

Algoritma & Pemrograman

Pertemuan 9:

- BRANCHING/PERCABANGAN dalam C
- LOOPING/PERULANGAN/ITERASI dalam C

Percabangan: IF dan IF-ELSE

- Perintah yang digunakan adalah `if` dan `if-else`
- Bentuk umum:

- Bentuk `if`

```
if (kondisi) {  
    //lakukan sejumlah action disini jika kondisi bernilai true  
}
```

- Bentuk `if-else`

```
if (kondisi) {  
    //lakukan sejumlah action disini jika kondisi bernilai true  
}  
else {  
    //lakukan sejumlah action disini jika kondisi bernilai false  
}
```

Percabangan: IF dan IF-ELSE

- Contoh soal: penentuan letak kuadran berdasarkan nilai koordinat (x,y)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int x,y;
6      printf("Input nilai x dan y:");
7      scanf("%d %d",&x,&y);
8      if (x>=0 && y>=0) {
9          printf("Kuadran I");
10     }
11     else if (x<0 && y>=0) {
12         printf("Kuadran II");
13     }
14     else if (x<0 && y<0) {
15         printf("Kuadran III");
16     }
17     else printf("Kuadran IV");
18     return 0;
19 }
```

Percabangan: SWITCH-CASE

- Perintah `switch-case` digunakan pada percabangan dengan 2 kasus atau lebih untuk menyederhanakan penulisan `if-else` yang bertingkat
- Bentuk umum:

Catatan: nilai_1, nilai_2, dst adalah nilai yang bertipe int atau char

```
switch (ekspresi) {  
    case nilai_1://lakukan sejumlah action di sini  
        break;  
    case nilai_2://lakukan sejumlah action di sini  
        break;  
    ...  
    case nilai_n://lakukan sejumlah action di sini  
        break;  
    default: //berfungsi sebagai else dalam if  
}
```

Contoh: Mencetak nama bulan sesuai nomor bulan

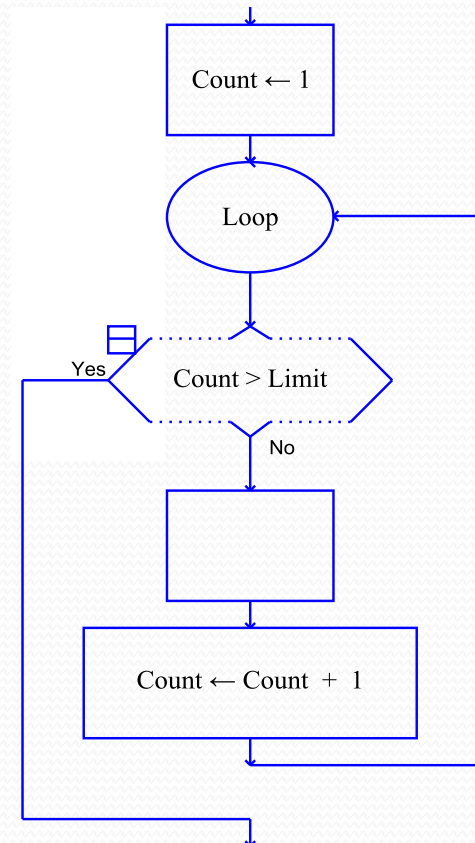
```
#include <stdio.h>

int main()
{
    int nobul;
    printf("Input nomor bulan:");
    scanf("%d",&nobul);
    switch(nobul) {
        case 1: printf("Januari");
                break;
        case 2: printf("Februari");
                break;
        case 3: printf("Maret");
                break;
        case 4: printf("April");
                break;
        case 5: printf("Mei");
                break;
        case 6: printf("Juni");
                break;
        case 7: printf("Juli");
                break;
        case 8: printf("Agustus");
                break;
        case 9: printf("September");
                break;
        case 10: printf("Oktober");
                break;
        case 11: printf("November");
                break;
        case 12: printf("Desember");
                break;
        default: printf("Bukan nomor bulan yang benar");
    }
}
```

Perulangan Dengan WHILE

- Pernyataan ini akan mengulang satu atau beberapa pernyataan, jika masih memenuhi kondisi. Pengecekan kondisi/syarat perulangan dilakukan di awal iterasi.
- Bentuk umum:

```
while (kondisi) {  
    //lakukan sejumlah action  
    //disini jika kondisi  
    //bernilai benar  
}
```



Perulangan Dengan WHILE

- Contoh: Mencetak nilai 1-10 ke layar

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i=1;
6      while (i<=10) {
7          printf("%d\n",i);
8          i+=1;
9      }
10     return 0;
11 }
```

