

F3.1 REKA BENTUK ARKITEKTUR

TAKLIMAT
01

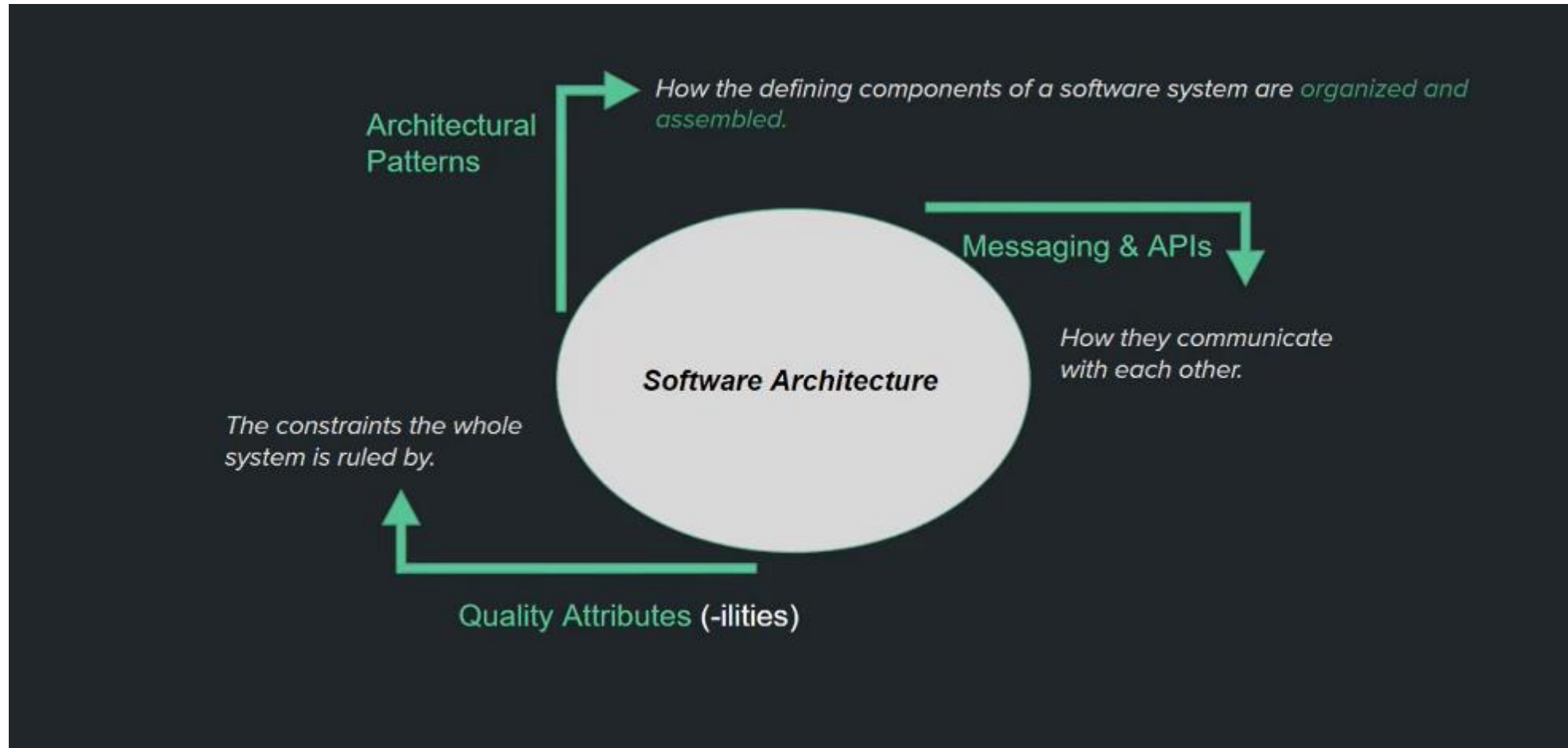


OBJEKTIF

- Menyediakan arkitektur sistem aplikasi yang terdiri daripada arkitektur keseluruhan sistem aplikasi, arkitektur aplikasi dan arkitektur pangkalan data berpandukan kepada keperluan-keperluan yang diperolehi di dalam fasa permulaan projek dan fasa analisis.

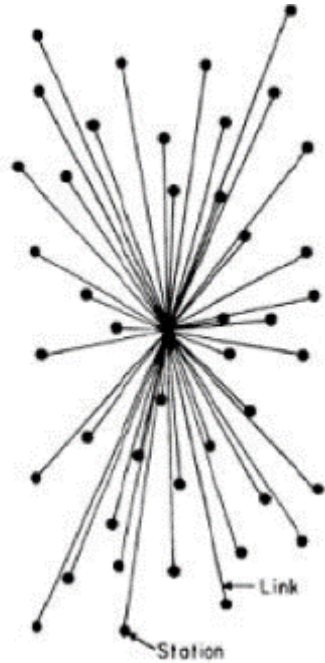
PENGENALAN

Reka bentuk arkitektur adalah penyusunan dan pengaturan struktur-struktur/komponen bagi sesuatu sistem aplikasi yang ingin dibangunkan.



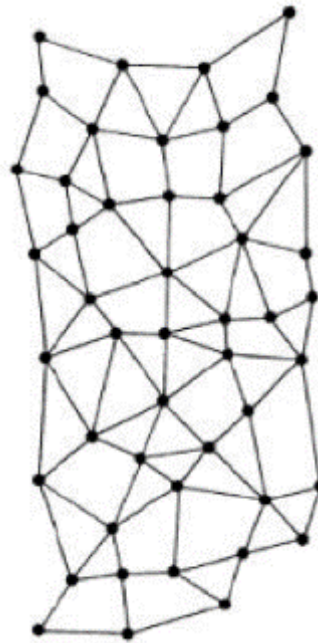
REKABENTUK ARKITEKTUR

Jenis-jenis arkitektur pengkomputeran :



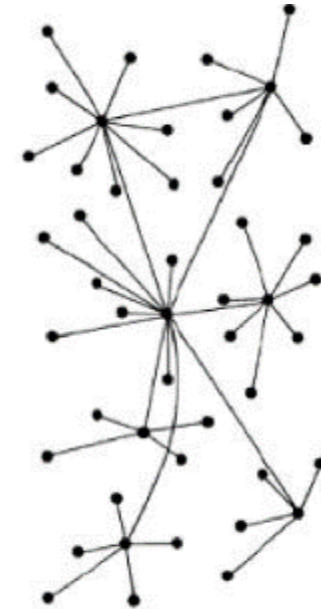
CENTRALIZED

Mengawal operasi dan aliran informasi secara berpusat.



DISTRIBUTED

Komputasi dilakukan dan dikongsi merentasi lokasi nod yang berbeza tanpa sebarang elemen kawalan pemusatan.



