

PERSEKITARAN KOD ARAHAN

STANDARD PEMBELAJARAN

- 3.1.1 Menggunakan jenis data (integer, boolean, double, char dan string) dalam kod segmen.
- 3.1.2 Membezakan antara pemboleh ubah dan pemalar dalam kod segmen.
- 3.1.3 Menghasilkan kod segmen menggunakan:
 - (i) fungsi input dan output.
 - (ii) operator perbandingan.
 - (iii) operator logik
- 3.1.4 Mengesan dan memperbaiki ralat pada kod segmen yang dihasilkan dalam penyelesaian masalah.
- 3.1.5 Menyelesaikan masalah yang melibatkan gabungan jenis data, pemboleh ubah, pemalar dan operator dalam kod segmen.

STANDARD PRESTASI

- 1 Mengenal pasti jenis data, pemalar atau operator.
- 2 Menjelaskan jenis data, pemalar dan operator
- 3 Menulis kod segmen yang melibatkan penggunaan pelbagai jenis data, pemboleh ubah, pemalar dan operator.
- 4 Menunjukkan keupayaan mengesan ralat dalam kod segmen.
- 5 Menunjukkan keupayaan memperbaiki ralat dalam kod segmen.
- 6 Menggabungkan jenis data, pemboleh ubah, pemalar dan operator dalam kod segmen untuk menyelesaikan masalah secara sistematik dan kreatif.

3.1**Persekitaran Kod Arah**

1. Kod arahan ialah satu arahan yang ditulis dalam bahasa pengaturcaraan untuk mengarahkan komputer melaksanakan sesuatu tugas.
2. Contoh bahasa pengaturcaraan komputer adalah seperti;
 - i. Scratch
 - ii. HTML
 - iii. Python
3. Persekitaran kod arahan merujuk kepada persediaan terhadap peralatan dan perisian yang diperlukan untuk menulis kod arahan.
4. Contoh keperluan kepada kod arahan adalah seperti;
 - i. Perisian aplikasi bahasa pengaturcaraan
 - ii. Perkakasan komputer
 - iii. Sambungan internet
 - iv. Pelayar web
5. Bahasa pengaturcaraan perlu diterjemah supaya komputer faham apa yang perlu dilakukan. Penterjemah terdiri daripada;
 - i. Penghimpun (assembly)
 - ii. Pengkompil (compiler)
 - iii. Pentafsir (interpreter)

3.1.1**Jenis Data Dalam Segmen Kod**

1. Setiap bahasa pengaturcaraan mempunyai satu senarai jenis data yang digunakan semasa pengisytiharan suatu pemboleh ubah.
2. Pemboleh ubah adalah ruang simpanan data supaya komputer dapat memproses data yang dimasukkan oleh pengguna.
3. Jenis data yang digunakan sebagai pemboleh ubah adalah;
 - i. Integer
 - ii. Double
 - iii. Boolean
 - iv. Char
 - v. String
4. **INTEGER** ialah nombor yang tidak mengandungi bahagian pecahan dan tidak mempunyai titik perpuluhan.
 - i. Integer terdiri daripada nombor negatif, sifar dan nombor positif.
 - ii. Pengisytiharan pemboleh ubah integer (int) dalam bahasa pengaturcaraan Python adalah seperti `>>> a = int nombor`

5. **DOUBLE** adalah sejenis data yang mengandungi bahagian pecahan atau nombor perpuluhan.
 - i. Double boleh terdiri dari nombor negatif dan nombor positif dalam bentuk nombor perpuluhan atau pecahan
 - ii. Bahasa pengaturcaraan Python tidak mempunyai jenis data double tetapi diganti dengan jenis data float.
 - iii. Apabila jenis data diisytiharkan sebagai double, memori sebanyak 64 bit akan diperuntukkan.

