KRISA

PANDUAN KEJURUTERAAN SISTEM APLIKASI SEKTOR AWAM

BPI BAHAGIAN PERUNDINGAN ICT

FASA PEMBANGUNAN

MENERANGKAN AKTIVITI-AKTIVITI PEMBANGUNAN SISTEM APLIKASI SERTA AMALAN TERBAIK DALAM PENGATURCARAAN, PEMBANGUNAN PANGKALAN DATA DAN PENGUJIAN YANG PERLU DILAKSANAKAN KE ATAS SISTEM APLIKASI

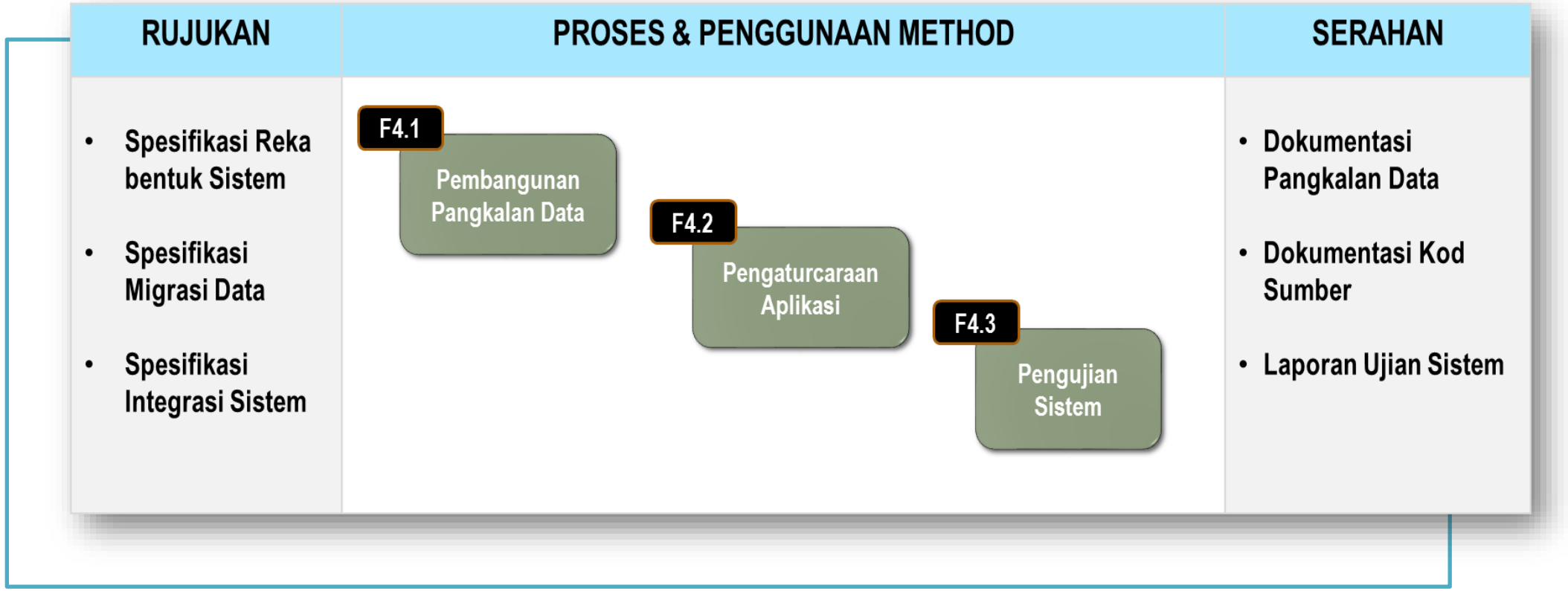
PENGENALAN

Fasa Pembangunan adalah fasa yang menterjemahkan atau merealisasikan **Spesifikasi Reka Bentuk Sistem (SDS)**, **Spesifikasi Migrasi Data dan Spesifikasi Integrasi Sistem** yang dihasilkan kepada **Pangkalan Data Fizikal**, **Kod Aturcara** dalam bahasa pengaturcaraan yang telah dipilih, memasang dan seterusnya melaksanakan **Pengujian Sistem** agar sistem yang dibangunkan bebas dari sebarang ralat, dapat berfungsi sepenuhnya dan berjaya memenuhi keperluan sebenar pembangunan.

Fasa Pembangunan menggariskan 3 aktiviti utama iaitu:

1. Pembangunan Pangkalan Data
2. Pengaturcaraan Aplikasi
3. Ujian Sistem

GAMBARAN KESELURUHAN



FASA PEMBANGUNAN

RUJUKAN

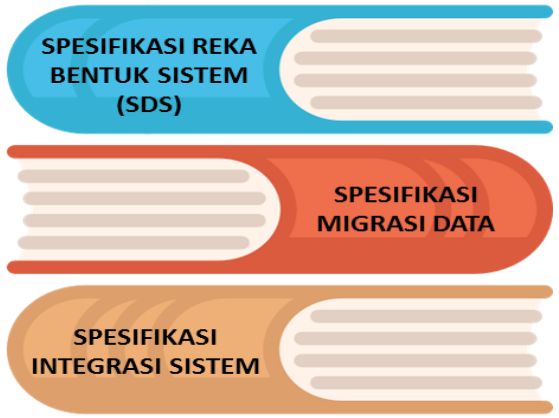
- **D04** SPESIFIKASI REKA BENTUK SISTEM (SDS)
- **D06** SPESIFIKASI MIGRASI DATA
- **D08** SPESIFIKASI INTEGRASI SISTEM

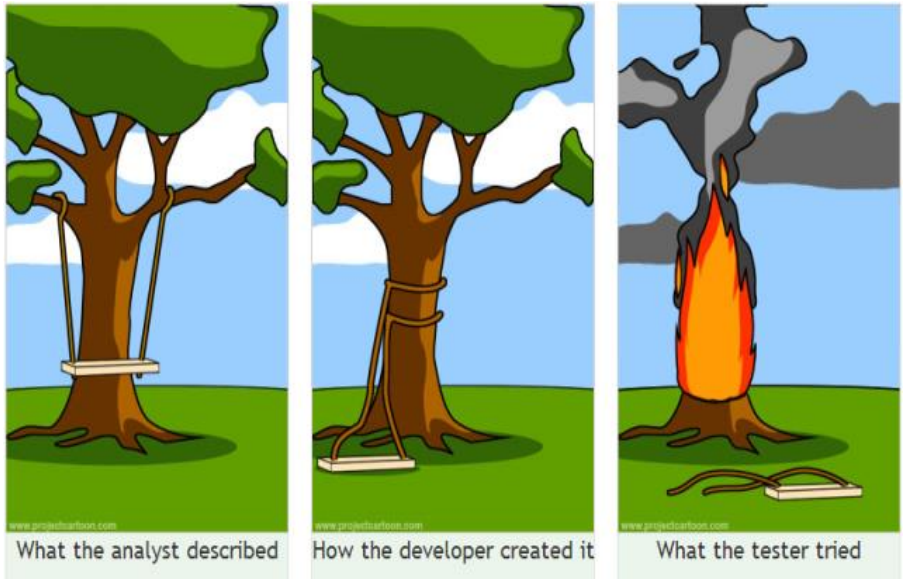
AKTIVITI

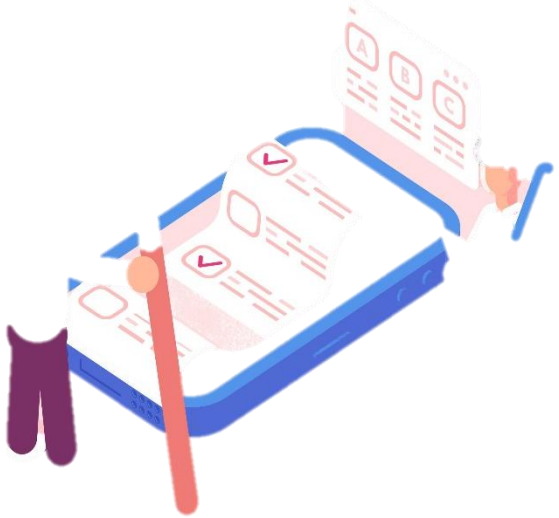
- **F4.1** PEMBANGUNAN PANGKALAN DATA
- **F4.2** PENGATURCARAAN APLIKASI
- **F4.3** UJIAN SISTEM

