

PEMROGRAMAN BERORIENTASI OBJEK

Percabangan

PERCABANGAN

- Salah satu permasalahan yang akan dijumpai dalam pembuatan program adalah ***percabangan***.
- ***Percabangan*** yang dimaksud adalah suatu *pemilihan statemen* yang akan dieksekusi dimana pemilihan tersebut didasarkan atas kondisi tertentu.

PERCABANGAN

- Didalam C++, terdapat dua jenis struktur (*blok program*) yang digunakan untuk mengimplementasikan suatu percabangan, yaitu dengan menggunakan struktur **if** dan struktur **switch**.

PERCABANGAN

- **Statemen - statemen** yang terdapat dalam sebuah **blok percabangan** akan dieksekusi hanya jika kondisi yang didefinisikan terpenuhi (**bernilai benar**)
- Artinya jika kondisi tidak terpenuhi (**bernilai salah**), maka **statemen - statemen** tersebut juga tidak ikut dieksekusi atau dengan kata lain akan diabaikan oleh compiler.

PERCABANGAN

- **Ilustrasi Pemahaman :**
“Jika saya lulus ujian maka saya akan diberikan hadiah oleh ayah”
- Pada kalimat diatas yang merupakan kondisi adalah lulus ujian. Pada kasus ini, hadiah hanya akan diberi jika saya lulus ujian.
- Sebaliknya , jika tidak lulus maka hadiah pun tidak akan diberikan.

STUKTUR IF SATU KONDISI

- Struktur ini merupakan struktur yang paling sederhana karena hanya melibatkan **satu buah ekspresi** yang akan diperiksa.
- Bentuk umum dari struktur percabangan yang memiliki satu kondisi adalah sebagai berikut :

STUKTUR IF SATU KONDISI

```
//jika terdapat lebih dari satu statement  
If(kondisi) {  
    Statement1;  
    Statement2;  
    ....  
}
```

```
//Jika hanya terdapat satu statement  
If(kondisi) Statement;
```

STUKTUR IF SATU KONDISI

```
#include <iostream>

using namespace std;

int main() {
    int nilai;

    // Memberikan informasi agar user memasukkan
    // sebuah bilangan bulat
    cout<<"Masukkan sebuah bilangan bulat : ";

    // Membaca nilai yang dimasukkan dari keyboard
    // dan disimpan ke variabel nilai
    cin>>nilai;

    // Menampilkan sebuah teks
    // jika nilai yang tersimpan
    // lebih besar dari nol
    if (nilai > 0)
        cout<<"Nilai yang Anda masukkan "
            <<"adalah bilangan positif";

    return 0;
}
```


STUKTUR IF SATU KONDISI

- Hasil dari program diatas bersifat ***dinamis***, artinya setiap proses eksekusi program akan memberikan hasil yang sama, karena hasilnya tentu tergantung kepada nilai yang dimasukan oleh user.

STUKTUR IF SATU KONDISI

- Kesimpulannya : program diatas akan menampilkan teks "Nilai yang anda masukan adalah bilangan positif" jika user memasukan nilai yang lebih besar dari nol.
- Dapat juga digunakan **operator** `||` dan `&&` dalam menentukan sebuah **ekspresi**

STUKTUR IF SATU KONDISI

```
#include <iostream>
using namespace std;
int main() {
    int bilangan;
    char huruf;
    // Meminta user untuk memasukkan sebuah bilangan
    cout<<"Masukkan sebuah bilangan bulat : ";
    cin>>bilangan;

    if ((bilangan > 0) && (bilangan < 10))
        cout<<bilangan
            <<" lebih besar dari nol"
            <<" dan lebih kecil dari sepuluh";
    // Meminta user untuk memasukkan sebuah huruf
    cout<<endl<<endl;
    cout<<"Masukkan sebuah huruf : ";
    cin>>huruf;
    if ((huruf == 'A') || (huruf == 'a') ||
        (huruf == 'I') || (huruf == 'i') ||
        (huruf == 'U') || (huruf == 'u') ||
        (huruf == 'E') || (huruf == 'e') ||
        (huruf == 'O') || (huruf == 'o')) {
        cout<<huruf
            <<" adalah salah satu huruf vokal";
    }

    return 0;
}
```

STUKTUR IF DUA KONDISI

- Struktur ini sedikit lebih kompleks bila dibandingkan dengan ***satu kondisi***.
- Sebenarnya konsepnya sangat sederhana, yaitu terdapat sebuah statement khusus yang berguna untuk mengatasi kejadian apabila kondisi yang didefinisikan tersebut tidak terpenuhi (***bernilai salah***).

STUKTUR IF DUA KONDISI

- Bentuk umum dari struktur percabangan yang memiliki dua kondisi adalah sebagai berikut :

```
If(kondisi) {  
    Statement_jika_kondisi_terpenuhi;  
}else{  
    Statement_jika_kondisi_tidak_terpenuhi;  
}
```

STUKTUR IF DUA KONDISI

```
#include <iostream>

using namespace std;

int main() {
    int bilangan;

    cout<<"Masukkan bilangan bulat "
         <<"yang akan diperiksa : ";
    cin>>bilangan;

    // Melakukan pengecekan bilangan apakah habis
    // dibagi dua atau tidak
    if (bilangan % 2 == 0) {
        cout<<bilangan<<" adalah bilangan genap";
    } else {
        cout<<bilangan<<" adalah bilangan ganjil";
    }

    return 0;
}
```

STUKTUR IF DUA KONDISI

- User diminta untuk memasukkan bilangan dan nilai tersebut akan disimpan dalam variabel `bilangan`.
- Program akan melakukan pemeriksaan terhadap nilai yang terdapat dalam variabel tersebut melalui ekspresi (`bilangan % 2 == 0`) maksudnya adalah melakukan pemeriksaan apakah nilai dari variabel tersebut habis dibagi dua (`sisanya=0`) atau tidak jika ya maka bilangan tersebut akan termasuk kedalam bilangan genap

STUKTUR IF DUA KONDISI

- Tetapi apabila tidak dapat habis dibagi dua maka program akan memilih statement yang terdapat pada bagian `else` yaitu bilangan ganjil.

