



PENGANTAR KOMPUTER & SOFTWARE II

PERCABANGAN

Tujuan Perkuliahan

- Mahasiswa memahami pengertian dan percabangan
- Mahasiswa dapat menggunakan notasi percabangan dengan benar
- Mahasiswa dapat memanfaatkan jenis-jenis percabangan dalam menyelesaikan persoalan sederhana yang diberikan

Percabangan

- Digunakan dalam satu kasus yang memerlukan pengambilan keputusan
- Percabangan melakukan pengujian terhadap suatu kondisi, kemudian menentukan apakah suatu baris atau blok instruksi akan diproses atau tidak.
- Pengujian tersebut didasarkan kepada pernyataan boolean (TRUE atau FALSE) dan dihubungkan dengan operator logika.

Percabangan

- Melakukan pengujian untuk mengambil suatu keputusan apakah suatu baris atau blok instruksi akan diproses atau tidak.
- Pengujian kondisi ini dilakukan untuk memilih salah satu dari beberapa alternatif yang tersedia.
- Pengujian didasarkan kepada pernyataan boolean (TRUE atau FALSE)
- Pengujian dihubungkan dengan operator logika.

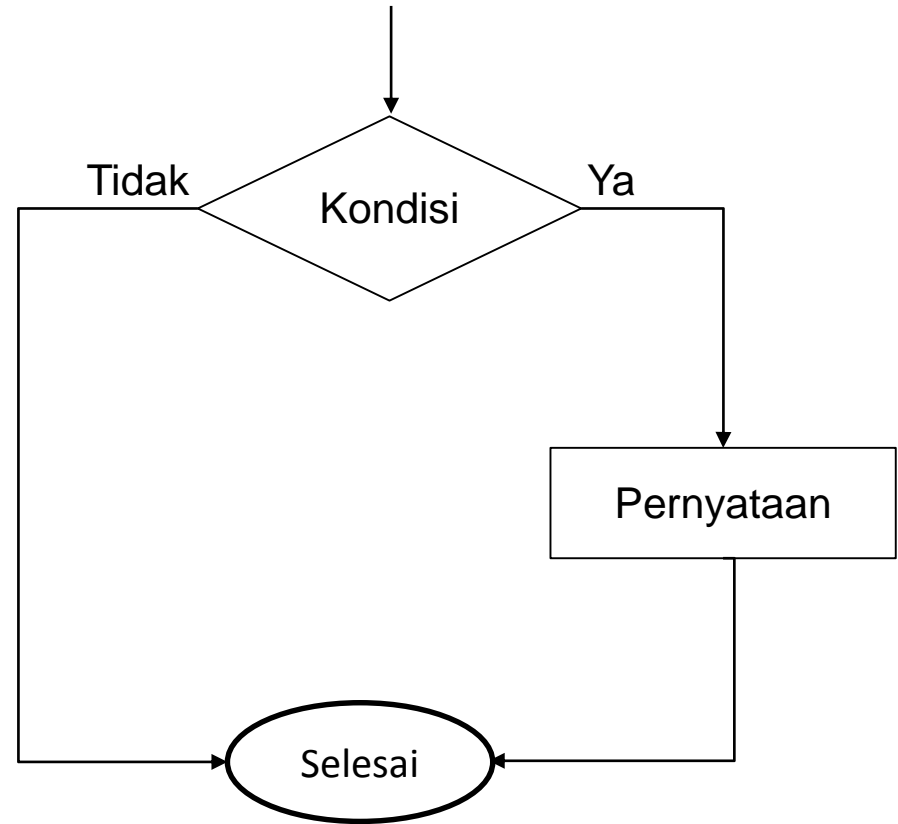
Percabangan

- IF
- IF – ELSE
- IF – ELSE – IF
- SWITCH CASE

Pernyataan IF

- Dalam pemrograman, percabangan digunakan untuk menentukan langkah kerja.
- Memutuskan mengeksekusi suatu blok pernyataan apabila sebuah kondisi terpenuhi.
- Jika tidak, blok pernyataan dapat diabaikan.
- Simbol *decision* digunakan untuk membandingkan suatu kondisi dan memiliki 2 keluaran yaitu “True / False” atau “Benar / Salah” atau “Ya / Tidak” atau “0 / 1”

