|  |  |  |
| --- | --- | --- |
| **BAHAGIAN PENDIDIKAN DAN LATIHAN TEKNIK VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN PERSEKUTUAN**  **KERTAS PENERANGAN**  ***(INFORMATION SHEET)***  C:\Users\norsu\Desktop\jata negara.png | | |
| **KOD DAN NAMA PROGRAM NOSS** | IT-010-3:2016 PEMBANGUNAN APLIKASI | |
| **TAHAP NOSS** | 3 | |
| **KOD, NAMA CU DAN WA NOSS** | CU02/WA2 – PLAN MODULE EXPECTED BEHAVIOUR | |
| **NAMA PROGRAM KV** | SISTEM PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB | |
| **KOD DAN NAMA KURSUS KSKV** | KPD2033 FUNDAMENTAL OF PROGRAMMING | |
| **NO DAN TAJUK STANDARD KANDUNGAN KSKV** | K2 PLAN MODULE EXPECTED BEHAVIOUR | |
| **NO. KOD NOSS** | IT-010-3:2016-C02/P(33/70) | Muka Surat : 1 Drp : 10 |
| **NO. KOD KSKV** | KPD2033/P(8/13) |

**TAJUK/***TITLE***:**

**KAEDAH TEST DRIVEN DEVELOPMENT (TDD)**

**TUJUAN/***PURPOSE* **:**

Kertas penerangan ini adalah bertujuan menerangkan mengenai :

1. Menyenaraiikan kaedah pembangunan pengujian
2. Kaedah pembangunan pengujian

**PENERANGAN/***INFORMATION* **:**

**KAEDAH TEST DRIVEN DEVELOPMENT (TDD)**

**Test Driven Development adalah Peringkat pembangunan yang dimulakan dengan menuliskan test terlebih dahulu kemudian  production code.**

**Terdapat empat kaedah yang digunakan dalam TDD iaitu:**

**i.Pseudo Kod**

**ii.Data Flow Diagram(DFD)**

**iii.Modul Aliran Proses**

**iv.Jadual Keputusan**

**i.Pseudo Kod**

**Pseudo kod ialah satu set aturan yang ditulis dalam bahasa tabii untuk menyelesaikan sesuatu masalah.**

1. Merupakan kod palsu yang mirip kod aturcara tetapi ditulis dalam bahasa pertuturan manusia dan mempunyai nombor turutan.
2. Ia menunjukkan langkah demi langkah untuk melaksanakan sesuatu tugas dari mula hingga tamat.
3. Dibuat sebelum pengkodan aturcara sebenar untuk pastikan kaedah penyelesaian masalah yang digunakan adalah betul.
4. Menggunakan ungkapan matematik, ungkapan logik dan struktur kawalan komputer seperti IF, ELSE, INPUT, OUTPUT.
5. Contoh pseudo kod bagi sebuah sistem untuk mengira luas bulatan apabila pengguna memasukkan nilai jejari bulatan.

1. Mula

2. INPUT jejari\_bulatan

3. luas\_bulatan = 3.142 \* jejari\_bulatan \* jejari\_bulatan

4. OUTPUT luas\_bulatan

5. Tamat

**Membezakan aliran urutan dan pilihan tunggal bagi pseudokod.**

1. Pseudokod aliran secara urutan
2. Pseudokod pilihan tunggal

|  |  |
| --- | --- |
| **a) Pseudokod aliran secara urutan** | **b) Pseudokod pilihan tunggal** |
| Mula  Masak air  Masukkan kopi ke dalam cawan  Masukkan gula  Tuangkan air panas ke dalam cawan  Tamat | Mula  Masak air  Masukkan kopi ke dalam cawan  Masukkan gula  Jika perlu krim  Tambah krim  Masukkan air panas ke dalam cawan  Tamat |

Jadual 1 Perbezaan antara aliran secara urutan dengan pilihan tunggal bagi pseudokod

**Situasi 1**:

Katakan anda hendak menghantar bahan bercetak, nilai bayaran yang dikenakan adalah berdasarkan berat bahan yang hendak dihantar itu. Jika bahan yang dihantar itu beratnya 50 gram, berapakah nilai yang perlu dibayar?

Contoh pseudokod bagi **situasi 1 adalah:**

Mula

Timbang berat bahan bercetak ( input )

Set nilai bayaran dikenakan (proses)

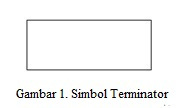
Papar nilai bayaran (output)

Tamat

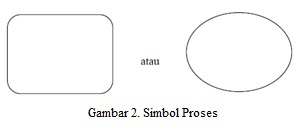
**ii.Data Flow Diagram(DFD)**

**Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data pada suatu sistem** atau menjelaskan proses kerja suatu sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logikal, tersruktur dan jelas. Secara singkatnya, DFD adalah alat pemodelan untuk memodelkan aliran kerja sistem.

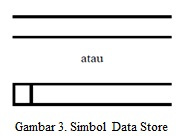
### .KOMPONEN DFD

****

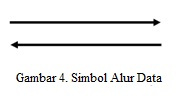
* **User / Terminator**: Kesatuan diluar sistem (external entity) yang memberikan input ke sistem atau menerima output dari sistem berupa orang, organisasi, atau sistem lain

****

* **Process**: Aktivitas yang mengolah input menjadi output.

****

* **Data Store**: Penyimpanan data pada *database*, biasanya berupa jadual.