DOKUMEN

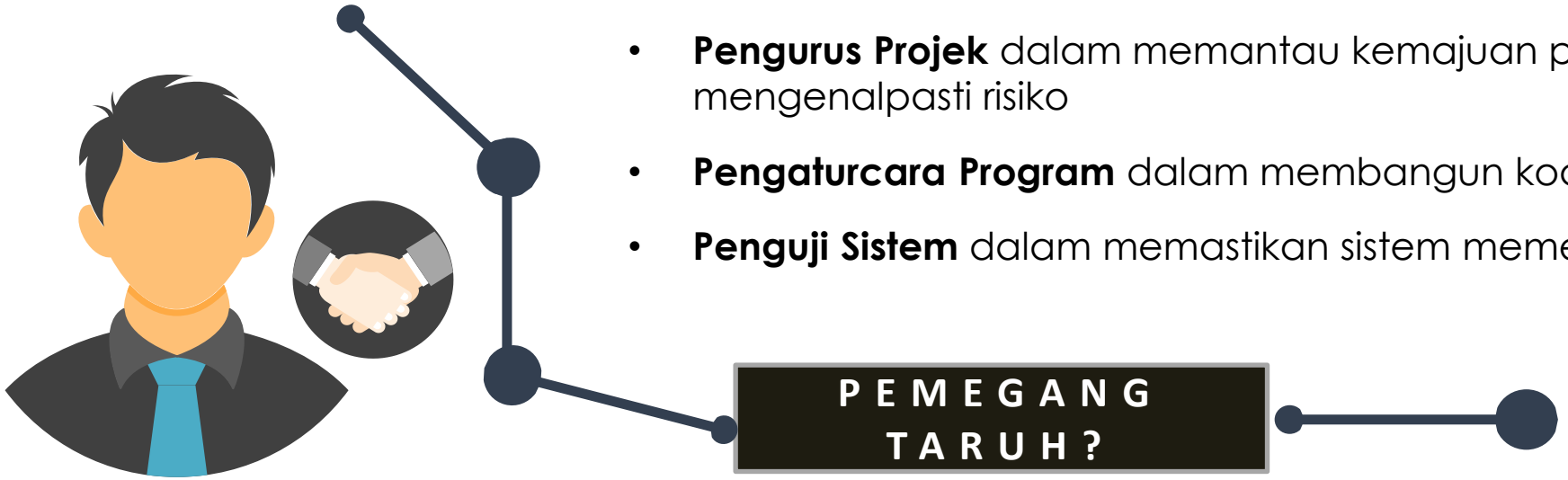
- **D09** DOKUMENTASI KOD PANGKALAN DATA
- **D10** DOKUMENTASI KOD SUMBER
- **D11** LAPORAN UJIAN SISTEM







PENGENALAN KEPADA REKA BENTUK



- **Pengurus Projek** dalam memantau kemajuan pembangunan dan mengenalpasti risiko
- **Pengaturcara Program** dalam membangun kod aturcara sistem aplikasi
- **Penguji Sistem** dalam memastikan sistem memenuhi spesifikasi.

FAKTOR KEJAYAAN

- ☑ Spesifikasi Reka bentuk Sistem (SDS) yang didokumenkan adalah lengkap dan memenuhi kehendak pengguna.
- ☑ Pasukan pengaturcara program berupaya menterjemahkan SDS kepada kod aturcara dan logik pemprograman, mempunyai kemahiran dan kepakaran dalam bahasa pengaturcaraan yang dipilih dan SQL (skill coding dan SQL query).
- ☑ Bilangan pengaturcara yang mencukupi dan bersesuaian dengan masa pembangunan.
- ☑ Kelengkapan tools dan persekitaran pembangunan yang sempurna.

FASA **PEMBANGUNAN**

TAKLIMAT

01

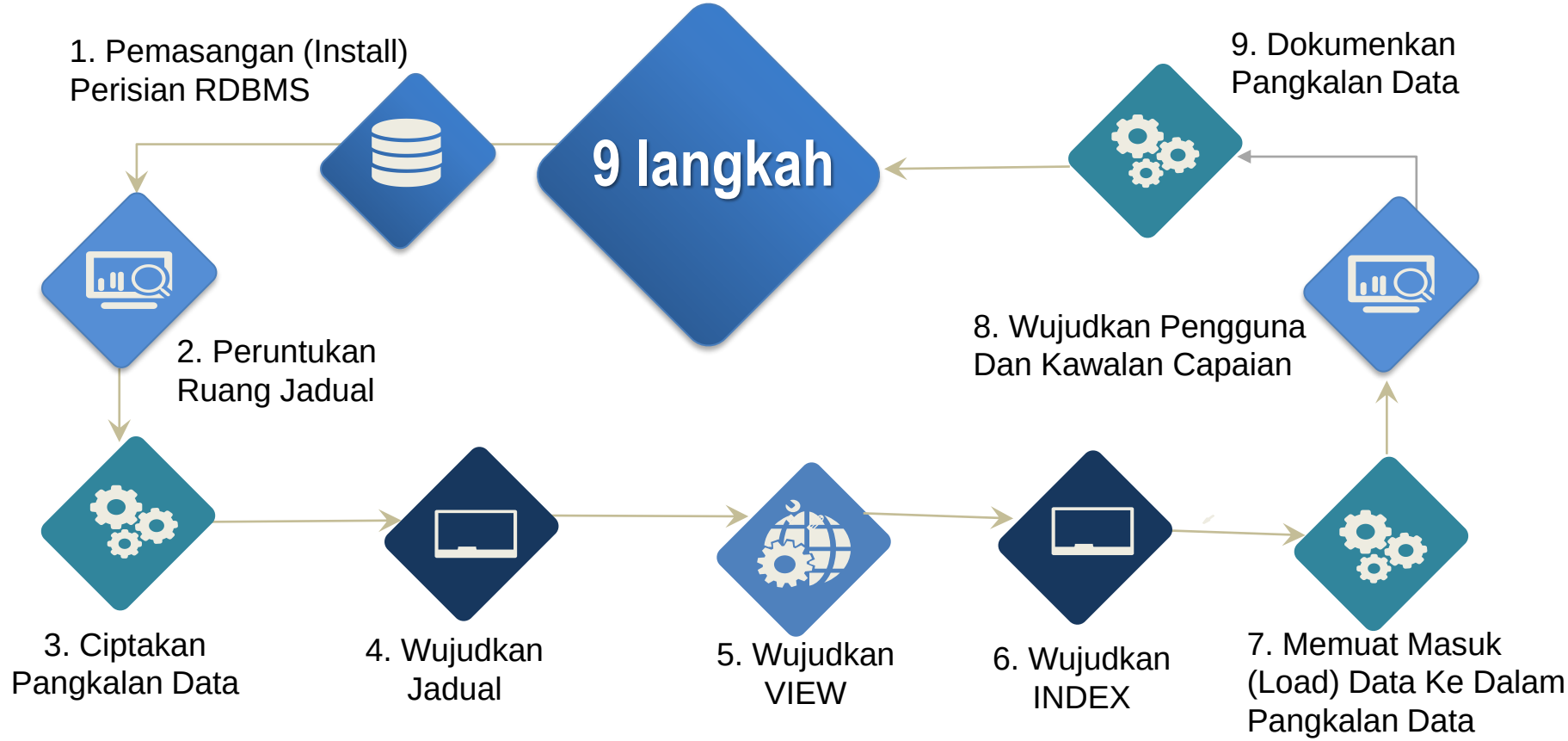
F4.1 PEMBANGUNAN PANGKALAN DATA

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.

PEMBANGUNAN PANGKALAN DATA

Pembangunan Pangkalan Data merupakan proses mewujudkan pangkalan data fizikal berdasarkan reka bentuk pangkalan data logikal dan arkitektur pangkalan data. Pembangunan Pangkalan Data dilaksanakan oleh Pentadbir Pangkalan Data atau lebih dikenali sebagai *Database Administrator* (DBA).



PEMBANGUNAN PANGKALAN DATA

Langkah 3: Ciptakan Pangkalan Data (*Create A Database*)

- Pengguna yang mempunyai capaian *root* dibenarkan untuk mencipta pangkalan data. Arahan yang digunakan untuk mencipta pangkalan data ialah:

CREATE DATABASE <nama pangkalan data>;

- Bagi melihat senarai pangkalan data yang telah dicipta, arahan yang digunakan adalah seperti berikut.

