

# SPM2102

# BAHASA PENGATURCARAAN 1

Tajuk:

**ATURCARA BERSTRUKTUR DALAM  
BAHASA PENGATURCARAAN C++**

Disediakan oleh:

**NORASYKIN MOHD ZAID  
NORAH MD NOOR**



# Struktur Pilihan *if..end if*

*if* ( syarat ) pernyataan;

- Syarat biasanya ialah satu perbandingan, tetapi boleh jadi juga hasil dari sesuatu operasi matematik atau suatu fungsi.
- Pernyataan ialah sebarang pernyataan atau set pernyataan-pernyataan C++.

Jika kita inginkan satu set pernyataan dilaksanakan, pernyataan-pernyataan ini perlulah di letakkan di dalam satu blok seperti ini:

```
if ( syarat )  
    {  
        pernyataan_1;  
        pernyataan_2;  
        .  
        .  
        pernyataan_n;  
    }
```

**Sintaks:**

***If (condition)  
statement;***

**Atau**

***If (condition)  
{  
 statement 1;  
 statement 2;  
 ... statement n;  
}***

# Struktur Pilihan *if..end if*

```
#include<iostream.h>

void main(){

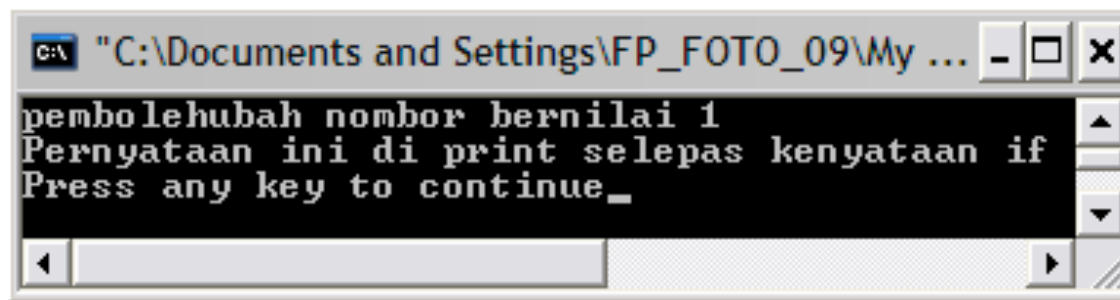
int nombor = 1;

if (nombor == 1)
    cout<<"pembolehkan nombor bernilai 1 "<<endl;

cout<<"Pernyataan ini di print selepas kenyataan if "<<endl;

}
```

Kenyataan if



Contoh output

# Struktur dwipilihan *if – then – else*

```
if ( syarat )  
    pernyataan1;  
else  
    pernyataan2;
```

- ***Jika syarat adalah benar, maka pernyataan1 akan dilaksanakan, dan jika tidak pernyataan2 akan dilaksanakan.***

**Sintaks:**

```
If (condition)  
    statement a;  
else  
    statement b;
```

**Atau**

```
If (condition)  
    {  
        statement a1;  
        statement a2;  
        ... statement an;  
    }  
else  
    {  
        statement b1;  
        statement b2;  
        ... statement bn;  
    }
```

# Struktur dwipilihan *if - then - else*

Contohnya:

```
if ( x == 0 )  
    cout<<"\n X adalah sifar";  
else  
    cout<<"\n X bukan sifar";
```