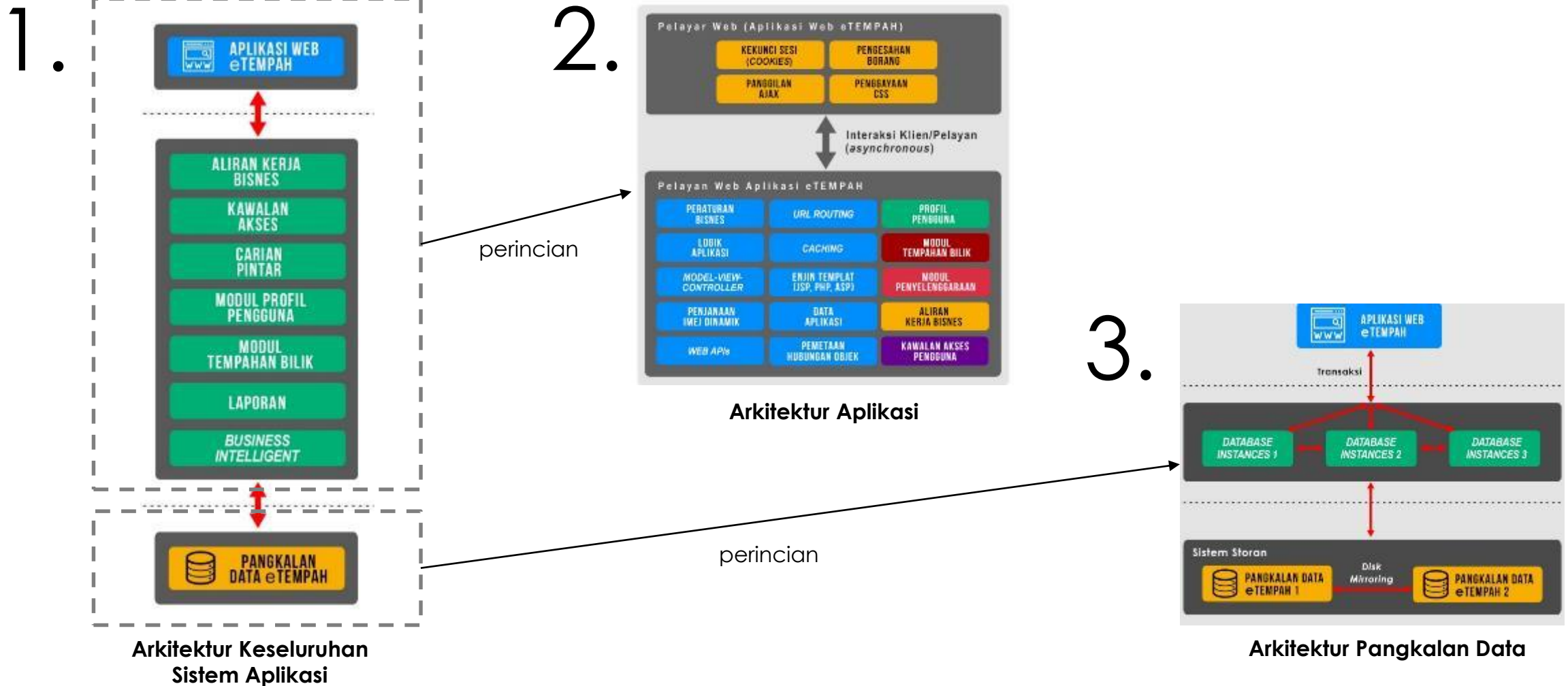
DECENTRALIZED

Elemen kawalan dipecahkan kepada beberapa pusat, di mana selalunya setiap nod mengandungi salinan sumber yang sama.

*Jenis-jenis arkitektur ini boleh digabungkan mengikut keperluan

REKABENTUK ARKITEKTUR

Menyediakan tiga arkitektur :



REKA BENTUK ARKITEKTUR

Kenal Pasti Keperluan Bisnes Dan Sistem

1

2

3

4

5

6

7

LANGKAH



- Rujuk dokumen-dokumen:
 - D02 Spesifikasi Keperluan Bisnes
 - D03 Spesifikasi Keperluan Sistem
 - Arkitektur *Enterprise*
- Dokumen-dokumen ini dirujuk bagi mengenal pasti keperluan fungsian dan bukan fungsian yang telah diperolehi daripada pemegang-pemegang taruh.

1

2

3

4

5

6

7

LANGKAH

a. Penentuan Jenis Aplikasi

Tertakluk kepada keperluan yang telah diperolehi dan batasan infrastruktur serta kemampuan teknologi sedia ada.

Contoh jenis-jenis aplikasi adalah seperti aplikasi mudah alih, client-server application, portal, aplikasi web tradisional atau aplikasi web moden (*Rich Internet Application*).

b. Penentuan Strategi Pelaksanaan

Dilakukan di dalam pelbagai jenis persekitaran di mana setiap persekitaran mempunyai kekangan-kekangan tersendiri.

Contohnya komponen-komponen yang perlu diasingkan secara fizikal dan perlu merentasi pelayan yang berbeza-beza, kekangan konfigurasi bagi peranti-peranti dan kekangan-kekangan yang lain.

c. Penentuan Ciri-ciri Kualiti

Ciri-ciri kualiti atau keperluan bukan fungsian seperti modular, keselamatan, prestasi, capaian, ketersediaan dan kebolehgunaan semula merupakan faktor-faktor yang perlu dipertimbangkan.

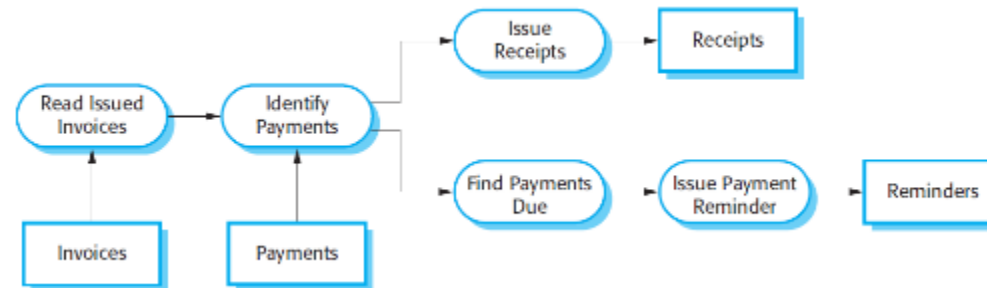
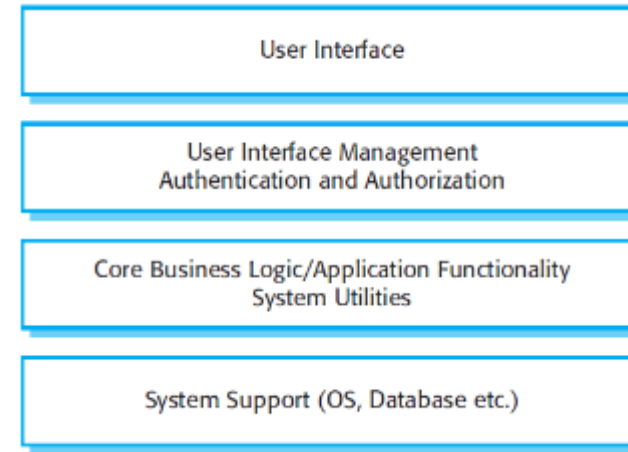
Perlu dilakukan terlebih dahulu bagi mengelakkan percanggahan di antara ciri-ciri kualiti yang perlu disediakan.

REKA BENTUK ARKITEKTUR

Kenal Pasti Corak Arkitektur Yang Sesuai

Corak-corak arkitektur:

- **Arkitektur Lapisan (Layered/N-Tier)**
- **Arkitektur (Client Server)**
- Arkitektur *Event Driven*
- Arkitektur Mikrokernel
- Arkitektur *Serverless*
- Arkitektur Repositori
- Arkitektur *Pipe and Filter*
- etc



1

2

3

4

5

6

7

Langkah 4 : Sediakan Arkitektur Keseluruhan Sistem Aplikasi

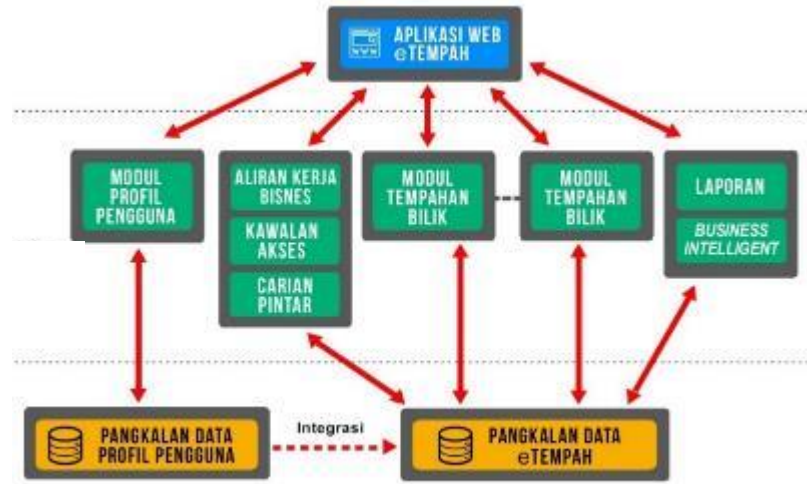
REKA BENTUK ARKITEKTUR

Sediakan Arkitektur Keseluruhan Sistem Aplikasi

1
2
3
4
5
6
7
LANGKAH



Arkitektur Monolitik



Arkitektur Mikroservis

Arkitektur Keseluruhan Sistem Aplikasi adalah merupakan:

- o komponen-komponen perisian iaitu antaramuka sistem, aplikasi dan respositori (data)
- o sifat-sifat luaran (*external properties*) komponen berkenaan
- o hubungan di antara satu komponen dengan komponen-komponen yang lain

Untuk Arkitektur Keseluruhan Sistem Aplikasi, corak **Arkitektur Lapisan** akan digunakan.