SHOW DATABASES;

Berikut adalah contoh paparan bagi arahan tersebut.

```
mysql> SHOW DATABASES;
+--------------------+
| Database |
+--------------------+
| information_schema |
| mysql |
| performance_schema |
| pdata1 |
+--------------------+
4 rows in set (0.01 sec)
```

Paparan di atas menunjukkan bahawa terdapat empat (4) pangkalan data yang terdapat dalam hos pelayan (*server host*).

PEMBANGUNAN PANGKALAN DATA

Langkah 4: Wujudkan Jadual (*Create Table*)

- Jadual adalah terdiri daripada baris dan lajur yang mengandungi rekod maklumat dalam pangkalan data. Entiti dalam rekabentuk pangkalan data adalah merujuk kepada jadual (table) bagi pangkalan data fizikal. Manakala atribut pula adalah medan (field).

- Jadual berikut adalah pemadanan secara teknologi antara reka bentuk pangkalan data logikal dan fizikal.

Istilah dalam rekabentuk logikal	Istilah dalam rekabentuk fizikal
Entiti	Jadual (table)
Atribut	Medan (column atau field)
Primary UID	Primary Key
Secondary UID	Unique Key
Relationship	Foreign Key

PEMBANGUNAN PANGKALAN DATA

Langkah 4: Wujudkan Jadual (*Create Table*)

- Arahan bagi mencipta jadual adalah seperti berikut.

CREATE TABLE <nama jadual> (<nama medan> <jenis medan> NOT NULL PRIMARY KEY <AUTO_INCREMENT sekiranya jenis medan adalah INT>,
<nama medan> <jenis medan>;

i. Untuk mengelakkan data tidak diisi ciri-ciri *NOT NULL* diberikan kepada medan tersebut.
ii. Ciri medan *AUTO_INCREMENT* adalah bagi menambahkan satu nilai seterusnya dalam medan dengan mengambil kira bahawa jenis medan adalah *INT*
iii. Medan yang ditakrif sebagai *PRIMARY KEY* adalah menunjukkan bahawa medan tersebut adalah kunci utama dalam jadual. *PRIMARY KEY* boleh ditakrifkan kepada satu atau lebih medan.

- Contoh arahan bagi mencipta jadual aset adalah seperti berikut.

CREATE TABLE aset (id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
nama VARCHAR(20), catatan VARCHAR(50);

PEMBANGUNAN PANGKALAN DATA

Langkah 5: Wujudkan VIEW (Create VIEW)

- VIEW adalah merujuk kepada jadual maya yang dihasilkan melalui arahan SQL.
- Dalam satu pangkalan data, view dan jadual berkongsi *tablespace*. Namun begitu, view dan jadual tidak boleh mempunyai nama yang sama.
- Beberapa fungsi SQL boleh dimasukkan ke dalam arahan seperti WHERE dan arahan JOIN daripada beberapa jadual lain kepada jadual maya yang mana fungsi arahan tersebut dipaparkan sebagai satu jadual.
- Arahan bagi mencipta view adalah seperti berikut.
  CREATE VIEW <nama_view> AS SELECT <medan1>, <medan2>, ... FROM <nama_jadual> WHERE condition;
- Contoh arahan bagi mencipta view bagi tempahan bilik oleh pengguna adalah seperti berikut.
  CREATE VIEW penggunaTempahBilik AS SELECT pengguna.nama, bilik_mesy.namabilik, bilik_mesy.lokasi FROM pengguna, bilik_mesy WHERE pengguna.nokp == bilik_mesy.nokp;

PEMBANGUNAN PANGKALAN DATA

Langkah 6: Wujudkan INDEX (Create INDEX)

- Kebiasaannya data disimpan tidak mengikut urutan. Data baru yang dimasukkan tidak disimpan mengikut susunan berdasarkan data yang dimasukkan terdahulu. Oleh yang demikian, tempoh untuk menemui data berdasarkan arahan adalah kurang pantas berbanding kemasukkan data. Dengan itu, index perlu dilakukan bagi membolehkan data ditemui dengan lebih cepat.

- Arahan *INDEX* digunakan untuk medan-medan tertentu dalam sesuatu jadual bagi mempercepatkan carian data. Lokasi sesuatu data lebih pantas ditemui berbanding carian satu-persatu baris yang terdapat dalam jadual jika tidak menggunakan *INDEX*.

- Arahan bagi mencipta *index* adalah seperti berikut.
  CREATE INDEX <nama index> ON <nama jadual> (<nama medan>);

- Contoh arahan bagi mencipta indeks no kad pengenalan pengguna adalah seperti berikut.
  CREATE INDEX nokp_idx ON pengguna(nokp);

PEMBANGUNAN PANGKALAN DATA

Langkah 7: Memuat Masuk (Load) Data Ke Dalam Pangkalan Data

- Sekiranya terdapat data daripada pangkalan data lama, data tersebut boleh dimuat masuk ke dalam pangkalan data yang baru dibina. Terdapat dua cara untuk memuat masuk data yang sedia ada ke dalam pangkalan data iaitu:

Pilihan A.

- Menggunakan Penyataan INSERT. Penyataan ini adalah untuk memasukkan rekod ke dalam jadual. Sintaks bagi penyataan INSERT adalah seperti berikut.

INSERT INTO <nama jadual> (<nama medan>) VALUES (data1), (data2);

- Contoh penyataan untuk memasukkan rekod ke dalam jadual adalah seperti berikut.

INSERT INTO pengguna (nama, emel, nombor_telefon, alamat, BAHAGIAN_id) VALUES ('Sanem Can Divit', 'sanem@erkenci.com.my', '03-88723038', 'No.1, Jalan Albatross, 62300 Putrajaya', 'BPI');

PEMBANGUNAN PANGKALAN DATA

Langkah 7: Memuat Masuk (Load) Data Ke Dalam Pangkalan Data

Pilihan B.

- Menggunakan Penyataan *LOAD DATA*. Penyataan *LOAD DATA* membolehkan data banyak yang terdapat dalam fail teks dimasukkan ke dalam pangkalan dengan data menggunakan satu arahan sahaja. Sintaks bagi penyataan *LOAD DATA* adalah seperti berikut.

  LOAD DATA INFILE <nama fail teks.txt> INTO TABLE <nama pangkalandata.nama jadual>;

- Contoh penyataan untuk memasukkan rekod ke dalam jadual adalah seperti berikut.

  LOAD DATA INFILE pengguna.txt INTO TABLE dbtempahan.pengguna;

PEMBANGUNAN PANGKALAN DATA

Langkah 8: Wujudkan Pengguna Dan Kawalan Capaian

- Pengguna diwujudkan untuk mengakses sesuatu pangkalan data. Dengan itu, pengguna perlu mempunyai capaian *root* untuk melaksanakan aktiviti-aktiviti pembangunan pangkalan data dan juga mencipta pengguna. Berikut adalah arahan yang digunakan untuk mencipta pengguna.

  CREATE USER '<nama pengguna> '@'<nama hos>' IDENTIFIED BY '<kata laluan>';

- Contoh arahan untuk mencipta pengguna nama1 adalah seperti berikut:

  mysql> CREATE USER 'nama1'@'localhost' IDENTIFIED BY 'katalaluan';

PEMBANGUNAN PANGKALAN DATA

Langkah 8: Wujudkan Pengguna Dan Kawalan Capaian

- Pentadbir pangkalan data (*Database Administrator* – DBA) boleh memberi kebenaran pengguna untuk mencapai beberapa akses melalui arahan *GRANT* seperti berikut:

  GRANT ALL PRIVILEGES ON database.table TO 'user'@'localhost';

- Arahan di atas memberikan semua kawalan akses *'GRANT ALL PRIVILEGES'* bagi semua pangkalan data dan jadual *'database.table'* kepada pengguna *'user''*. Berikut adalah senarai arahan kebenaran yang digunakan mengikut kesesuaian capaian.

