

# Algoritma & Pemrograman

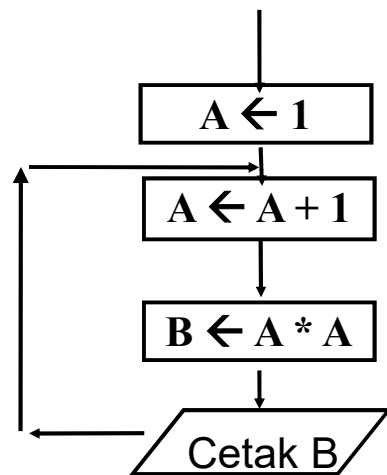
## PENGULANGAN

**Pengampu : Agus Priyanto, M.Kom**



**INSTITUT TEKNOLOGI TELKOM**  
**Smart, Trustworthy, And Teamwork**

Looping atau pemutaran kembali, terjadi ketika mengalihkan arus diagram alur kembali ke atas, shg bbrp alur berulang bbrp kali.



- (1) Variabel A diberi harga 1
  - (2) Var. A berubah hrg menjadi 2
  - (3) Var. B diberi hrg sebesar  
    hrg A dikalikan hrg A
  - (4) Harga B dicetak
- Lalu ke (2), (3), (4) & kembali

lagi ke (2) dstnya... Jadi yang akan tercetak adalah harga-harga 4, 9, 16, ... dst



# Struktur Perulangan

Struktur perulangan secara umum ada dua bagian:

- ***Kondisi perulangan***

- ☐ Ekspresi boolean yang harus dipenuhi untuk melaksanakan perulangan

- ***Badan perulangan***

- ☐ Bagian algoritma yang diulang

Struktur perulangan disertai pula:

- ***Inisialisasi***

- ☐ Aksi yang dilakukan sebelum perulangan dilakukan pertama kali

- ***Terminasi***

- ☐ Aksi yang dilakukan setelah perulangan selesai dilaksanakan



# Struktur perulangan

- Struktur perulangan secara umum:

<inisialisasi>

awal perulangan

badan perulangan

akhir perulangan

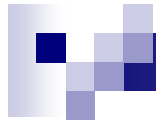
<terminasi>



# Struktur perulangan

Beberapa hal yang harus diperhatikan:

- Inisialisasi dan terminasi *tidak selalu harus ada*, namun dalam beberapa kasus inisialisasi biasanya diperlukan.
- **Suatu perulangan harus berhenti!**
- Beberapa struktur perulangan dapat dipakai untuk masalah yang sama, namun ada notasi perulangan yang hanya cocok dipakai untuk masalah tertentu.



# Struktur perulangan

Notasi struktur perulangan:

- Struktur FOR
- Struktur WHILE
- Struktur DO-WHILE



# Struktur FOR

- Struktur perulangan FOR digunakan untuk menghasilkan perulangan sejumlah kali yang dispesifikasikan.
- Jumlah perulangan diketahui atau dapat ditentukan sebelum eksekusi.
- Diperlukan *variabel counter/pencacah*.
- Jika cacah perulangan sudah mencapai jumlah yang dispesifikasikan, maka proses perulangan berhenti.

# Struktur FOR

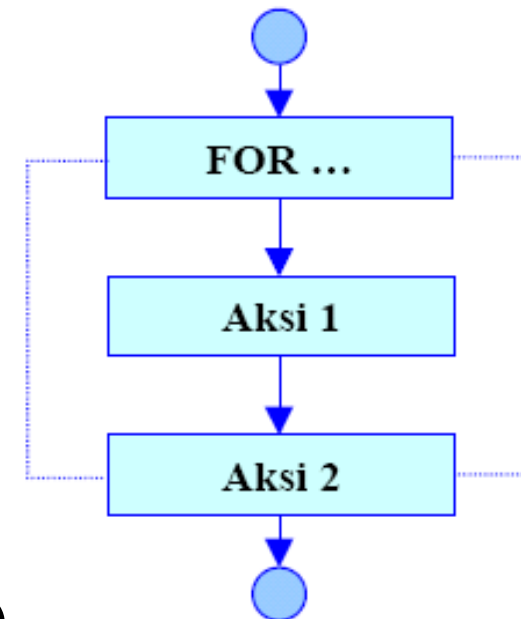
- Bentuk umum struktur FOR:

```
for(<ekspresi1>; <ekspresi2>; <ekspresi3>)  
{  
    aksi1;  
    aksi2;  
}
```

Ekspresi1: inisialisasi counter

Ekspresi2: syarat pelaksanaan

Ekspresi3: naikkan counter (inkremen)







# Struktur FOR

- Bentuk umum struktur FOR menaik (ascending):

```
for(var=nilai_awal; nilai_awal<=nilai_akhir; var++)  
{  
    aksi1;  
    aksi2;  
}
```