# Struktur dwipilihan *if - then - else*

```
#include<iostream.h>

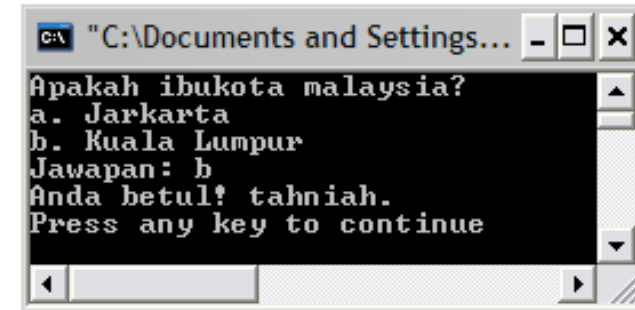
void main(){

char jawapan;

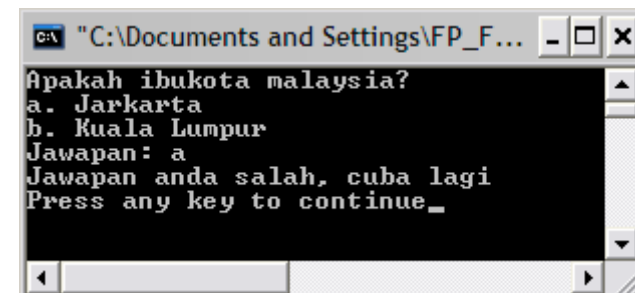
        cout<<"Apakah ibukota malaysia? "<<endl;
        cout<<"a. Jarkarta "<<endl;
        cout<<"b. Kuala Lumpur "<<endl;
        cout<<"Jawapan: ";
        cin>>jawapan;

        if (jawapan == 'b')
                cout<<"Anda betul! tahniah. "<<endl;
        else
                cout<<"Jawapan anda salah, cuba lagi "<<endl;
}
```

Kenyataan  
if



Contoh output jawapan betul



Contoh output jawapan salah

# Tambahan...

if(  $a \neq 0$  ) adalah setara dengan if ( a )

if (1) -- akan sentiasa memberikan jawapan "benar".



# Struktur multipilihan *if-else-if* atau nested *if-else*

Pernyataan ini membolehkan lebih daripada 2 pilihan seperti berikut:

```
if ( x == 0 )
    cout<<"\n X adalah sifar";
else if ( x > 0 )
    cout<<"\n X adalah positif";
else
    cout<<"\n X adalah negatif";
```

**Sintaks:**

```
If (syarat 1)
    statement a;
else if (syarat 2)
    statement b;
else if (syarat 3)
    statement c;
```

...

**Atau**

```
If (syarat 1)
{
    statement a1;
    ... statement an;
}
else if (syarat 2)
{
    statement b1;
    ... statement bn;
}
```

...

```

1 // Fig. 1.14: fig01 14.cpp
2 // Using if statements, relational
3 // operators, and equality operators
4
5
6
7 #include <iostream.h>
8
9
10 int main()
11 {
12     int num1, num2;
13
14     cout << "Enter two integers, and I will tell you\n"
15         << "the relationships they satisfy: ";
16     cin >> num1 >> num2; // read
17
18     if ( num1 == num2 )
19         cout << num1 << " is equal to " << num2 << endl;
20
21     else if ( num1 != num2 )
22         cout << num1 << " is not equal to " << num2 << endl;
23
24     else if ( num1 < num2 )
25         cout << num1 << " is less than " << num2 << endl;
26
27     else if ( num1 > num2 )
28         cout << num1 << " is greater than " << num2 << endl;
29
30     else if ( num1 <= num2 )
31         cout << num1 << " is less than or equal to "
32             << num2 << endl;
33

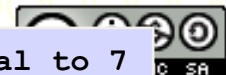
```

Enter two integers, and I will tell you  
the relationships they satisfy: 3 7

The **if** statements test the truth of the condition. If it is **true**, the body is executed. If not, body is skipped.

3 is less than 7  
To include multiple statements in a body, delineate them with braces { }.

3 is less than or equal to 7



# Struktur multipilihan *if-else-if*

```
if ( markah < 40)
    gred = "F";
else if ( markah < 50)
    gred = "E";
else if (markah < 60)
    gred = "D";
else if ( markah < 70)
    gred = "C";
else if (markah < 80)
    gred = "B";
else
    gred = "A";
end if
```

# Struktur multipilihan switch dan case

- digunakan untuk menggantikan pernyataan-pernyataan if-else if
- akan lebih berstruktur dan memudahkan pengaturcaraan

