|  |  |  |
| --- | --- | --- |
| **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/WA1 – INTERPRET APPLICATION MODULE DEVELOPMENT REQUIREMENT** | |
| **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** | K1 INTERPRET APPLICATION MODULE DEVELOPMENT REQUIREMENT | |
| **NO. KOD NOSS** | IT-010-3:2016-C02/P(12/70) | Muka Surat : 1 Drp : 10 |
| **NO. KOD KSKV** | KPD2033/P(4/13) |

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

**KONSEP PENGATURCARAAN**

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

Kertas penerangan ini adalah bertujuan menerangkan mengenai :

* Konsep pengaturcaraan.
* Tahap dan generasi bahasa pengaturcaraan

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

1. **PENGENALAN KEPADA KONSEP PENGATURCARAAN**

* **Programming** atau **bahasa pengaturcaraan** ialah **satu set perkataan, simbol-simbol dan kod-kod yang membolehkan manusia berkomunikasi dengan komputer-komputer.**
* Ia juga bahasa buatan yang digunakan untuk menulis arahan yang boleh diterjemahkan ke dalam bahasa mesin dan kemudian dilaksanakan oleh komputer.

|  |  |
| --- | --- |
| Pengguna | Orang yang akan menggunakan aturcara yang ditulis. |
| Komputer | Jenis komputer dan sistem pengoperasian yang akan digunakan, samaada ia berdiri sendiri atau rangkaian komputer pelanggan / pelayan. |
| Pengaturcara | * Pengetahuan dalam menulis aturcara sama ada akan melibatkan pengaturcara lain. * Orang yang menulis kod aturcara. |

Rajah 1: perkara penting yang perlu dipertimbang sebelum menulis aturcara

|  |  |  |
| --- | --- | --- |
| C | Pascal | Java |
| C++ PHP | Basic | HTML |
| Phyton | COBOL | SQL |

Jadual 1: Contoh bahasa pengaturcaraan

* **Pengaturcara** ialah **orang yang mahir dalam bahasa pengaturcaraan, menggunakan bahasa ini untuk menghasilkan perisian komputer**.
* Setiap bahasa yang dicipta mempunyai tujuan untuk memberi penyelesaian kepada masalah tertentu.
* Jenis bahasa pengaturcaraan yang digunakan, ada satu kesamaan yang nyata, iaitu mereka akan ditukarkan kepada bentuk digital komputer 0 dan 1 yang mewakili tenaga elektronik tutup dan buka.
* Konsep bahasa pengaturcaraan pangkalan data.
* SQL atau Structured Query Language merupakan bahasa yang digunakan dalam proses meminta data dari pangkalan data dan menerima semula hasilnya.
* Aturcara berstruktur mempunyai struktur aturcara seperti berikut :

|  |  |
| --- | --- |
| Deklarasi | Sekumpulan pembolehubah yang mempunyai jenis data yang spesifik. Semua p/u mesti dideklarasi sebelum dilaksanakan.  int a, b, c;  char nama, no\_pend; |
| Masukan | Sekumpulan maklumat yang akan dimasukkan ke dalam komputer melalui peranti input seperti keyboard dll. Input akan diproses menghasilkan output yang diingini. |
| Storan | Setiap maklumat akan disimpan di dalam memori komputer dan dikodkan dengan kombinasi unik kosong dan satu. |
| Operasi | 2 jenis iaitu :  i. batch mode – simpan dan proses sekali  ii. Interactive mode – pengguna dan komputer boleh berinteraksi sesama sendiri semasa proses pengiraan dilaksanakan |
| Kawalan | Susunan pelaksanaan arahan dalam program samada secara berjujukan (satu-per-satu iaitu dari atas ke bawah) atau secara tidak berjujukan |
| Output | Hasil daripada pemprosesan input samada dipaparkan pada skrin ataupun kertas |

Jadual 2: struktur aturcara

* Langkah pengenalpastian dalam pengaturcaran

Merangka pembolehubah

1. Untuk melancarkan perjalanan aturcara
2. Rujukan untuk lokasi memori

Melukis carta alir

1. Mempersembahkan algoritma dalam bentuk grafik (simbol grafik)

Menulis program

1. Ditulis dalam bentuk yang mudah difahami tetapi jelas, boleh dibaca dan mengikut ciri-ciri aturcara.
2. Dalam susun atur yang baik dan ringkas

Menguji & melaksanakan program

1. Mesti dikompil dan dilaksanakan oleh editor dan compiler
2. Kenalpasti ralat kesalahan

Mendokumentasi program

1. Perlu dokumen untuk rujukan penyelenggaraan utk masa depan dan proses pengurusan.
2. Mesti mengandungi :
   * 1. Spesifikasi keperluan yang tepat
     2. Input, output, kekangan dan formula yang terperinci untuk masalah yang ingin diselesaikan.
     3. Algoritma dalam bentuk cartalir atau kod pseudo
     4. Aturcara sumber yang lengkap dengan komen
     5. Contoh output program dan pengujian data