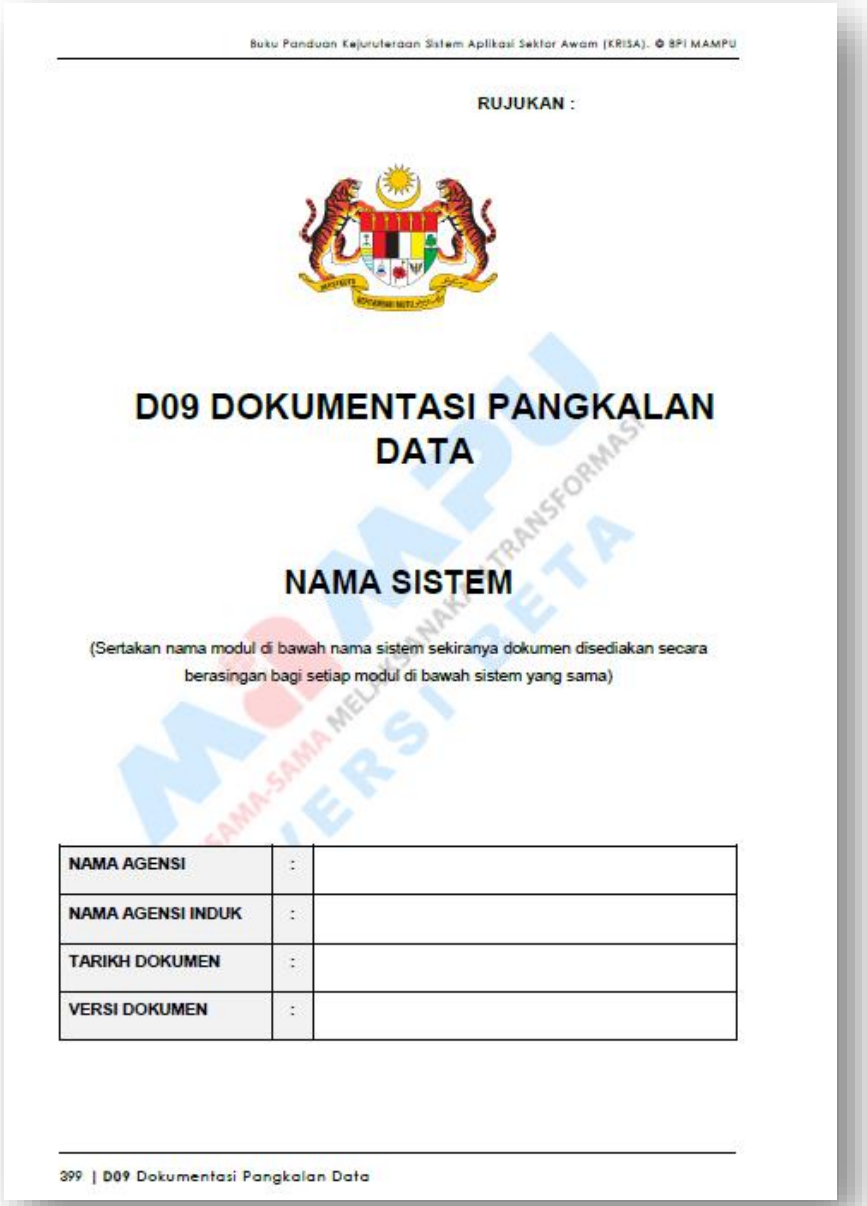
  i. *ALL PRIVILEGES* – membenarkan semua aktiviti
  ii. *CREATE* – membenarkan pengguna mencipta pangkalan data dan jadual
  iii. *DROP* – membenarkan pengguna menghapus pangkalan data dan jadual
  iv. *DELETE* – membenarkan pengguna menghapuskan baris rekod dalam jadual
  v. *INSERT* – membenarkan pengguna menambah baris rekod dalam jadual
  vi. *SELECT* – membenarkan pengguna membaca rekod dalam pangkalan data
  vii. *UPDATE* – membenarkan pengguna mengemaskini rekod dalam jadual

- Pentadbir pangkalan data (*Database Administrator* – DBA) boleh memberi menarik semula kebenaran pengguna melalui arahan REVOKE. Contoh - menarik balik semua kebenaran kawalan akses *'REVOKE ALL PRIVILEGES'* bagi semua pangkalan data dan jadual *'database.table'* kepada pengguna *'user''*.
  REVOKE ALL PRIVILEGES ON database.table TO 'user'@'localhost';

PEMBANGUNAN PANGKALAN DATA

Langkah 9: Dokumenkan Pangkalan Data

- Dokumenkan maklumat pangkalan data fizikal yang dibangunkan ke dalam **D09 Dokumen Pangkalan Data**. Dokumentasi mengikut susunan berikut:
    a. Ringkasan maklumat pangkalan data fizikal.
    b. Skrip yang mengandungi arahan-arahan *SQL*



Buku Panduan Kejuruteraan Sistem Aplikasi Sektor Awam (KRISA). © BPI MAMPU

RUJUKAN :

D09 DOKUMENTASI PANGKALAN DATA

NAMA SISTEM

(Sertakan nama modul di bawah nama sistem sekiranya dokumen disediakan secara berasingan bagi setiap modul di bawah sistem yang sama)

NAMA AGENSI	:	
NAMA AGENSI INDUK	:	
TARIKH DOKUMEN	:	
VERSI DOKUMEN	:	

399 | D09 Dokumentasi Pangkalan Data

FASA **PEMBANGUNAN**

TAKLIMAT

02

F4.2 PENGATURCARAAN APLIKASI

OBJEKTIF

- Menulis kod dalam struktur yang tersusun dan ringkas supaya mudah untuk dibaca dan difahami, diubahsuai / ditambahbaik dan diselengara.
- Menukarkan dokumen SDS kepada kod aturcara dan membuat ujian unit ke atas kod yang dibangunkan. Seterusnya menghasilkan produk iaitu Sistem Aplikasi yang memenuhi kehendak pengguna.

FASA PEMBANGUNAN

PENGATURCARAAN APLIKASI

PENGENALAN

- Proses untuk menulis kod aturcara bagi menghasilkan satu sistem aplikasi. Kod ini menentukan tindakan yang perlu dilaksanakan oleh system
- Fasa ini sangat penting kerana ia mempengaruhi fasa seterusnya iaitu fasa pengujian dan penyelenggaraan. Satu kod yang ditulis dengan baik mampu mengurangkan kerja-kerja pada kedua-dua fasa tersebut.
- Sehubungan itu, tumpuan harus diberikan semasa fasa pembangunan ini supaya dapat menghasilkan kod aturcara yang berkualiti dan memenuhi keperluan pengguna.

01

TEKNIK PENGATURCARAAN

- Tidak Berstruktur
- Berstruktur
- Berorientasikan Objek
- Web
- Stored Procedure

02

LANGKAH-LANGKAH PENGATURCARAAN

- Sediakan Keperluan Pra Pembangunan
- Pertimbang Dan Laksanakan Amalan Terbaik Dalam Pengaturcaraan
- Membangunkan Sistem
- Menyediakan Dokumentasi Kod Sumber

PENGATURCARAAN APLIKASI

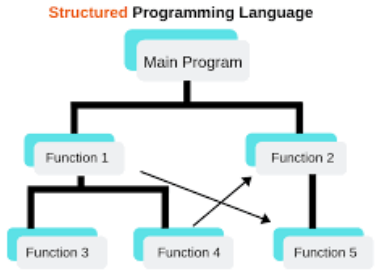
TEKNIK PENGATURCARAAN

1 TIDAK BERSTRUKTUR

Kod ditulis tanpa berstruktur, semua arahan ditulis dalam fungsi main().

2 BERSTRUKTUR

Pengaturcaraan bertatacara. Kod yang besar dipecahkan kepada kaedah-kaedah pendek (juga dikenali sebagai fungsi atau tatacara) yang lebih kecil agar mudah difahami.



3 BERORIENTASIKAN OBJEK

Aturcara komputer yang terdiri daripada sekumpulan unit-unit atau objek. Setiap objek berupaya untuk menerima dan menghantar mesej (pesanan) kepada objek lain

4 WEB

Penulisan, markup dan pengekodan yang terlibat dalam pembangunan web, ia termasuk kandungan web, pelanggan web, skrip pelayan dan keselamatan rangkaian

5 STORED PROCEDURE

Satu set arahan komputer yang disimpan dalam pangkalan data untuk membuat capaian data dari pangkalan data.

PENGATURCARAAN APLIKASI

LANGKAH - LANGKAH PENGATURCARAAN

LANGKAH 01

Sediakan Keperluan Pra Pembangunan

Menyediakan keperluan teknikal berdasarkann Penyediaan Pelan Pembangunan Sistem [F1.1] Langkah 4: Proses Teknikal.