Flowchart pernyataan IF



Bentuk Umum

```
if (kondisi) {  
    pernyataan  
    . . .  
}
```


Contoh

```
#include <iostream>
using namespace std;

int main(){
    int nilai;

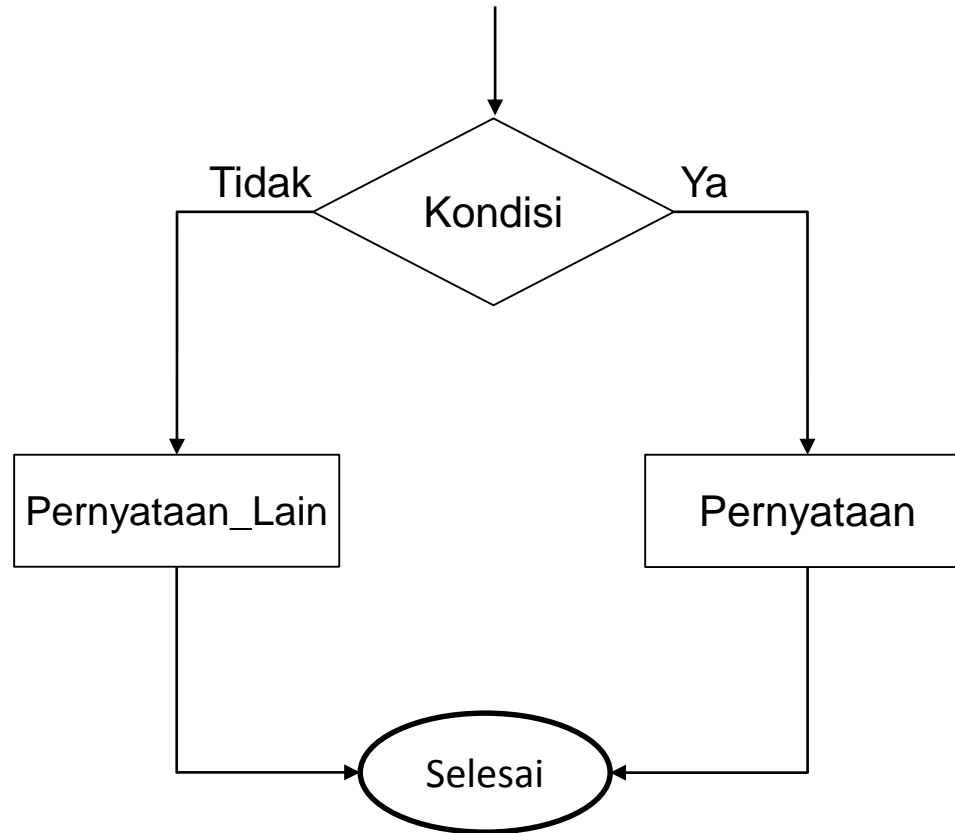
    cout << "Masukkan nilai = ";
    cin >> nilai;

    if(nilai >= 60){
        cout << "Selamat anda lulus.";
    }
}
```

Pernyataan IF-ELSE

- Memutuskan mengeksekusi suatu blok pernyataan apabila sebuah kondisi terpenuhi.
- Namun jika tidak terpenuhi, program akan mengeksekusi blok pernyataan yang lain.

Flowchart



Bentuk Umum

```
if (kondisi) {  
    pernyataan  
    ...  
} else {  
    pernyataan_lain  
    ...  
}
```

