

PRAKTIKUM 2

Pemilihan Kondisi

Tujuan Praktikum :

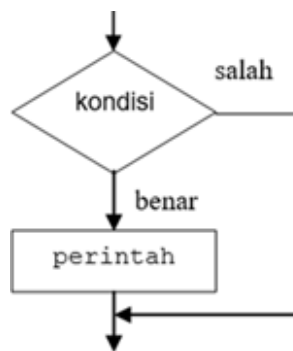
Mempraktekkan struktur dasar algoritma pemilihan kondisi/percabangan/seleksi :

- IF .. THEN
- IF .. THEN .. ELSE
- IF .. THEN .. ELSE IF
- NESTED IF
- .. SWITCH .. CASE ..

Struktur kontrol pemilihan adalah pernyataan yang mengizinkan user untuk memilih dan mengeksekusi blok kode spesifik dan mengabaikan blok kode yang lain.

Percobaan 2.1: Analisa Kasus Tunggal (IF .. THEN)

Pernyataan *if then* mempunyai pengertian, “Jika kondisi bernilai benar, maka perintah akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan”. Dari pengertian tersebut dapat dilihat dari diagram alir berikut:



Bentuk umum dari pernyataan *if*

```
if ( kondisi )
{
    perintah;
}
```

Penulisan ***kondisi*** berada di dalam tanda kurung, jika pemakaian *if* diikuti dengan pernyataan majemuk, maka bentuk penulisannya sebagai berikut:

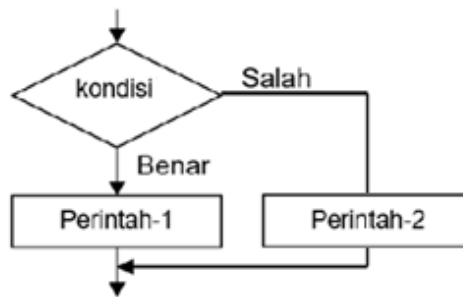
```
if ( kondisi )
{
    perintah 1;
    perintah 2;
    .....
}
```

```
#include <iostream>
using namespace std;
main() {
    int n,d;
    cout << "Masukkan bil. Pertama :"; cin >> n;
    cout << "Masukkan bil. Kedua :"; cin >> d;

    if(n%d==0) {
        cout << n << " habis dibagi oleh " << d << endl;
    }
}
```

Percobaan 2.2: Analisa 2 Kasus Komplementer (IF .. THEN .. ELSE)

Pernyataan *if-else* mempunyai pengertian, “Jika kondisi bernilai benar, maka perintah-1 akan dikerjakan dan jika tidak memenuhi syarat maka akan mengerjakan perintah-2”. Dari pengertian tersebut dapat dilihat dari diagram alir berikut:



Bentuk umum dari pernyataan *if-else*

```
if ( kondisi ) {
    perintah-1;
}
else {
    perintah-2;
}
```

Perintah-1 dan perintah-2 dapat berupa sebuah pernyataan tunggal, pernyataan majemuk atau pernyataan kosong. Jika pemakaian *if-else* diikuti dengan pernyataan majemuk, maka bentuk penulisannya sebagai berikut:

```
if ( kondisi ) {
    perintah-1;
    ...
}
else {
    perintah-2;
    ...
}
```

```
#include <iostream>
using namespace std;
main(){
    int i;
    cout << "Masukkan nilai i : ";
    cin >> i;

    if(i%2==0){
        cout << "i = " << i << " adalah bilangan genap";
    }
    else {
        cout << "i = " << i << " adalah bilangan ganjil";
    }
}
```

Percobaan 2.3: Analisa Banyak Kasus (IF .. THEN .. ELSE IF)

Struktur percabangan *if* lebih dari dua kondisi atau *if-else* majemuk sebenarnya merupakan pengembangan dari struktur *if* dua kondisi, karena percabangan ini akan menambahkan (menyisipkan) satu atau lebih kondisi ke dalamnya. Artinya dalam pernyataan *if-else if* lebih dari dua kondisi: *“Jika kondisi 1 bernilai benar, maka perintah-1 akan dikerjakan, jika kondisi 1 salah maka akan mengecek kondisi 2 dan jika kondisi 2 bernilai benar, maka perintah-2 akan dikerjakan, jika kondisi 2 juga salah maka akan mengecek kondisi berikutnya dan akan mengerjakan perintah pada struktur blok yang memiliki kondisi bernilai benar”*.

Seandainya tidak ada kondisi yang bernilai benar, maka akan mengerjakan perintah yang berada pada struktur blok *else*. untuk lebih jelasnya mari kita perhatikan struktur *if-else if* majemuk dibawah ini.

```
if ( kondisi )
{
    ... perintah 1;
    ... perintah 2;
}
else if ( kondisi )
{
    ... perintah 1;
    ... perintah 2;
}
else
{
    ... perintah 1;
    ... perintah 2;
}
```

```
#include <iostream>
using namespace std;
main() {
    int gol;
    cout << "Masukkan nilai gol : ";
    cin >> gol;

    if(gol==1){
        cout << "Gaji = 100";
    }
    else if(gol==2){
        cout << "Gaji = 200 ";
    }
    else if(gol==3){
        cout << "Gaji = 300";
    }
    else{
        cout << "Golongan Salah";
    }
}
```

Percobaan 2.4: Kasus Bersarang (NESTED IF)

Nested if merupakan pernyataan *if* yang berada didalam pernyataan *if* yang lainnya.

Bentuk penulisan pernyataan Nested if adalah :

```
If ( kondisi ) {
    If ( kondisi ) {
        ... perintah;
    } else {
        ... perintah;
    }
}
else
{
    If ( kondisi ) {
        ... perintah;
    } else {
        ... perintah;
    }
}
```

```
#include <iostream>
using namespace std;
main() {
    int a, b, c, max;
    cout << "Masukkan 3 buah bilangan : ";
    cin >> a; cin >> b; cin >> c;

    if (a > b) {
        if (a > c)
            max = a; //a > b and a > c
        else
            max = c; //c >= a > b
    }
    else {
        if (b > c)
            max = b; //b >= a and b > c
        else
            max = c; // c >= b >= a
    }

    Cout << "Maksimum adalah : "<< max << endl;
}
```

Percobaan 2.5: Pernyataan Banyak Kasus (.. SWITCH .. CASE)

Bentuk dari *switch - case* merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah atau banyak alternatif penyelesaian. Pernyataan *switch - case* ini memiliki kegunaan sama seperti *if - else* bertingkat, tetapi penggunaannya untuk memeriksa data yang bertipe **karakter** atau **integer**. Bentuk penulisan perintah ini sebagai berikut:

```
switch (ekspresi integer atau karakter )
{
    case konstanta-1 :
        ... perintah 1;
        ... perintah 2;
        break;
    case konstanta-2 :
        ... perintah 1;
        ... perintah 2;
        break;
    default :
        ... perintah 1;
        ... perintah 2;
}
```



```
#include <iostream>
using namespace std;
main() {
    int gol;
    cout << "Masukkan nilai gol : "; cin >> gol;

    switch (gol) {
        case 1 :
            cout << "Gaji = 100";
            break;
        case 2 :
            cout << "Gaji = 200 ";
            break;
        case 3 :
            cout << "Gaji = 300";
            break;
        default :
            cout << "Golongan Salah";
            break;
    }
}
```