Memahami dengan mendalam Spesifikasi Keperluan Sistem dan Spesifikasi Rekabentuk Sistem

- Pendekatan
- Teknik
- *Tools*
- Bahasa pengaturcaraan
- Pangkalan data

LANGKAH 02

Pertimbang Dan Laksanakan Amalan Terbaik Dalam Pengaturcaraan

LANGKAH 03

Membangunkan Sistem

LANGKAH 04

Menyediakan Dokumentasi Kod Sumber

PENGATURCARAAN APLIKASI

LANGKAH 2 : PERTIMBANG DAN LAKSANAKAN AMALAN TERBAIK DALAM PENGATURCARAAN

- ✓ Satu set peraturan tidak rasmi yang dihasilkan daripada pengalaman dan pembelajaran oleh komuniti pembangun sistem dari semasa ke semasa.
- ✓ Setiap bahasa pengaturcaraan mempunyai set peraturan yang tersendiri.
- ✓ Teknik dan amalan pengaturcaan yang baik dapat meningkatkan kualiti dan prestasi perisian yang dihasilkan.

LANGKAH 2 : PERTIMBANG DAN LAKSANAKAN AMALAN TERBAIK DALAM PENGATURCARAAN

PENGGUNAAN KONVENSYEN ULASAN

Bahagian atas setiap fail kod sumber	Fungsi / Method	In line	Classes and Interfaces	Pembolehubah
1. Pengaturcara asal 2. Tarikh 3. Tujuan 4. Algorithma yang digunakan (merujuk kepada SDS mana) 5. Senarai pengubahsuaian kepada kod aturcara	1. Purpose of method 2. Argument descriptions 3. Result descriptions 4. Exceptions thrown	Kod yang rumit dan kurang jelas, diletakkan komen pada bahagian atas sebelum kod atau sebaris dengan kod yang ditulis, untuk memberi penerangan mudah berkaitan kod yang ditulis.	1. Nama Kelas 2. Keterangan berkaitan kelas dan tujuannya 3. Versi Semakan 4. Nama pengaturcara asal 5. Version	Ulasan untuk pembolehubah sepatutnya ringkas sahaja,menerangkan secara ringkas apa kegunaannya.

LANGKAH 2 : PERTIMBANG DAN LAKSANAKAN AMALAN TERBAIK DALAM PENGATURCARAAN

PENGGUNAAN KONVENSYEN PEMFORMATAN

Indentation dan Layout

- Tetapkan saiz yang standard untuk inden. Secara global saiz inden ditetapkan kepada 4 ruang (4 spaces) untuk setiap peringkat.
- Baris kod tidak terlalu panjang, elakkan penggunakan horizontal scroll. Pecahkan kod-kod yang panjang (Break up long lines) kepada yang lebih pendek.

Bracketing

- Selaraskan susunan penggunaan '{' dan '}' supaya selari, atau menggunakan slanting style, di mana '{' adalah dihujung kod manakala '}' selari dengan permulaan kod.

Penulisan HTML

- Tetapkan format untuk tags dan attributes, seperti menggunakan huruf besar untuk semua tags dan huruf kecil untuk attributes

Penulisan penyataan SQL

- Gunakan huruf besar untuk kata kunci (SELECT, DELETE, UPDATE, WHERE, ORDER BY) dan huruf kecil elemen pangkalan data seperti nama tables, columns dan views.
- Susun setiap klausa SQL yang utama pada baris yang berasingan supaya kenyataan ini adalah lebih mudah untuk dibaca dan dipinda.

LANGKAH 2 : PERTIMBANG DAN LAKSANAKAN AMALAN TERBAIK DALAM PENGATURCARAAN

PENGGUNAAN KONVENSYEN PENAMAAN

Tips #1	Tips #2	Tips #3	Tips #4
Gunakan nama yang memberi makna, mudah difahami dan sesuai dengan tujuannya. Contoh fungsi Cetak(), melaksanakan fungsi untuk mencetak.	Penggunaan nama yang baik mampu menggambarkan kandungan/ tujuan entiti tersebut (Self Describing) dan mudah dicari.	Elakkan penggunaan nama-nama yang hanya berbeza pada aksara, contoh cetak dan Cetak tetapi berbeza tujuan.	Elakkan penggunakan nama yang boleh mengelirukan, contohnya x. Tidak memberi sebarang makna.

LANGKAH 2 : PERTIMBANG DAN LAKSANAKAN AMALAN TERBAIK DALAM PENGATURCARAAN

Jenis	Konvensyen Penamaan	Contoh
Classes	UpperCamelCase: bermula dengan huruf besar dan pada permulaan setiap perkataan baru	DaftarAset(); PermohonanAset(); CleverClassName();

Methods	lowerCamelCase: bermula dengan huruf kecil dan huruf besar pada permulaan setiap perkataan baru	main();
		kiraJumlah();
		aUsefulMethod();

Atrribute / Variables	• lowerCamelCase: bermula dengan huruf kecil dan huruf besar pada permulaan setiap perkataan baru • nama-nama pembolehubah tidak harus bermula dengan garis bawah (_) atau tanda dolar ($) walaupan dibenarkan • nama pembolehubah sepatutnya pendek tetapi memberi makna • penggunaan 1 aksara sebagai pembolehubah perlu dielakkan kecuali pada pembolehubah sementara, untuk kegunaan loop / for yang digunakan sebagai pembolehubah bilangan sahaja. Kebiasaan pembolehubah sementara menggunakan aksara i,j,k dan m untuk int dan c,d dan e untuk char.	Pembolehubah sementara: int i; char c; myVariable; aClassAttribute; float jumlahMarkah;
Constants / Magic Number	• UpperCase • Dipisahkan dengan '_' setiap perkataan	int MAX_PESERTA = 10;

LANGKAH 2 : PERTIMBANG DAN LAKSANAKAN AMALAN TERBAIK DALAM PENGATURCARAAN

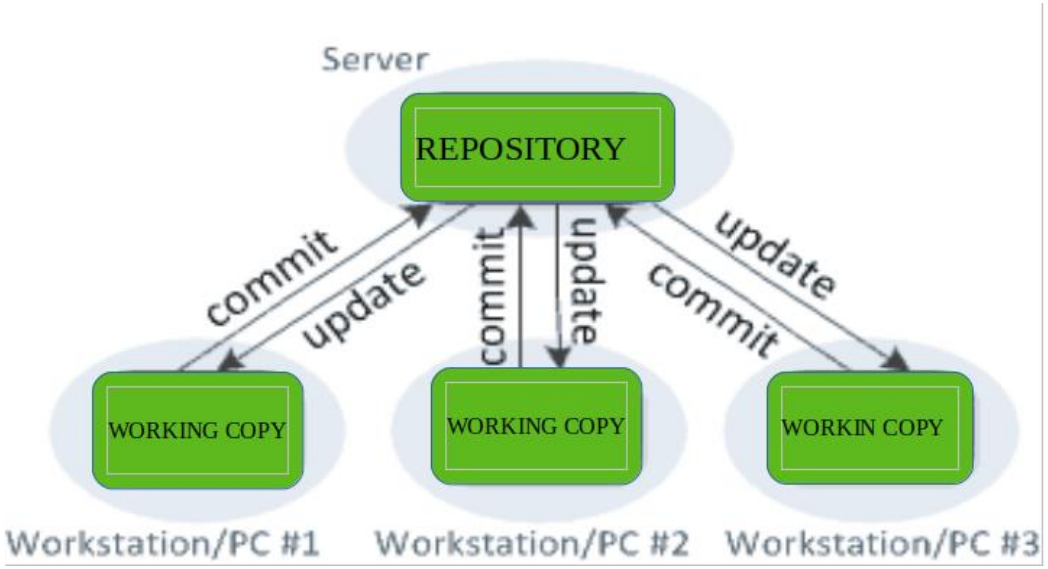
PENGGUNAAN KONVENSYEN PENGATURCARAAN

Kod Sumber

- Kod yang ditulis perlu mudah dan jelas - elakkan penulisan kod yang panjang, contohnya melebihi 20 baris. Amalkan penggunaan pengaturcaraan berstruktur / bertatacara.
- Elakkan penggunaan nilai hard-coded / magic numbers – kod sumber seharusnya tidak menggunakan hard-coded untuk merujuk kepada mana-mana parameter seperti paths, file, host, alamat IP, URLs, ports dan lain-lain.
- Mewujudkan Program Versioning
- Mewujudkan fungsi / method yang boleh dikongsi
- Security - validation, code hard (SQL injection)
- Direktori simpanan - Simpan fail dalam folder secara teratur untuk mrmudahkan carian
- Menyediakan mesej ralat yang menggambarkan ralat sebenar, agar mudah untuk mengesan punca ralat.
- Elakkan penggunaan penyataan bersarang yang melebihi 3 peringkat (deeply nested control statements).