*Sila rujuk buku KRISA Bab 4.5 bagi penerangan lanjut bagi Arkitektur Monolitik dan Mikroservis

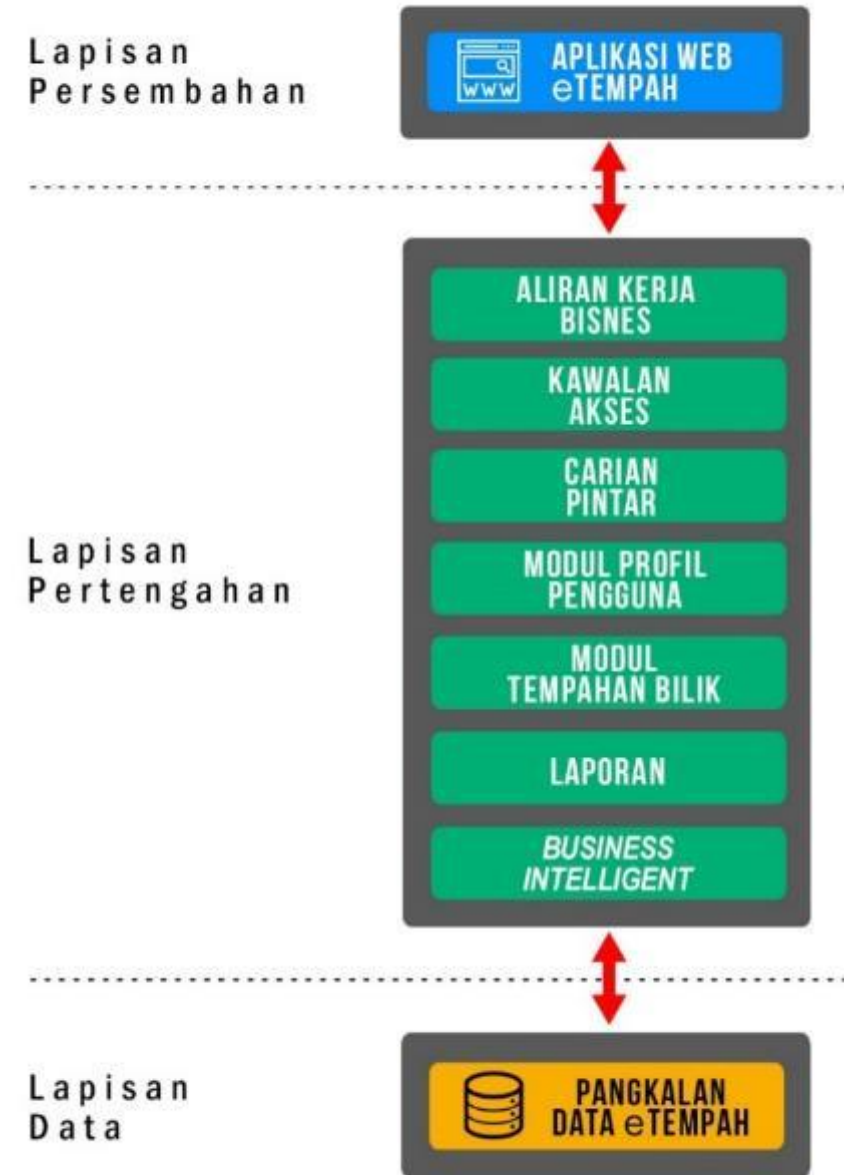
FASA REKA BENTUK

REKA BENTUK ARKITEKTUR

Sediakan Arkitektur Keseluruhan Sistem Aplikasi

Secara amnya, Arkitektur Lapisan merangkumi lapisan-lapisan seperti:

- **Lapisan Persembahan (Presentation Tier/Layer)**
- **Lapisan Pertengahan (Middle Tier/Layer)** atau dikenali juga sebagai **Lapisan Aplikasi** atau **Lapisan Bisnes**
- **Lapisan Data (Data Tier/Layer)**



REKA BENTUK ARKITEKTUR

Sediakan Arkitektur Keseluruhan Sistem Aplikasi

Arkitektur Lapisan – Lapisan Persembahan

Merupakan lapisan di mana pengguna berinteraksi dengan antaramuka aplikasi sama ada komponen-komponen tersebut merupakan web aplikasi, aplikasi mobil, portal web dll.



1

2

3

4

5

6

7

LANGKAH

REKA BENTUK ARKITEKTUR

Sediakan Arkitektur Keseluruhan Sistem Aplikasi

Arkitektur Lapisan – Lapisan Pertengahan/Aplikasi/Bisnes

Merupakan lapisan pengantaraan yang menghubungkan di antara lapisan persembahan dan lapisan data.

Lapisan pertengahan ini terdiri dari komponen-komponen yang melibatkan

- a. **Logik bisnes aplikasi** seperti modul aplikasi.
- b. **Perkhidmatan aplikasi** seperti pelaporan (SQL Server Reporting Services) dan *dashboard/business intelligence* (Pentaho/Power BI).
- c. **Perisian utiliti (perisian bukan web)** seperti *antivirus, disk tools, backup software* dll.



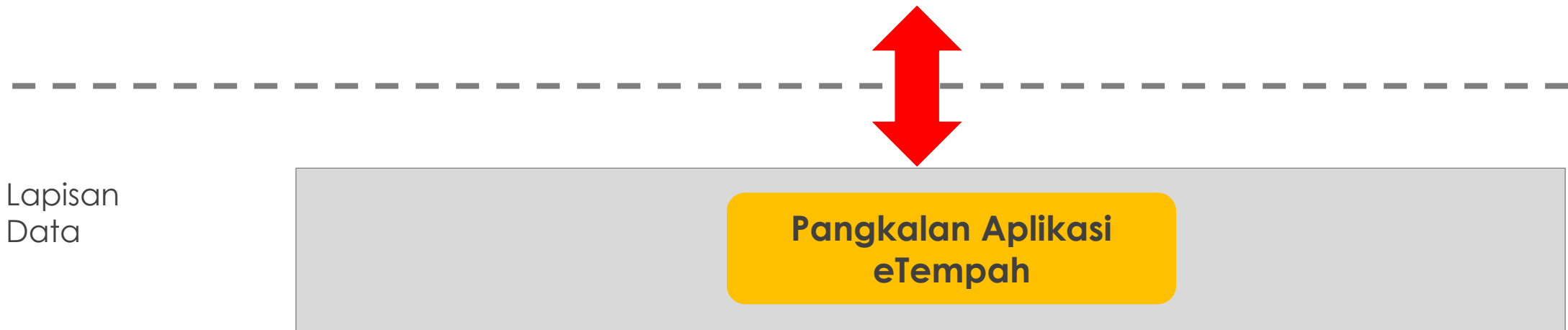
FASA REKA BENTUK

REKA BENTUK ARKITEKTUR

Sediakan Arkitektur Keseluruhan Sistem Aplikasi

Arkitektur Lapisan – Lapisan Data

Merupakan lapisan di mana maklumat-maklumat aplikasi disimpan di dalam pangkalan data.



Lapisan
Data

Pangkalan Aplikasi
eTempah

1

2

3

4

5

6

7

LANGKAH

1

2

3

4

5

6

7

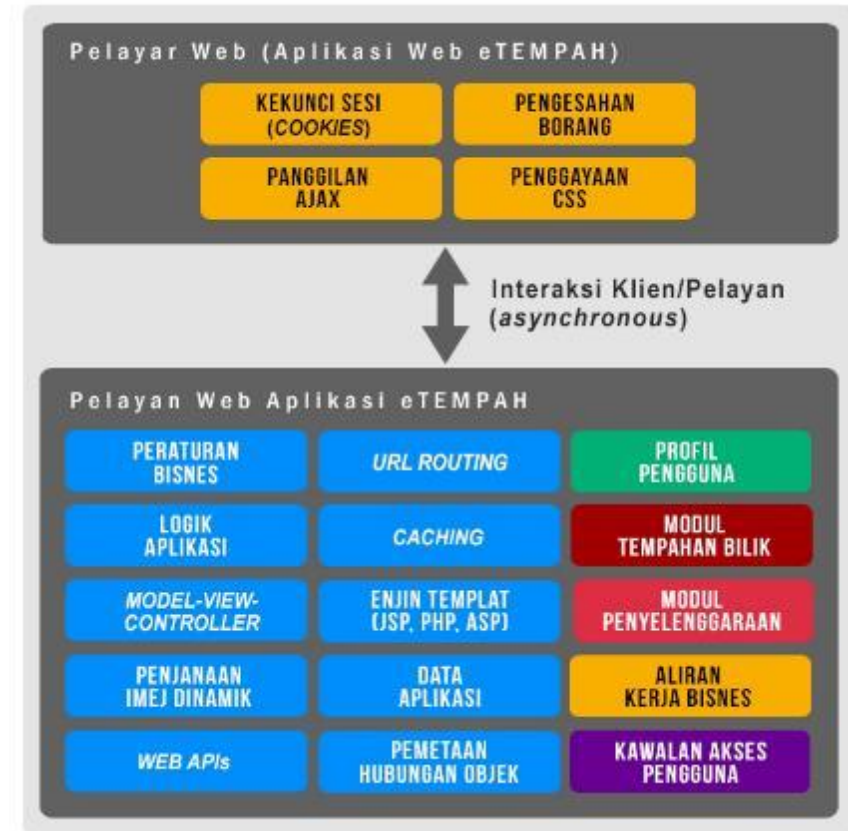
Langkah 5 : Sediakan Arkitektur Aplikasi

LANGKAH

REKA BENTUK ARKITEKTUR

Sediakan Arkitektur Aplikasi

- 1
 - 2
 - 3
 - 4
 - 5
 - 6
 - 7
- Arkitektur Aplikasi merupakan subset dan perincian kepada Arkitektur Keseluruhan Sistem Aplikasi di mana ia tertumpu kepada **penyusunan dan pengaturan sistem aplikasi sahaja**.
 - Arkitektur Aplikasi menerangkan susun atur komponen/subkomponen aplikasi yang terdiri dari:
 - *front end, back end* atau *full stack frameworks*,
 - logik bisnes aplikasi (modul dan submodul),
 - perkhidmatan aplikasi (dashboard / pelaporan)
 - Corak arkitektur yang digunakan adalah **Arkitektur Client-Server**.



