pemetaan Struktur Data

Id Servis	
SI-BM-PB-AK-01	
Nama Servis	
Aduan Kerosakan	
Nama Sistem Pemilik (Sistem sumber)	Nama Sistem Penerima (Sistem Destinasi)
Sistem Pengurusan Tempahan Bilik Mesyuarat	Sistem Selenggara Aset

Nama Medan	Jenis	Saiz	Keterangan	Nama Medan	Jenis	Saiz	Keterangan
id_aduan	number	12		no_aduan	integer		
id_bilik_mesy	number			id_bilik_mesyuarat	integer		
id_jenis_kerosakan	number			id_jenis_kerosakan	integer		
no_ic_daftar_aduan	string			id_pengguna	varchar	12	
tarikh_daftar_aduan	string		dd/MM/yyyy	tarikh_aduan	date		dd/MM/yyyy
keterangan_aduan	string	255		keterangan	text		
status_aduan	number			status_aduan	integer		

Jadual 62 menunjukkan pemetaan data aduan kerosakan bilik mesyuarat antara Sistem Selenggara Aset dengan Sistem Pengurusan Tempahan Bilik Mesyuarat. Ini untuk memastikan data yang diterima daripada Sistem Pengurusan Tempahan Bilik Mesyuarat memenuhi keperluan Sistem Selenggara Aset.

Langkah 7 : Sediakan Reka Bentuk Arkitektur Integrasi

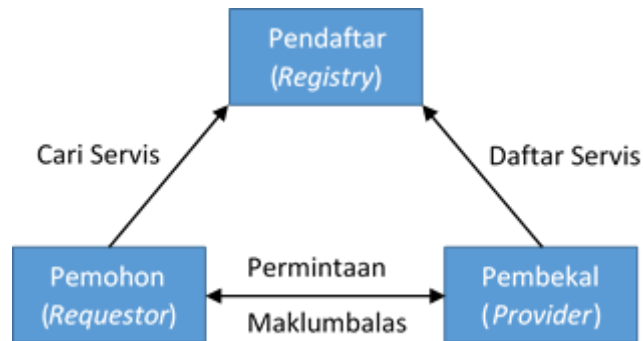
Arkitektur integrasi adalah seni bina yang menggambarkan aliran data antara pelbagai sistem/aplikasi untuk berkongsi data. Reka bentuk arkitektur integrasi adalah bergantung kepada cara dan peraturan integrasi tersebut dilakukan.

Pelaksanaan integrasi sistem menggunakan protokol yang standard bagi memenuhi keperluan pertukaran data antara sistem dan aplikasi yang berbeza. Protokol ini menentukan tatacara data diminta (requested), dihantar dan diproses antara sistem yang berintegrasi. Protokol dan arkitektur yang digunakan bagi pertukaran data dan mekanisme untuk penghantaran dan komunikasi data antara sistem dan aplikasi yang berbeza adalah seperti HTTP, SOAP dan REST. Piawaian ini adalah bagi memastikan keselamatan, integriti dan kebolehpercayaan data dipatuhi semasa pelaksanaan integrasi.

a) SOAP

SOAP bermaksud *Simple Object Access Protocol*. Terdapat tiga peranan utama dalam protokol pemesejan standard *web service*:

- i) Pembekal (*Provider*) - mewujudkan servis dan menjadikannya tersedia untuk sistem aplikasi yang ingin menggunakannya.
- ii) Pemohon (*Requestor*) - menggunakan servis sedia ada dengan menghantar permintaan kepada pembekal servis.
- iii) Pendaftar (*Registry*) – menyimpan maklumat servis yang disediakan oleh pembekal servis.



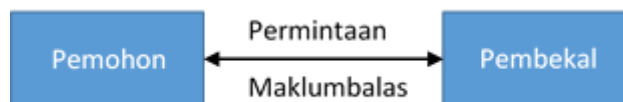
Rajah 4.54 : Arkitektur SOAP

Pembekal servis akan memaklumkan kepada pendaftar tentang servis yang disediakan dan cara menggunakannya. Pemohon akan mencari servis yang dikehendaki dalam pendaftar. Setelah mendapat maklumat mengenai servis yang diperlukan, pemohon akan menghantar permintaan kepada pembekal servis tersebut. Pembekal akan memberi maklum balas berdasarkan jenis permintaan tersebut.

b) **REST**

REST bermaksud *REpresentational State Transfer*. Ia mempunyai ciri-ciri berikut:

- i) Setiap sumber seperti maklumat boleh diakses melalui URL.
- ii) Kata arahan GET, POST, PUT atau DELETE akan digunakan semasa mengakses sumber tersebut.



Rajah 4.55 : Arkitektur RESTful

Arkitektur *RESTful* berkonsepkan *client-server*. Permintaan akan dilakukan oleh klien/pemohon dengan cara menghantar URL sumber yang diperlukan dan kata arahan. Pelayan/pembekal servis akan memberi maklum balas berdasarkan URL dan kata arahan permintaan tersebut.