PENGATURCARAAN APLIKASI

Kawalan versi secara berpusat

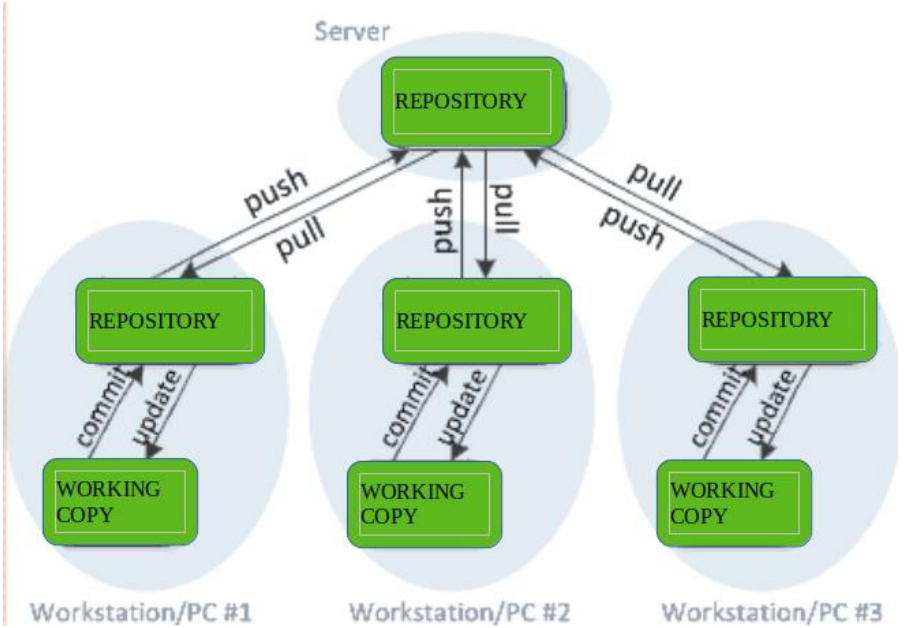


CVS

SUBVERSION

Microsoft Visual SourceSafe 6.0

Kawalan versi secara teragih

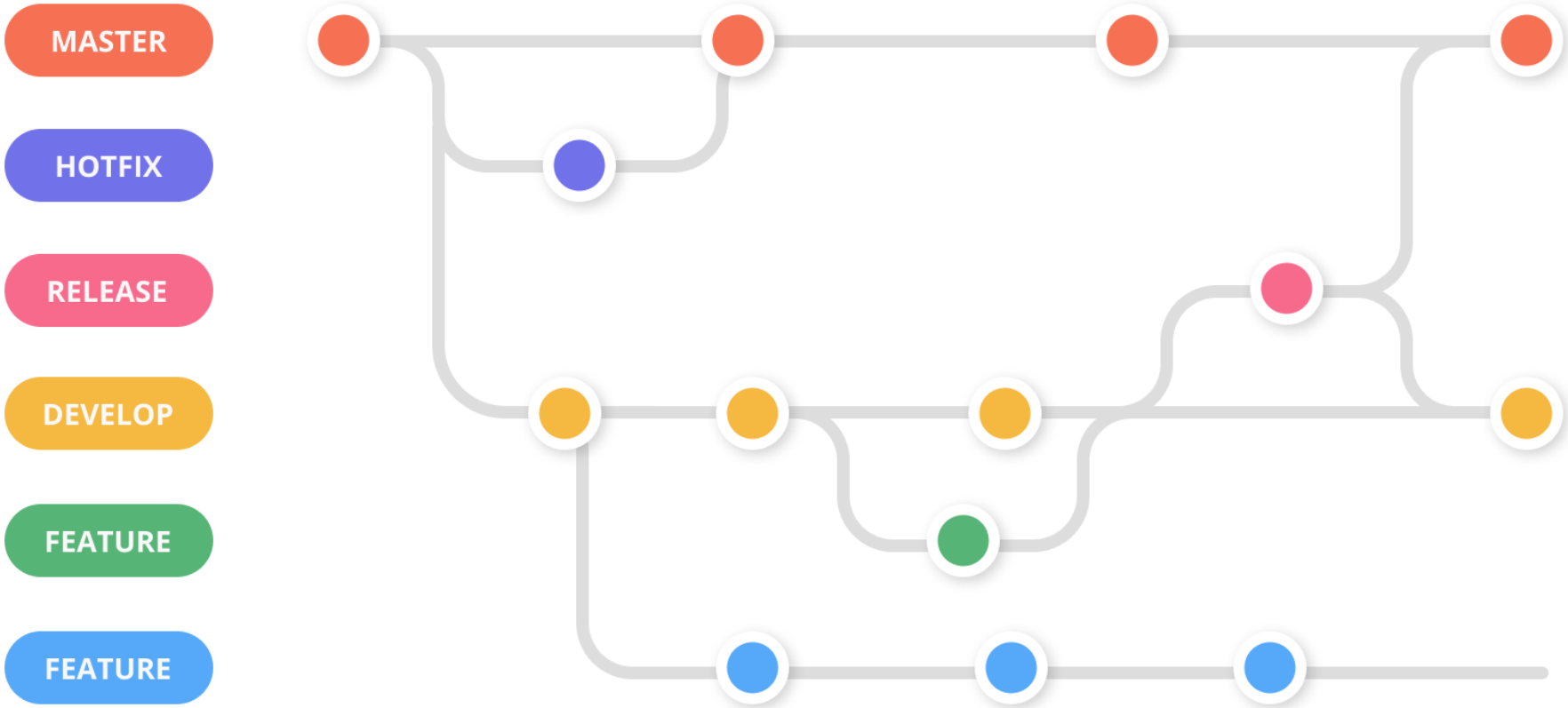


git

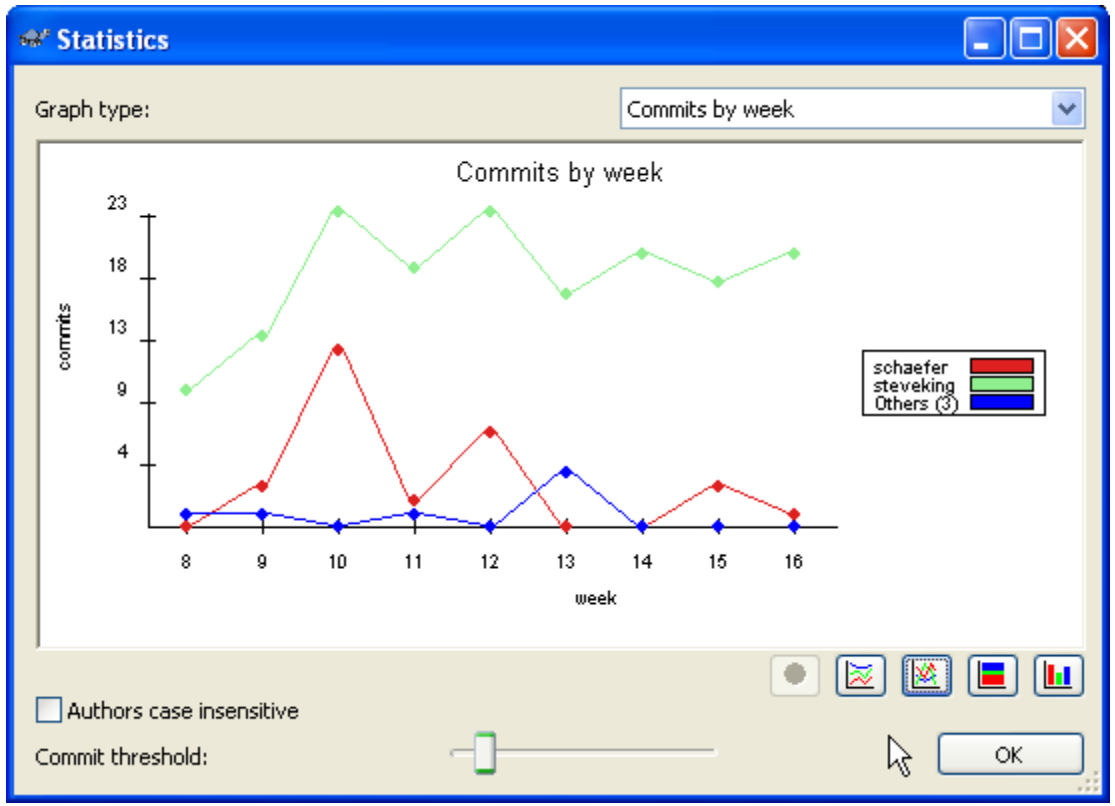
mercurial

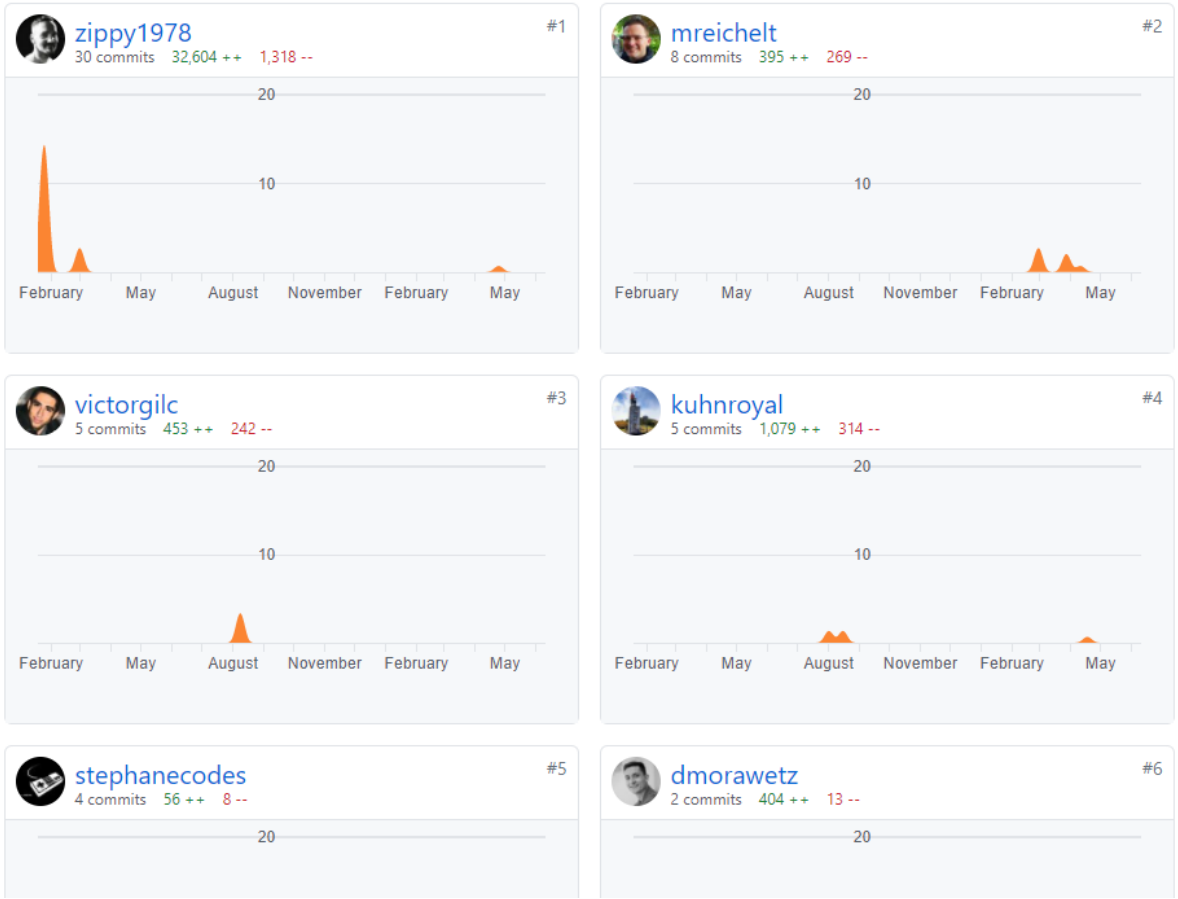
Bazaar

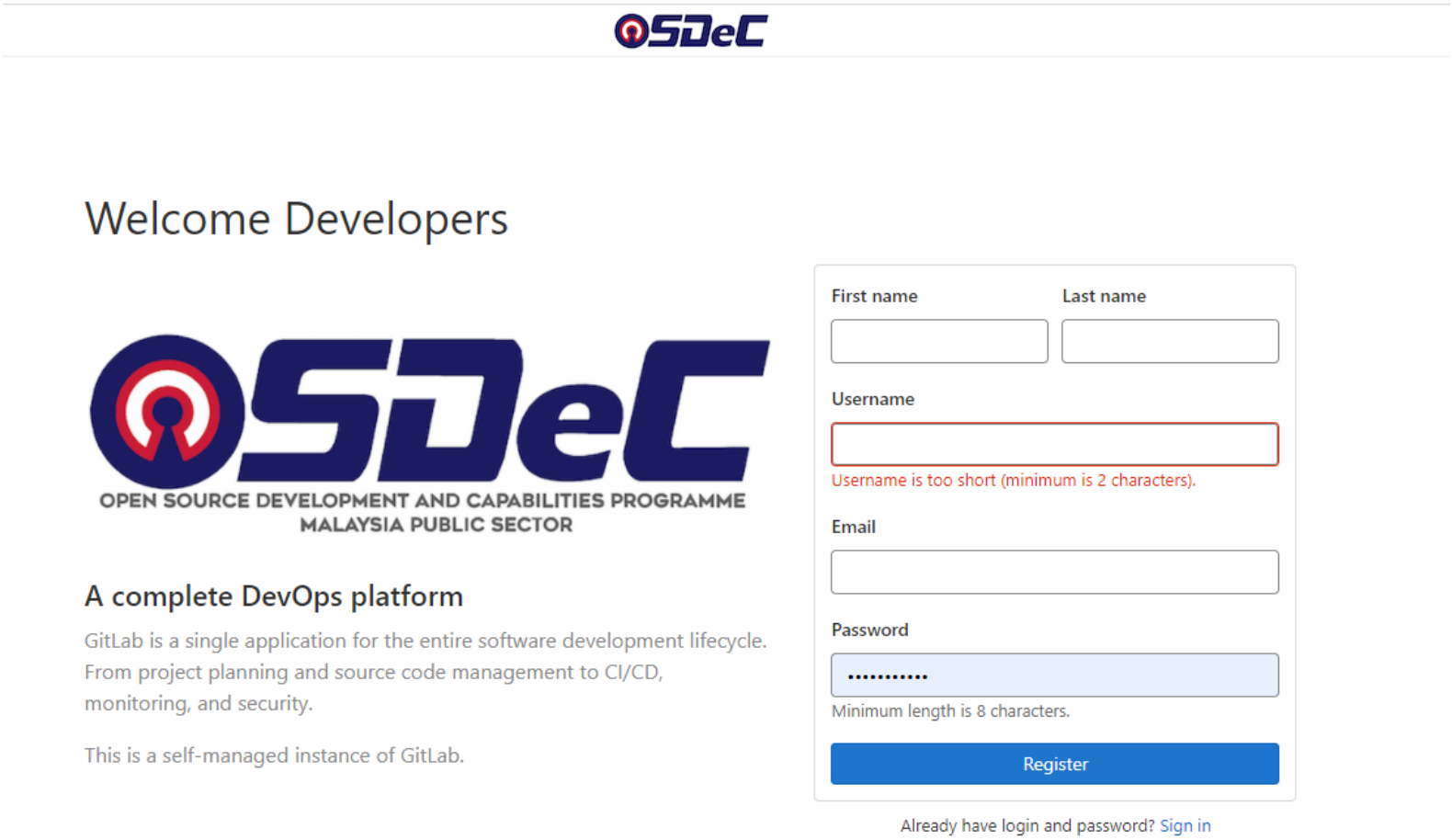
Teknik Percabangan



FASA PEMBANGUNAN



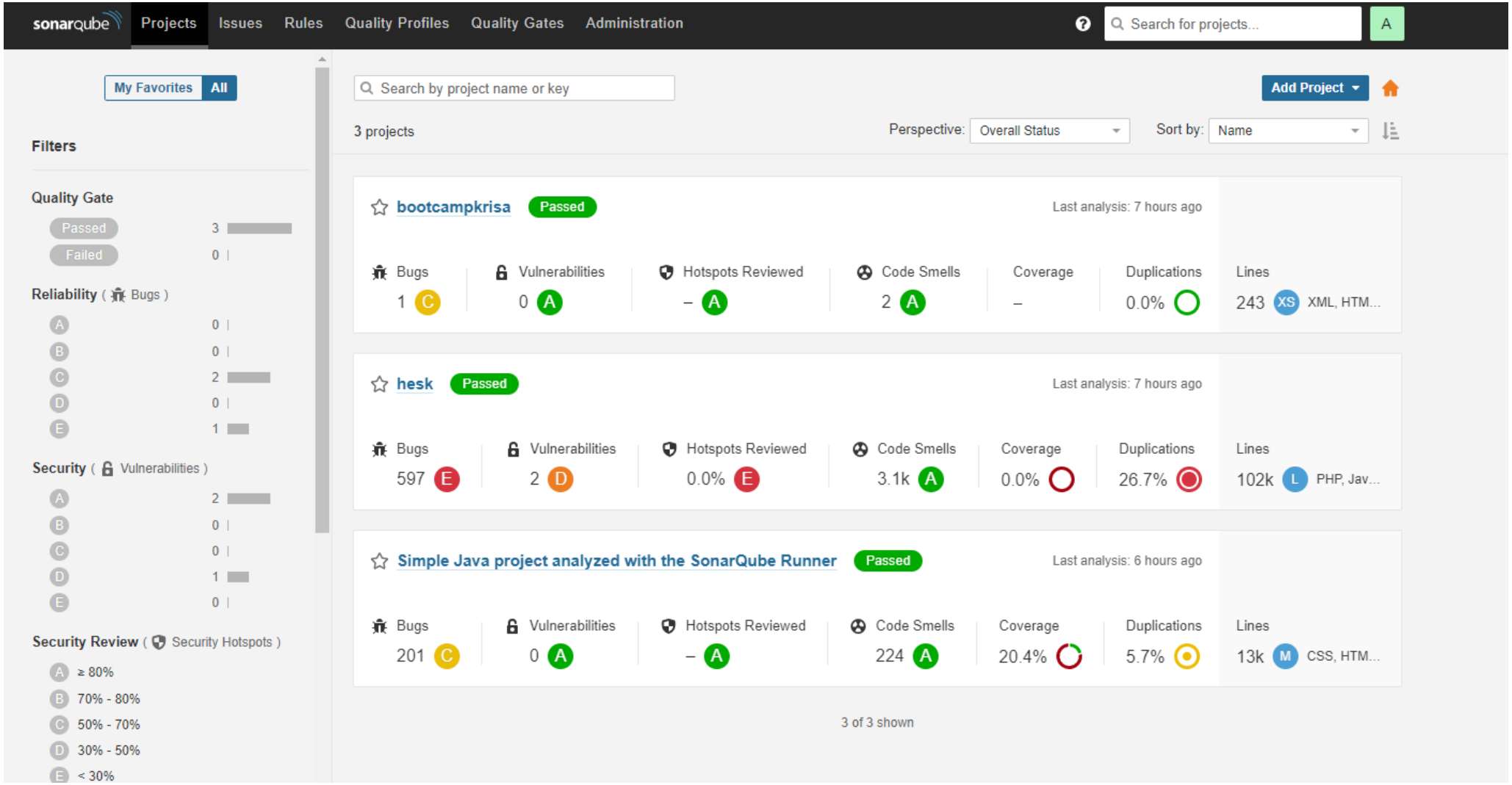




https://git.osdec.gov.my/users/sign_up

FASA PEMBANGUNAN

sonarqube



LANGKAH 2 : PERTIMBANG DAN LAKSANAKAN AMALAN TERBAIK DALAM PENGATURCARAAN

PENGGUNAAN KONVENSYEN PENGATURCARAAN

Pengaturcaraan Pangkalan Data

Elakkan penggunaan SELECT *, Biasakan menulis dengan jelas kolum-kolum yang dikehendaki.

Gunakan stored procedure untuk menggantikan penyataan SQL dalam kod sumber bagi meningkatkan prestasi capaian data.

Melakukan data validation pada client semasa data entry. Ini dapat mengelakkan capaian ke pangkalan data berulang kali dengan data yang tidak sah.

LANGKAH 3: MEMBANGUNKAN SISTEM

Menyediakan persekitaran pembangunan (*development environment*)

- Membina dan memasang perisian, perkakasan (*server*), rangkaian, *tools* dan lain-lain peralatan yang berkaitan.
- Menguji persekitaran pembangunan supaya memenuhi keperluan pembangunan.

Membangunkan Sistem

Mewujudkan dokumentasi *coding standard* (Rujuk Dokumentasi Kod sumber)

Memulakan penulisan kod aturcara (Rujuk Amalan Baik dalam Pengaturcaraan)

Melaksanakan ujian unit bagi setiap komponen sistem dan mendokumenkan keputusan dalam Laporan Ujian (Keterangan lanjut dalam perenggan Ujian Sistem)