REKA BENTUK ARKITEKTUR

Sediakan Arkitektur Aplikasi

Arkitektur *Client Server* – Lapisan Klien (Pelayar Web)

Lapisan Klien merangkumi komponen/subkomponen aplikasi yang beroperasi di pihak klien (*front end*).



Contoh Lapisan Klien (Ringkas)



Contoh Lapisan Klien (Terperinci)

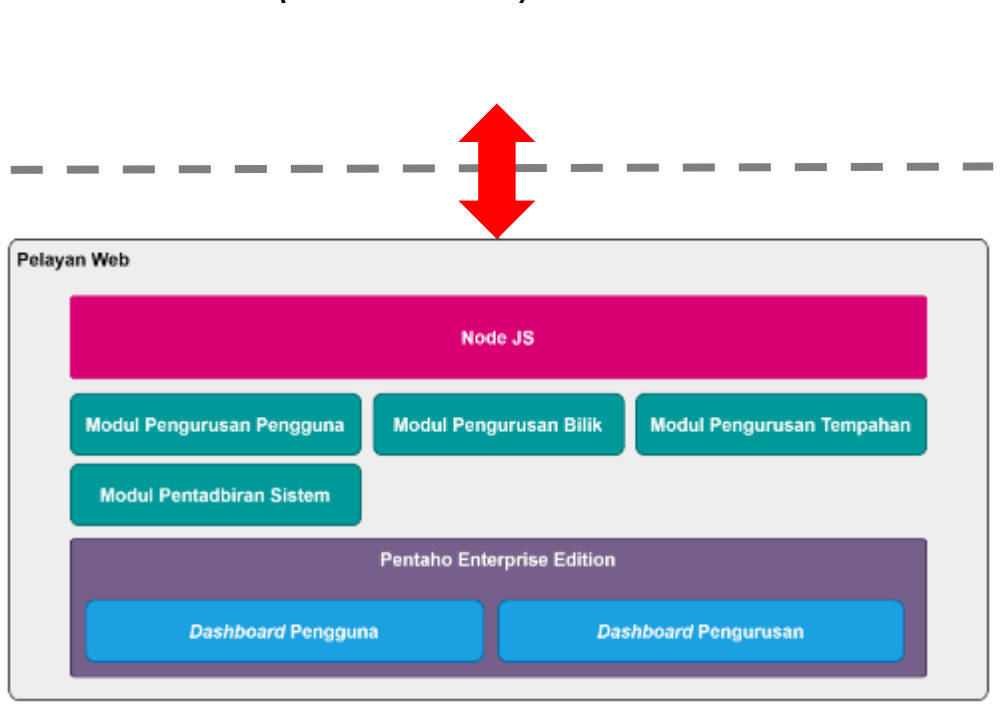
REKA BENTUK ARKITEKTUR

Sediakan Arkitektur Aplikasi

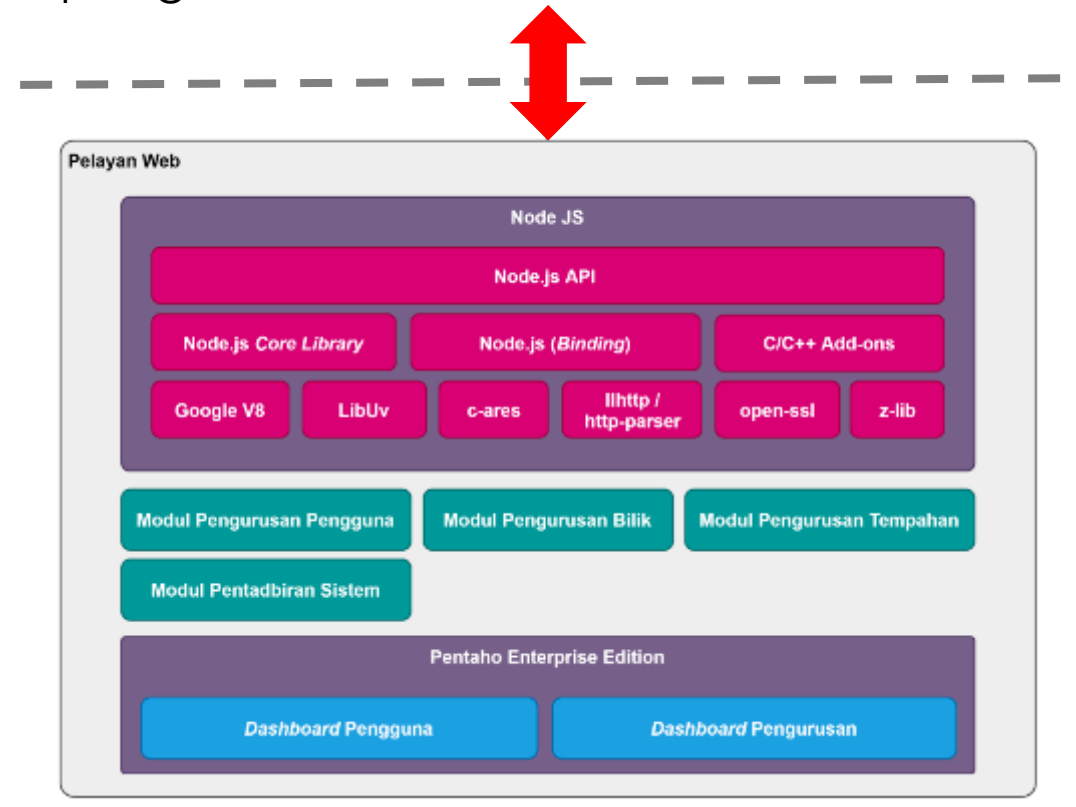
Arkitektur *Client Server* – Lapisan Pelayan (Server)

Lapisan Server merangkumi komponen/subkomponen aplikasi yang beroperasi di pihak server (*back end*), tidak termasuk komponen pangkalan data.

1
2
3
4
5
6
7
LANGKAH



Contoh Lapisan Pelayan (Ringkas)



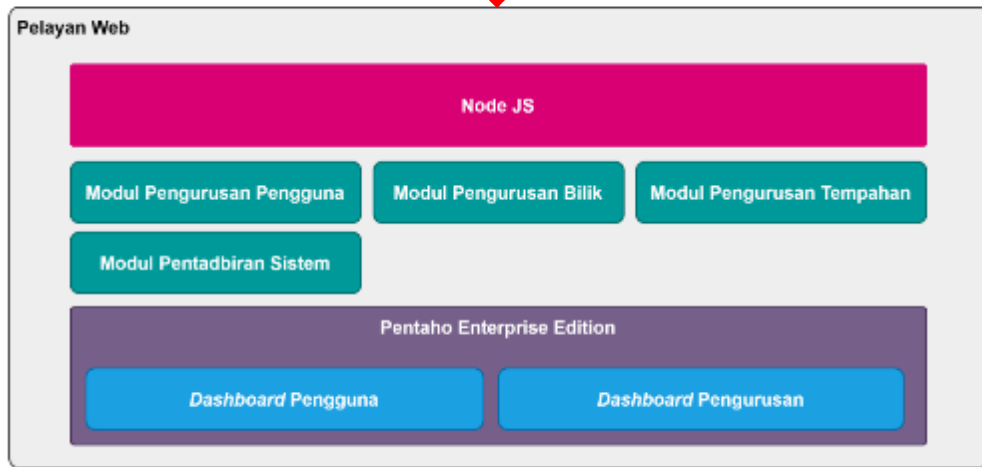
Contoh Lapisan Server (Terperinci)

FASA REKA BENTUK

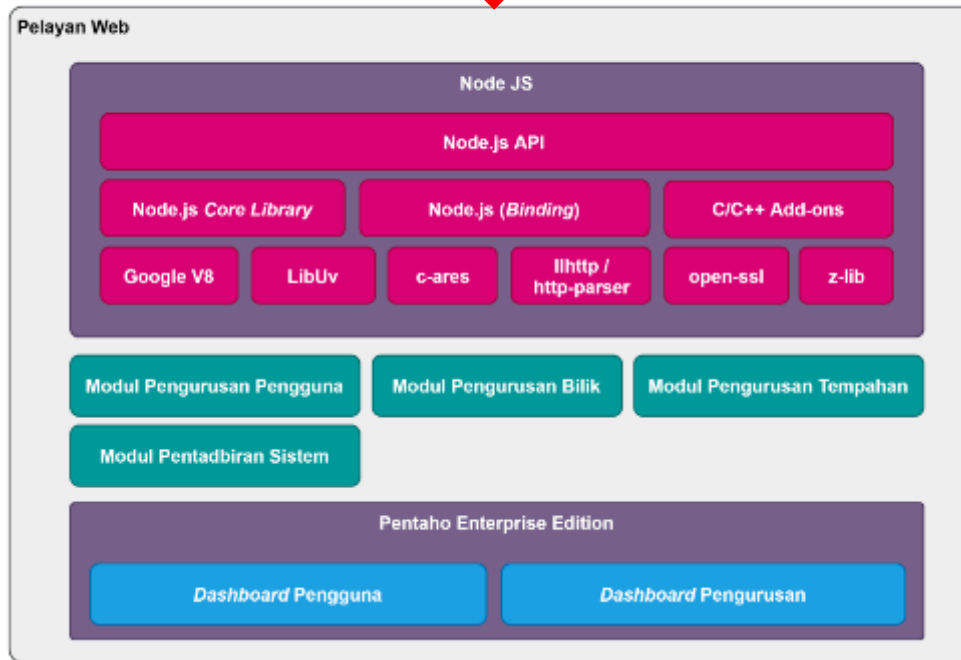
REKA BENTUK ARKITEKTUR

Sediakan Arkitektur Aplikasi

Contoh Arkitektur Aplikasi



Contoh Arkitektur Aplikasi (Ringkas)