STUKTUR IF DUA KONDISI

```
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    double X, Y, Z;
    // Meminta user untuk memasukkan bilangan
    cout<<"Masukkan bilangan yang akan dibagi : ";
    cin>>X;
    cout<<"Masukkan bilangan pembagi          : ";
    cin>>Y;
    // Menghindari terjadinya pembagian dengan 0
    if (Y == 0) {
        cout<<"Kesalahan: "
            <<"Bilangan pembagi tidak boleh NOL";
    } else {
        // Melakukan pembagian bilangan X dengan Y
        Z = X/Y;
        // Menentukan presisi dari hasil bagi
        // dengan nilai dua angka
        // di belakang koma
        cout.setf(ios::fixed, ios::floatfield);
        cout.precision(2);
        // Menampilkan hasil bagi
        cout<<"Hasil bagi = "<<Z;
    }
    return 0;
}return 0;
}
```

STUKTUR IF TIGA KONDISI

- Struktur ini merupakan **struktur percabangan** yang biasanya membingungkan para programmer pemula.
- Percabangan jenis ini merupakan perluasan dari struktur yang memiliki **dua kondisi** yaitu dengan menyisipkan (menambahkan) satu atau beberapa kondisi ke dalamnya dan bentuk umum penulisannya :

STUKTUR IF TIGA KONDISI

```
If(kondisi) {  
Statement_jika_kondisi1_terpenuhi;  
}else if(kondisi2){  
Statement_jika_kondisi2_terpenuhi;  
}else if(kondisi3){  
Statement_jika_kondisi3_terpenuhi;  
}  
....  
}else {  
Statement_jika_semua_kondisi_tidak_terpenuhi;  
}
```

STUKTUR IF TIGA KONDISI

```
#include <iostream>

using namespace std;

int main() {
    int bil;

    cout<<"Masukkan sebuah bilangan bulat "
         <<"yang akan diperiksa : ";
    cin>>bil;

    if (bil > 0) {
        cout<<bil<<" adalah bilangan POSITIF";
    } else if (bil < 0) {
        cout<<bil<<" adalah bilangan NEGATIF";
    } else {
        cout<<"Anda memasukkan bilangan NOL";
    }

    return 0;
}
```

STUKTUR IF TIGA KONDISI

```
#include <iostream>
using namespace std;
int main() {
    double nilai;
    char indeks;
    // Meminta user untuk menentukan nilai
    // yang diperoleh dalam bentuk bilangan
    cout<<"Masukkan nilai yang diperoleh : ";
    cin>>nilai;
    // Melakukan konversi nilai numerik
    // menjadi nilai indeks
    if (nilai >= 85) {
        indeks = 'A';
    } else if (nilai >= 70) {
        indeks = 'B';
    } else if (nilai >= 55) {
        indeks = 'C';
    } else if (nilai >= 40) {
        indeks = 'D';
    } else {
        indeks = 'E';
    }
    // Menampilkan nilai indeks yang didapatkan
    cout<<"Nilai indeks dari nilai "
        <<nilai<<" adalah "<<indeks;
    return 0;
}
```

STATEMENT SWITCH

- Bentuk umum penggunaan statement switch adalah sebagai berikut :

```
switch(ekspresi) {  
  case nilai_konstant1 : Statement_statement; break;  
  case nilai_konstant2 : Statement_statement; break;  
  ....  
  case nilai_konstantN : Statement_statement; break;  
  
  default  
  Statement_statement_alternatif;  
  ;
```

STATEMENT SWITCH

- Tipe data dari ekspresi diatas harus berupa tipe ordinal, seperti : atau karakter.
- Statement **default** berguna untuk mengeksekusi statement alternatif, yaitu jika nilai yang kita masukan ternyata tidak sesuai dengan nilai-nilai konstan yang telah didefinisikan.
- Nilai konstan dapat didefinisikan menggunakan statement **case**.

STATEMENT SWITCH

- Dalam bahasa C standar yang diizinkan untuk menuliskan 257 buah statement **case** dalam sebuah struktur percabangan yang menggunakan statement **switch** sedangkan dalam bahasa C++ mengizinkan 16.384

STATEMENT SWITCH

- Tetapi sebaiknya harus dibatasi pemilihan tersebut untuk alasan efisien.
- Statement **break** diatas berfungsi sebagai statement peloncatan.

STATEMENT SWITCH

```
#include <iostream>
using namespace std;
int main() {
    int nohari;
    cout<<"Masukkan nomor hari (1..7): ";
    cin>>nohari;
    switch (nohari) {
        case 1:
            cout<<"Hari ke-"<<nohari<<": adalah MINGGU"; break;
        case 2:
            cout<<"Hari ke-"<<nohari<<": adalah SENIN"; break;
        case 3:
            cout<<"Hari ke-"<<nohari<<": adalah SELASA"; break;
        case 4:
            cout<<"Hari ke-"<<nohari<<": adalah RABU"; break;
        case 5:
            cout<<"Hari ke-"<<nohari<<": adalah KAMIS"; break;
        case 6:
            cout<<"Hari ke-"<<nohari<<": adalah JUMAT"; break;
        case 7:
            cout<<"Hari ke-"<<nohari<<": adalah SABTU"; break;
        default:
            cout<<"Tidak terdapat nama hari ke-"<<nohari;
    }
    return 0;
}
```