Contoh

```
#include <iostream>
using namespace std;

int main(){
    int nilai;

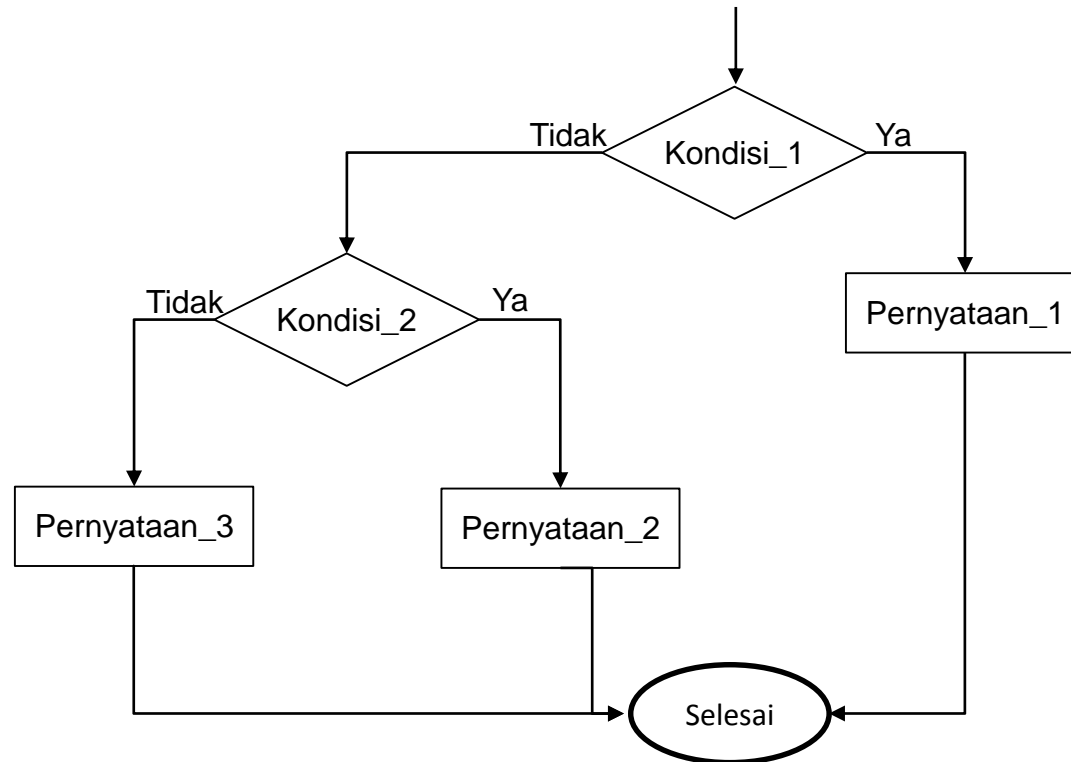
    cout << "Masukkan nilai = ";
    cin >> nilai;

    if(nilai >= 60){
        cout << "Selamat anda lulus.";
    } else {
        cout << "Maaf, anda belum lulus.";
    }
}
```