# Struktur multipilihan switch dan case

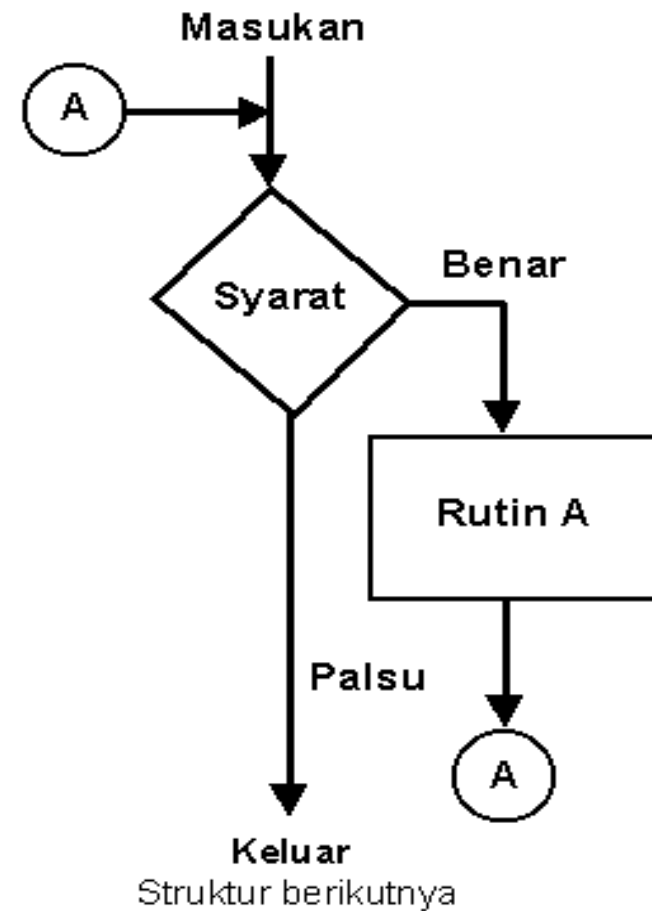
```
switch ( pilihan )  
{  
    case 1 : cout<<"Anda pilih nombor 1";  
             break;  
    case 2 : cout<<"Anda pilih nombor 2";  
             break;  
    case 3 : cout<<"Anda pilih nombor 3";  
             break;  
    default : cout<<"Anda pilih nombor selain daripada  
1,2 atau 3";  
}
```

# Kenyataan lompatan-Break & Continue

- Melakukan perubahan pada aliran kawalan (flow of control)
  - Break
    - Kenyataan break dalam sesuatu aturcara akan memaksa sesuatu proses pengulangan dihentikan serta merta. Baris aturcara berikutnya selepas gelung akan dilaksanakan (Immediate (premature) exit from loop)
  - Continue
    - Kenyataan continue pula akan memaksa supaya pengulangan dibuat lebih awal dengan mengabaikan struktur kawalan biasa dalam sesuatu gelung (Skip the remaining statements in loop body)

# Struktur Gelung

- Terdapat 3 jenis sintak yang digunakan dalam bahasa C++ untuk struktur pengulangan
  - While
  - Do-While
  - For



# Struktur pengulangan *while*

***Bentuk penggunaan pernyataan -  
while ( syarat ) pernyataan;***

```
i = 0;  
while ( i < 10 )  
{  
    cout<<i;  
    i++;  
}
```

Dalam contoh ini, satu senarai nombor dari sifar hingga ke sembilan dipaparkan pada paparan.



# Struktur pengulangan *while*

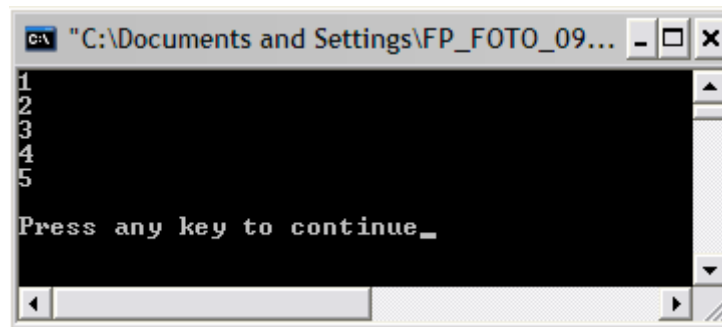
```
#include <iostream.h>
int x = 12;
int x2;
main()
{
    while ( x <= 21 )
    {
        x2 = x * x;
        cout<<"\n"<<x<<" kuasa dua = " <<x2;
        x++;
    }
}
```

Dalam contoh ini, satu senarai output jadual kuasa dua bagi nilai 12 hingga 21 dipaparkan pada paparan.