Contoh Arkitektur Aplikasi (Terperinci)

1
2
3
4
5
6
7
LANGKAH

1

2

3

4

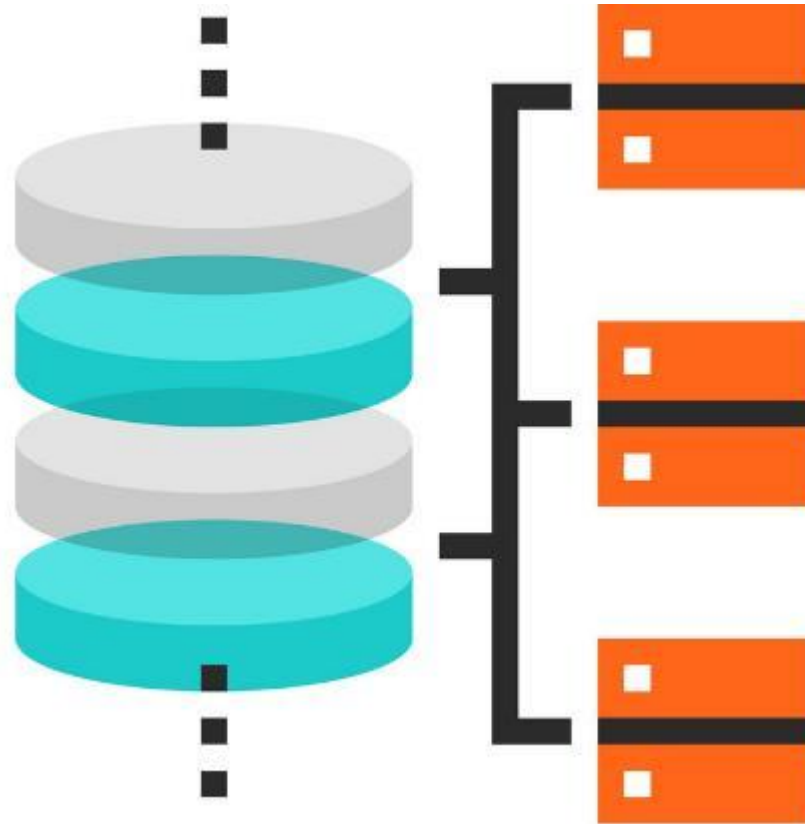
5

6

7

LANGKAH

Langkah 6 : Sediakan Arkitektur Pangkalan Data

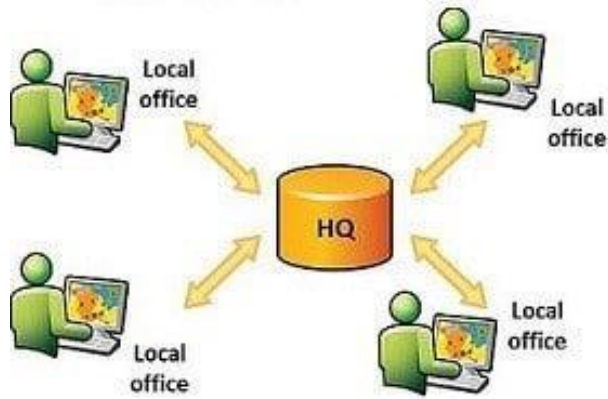


- Merupakan subset kepada arkitektur keseluruhan sistem aplikasi di mana ia tertumpu kepada penyusunan dan pengaturan **komponen pangkalan data dan pengklusteran/grid data sahaja**.
- Arkitektur Pangkalan Data diwujudkan bagi melihat struktur dan penyusunan di antara aplikasi, pangkalan-pangkalan data dan storan-storan disk yang terlibat.
- Penyediaan arkitektur ini adalah berdasarkan kepada keperluan fungsian, bukan fungsian dan corak arkitektur yang telah ditentukan.

REKA BENTUK ARKITEKTUR

Sediakan Arkitektur Pangkalan Data

Penentuan Pangkalan Data Berpusat atau Teragih



Pangkalan Data Berpusat (*Centralized*)

Merupakan pangkalan data di mana fail-failnya disimpan dan diselenggara di satu lokasi storan.

Kelebihan : Mudah diselenggara, jimat kos

Kekurangan : Isu prestasi bagi utiliti tinggi



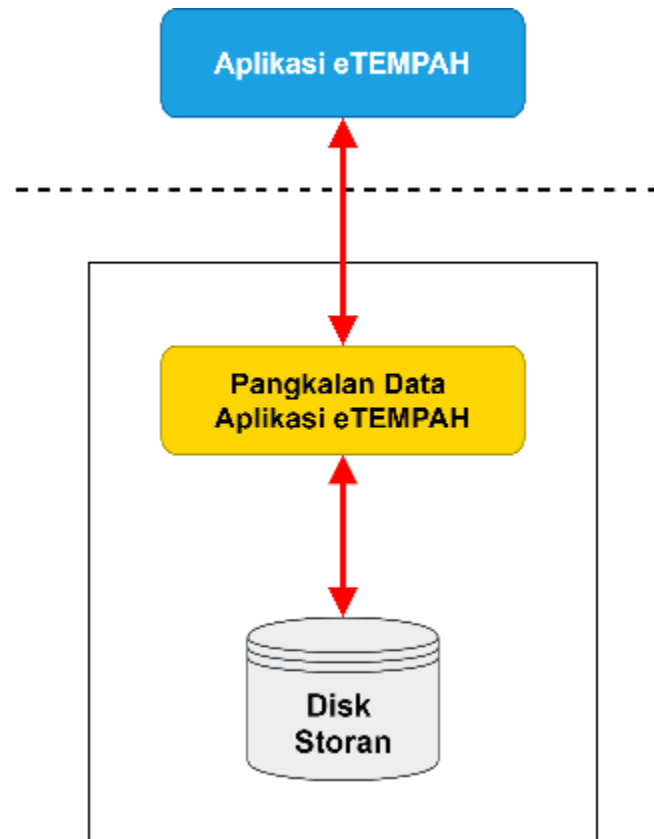
Pangkalan Data Teragih (*Decentralized*)

Merupakan pangkalan data di mana sebahagian atau keseluruhan fail-failnya disimpan dan diselenggara di satu lokasi berasingan.