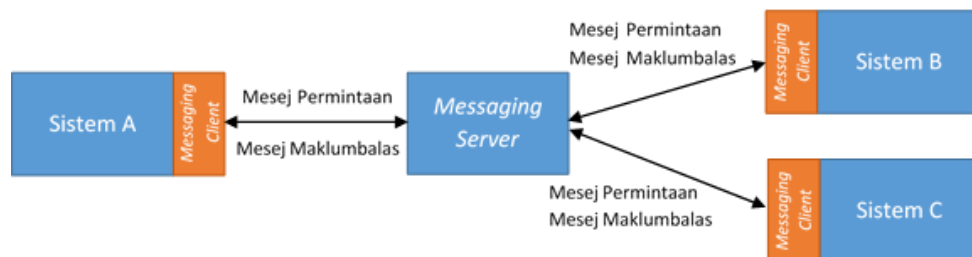
c) **Arkitektur Messaging**

Penghantaran Mesej (*Messaging*) membolehkan setiap sistem aplikasi berkomunikasi menggunakan mesej melalui saluran mesej (*message channel*) yang sama. Mesej yang dihantar oleh sistem aplikasi akan diuruskan oleh sistem penghantaran mesej (*messaging system*). Perkongsian data dan arahan adalah menggunakan mesej.



Rajah 4.56 : Arkitektur Messaging

Setiap sistem aplikasi akan dipasang dengan *messaging client* untuk membolehkannya menghantar dan menerima mesej. *Messaging server* bertindak sebagai orang tengah yang menguruskan penghantaran mesej.

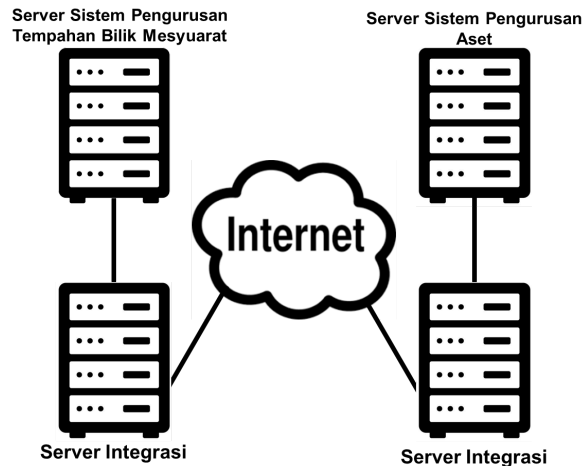


Rajah 4.57 : Arkitektur Messaging untuk Pengintegrasian Berbilang Server

Arkitektur *messaging* memudahkan pengintegrasian yang melibatkan proses berbilang langkah (*multi step*) kerana ia menyokong pengintegrasian berbilang sistem. Proses integrasi tersebut akan dibangunkan dalam *messaging server*.

Kenal pasti arkitektur integrasi dan terangkan arkitektur tersebut dengan menjelaskan komponen-komponen yang berkaitan seperti aliran data antara dua atau lebih sistem atau aplikasi atau komponen yang berintegrasi, protokol, kaedah integrasi dan metod yang digunakan.

Berdasarkan contoh kajian kes di Langkah 4, berikut adalah contoh arkitektur integrasi sistem bagi integrasi sistem antara Sistem Pengurusan Tempahan Bilik Mesyuarat dengan Sistem Pengurusan Aset.



Rajah 4.58 : Contoh Arkitektur Integrasi Sistem

Langkah 8 : Dokumentasikan Spesifikasi Integrasi sistem

Rekodkan hasil langkah-langkah yang telah dilaksanakan ke dalam dokumen **D08 Spesifikasi Integrasi Sistem**.

Jadual 4.45 : Kandungan Dokumen Spesifikasi Integrasi Sistem

Kandungan	Keterangan
Tujuan	Terangkan tujuan dokumen ini dihasilkan.
Rekabentuk Arkitektur Integrasi	Terangkan reka bentuk arkitektur integrasi menggunakan rajah yang bersesuaian.
Keperluan Servis Integrasi	Terangkan servis dan pendekatan integrasi yang telah ditetapkan seperti di Langkah 2 dan 3.
Aliran Proses Integrasi	Terangkan aliran proses integrasi bagi setiap servis integrasi yang dibangunkan.
Format Pertukaran dan Pemetaan Data	Terangkan format pertukaran data yang digunakan seperti di Langkah 5 dan pemetaan struktur data seperti di Langkah 6.

Spesifikasi Integrasi Sistem perlu dikemaskini dari semasa ke semasa mengikut keperluan yang telah dipersetujui.

Langkah 9 : Dapatkan Pengesahan Pengguna

Spesifikasi Integrasi Sistem yang didokumenkan perlu dibentang dan mendapat pengesahan pemilik sistem bagi memastikan kesahihan dan spesifikasi yang dihasilkan memenuhi keperluan integrasi sistem.

Rujukan

1. Gregor Hohpe & Bobby Woolf (2003). Enterprise Integration Patterns. Addison-Wesley. ISBN: 0321200683.
2. MSC CFI Architecture – System Integration Architecture Version 3.0 (2004).
3. Dokumen uCustoms Penang Port Sdn Bhd (PPSB) Integration Specification Issue 1.0 (2015).
4. <https://riverty.io/data-learning-center/data-integration-techniques-and-strategies/>
5. Principle of Data Integration (2014) Alon Halevy, AnHai Doan, and Zachary Ives.