# Struktur pengulangan *while*

```
#include<iostream.h>
void main(){
  int kira = 1;
  while (kira <= 5)
  {
    cout<<kira<<endl;
    kira = kira + 1;
  }
  cout<<endl;
}
```

While loop



Contoh output

# Struktur gelung *do-while*

***Bentuk penggunaan pernyataan -  
do pernyataan;  
while ( syarat );***

```
i = 0;  
do  
{  
    cout<<i;  
    i++;  
} while ( i < 10 );
```

Apakah bezanya berbanding gelung while?

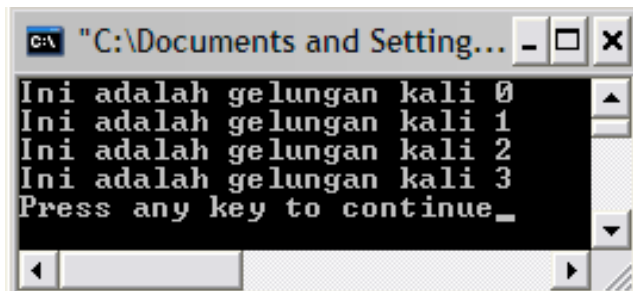
```
#include <stdio.h>
int x = 12;
int x2;
main()
{
    puts(" Sebuah Jadual Kuasa Dua ");
    do
    {
        x2 = x * x;
        cout<<"\n"<<x<<" kuasa dua = " <<x2;
        x += 2;
    } while ( x <= 21 );
}
```

Dalam contoh ini, satu senarai output jadual kuasa dua bagi nilai 12 hingga 21 dipaparkan pada paparan.

# Struktur pengulangan *do-while*

```
#include<iostream.h>
void main(){
    char jawapan;
    int bilangan = 0;
    do{
        cout<<"Ini adalah gelungan kali "<<bilangan<<endl;
        ++bilangan;
    }while(bilangan < 4);
}
```

Do-While loop



```
C:\Documents and Setting...
Ini adalah gelungan kali 0
Ini adalah gelungan kali 1
Ini adalah gelungan kali 2
Ini adalah gelungan kali 3
Press any key to continue_
```

Contoh output

# Struktur gelung *for*

*digunakan untuk melakukan gelungan dengan bilangan tertentu*

*Bentuk penggunaan pernyataan -*

*for ( pememulaan; ; perubahan syarat)  
pernyataan;*

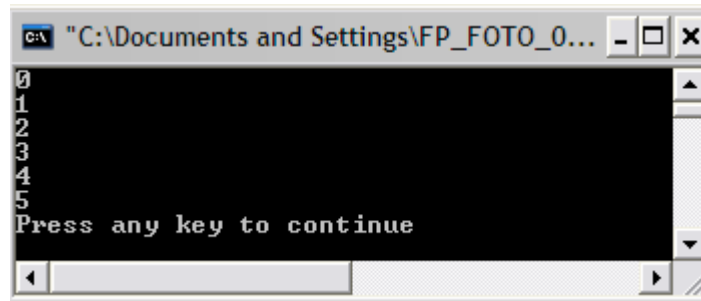
Contoh 1

```
for ( i = 0; i < 10; i++)  
    cout<<i;
```

# Struktur pengulangan *for*

```
#include<iostream.h>
void main(){
for (int nombor = 0; nombor < 6; ++nombor){
    cout<<nombor<<endl;
}
}
```

For loop



```
C:\Documents and Settings\FP_FOTO_0...
0
1
2
3
4
5
Press any key to continue
```

Contoh output

# Rujukan

- Jabatan Multimedia Pendidikan (2004), *Pengenalan kepada Bahasa Pengaturcaraan dalam Konteks Bahasa C++* , Faculti Pendidikan, UTM ( unpublished )
- Abdullah Zawawi Talib, Ahamad Tajudin Khader, Maziani Sabudin, Wahidah Husain (2002), *Prinsip-Prinsip Pengaturcaraan Menggunakan C++ : Edisi kedua*, Pearson Malaysia, ISBN 983-9236-87-3
- Bakar, M. A et al (2002). Pengaturcaraan C. Malaysia. Prentice Hall
- Deitel, P. J., & Deitel, H. M. (2005). *C++ How to Program (5th Edition)*, Prentice Hall