6. **CHAR** ialah jenis data yang digunakan untuk mewakili aksara (character) yang terdiri daripada huruf abjad, angka dan simbol.
7. Bahasa pengaturcaraan Python tidak mempunyai jenis data Char tetapi diganti dengan String.
8. **STRING** ialah satu perkataan yang dibina daripada aksara. String terdiri daripada satu aksara yang berjujukan yang membentuk suatu rentetan. Contoh :

K + O + M + P + U + T + E + R = KOMPUTER

9. Jadual dibawah menunjukkan fungsi yang memanipulasikan nilai String dan tujuannya.

Fungsi	Tujuan
Tanda [] atau [:]	Mencapai satu aksara atau lebih dalam string
len ()	Menghitung bilangan aksara dalam string
Simbol +	Menggabungkan dua string tanpa ruang kosong
Simbol ,	Menggabungkan dua string dengan satu ruang kosong
Simbol *	Mengulang string dengan bilangan yang dinyatakan
Join ()	Menambah aksara diantara setaip aksara

10. Contoh kod aturacara yang menggunakan jenis data string

```

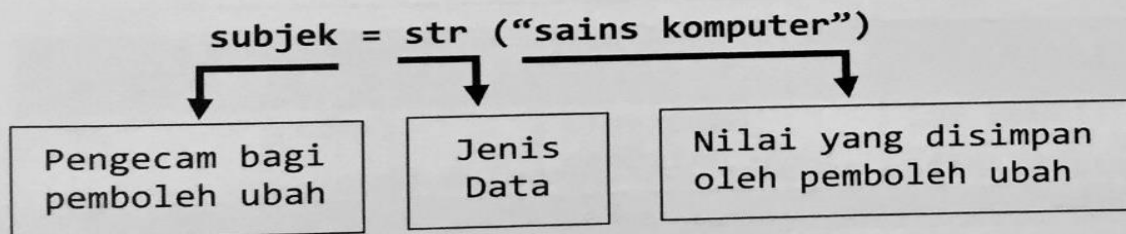
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.:
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a = str ("Helo")
>>> b = str ("Dunia")
>>> c = str ("!")
>>> print (a , b + c)
Helo Dunia!
>>> print (a * 4)
HeloHeloHeloHelo
>>>
    
```

11. **BOOLEAN** mempunyai dua nilai iaitu benar dan palsu yang digelar nilai logik.
- Nilai logik boolean diwakili oleh 0 untuk palsu, 1 untuk benar.
 - Nilai logik digunakan untuk pernyataan bersyarat dalam struktur kawalan pilihan dan struktur kawalan ulangan
 - Pernyataan bersyarat ini disebut ungkapan boolean
 - Ungkapan booleh menggunakan operator perbandingan iaitu seperti >, <, >=, <=

3.1.2

Perbezaan antara Pbolehkan Ubah dan Pemalar

- Pbolehkan ubah ialah storan ingatan dalam komputer yang digunakan oleh atur cara untuk mewakili data yang dimasukkan oleh pengguna.
- Pbolehkan ubah dapat digunakan oleh pengaturcara untuk memanipulasikan data dengan menggunakan kod arahan yang dibangunkan.
- Pbolehkan ubah yang diisytiharkan akan diberi satu label yang disebut pengecam yang sesuai dan disertakan dengan nilai.
- Pengecam suatu pemboleh ubah perlu dilabelkan dengan nama yang sesuai bagi membolehkan pengecam dapat dirujuk semula dalam kod arahan.
- Semasa pengisytiharan pemboleh ubah, ruang ingatan dalam komputer ditempah berdasarkan jenis data yang diberi kepada pemboleh ubah.
- Contoh pengisytiharan pemboleh ubah;



- Syarat untuk mengisytiharkan pemboleh ubah
 - Nama pemboleh ubah hanya boleh mengandungi aksara dan digit
 - Nama pemboleh ubah harus menggambarkan nilai yang dipegang
 - Nama pemboleh ubah tidak boleh bermula dengan nombor
 - Nama pemboleh ubah tidak boleh ada ruang kosong