****

* **Data Flow**: Aliran data pada sistem (antar proses, antara terminator & proses, serta antara proses & data store).

**iii. MODUL ALIRAN PROSES**

**CARTA ALIR (*FLOWCHART*)**

**Carta Alir ialah kaedah grafik yang menunjukkan langkah-langkah bagi penyelesaian masalah dan hubung kait antara satu sama lain. Carta alir mengandungi simbol geometrik yang dihubungkan menggunakan anak panah. Setiap simbol mewakili aktiviti bagi langkah yang akan dilaksanakan oleh program. Aktiviti ini akan dilaksanakan mengikut turutan dari atas ke bawah.**

1. Menggunakan simbol grafik untuk mewakili arahan-arahan penyelesaian masalah.
2. Terdiri daripada nod dan garis penghubung.
3. Setiap elemen dalam carta alir mempunyai simbol standard dan fungsi tertentu.
4. Berikut merupakan simbol standard yang lazim digunakan:

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama Nod** | **Fungsi** |
|  | Terminal  *Terminator* | Permulaan / penamatan carta alir |
|  | Input / Output | Membaca data yang dimasukkan (input) atau memaparkan data / hasil (output) |
|  | Proses  *Process* | Apa-apa jenis operasi pemprosesan data seperti operasi matematik, operasi logik, perpindahan data, penghapusan data dan lain-lain |
|  | Syarat / Pengujian  *Decision* | Menguji syarat yang terkandung dalam simbol berkenaan. Aliran akan ditentukan melalui keputusan syarat sama ada Benar (*True*) atau Palsu (*False*) |
|  | Penghubung  *Connector* | Titik sambungan yang menghubungkan carta alir yang terpisah dalam satu muka surat yang sama. |
|  | Penghubung  *Off-page Connector* | Titik sambungan yang menghubungkan carta alir yang terpisah ke muka surat yang berlainan. |
|  | Aliran Aktiviti  *Flow Line* | Menghubungkan nod-nod untuk menunjukkan aliran proses. |

1. Contoh carta alir bagi sebuah sistem untuk mengira luas bulatan apabila pengguna memasukkan nilai jejari bulatan:

Mula

Tamat

OUTPUT

luas\_bulatan

INPUT

jejari\_bulatan

luas\_bulatan = 3.142 \* jejari\_bulatan \* jejari\_bulatan

**Situasi: Membancuh kopi**

Contoh carta alir bagi **Proses Membancuh Kopi**

Masuk

kan

kopi ke dalam

cawan

Mula

Masak Air

Tamat

Masukkan

gula

Tuangkan air

panas ke dalam

cawan

Kacau a

ir

Rajah 1: Carta Alir proses membancuh kopi

#### Membezakan aliran urutan dan pilihan tunggal bagi carta alir

a) Aliran secara urutan (carta alir)

dan Pilihan tunggal (carta alir)

|  |  |
| --- | --- |
| **a) Carta alir secara urutan** | **b) Carta alir pilihan Tunggal** |
|  | Mula |
| Masukkan    kopi ke dalam cawan    Masukkan    gula    Mula    Masak Air    Tamat      Kacau air    Tuangkan air    panas ke dalam | Masak Air    Perlu      Krim  ?    Tuangkan air    panas ke dalam cawan    Tidak    Ya    Masukkan    kopi ke dalam cawan    Masukkan    gula    Masukkan krim    Tamat    Kacau air |

Jadual 2. Perbezaan antara aliran secara urutan dengan pilihan tunggal bagi carta alir

**iv.Jadual Keputusan**

* + 1. Menunjukkan struktur logikal yang menerangkan logik proses
    2. Setiap kombinasi logikal ditunjukkan
    3. Hasil digabungkan dan dimudahkan
    4. Pengaturcara boleh menggunakan jadual keputusan untuk membangunkan kod.

****

Rajah di atas menggambarkan polisi check-cashing yang muncul di belakang kad check-cashing untuk kedai runcit. Dasar yang sama ini telah didokumenkan dengan jadual keputusan. Tiga keadaan memberi kesan kepada keputusan check-cashing: jenis cek, sama ada jumlah cek itu melebihi had maksimum, dan sama ada syarikat itu diakreditasi oleh kedai. Tindakan (keputusan) sama ada untuk membayar cek atau menolak wang tunai cek. Perhatikan bahawa setiap gabungan syarat mentakrifkan suatu peraturan yang menghasilkan suatu tindakan, yang ditandakan oleh X.

**SOALAN/***QUESTION* **:**

1. Apakah **Test Driven Development?**.
2. Senaraikan Empat kaedah dalam TDD.
3. Senaraikan notasi dan keterangannya yang terdapat dalam Jadual Keputusan.

**RUJUKAN/***REFERENCE* **:**

1. Keith Cooper, Linda Torczon(2011), Engineering a Compiler (2nd Edition), Morgan Kaufmann, ISBN: 978-0120884780
2. Todd Zaki Warfel (2009), Prototyping: A Practitioner's Guide, Louis Rosenfeld Media, LLC, ISBN I-933820-21-7
3. Steve McConnell (2004), Code Complete: A Practical Handbook of Software Construction, (2nd Edition),Microsoft Press, ISBN: 978-0-7356-1967-8
4. Michael L. Scot (2009) , Programming Language Pragmatics, Third Edition, Morgan Kaufmann, ISBN-13: 978-0123745149
5. Simon Marlow (2013), Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multithreaded Programming (1st Edition), O'Reilly Media, ISBN-13: 978-1449335946