# Struktur FOR

*Yang harus diperhatikan adalah:*

- Pencacah harus dari tipe data yang memiliki *predecessor* dan *successor*, yakni *integer* atau *karakter*.
- Aksi adalah satu atau lebih instruksi yang diulang
- **nilai\_awal** harus lebih kecil atau sama dengan **nilai\_akhir**
- Pada awalnya, pencacah diinisialisasi dengan **nilai\_awal**.
- Nilai pencacah secara otomatis **bertambah** satu setiap kali aksi perulangan dimasuki, sampai akhirnya nilai pencacah sama dengan **nilai\_akhir**
- Jumlah perulangan yang terjadi:  
**nilai\_akhir - nilai\_awal + 1**



# Struktur FOR

## ■ Contoh:

```
/* mencetak kata 'Halo' sebanyak 10  
kali */
```

```
int i;
```

```
for(i=1;i<=10;i++)  
    cout<<"Halo\n", i;
```



# Struktur FOR

- Bentuk umum struktur FOR menurun (descending):

```
for(var=nilai_awal; nilai_awal>=nilai_akhir; var--)  
{  
    aksi1;  
    aksi2;  
}
```



# Struktur FOR

Yang harus diperhatikan adalah:

- Pencacah harus dari tipe data yang memiliki *predecessor* dan *successor*, yakni *integer* atau *karakter*.
- Aksi adalah satu atau lebih instruksi yang diulang
- **nilai\_awal** harus lebih besar atau sama dengan **nilai\_akhir**
- Pada awalnya, pencacah diinisialisasi dengan **nilai\_awal**.
- Nilai pencacah secara otomatis **berkurang** satu setiap kali aksi perulangan dimasuki, sampai akhirnya nilai pencacah sama dengan **nilai\_akhir**
- Jumlah perulangan yang terjadi:  
**nilai\_awal - nilai\_akhir + 1**



# Struktur FOR

## ■ Contoh:

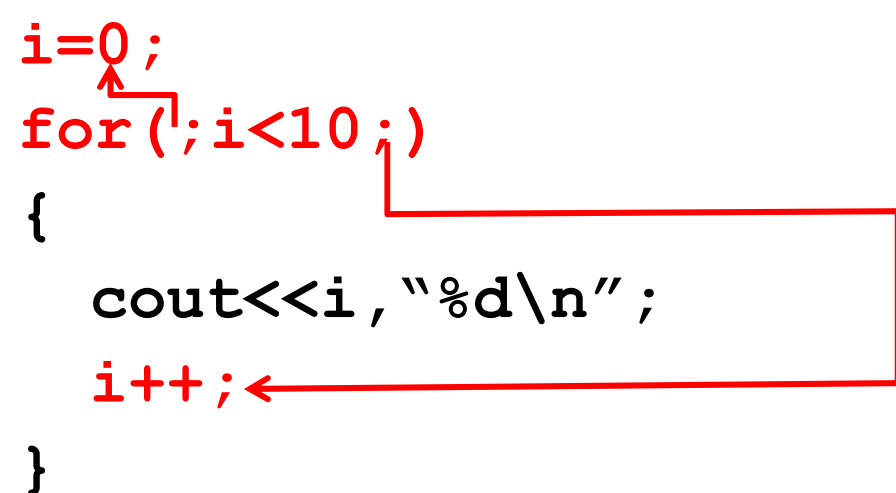
```
/* mencetak angka: 9 8 7 6 5 4 3 2 1 0 */
```

```
int i;
```

```
for(i=9;i>=0;i--)  
    cout<<i ,"\n";  
return 0;
```

# Contoh Lain FOR

```
int i;  
i=0; //inisialisasi  
for(;i<10;) //kondisi  
{  
    cout<<i, "%d\n"; //proses  
    i++; //inkremen  
}  
cout<<"-----\n";  
Cout<<i, "%d\n"; //terminasi
```





# Contoh Lain FOR

## Contoh :

for (a =0; a<=10; ++a)

**perulangan positif (+1)**

for (a =0; a<=10; a+=2)

**perulangan positif (+2)**

for (a=10; a>=0; --a)