Kelebihan : Berprestasi tinggi, *high scalability*

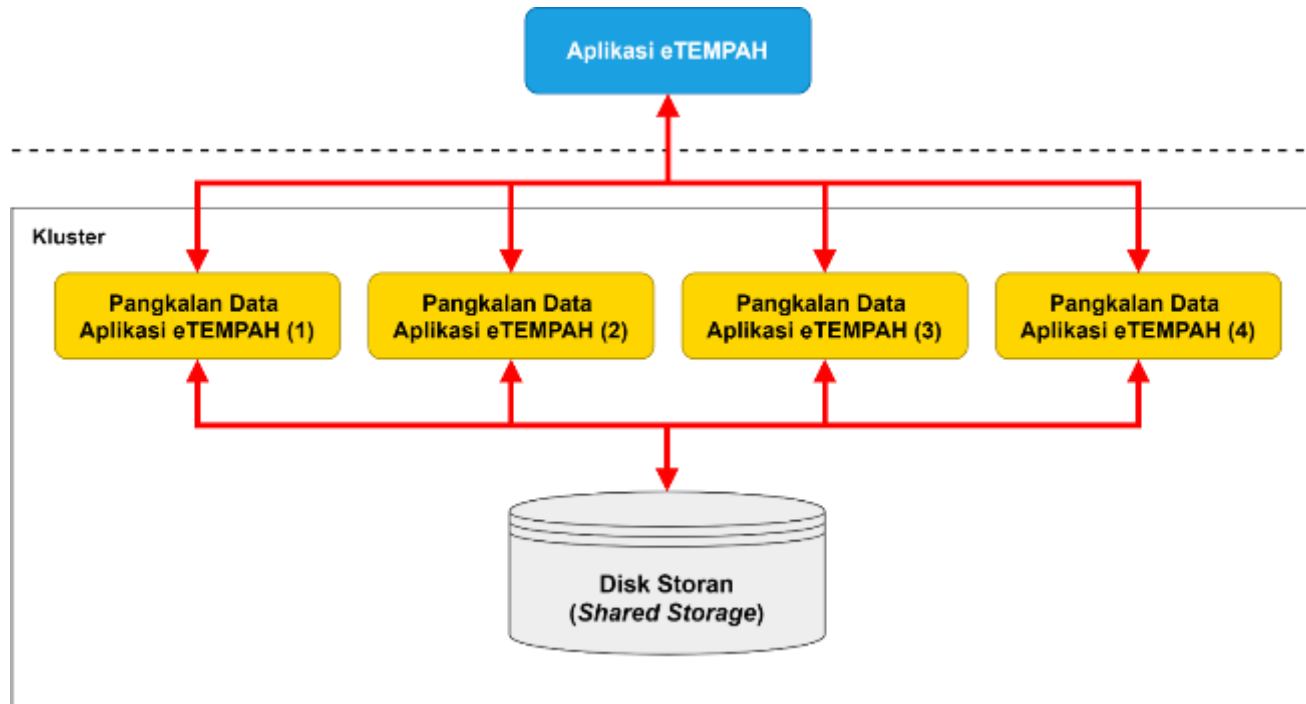
Kekurangan : Kos tinggi

Arkitektur Pangkalan Data Berpusat (*Centralized*)



- Masukkan komponen aplikasi yang berkomunikasi dengan pangkalan data.
- Untuk arkitektur ini, ia hanya melibatkan satu komponen pangkalan data dan disk storan sahaja, dan juga memori dan pemprosesan sendiri.
- Komponen pangkalan data dihubungkan dengan disk storan yang terlibat.
- Arkitektur boleh diperincikan lagi dengan memasukkan subkomponen seperti *memory*, *filesystem*, *queue*, *cache* dll.

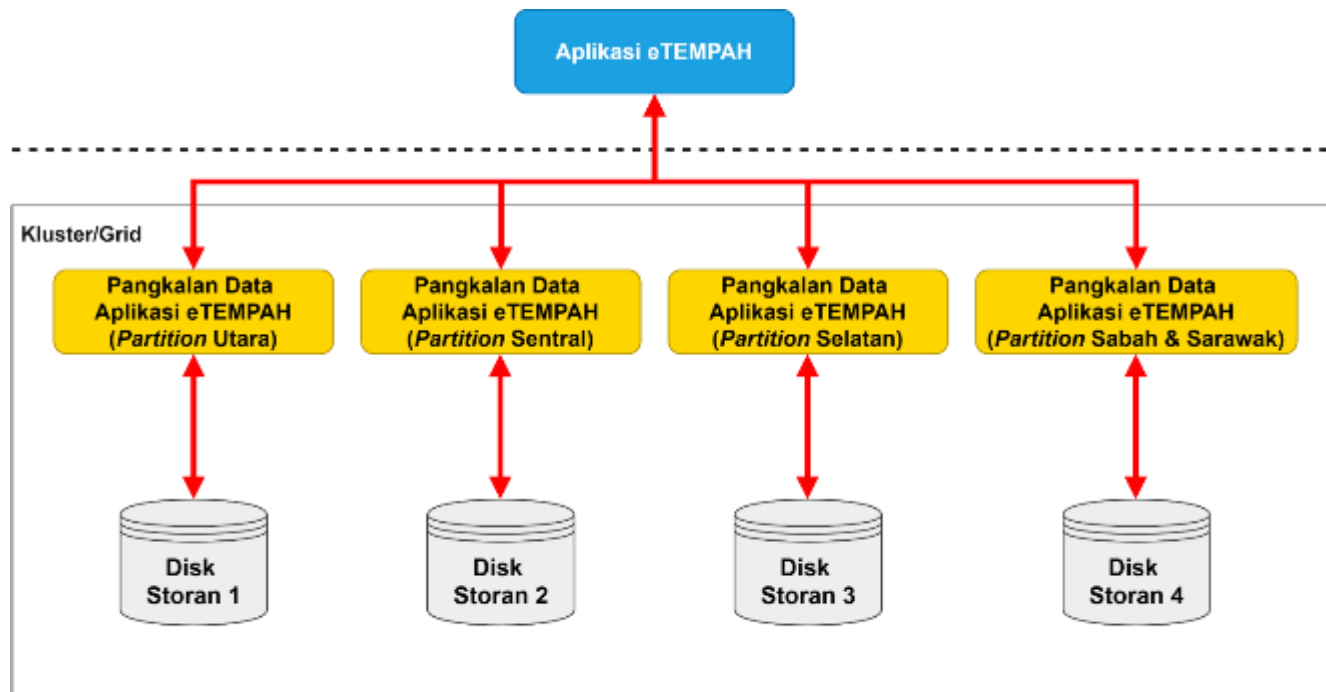
Arkitektur Pangkalan Data Teragih (*Shared Disk/Storage*)



- Arkitektur ini membolehkan semua pangkalan data berkongsi maklumat (disk storan) dan memori di antara satu sama lain.
- Salinan data bagi semua pangkalan data adalah sama di mana data-data tersebut tidak diasingkan mengikut *partition*.

*Sila rujuk buku KRISA Bab 4.5 bagi penerangan lanjut bagi Arkitektur *Shared Disk* dan perbezaannya dengan Arkitektur *Shared Nothing*

Arkitektur Pangkalan Data Teragih (*Shared Nothing*)



- Arkitektur ini mengasingkan maklumat (disk storan) dan memori setiap pangkalan data yang terlibat.
- Salinan data bagi semua adalah tidak sama di mana data-data tersebut diasingkan mengikut *partition*.

*Sila rujuk buku KRISA Bab 4.5 bagi penerangan lanjut bagi Arkitektur *Shared Nothing* dan perbezaannya dengan Arkitektur *Shared Disk*

REKA BENTUK ARKITEKTUR

Sediakan Arkitektur-Arkitektur Secara Iteratif

1

2

3

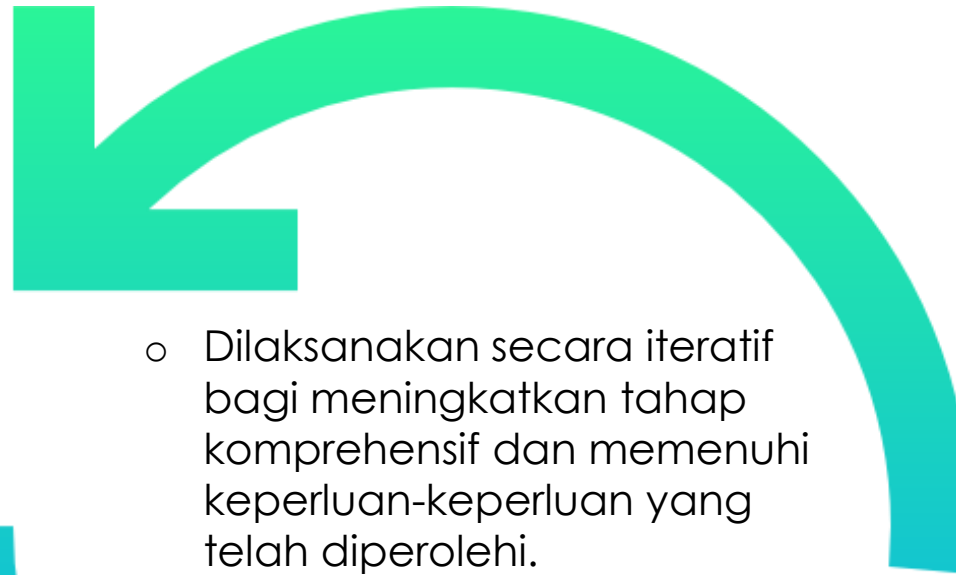
4

5

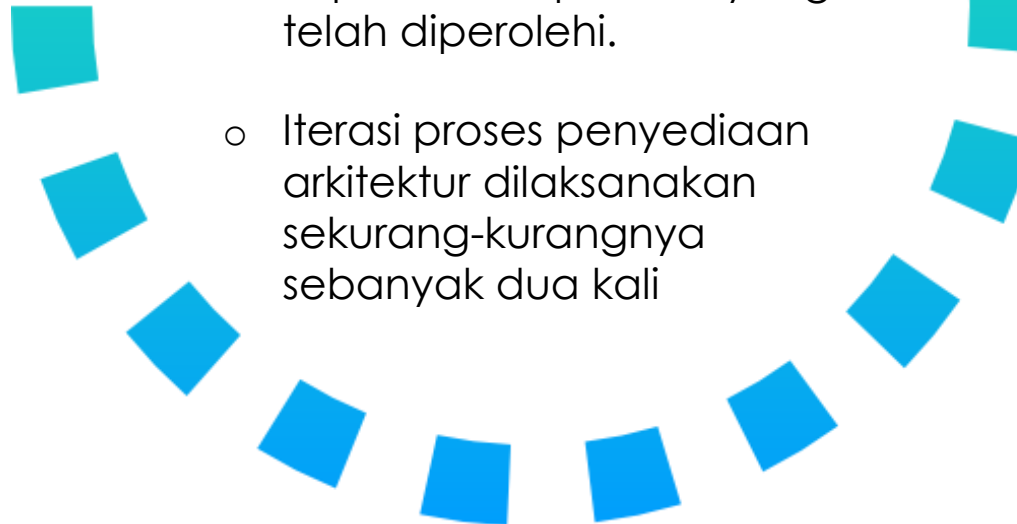
6

7

LANGKAH



- Dilaksanakan secara iteratif bagi meningkatkan tahap komprehensif dan memenuhi keperluan-keperluan yang telah diperolehi.
- Iterasi proses penyediaan arkitektur dilaksanakan sekurang-kurangnya sebanyak dua kali