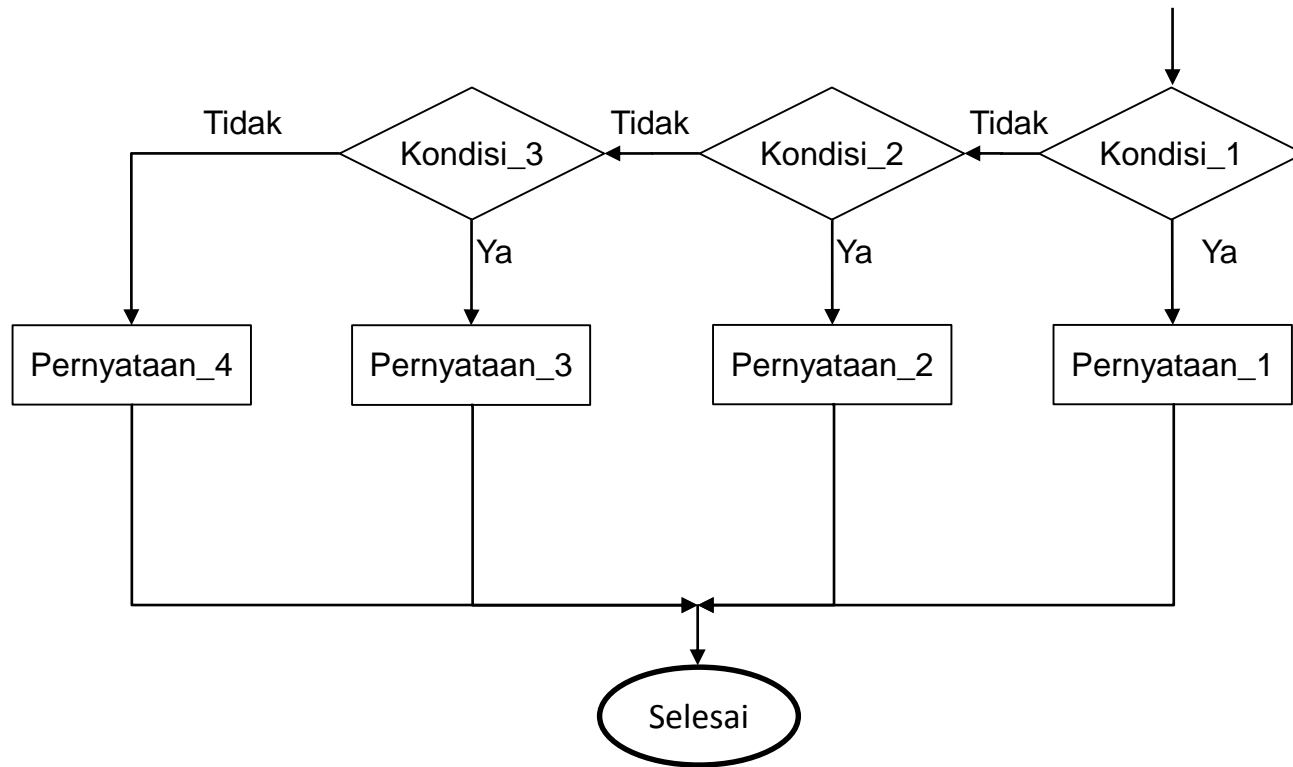
Pernyataan IF-ELSE-IF

- Digunakan jika terdapat lebih dari dua kondisi dalam memutuskan sesuatu.

Flowchart



Flowchart Alternatif 2



Bentuk Umum

```
if(kondisi_1) {  
    pernyataan_1  
    ...  
} else if(kondisi_2) {  
    pernyataan_2  
    ...  
} else {  
    pernyataan_lain  
    ...  
}
```

Contoh

```
#include <iostream>
using namespace std;
int main(){
    int nilai;
    string index;

    cout << "Masukkan nilai = ";
    cin >> nilai;
    if(nilai >= 80){
        index = "A";
    } else if(nilai >= 70){
        index = "B";
    } else if(nilai >= 60){
        index = "C";
    } else if(nilai >= 40){
        index = "D";
    } else {
        index = "E";
    }
    cout << "Nilai anda = " << nilai << endl;
    cout << "Index = " << index << endl;
}
```

Percabangan Operator Boolean

- Digunakan ketika sebuah blok instruksi meminta dua buah kondisi atau lebih secara bersamaan.
- Blok instruksi hanya bisa dijalankan ketika kondisi-kondisi yang diminta terpenuhi.

Operator Boolean

Operator	Keterangan	Simbol dalam C++
AND	Konjungsi	&&
OR	Disjungsi	
NOT	Negasi	!
XOR	Exclusive OR	^

Contoh

```
#include <iostream>
using namespace std;

int main(){
    int nilai, umur;

    cout << "Masukkan nilai tes = ";
    cin >> nilai;
    cout << "Masukkan umur = ";
    cin >> umur;

    if(nilai >= 250 && umur <= 30){
        cout << "Selamat, anda diterima sebagai pegawai.";
    } else {
        cout << "Mohon maaf, anda belum memenuhi syarat.";
    }
}
```

Percabangan Bersarang

- Digunakan apabila terdapat pernyataan percabangan dibawah sebuah pernyataan percabangan lainnya.

Bentuk Umum

```
if(kondisi_1) {  
    if(sub_ondisi_1) {  
        sub_pernyataan_1  
        ...  
    } else {  
        sub_pernyataan_lain  
        ...  
    }  
} else {  
    pernyataan_lain  
    ...  
}
```

Contoh

```
#include <iostream>
using namespace std;

int main(){
    int nilai;
    string index, keterangan;

    cout << "Masukkan nilai = ";
    cin >> nilai;

    if(nilai >= 60){
        keterangan = "Selamat anda lulus.";

        if(nilai >= 80){
            index = "A";
        } else if(nilai >= 70){
            index = "B";
        } else {
            index = "C";
        }
    }
}
```