8. Pemalar bertindak seakan-akan pemboleh ubah dan boleh menyimpan sesuatu nilai dengan tetap dalam ingatan komputer.
9. Perbezaan utama ialah nilai sesuatu pemalar tidak akan berubah menjadi nilai yang lain semasa pelaksanaan atur cara.
10. Contoh penggunaan pemalar dalam kod aturcara;

```

// 1.) util java untuk input user.
import java.util.Scanner;

// 2.) Nama class = nama fail
public class LuasBulatan{

    public static void main(String[] args) {

        Scanner data = new Scanner(System.in);
        // 3.) Input jejari daripada pengguna
        System.out.println("Masukkan panjang jejari : ");
        // 4.) Isytihar pembolehubah jejari, pie, dan LuasBulatan.
        double jejari = data.nextDouble();
        double pie, LuasBulatan ;
        pie = 3.142;
        // 5.) Proses arithmetik pengiraan luas sebuah bulatan.
        LuasBulatan = pie * jejari * jejari;

        // 6.) Papar/cetak Luas bulatan
        System.out.println("Luas Bulatan Adalah " + LuasBulatan);
    }
}
    
```

PEMALAR

11. Perbezaan antara pemalar dan pemboleh ubah

PEMBOLEH UBAH	CIRI-CIRI	PEMALAR
Nilai yang boleh berubah	Perubahan nilai	Nilai yang tidak boleh berubah
Perlu ingat label dan nilai yang digunakan	Nilai yang diingat	Tidak perlu untuk ingat label dan nilai
Tiada standard	Standard penggunaan nilai	Nilai yang standard

3.1.3 (i)

Menghasilkan Segmen Kod Menggunakan Fungsi Input Dan Fungsi Output

1. Dalam bahasa pengaturcaraan, fungsi input dan fungsi output merupakan kaedah interaktif yang digunakan dalam interaksi manusia dan komputer.
2. Fungsi input dan output membolehkan mesej yang hendak disampaikan dan diterima dapat difahami oleh pengguna dan komputer.
3. Interaksi manusia dan komputer melibatkan cara manusia berkomunikasi dengan komputer dengan ciri-ciri perantaraan dalam aspek yang lebih luas seperti;
 - i. Faktor komputer (sistem pengoperasian, grafik komputer, dan persekitaran pembangunan)
 - ii. Faktor manusia (komunikasi, sains sosial, bahasa)
4. Perantaraan merujuk kepada medium seperti antara muka suatu atur cara, cara data dimasukkan dan reka bentuk grafik yang mesra pengguna supaya komunikasi antara manusia dengan komputer dapat berlaku dengan betul dan tepat.

A. FUNGSI INPUT

1. Fungsi input digunakan untuk mendapatkan data daripada pengguna melalui segmen kod yang dibina.
2. Data boleh dimasukkan melalui papan kekunci tertakluk kepada segmen kod yang dibina.
3. Sintaks bagi setiap bahasa pengaturcaraan adalah berbeza. Contoh sintaks beberapa bahasa pengaturcaraan untuk mendapatkan input dari pengguna.

Bahasa aturcara	Sintaks umum	Segmen kod	contoh
C++	cin>>	cin>>nama_ pemboleh_ubah	int umur; cin>>umur
C	scanf()	scanf(nama_ pembolehubah);	Int umur; scanf(umur);
Java	System.in		Scanner umur=new Scanner(system.in);
Python	Input ()	Input (nama_ pemboleh_ubah)	>>>umur =input ("umur")

B. FUNGSI OUTPUT

1. Fungsi output merujuk kepada maklumat yang dipaparkan pada skrin komputer yang dihasilkan melalui segmen kod atau data yang dimasukkan oleh pengguna.
2. Fungsi output akan diaktifkan oleh sintaks umum mengikut bahasa pengaturcaraan yang digunakan.
3. Monitor adalah peranti output yang digunakan untuk memaparkan maklumat atau mesej yang dihasilkan melalui proses oleh segmen kod dalam komputer.
4. Contoh sintak fungsi output dari pelbagai bahasa pengaturcaraan