Contoh Penyelesaian

REKABENTUK ARKITEKTUR

CONTOH PENYELESAIAN - SOALAN

- 1) Agensi XYZ ingin membangunkan satu aplikasi web bagi tempahan kenderaan jabatan (transportMe). Aplikasi tersebut mempunyai fungsi-fungsi seperti berikut:
 - a) **Modul Profil Pengguna**
 - b) **Modul Pentadbiran Sistem**
 - c) **Modul Tempahan Kenderaan**
 - d) **Modul Penetapan Jadual Pemandu**
 - e) **Dashboard Statistik Tempahan (*embed with PowerBI*)**

- 2) Agensi XYZ memutuskan bahawa aplikasi ini akan menggunakan **Angular 12 Framework** di *frontend*, di samping menggunakan **HTML5, javascript** dan **CSS3, Bootstrap 5** sebagai struktur, elemen interaktif dan pengayaan kepada web.

- 3) Manakala **Spring Boot 2.2.1** akan digunakan sebagai *back end framework*.

- 4) Agensi bersetuju menggunakan **pangkalan data secara berpusat** menggunakan **PostgreSQL 12** oleh kerana penggunaan aplikasi ini hanya dilakukan secara dalaman agensi dan jumlah transaksi/trafik tidaklah begitu tinggi.

REKABENTUK ARKITEKTUR

CONTOH PENYELESAIAN - JAWAPAN

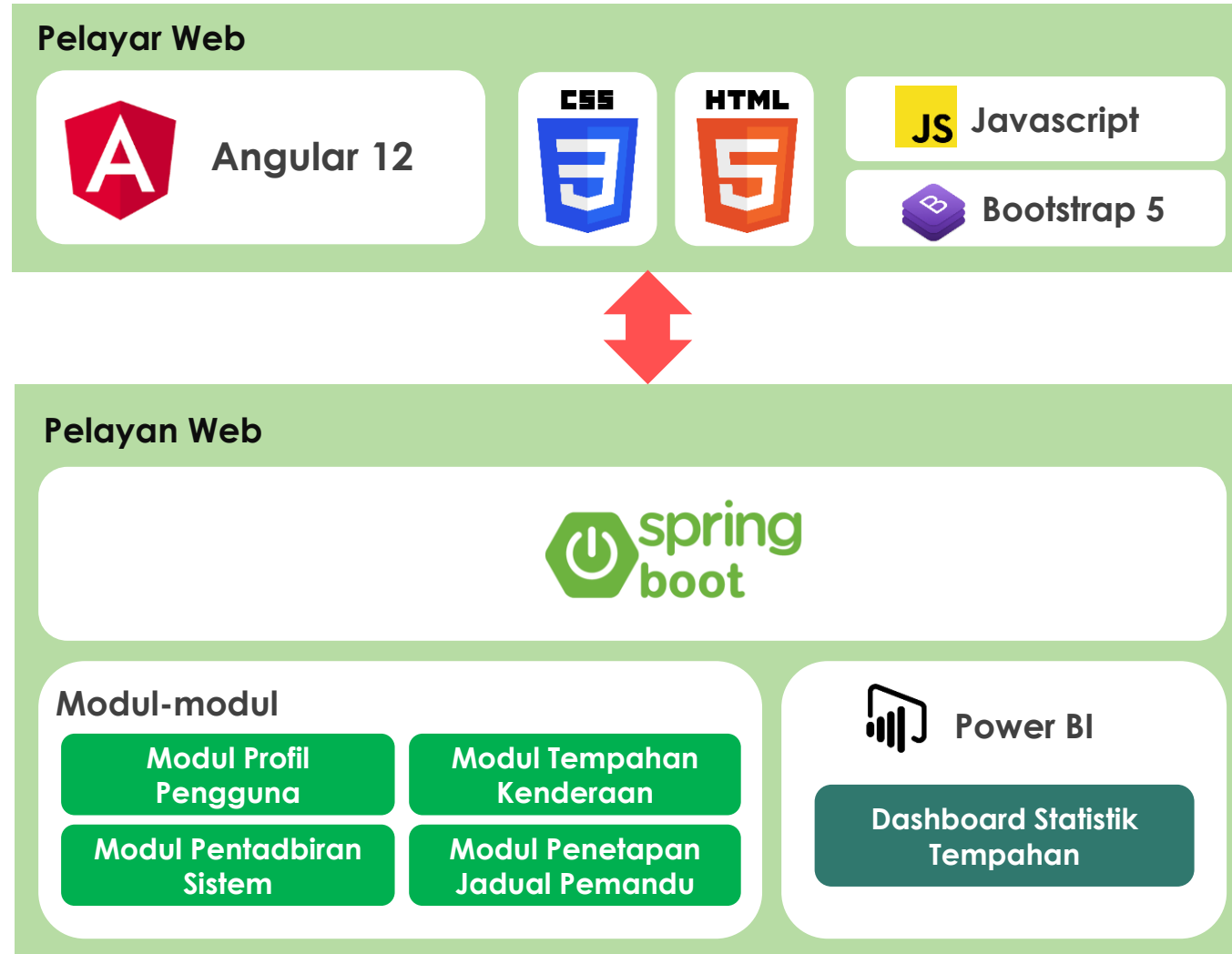
1. Arkitektur Keseluruhan Sistem Aplikasi