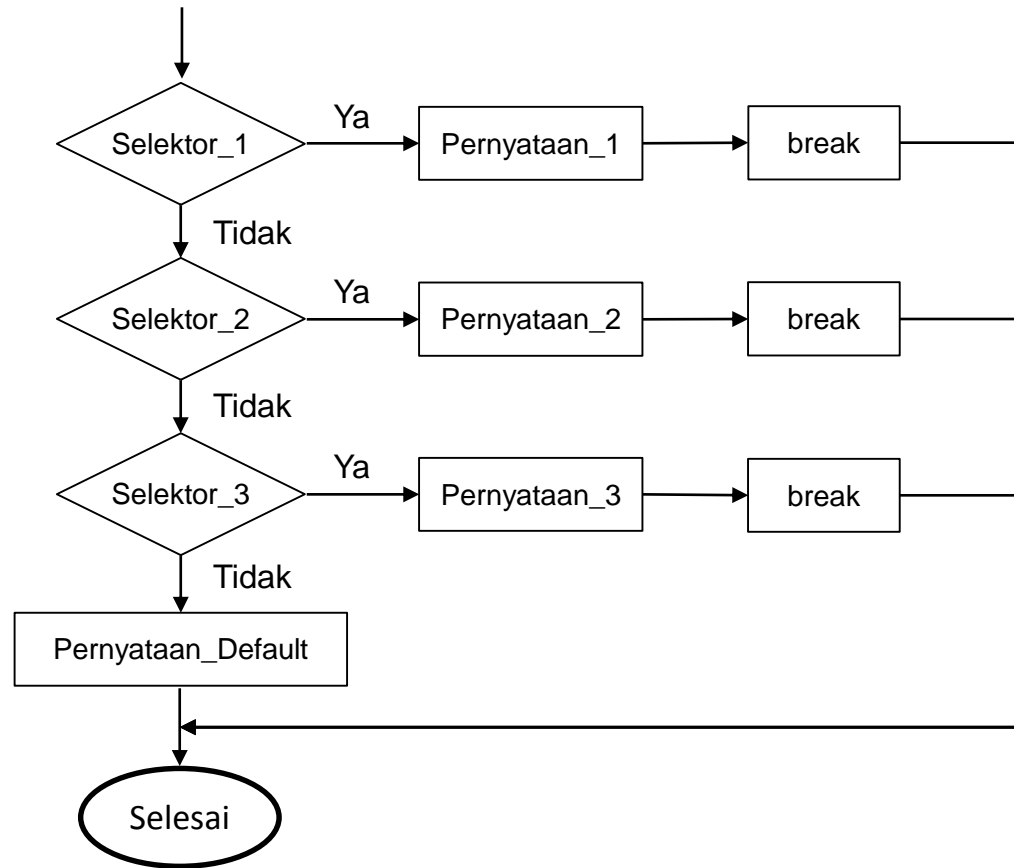
```
    else {
        keterangan = "Belum lulus.";

        if(nilai >= 40){
            index = "D";
        } else {
            index = "E";
        }
    }
}
```


Pernyataan SWITCH-CASE

- Bentuk alternatif dari IF ELSE IF
- Digunakan hanya untuk membandingkan variabel tunggal dengan dengan beberapa kemungkinan nilai-nilai.
- Digunakan untuk pilihan berjumlah banyak.
- Tidak dianjurkan untuk pilihan yang melibatkan jangkauan (range).

Flowchart



Bentuk Umum

```
switch(variable) {  
    case value_1:  
        statement 1;  
        break;  
    case value_2:  
        statement 2;  
        break;  
    case value_3:  
        statement 3;  
        break;  
    default:  
        statement 4;  
        break;  
}
```

Contoh

```
#include <iostream>
using namespace std;

int main(){
    char index;
    string keterangan;

    cout << "Masukkan index nilai:";
    cin  >> index;
```

```
switch(index){
    case 'A':
        keterangan = "Excelent";
        break;
    case 'B':
        keterangan = "Good";
        break;
    case 'C':
        keterangan = "Fair";
        break;
    case 'D':
        keterangan = "Poor";
        break;
    default:
        keterangan = "Tidak diketahui";
}

cout << "Keterangan index nilai = ";
cout << keterangan << endl;
}
```

Tugas 1

- Suatu bilangan dikatakan genap apabila nilainya habis dibagi dengan 2. Apabila sisa bagi sama dengan 1, maka ia dikatakan bilangan ganjil.
- Buatlah flowchart dan programnya!
- Input: bilangan. Output: kategori bilangan.

Tugas 2

- Suatu pengelompokan bilangan positif, negatif, dan bilangan Nol.
- Buatlah flowchart dan programnya!
- Input: bilangan. Output: kelompok bilangan.

TERIMA KASIH