Panduan untuk mengguna program

1. **TAHAP – TAHAP BAHASA PENGATURCARAAN**



Rajah 2: Generasi dan Aras Bahasa pengaturcaraan

* Seperti mana-mana komputer dan perisian, bahasa pengaturcaraan juga mengalami proses evolusi. Proses ini menghasilkan **5 generasi bahasa** pengaturcaraan iaitu :
* Generasi pertama **bahasa Mesin**.
* Generasi kedua dikenali sebagai **bahasa Penghimpun**.
* Generasi ketiga dinamakan sebagai **bahasa Peringkat Tinggi/prosedur**
* Generasi keempat dikenali sebagai bahasa Peringkat **Sangat Tinggi/bukan prosedur**
* Generasi keempat dan kelima dikenali **bahasa Lazim / Tabii / Natural.**
* Aras dan generasi bahasa pengaturcaraan boleh diringkaskan seperti gambarajah dibawah :

**Aras dan Generasi Bahasa Pengaturcaraan**

Bahasa Aras Rendah

Bahasa Aras Tinggi

Generasi Pertama

**Bahasa Mesin**

Generasi Kedua **Bahasa Penghimpun**

Generasi Ketiga

**Bahasa Prosedur**

Generasi Keempat

**Bahasa Bukan Prosedur**

Generasi Kelima

**Bahasa Tabii / Lazim**

Rajah 3 : Aras dan generasi bahasa pengaturcaraan

* Bahasa pengaturcaraan dikategorikan sebagai peringkat rendah apabila ia hampir dengan bahasa yang difahami komputer iaitu perwakilan 0 dan 1.
* Bahasa Peringkat Tinggi pula hampir kepada bahasa yang manusia gunakan setiap hari atau menghampiri bahasa Inggeris lazim.

* [Generasi Pertama: Bahasa Mesin (1945)](http://www.blogger.com/blogger.g?blogID=3147656342096712602)
* Bahasa mesin merupakan bahasa asas bagi komputer dengan perwakilan data sebagai 0 dan 1.
* Aturcara bahasa mesin berbeza dari sebuah komputer ke komputer yang lain, iaitu bersandarkan kepada mesin. Ia merupakan satu-satunya bahasa yang difahami oleh komputer.
* Generasi Kedua : Bahasa Himpunan ([Pertengahan 1950an)](http://www.blogger.com/blogger.g?blogID=3147656342096712602)
* Bahasa Penghimpun adalah bahasa peringkat rendah yang membolehkan pengaturacara menggunakan singkatan atau perkataan yang mudah diingati bagi menggantikan nombor.
* Bahasa Penghimpun di kenali sebagai bahasa simbolik yang menggunakan kod mnemonic .
* Pengaturcara dapat menulis arahan secara lebih mudah berbanding bahasa Mesin. Walaupun demikian bahasa ini susah untuk dipelajari. Tambahan pula bahasa ini mempunyai kekangan kerana ia bergantung kepada mesin tertentu. Bentuk bahasa ini akan berbeza untuk arahan kepada mesin yang berlainan. Oleh kerana kekangan tersebut, pakar komputer telah memperkenalkan Penterjemah Bahasa.
* Disebabkan komputer hanya boleh menjalankan arahan dalam bahasa Mesin, Penterjemah diperlukan jika aturcara ditulis dalam bahasa lain. Penterjemah bahasa adalah sejenis perisian sistem yang akan menterjemahkan aturcara yang ditulis dalam bahasa kedua, ketiga atau lebih tinggi kepada bahasa mesin.
* Contohnya A mewakili ADD manakala S mewakili SUM, huruf MP digunakan bagi menggantikan arahan MULTIPLY dan STO mewakili STORE.
* Terdapat 3 jenis penterjemah bahasa iaitu Penghimpun, Pengkompil dan pentakfsir.

* [Generasi Ketiga: Bahasa Peringkat Tinggi / Bahasa Prosedur (Awal 1960an)](http://www.blogger.com/blogger.g?blogID=3147656342096712602)
* Bahasa peringkat tinggi atau bahasa generasi ketiga adalah bahasa seakan bahasa Inggeris. Contohnya COBOL yang digunakan dalam aplikasi bisnes, Fortran, Basic, Pascal, C dan Ada.
* Bahasa ini tidak bersandar kepada mesin, iaitu ia boleh digunakan pada lebih daripada satu jenis komputer.
* Bahasa generasi ketiga yang diterangkan di atas juga dikenali sebagai bahasa prosedur/bertatacara. Ia bertatacara kerana satu siri tatacara dibentuk terlebih dahulu sebelum menulis aturcara. Tatacara ini merupakan siri arahan yang mesti diikuti oleh pengaturcara apabila menulis arahan dalam bahasa tersebut.
* Penterjemah bahasa peringkat tinggi bergantung kepada bahasa yang digunakan samaada suatu pengkompil atau pentafsir.