REKABENTUK ARKITEKTUR

CONTOH PENYELESAIAN - JAWAPAN

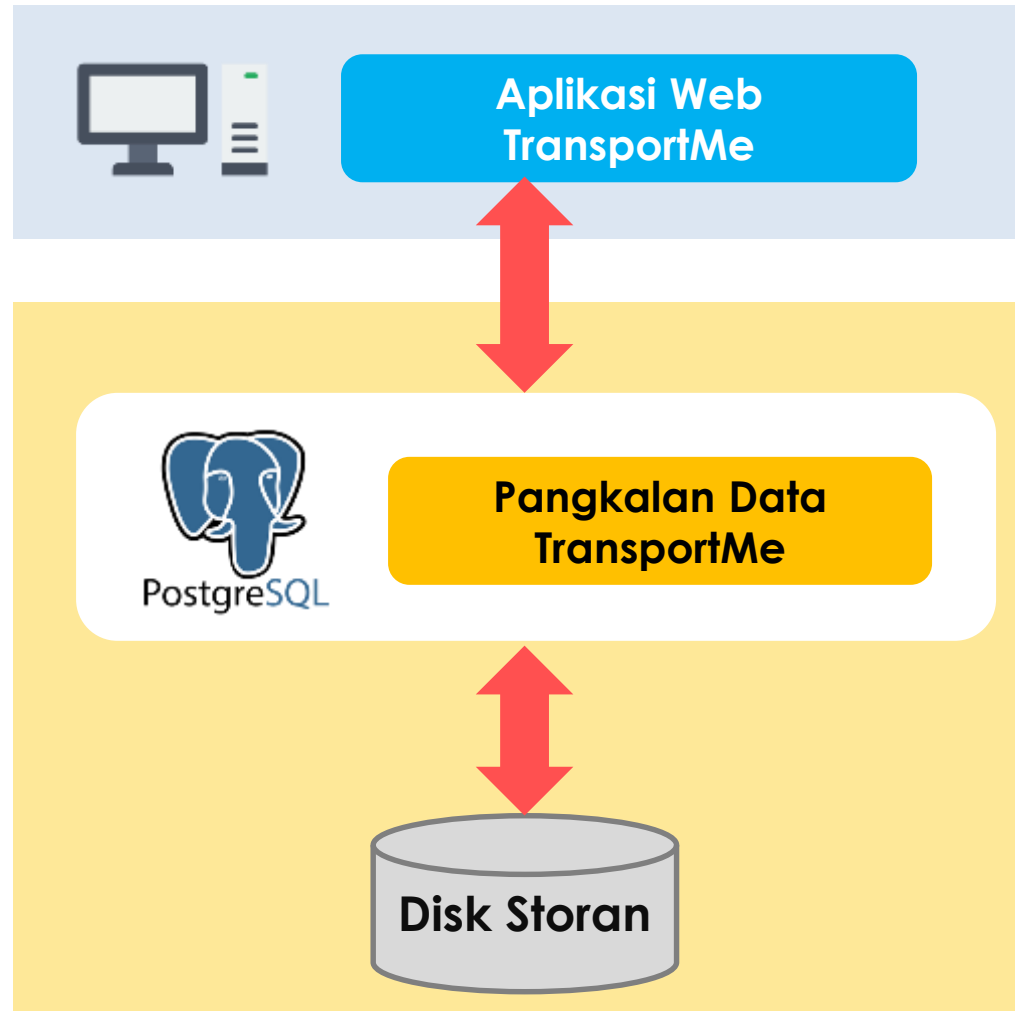
2. Arkitektur Aplikasi



REKABENTUK ARKITEKTUR

CONTOH PENYELESAIAN - JAWAPAN

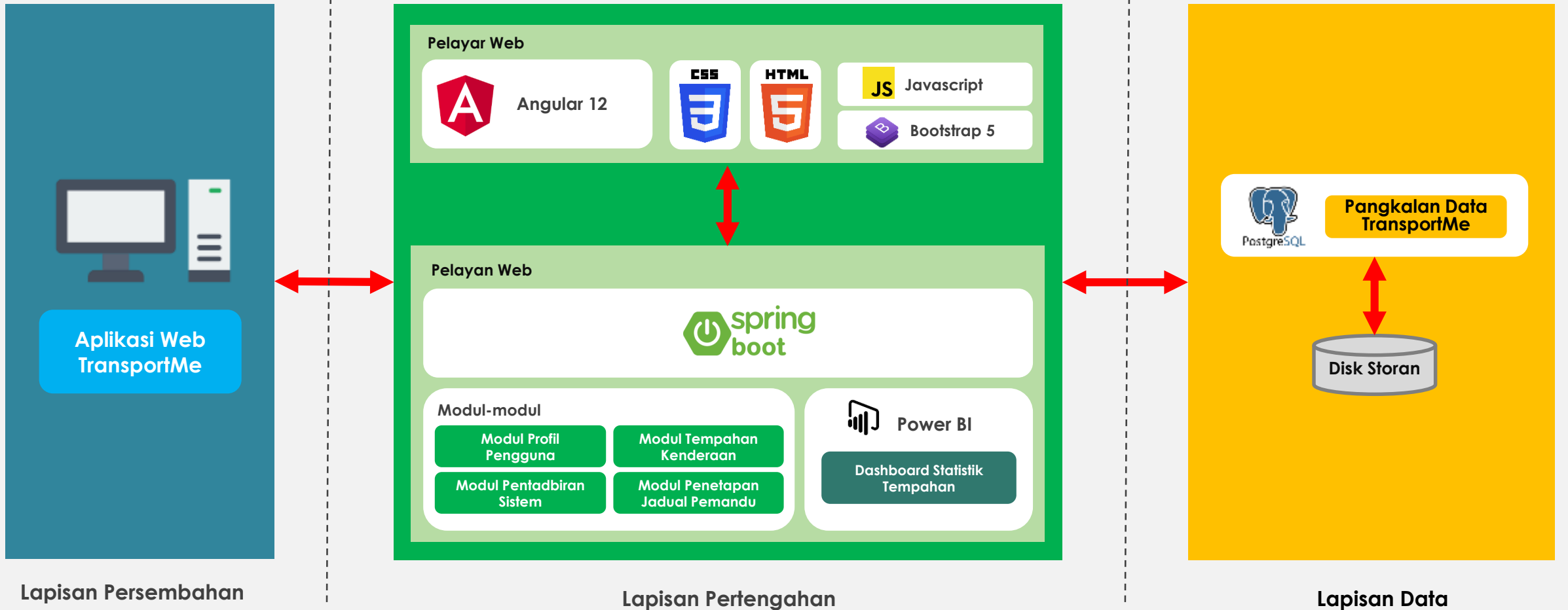
3. Arkitektur Pangkalan Data



REKABENTUK ARKITEKTUR

CONTOH PENYELESAIAN – JAWAPAN ALTERNATIF

Arkitektur Keseluruhan Sistem Aplikasi



F3.2 PENENTUAN TEKNOLOGI

TAKLIMAT
02



PENENTUAN TEKNOLOGI

PENGENALAN

- Teknologi dan *tool* yang dipilih akan digunakan bagi memandu dalam reka bentuk antaramuka, reka bentuk proses dan reka bentuk logikal pangkalan data.
- Teknologi yang dipilih juga akan digunakan semasa pelaksanaan fasa pembangunan, pengujian, pengoperasian dan penyelenggaraan sistem aplikasi.
- Beberapa aspek perlu diambil kira:
 - Memenuhi dan mematuhi reka bentuk arkitektur sistem aplikasi
 - Keserasian dengan keperluan fungsian dan bukan fungsian
 - Selaras dengan visi dan misi organisasi
 - Teknologi-teknologi yang mudah diperolehi dan diselenggara



PENENTUAN TEKNOLOGI

MATRIKS ALTERNATIF

- Satu kaedah yang digunakan untuk memilih teknologi yang memenuhi keperluan.
- Digunakan juga sebagai kaedah untuk menentukan reka bentuk, membantu dalam mengurus dan menyusun reka bentuk.
- Matriks alternatif menggabungkan beberapa analisa kebolehlaksanaan seperti berikut:
 - **Kebolehlaksanaan Teknikal** - penilaian kematangan atau keupayaan teknologi untuk berfungsi dengan teknologi yang lain.
 - **Kebolehlaksanaan Operasi** - keselesaan dan kesesuaian pihak pengurusan, pegawai dan pengguna dengan teknologi yang dicadangkan.
 - **Kebolehlaksanaan Ekonomi** - penilaian sama ada teknologi yang digunakan berpatutan dan kos efektif.

No.	Kriteria	Bahasa Pengaturcaraan		
		Java	C#	Python
		Nilai Skor		
1	Pengetahuan, pengalaman dan kemahiran pasukan pembangun	8	6	1
2	Ketersediaan (<i>availability</i>) pengaturcara	9	5	1
3	Ketersediaan <i>Integrated Development Environment</i> (IDE) dan <i>tools</i> di pasaran	8	8	6
4	Kemudahan integrasi	8	8	8
5	Penjimatan kos	9	6	9
6	Prestasi	8	8	8
7	Sekuriti	8	8	8
8	Sokongan dan komuniti	8	9	6
9	Keanjalan bahasa pengaturcaraan	8	8	8
10	Tren semasa	9	6	4
JUMLAH		83	72	59

Contoh Jadual Matriks Alternatif (Ringkas)

No.	Kriteria	Pemberat	Front End Framework				Catatan
			Angular	React	Angular	React	
			Nilai Skor		Nilai Skor * Pemberat		
1	Kebolehlaksanaan Teknikal						
1.1	Prestasi	0.05	9	10	0.45	0.5	React uses a Virtual DOM that helps to increase general speed and application's performance. - https://becominghuman.ai
1.2	Ketersediaan Integrated Development Environment (IDE) dan tools di pasaran	0.05	10	10	0.5	0.5	
1.3	Kemudahan Integrasi	0.05	10	10	0.5	0.5	
1.4	Kemudahan Pengujian	0.05	10	8	0.5	0.4	Angular - Possible with a single tool React - Requires a set of tools for performing various types of testing (https://aglowiditsolutions.com/)
1.5	Backward Compatibility	0.05	8	10	0.4	0.5	React is fully compatible between all versions. - https://aglowiditsolutions.com
1.6	Tren Semasa (Populariti)	0.05	4	10	0.2	0.5	- https://www.simform.com/
2	Kebolehlaksanaan Operasi						
2.1	Pengetahuan, pengalaman dan kemahiran pasukan pembangun dengan framework	0.2	8	2	1.6	0.4	Pihak pembangunan mempunyai pengalaman menggunakan Angular
2.2	Pengetahuan, pengalaman dan kemahiran pasukan pembangun dengan bahasa pengaturcaraan framework	0.2	8	5	1.6	1	Pengaturcara mempunyai pengalaman dengan bahasa Typescript (Angular) daripada Javascript(React)
3	Kebolehlaksanaan Ekonomi						
3.1	Penjimatan Kos	0.3	10	10	3	3	Kedua-dua framework adalah open source
Jumlah Kebolehlaksanaan		1			8.75	7.3	

Contoh Jadual Matriks Alternatif (Terperinci)



MAMPU

Unit Pemodenan Tadbiran dan Perancangan Pengurusan Malaysia

All information incorporated within this slide is created for
Malaysian Administrative Management and Planning Unit (MAMPU),
Prime Minister's Department, Malaysia.

All information is the property of MAMPU and any unauthorized
reproduction is prohibited

**TERIMA
KASIH**