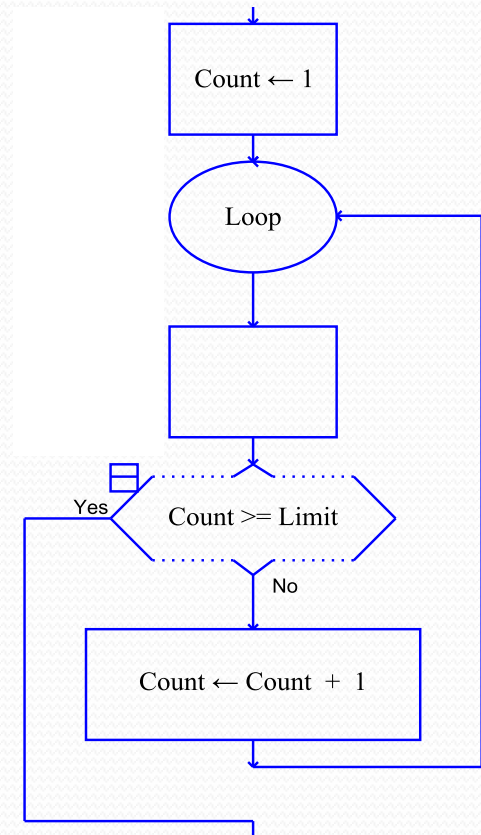
Output:

```
1
2
3
4
5
6
7
8
9
10
```

Perulangan Dengan DO-WHILE

- Seperti halnya while, perintah ini menyatakan perulangan proses selama kondisi tertentu. Pengecekan kondisi/syarat perulangan dilakukan di akhir iterasi.
- Dengan menggunakan do-while, sebuah statement dieksekusi setidaknya 1 kali
- Bentuk umum:

```
do {  
    //lakukan sejumlah action disini jika  
    //kondisi bernilai benar  
}  
while (kondisi)
```



Perulangan Dengan DO-WHILE

- Contoh: Pengecekan validitas input usia (0-130)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int usia;
6      do {
7          printf("Masukkan usia Anda:");
8          scanf("%d",&usia);
9      }
10     while (usia<0 or usia>130);
11 }
```

Perulangan Dengan FOR

- Perintah FOR melakukan hal yang sama dengan perintah perulangan yang lain, tetapi di awal perulangan terdapat deklarasi range dari perulangan yang akan dilakukan
- Perulangan dengan FOR dilakukan jika jumlah yang perulangan akan dilakukan telah diketahui sebelumnya
- Bentuk umum:

```
for (inisialisasi_cacah;kondisi;perubahan_nilai_cacah) {  
    //lakukan sejumlah action disini selama kondisi  
    //bernilai benar  
}
```

Perulangan Dengan FOR

- Contoh: Mencetak 1-10 dengan `for`

```
1  #include <stdio.h>
2
3  int main()
4  {
5      for(int i=1;i<=10;i++) {
6          printf("%d\n",i);
7      }
8      return 0;
9  }
```

Deklarasi nilai
awal var cacah

Kondisi var
cacah

Perubahan nilai
var cacah

Output:

```
1
2
3
4
5
6
7
8
9
10
```

Latihan

- Buatlah program untuk menampilkan bilangan genap dengan batasan nilai lebih kecil sama dengan N. N adalah input dari user. Fasilitas: user bisa mengulangi input N tanpa harus menjalankan/ meng-compile ulang program.
- Buatlah program untuk menentukan sisa hasil pembagian antara bilangan yang dimasukkan dengan bilangan pembagi dengan menggunakan konsep looping. Apabila sisa baginya = 0 maka dicetak tidak ada dan kalau ada sisa baginya, maka sisa bagi tersebut ditampilkan. Tidak boleh memakai operator %
- Buatlah program yang mempunyai fungsi berparameter untuk menghitung N! (N faktorial).

$$N! = N \times (N - 1) \times (N - 2) \times \dots \times 1.$$

$$\text{Contoh: } 4! = 4 \times 3 \times 2 \times 1 = 24$$

N adalah input dari user. Nilai maksimal N adalah 7. Hasil perhitungan ditampilkan ke layar. Buat menu pilihan untuk menawarkan kepada user apakah ingin mengulangi lagi prosesnya ataukah tidak.

- Buatlah program yang memiliki fungsi untuk menampilkan N (= input user) bilangan pertama deret Fibonacci. Deret Fibonacci adalah: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

$$\text{Fibo}(N) = \text{Fibo}(N - 1) + \text{Fibo}(N - 2)$$

Default: Fibo ke-1 = 1, Fibo ke-2 = 1

Contoh: Input : N = 6
 Output : 1 1 2 3 5 8