|  |
| --- |
| **Generasi Pertama - Bahasa Mesin**    11110010     01110011     1101     001000010000     0111     000000101011  11110010     01110011     1101     001000011000     0111     000000101111  11111100     01010010     1101     001000010010     1101     001000011101 |
| **Generasi Kedua - Bahasa Assembly**    PACK     210(8,13) , 02B(4,7)  PACK     218(8,13) , 02F(4,7)  MP        212(6,13) , 21D(3,13)  SRP      213(5,13) , 03E(0,5)  UNPK    050(5,7) , 214(4,13)  OI        054(7) , X'FO' |
| **Generasi Ketiga - COBOL (cth)**    MULTIPLY HOURS-WORKED BY DAY-RATE GIVING GROSS-PAY ROUNDED |

Rajah 4: Perbezaan Antara 3 Generasi Bahasa Pengaturcaraan Terawal

* [Generasi Keempat: Bahasa peringkat Tinggi](http://www.blogger.com/blogger.g?blogID=3147656342096712602)
* Bahasa peringkat tertinggi atau bahasa generasi keempat dikenali juga dengan nama 4GL. Ia lebih berorientasikan pengguna dan membenarkan pengaturcara untuk membangunkan aturcara dengan sedikit arahan berbanding dengan bahasa generasi ketiga.
* 4GL juga dipanggil bahasa tak berprosedur kerana pengaturcara boleh menulis aturcara lengkap yang hanya perlu memberitahu komputer apa yang hendak dilakukan sahaja, dan tidak perlu menyatakan semua prosedur bagi melakukannya.
* Bahasa peringkat ini mempunyai tujuan-tujuan khas kepada pengguna seperti:

1. **Penjana laporan** - Penjana laporan dipanggil juga penulis laporan. Ia adalah aturcara bagi pengguna akhir yang digunakan untuk menghasilkan laporan.
2. **Bahasa pertanyaan** - Bahasa pertanyaan adalah bahasa yang mudak digunakan untuk mendapatkan semula data daripada sistem pengurusan pangkalan data.
3. **Penjana Aplikasi** - Penjana applikasi adalah alat pengaturcaraan yang membenarkan seseorang memberi data secara terperinci penerangan data yang diperlukan untuk diproses. Perisian kemudia menjana kod yang diperlukan untuk menghasilkan aturcara bagi melakukan kerja-kerjanya. Kebaikannya, pengarucara tidak perlu menentukan bagaimana sepatutnya diproses. Contoh, sistem pengurusan pangkalan data berhubungan Nomad dan Focus.

* [Generasi Kelima - Bahasa Tabie](http://www.blogger.com/blogger.g?blogID=3147656342096712602) / Bahasa Lazim
* Bahasa tabie adalah sebahagian daripada bidang kepintaran buatan. Kepintaraan buatan adalah sekumpulan teknologi yang berkaitan, yang cuba untuk membangunkan mesin bagi menjejak kualiti seperti manusia.
* Bahasa tabie atau bahasa genarasi kelima terdiri daripada dua jenis:

1. Bahasa manusia - English, Spanish, dan lain-lain.
2. Bahasa pengaturcaraan yang menggunakan bahasa manusia untuk membolehkan manusia berhubung dengan komputer.

|  |  |  |
| --- | --- | --- |
| Tahun | Bahasa  Pengaturcaraan | Penerangan |
| 1970 | Pascal | Bahasa mudah |
| 1970 | Smalltalk | Bahasa pengaturcaraan orientasi objek yg pertama |
| 1971 | FORTH | Untuk pemprosesan data dengan pantas |
| 1972 | PROLOG | Pengaturcaraan logik utk aplikasi kecerdasan buatan (AI) |
| 1972 | C | Untuk penggunaan saintifik |
| 1979 | Ada | Satu permulaanpiawaian baru dlm pengaturcaraan |
| 1981 | Modula-2 | Bahasa seakan-akan Pascal yg digunakan bagi menulis perisian sistem |
| 1982 | dBase | Bahasa pengaturcaraan pangkalan data |
| 1984 | Turbo Pascal | Versi Pascal bagi persekitaran pengaturcaraan orientasi objek |
| 1987 | HyperCard | Bahasa hiperteks Macintosh |

Jadual 3 : Sebahagian Evolusi Bahasa Pengaturcaraan

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

1. Terangkan maksud **Bahasa pengaturcaraan**
2. Nyatakan Langkah pengenalpastian dalam pengaturcaran
3. Berikan Contoh bahasa pengaturcaraan.
4. Senaraikan 5 Generasi Bahasa Pengaturcaraan
5. Bahasa peringkat tinggi merupakan generasi peringkat berapa……
6. **Penjana laporan** bertujuan sebagai……………………………..

**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