DOKUMENTASI KOD SUMBER

- Menerangkan tujuan dokumen ini dihasilkan. Senaraikan kumpulan sasar dokumen ini. Nyatakan andaian, batasan dan kekangan dalam menyediakan dokumentasi kod sumber ini.

Cara penulisan kod
i) File Name
ii) ii) Class Headers and Declaration
iii) Method Headers and Declaration
iv) Indentation
v) Inline Comments
vi) Variable Names
vii) Use of Braces
viii) Line Length
ix) Spacing
x) Program Statement

TUJUAN DOKUMEN

SKOP DOKUMEN

PIAWAIAN KOD SUMBER

LAMPIRAN

Kembali

- Menyatakan skop bagi penyediaan kod sumber. Berikut adalah contoh maklumat yang boleh dinyatakan:
- i) Nama sistem yang terlibat
- ii) Nama modul yang terlibat
- iii) Nama pasukan pembangunan sistem yang terlibat

- Seksyen ini merupakan ruangan untuk menyertakan dokumen-dokumen sokongan yang perlu dirujuk seperti format borang fizikal, format laporan dan pelbagai lagi dokumen-dokumen lain yang berkaitan.

Langkah 4 : Menyediakan Dokumentasi Kod Sumber

1

Dokumentasi kod sumber adalah dokumen yang mengandungi senarai kod aturcara yang meliputi struktur direktori dan hierarki fail serta garis panduan yang mengesyorkan perkara berkaitan gaya pengaturcaraan (*style*), konvensyen penamaan, *indentation*, ulasan / komen, pengistiharan pembolehubah, penyataan SQL dan lain-lain yang perlu dipatuhi oleh semua pengaturcara.

2

Tujuan pengwujudan dokumen ini adalah untuk mewujudkan habit / kebiasaan yang baik dalam gaya penulisan kod dan mewujudkan keseragaman kepada semua pengaturcara.

3

Pengaturcara sangat digalakkan untuk mengikuti garis panduan ini supaya:

i) meningkatkan kebolehbacaan dan kefahaman ke atas kod sumber mereka.
ii) memudahkan semakan, penambahbaikan dan penyelenggaraan ke atas perisian.
iii) proses pemindahan teknologi (*Transfer Of Technology*) kepada ahli pasukan lain/ baru menjadi mudah dan lebih cepat.
iv) memudahkan pasukan pengaturcara membuat carian kepada fungsi-fungsi atau kelas dalam kod yang ditulis, seterusnya menggalakkan penggunaan semua kod sedia ada (*reusability*) dan
v) sebagai rujukan pada masa depan.

PENGATURCARAAN APLIKASI

Langkah 4 : Menyediakan Dokumentasi Kod Sumber

4 Salah satu cabaran dalam mendokumentasikan kod sumber adalah memastikan bahawa ulasan / komen dikemaskini selari dengan kod aturcara yang ditulis. Kekangan masa menyebabkan pengaturcara hanya fokus kepada kod yang tulis dan sangat sedikit masa diberikan kepada penulisan dokumen kod sumber.

5 Menjana dokumen secara automatik (***Documentation Generator***). Terdapat *tool* yang membolehkan dokumen ini dijana secara automatik dari kod sumber yang ditulis. Ia dikenali sebagai *documentation generator*.

6 *Documentation generator* dijana dari kod sumber seperti ulasan (***comment***) yang ditulis oleh pengaturcara, ini menjadikan dokumen mudah dikemaskini dari masa ke masa selaras dengan kod yang ditulis.

7 Setiap Bahasa pengaturcaraan mempunyai *documentation generator* sendiri. Contoh: Doxygen, NDoc, javadoc, EiffelStudio, Sandcastle, ROBODoc, POD, TwinText, atau Universal Report dan lain-lain.

Unit Pemodenan Tadbiran dan Perancangan Pengurusan Malaysia

TERIMA KASIH