Bahasa aturcara	Sintaks umum	Segmen kod	contoh
C++	cout<>	Cout<<nama_ pemboleh_ubah	Cout<<umur;
C	printf()	printf(nama_ pembolehubah);	print(umur);
Java	System.out	System.out.println	System.out.println (umur);
Python	print ()	print (nama_ pemboleh_ubah)	print("umur")

3.1.3 (ii)

Operator Perbandingan

1. Operator perbandingan digunakan untuk membandingkan nilai di sebelah kiri operator dengan nilai di sebelah kanan operator.
2. Hasil kepada perbandingan ini adalah nilai logik benar atau palsu.
3. Operator perbandingan digunakan dalam ungkapan boolean untuk pernyataan bersyarat bagi struktur kawalan pilihan dan struktur kawalan ulangan.
4. Jadual dibawah menunjukkan pelbagai operator perbandingan

Operator	Maksud
==	Sama dengan. Contoh a==b
!=	Tidak sama dengan. Contoh a!=b
>	Lebih besar daripada. Contoh a>b
<	Lebih kecil daripada. Contoh a=	Lebih besar atau sama dengan. Contoh a>=b
<=	Lebih kecil atau sama dengan. Contoh a<=b

3.1.3 (iii)

Operator Logiik

1. Operator logik digunakan dalam suatu ungkapan algebra ringkas untuk mendapatkan nilai boolean, iaitu benar atau palsu.
2. Jadual dibawah menunjukkan operator logik

Operator logik	Penerangan
AND (DAN)	Keadaan menjadi benar jika kedua-dua nilai operan adalah benar
OR (ATAU)	Keadaan menjadi benar jika salah satu nilai operan adalah benar
NOT (TAK)	Menyongsangkan seluruh keadaan logik operan

3. Operator tambahan untuk operasi matematik dalam kod aturcara

Operator	Nama	Penerangan
%	Modulus	Operan kiri dibahagikan dengan operan kanan. Baki pembahagian dipaparkan
**	Eksponen	Operan kiri dikuasakan mengikut nilai operan kanan
//	Floor division	Operan kiri dibahagikan dengan operan kanan. Hasil pembahagian tanpa nilai baki dipaparkan

3.1.3

Mengesan dan Mbaiki Ralat

1. Ralat ditakrifkan sebagai kesilapan atau kesalahan dalam proses pembangunan aturcara.
2. Ralat sintaks lazimnya berlaku disebabkan oleh kesilapan dalam penggunaan bahasa pengaturcaraan.
3. Antara sebab ralat sintaks terjadi adalah disebabkan oleh;
 - i. Penggunaan sintaks yang tidak betul
 - ii. Ejaan sintaks tidak betul
 - iii. Penggunaan simbol dan tatatanda yang salah
 - iv. Meletakkan sintaks pada tempat yang salah
 - v. Penggunaan inden yang salah
4. Teknik langkah demi langkah boleh digunakan untuk mengesan dan membaiki ralat.

5. Ralat masa larian akan menyebabkan pelaksanaan segmen kod terhenti secara tiba-tiba tanpa diduga.
6. Ralat masa larian adalah disebabkan oleh;
 - i. Pembahagian dengan sifar
 - ii. Melaksanakan operasi untuk dua pemboleh ubah yang berlainan jenis data
 - iii. Menggunakan pemboleh ubah, fungsi, modul atau objek yang belum ditakrifkan
 - iv. Mengakses nilai pustaka atau objek atribut yang tidak wujud
 - v. Mengakses fail yang tida wujud.
7. Ralat logik tidak mengganggu pelaksanaan segmen kod. Ralat logik disebabkan oleh kecuaiian pengaturcara.
8. Ralat logik disebabkan oleh;
 - i. Pemanggilan pemboleh ubah yang salah
 - ii. Penggunaan nombor bulat atau nombor perpuluhan yang tidak betul atau bukan di tempat yang sepatutnya
 - iii. Penggunaan operator pengendali yang salah
 - iv. Kesilapan dalam ungkapan boolean
 - v. Penggunaan ruang jarak yang salah