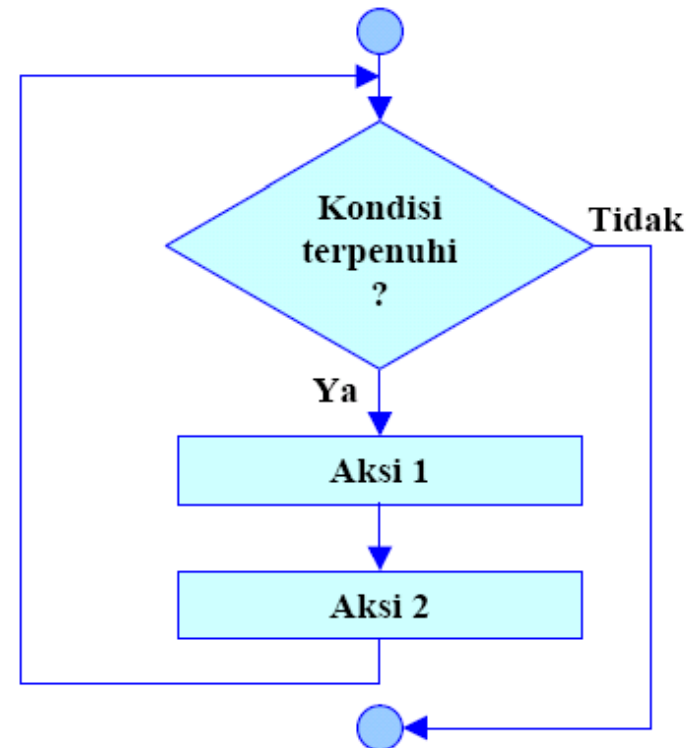
**perulangan negatif(-1)**

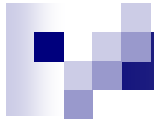


# Struktur WHILE

## ■ Bentuk umum struktur WHILE:

```
inisialisasi;  
while(<ekspresi> )  
{  
    aksi1;  
    aksi2;  
}  
terminasi;
```





# Struktur WHILE

Beberapa hal yang harus diperhatikan:

- **Aksi** atau runtunan aksi akan dilaksanakan berulang kali selama **<ekspresi>** bernilai **true**. Jika kondisi bernilai **false**, badan perulangan tidak akan dilaksanakan yang artinya perulangan selesai.



# Struktur WHILE

## ■ Contoh:

*/\* mencetak kata 'Halo' sebanyak 10 kali \*/*

```
int i;
```

```
i=0;
```

```
while(i<=10)
```

```
{
```

```
    cout<<"Halo\n";
```

```
    i++;
```

```
}
```



# Struktur WHILE

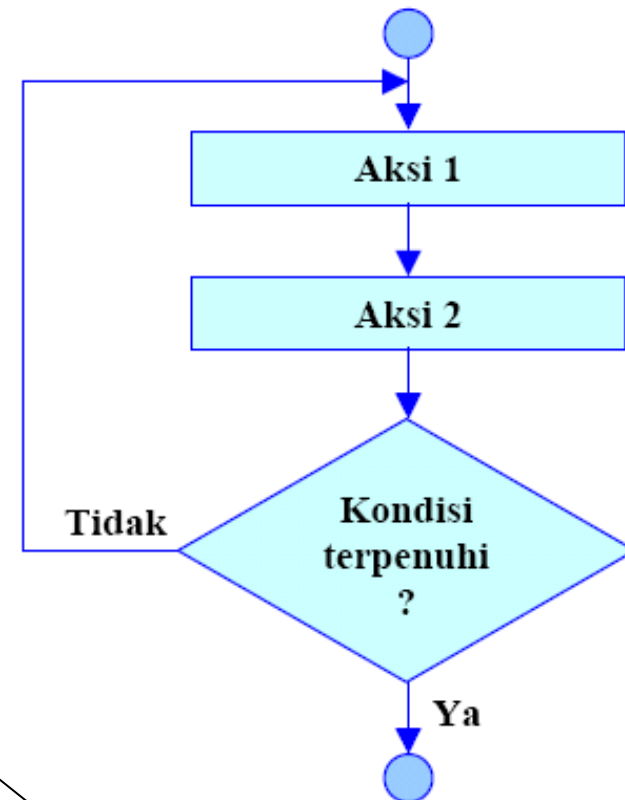
Hal penting yang tidak boleh dilupakan dalam program contoh:

- Melupakan inisialisasi: `i=1;`
- Tidak menuliskan instruksi yang mengubah kondisi: `i++;`

# Struktur DO-WHILE

## ■ Bentuk umum:

```
inisialisasi;  
do  
{  
    aksi1;  
    aksi2;  
} while(<ekspresi>);  
terminasi;
```



Jangan lupa  
titik koma



# Struktur DO-WHILE

Beberapa hal yang perlu diperhatikan:

- **Aksi** di dalam badan kalang diulang sampai kondisi boolean bernilai **false**. Jika kondisi boolean masih **true**, perulangan masih terus dilakukan.
- Karena proses perulangan suatu saat harus berhenti, maka di dalam badan perulangan harus ada **aksi** yang mengubah nilai peubah **<ekspresi>**.



# Struktur DO-WHILE

## ■ Contoh:

*/\* mencetak kata 'Halo' sebanyak 10 kali \*/*

```
int i;
```

```
i=1;
```

```
do
```

```
{
```

```
    cout<<"Halo\n";
```

```
    i++;
```

```
} while(i<=10);
```



## WHILE atau DO-WHILE ?

- Meskipun kadang bisa digunakan untuk fungsi yang sama, ketiga struktur perulangan sebaiknya digunakan sesuai dengan kasus yang dihadapi:
  - Jika banyaknya perulangan dapat dipastikan  
→gunakan struktur FOR
  - Jika perulangan dihentikan jika kondisi tertentu dipenuhi  
→gunakan WHILE atau DO-WHILE





## WHILE atau DO-WHILE?

WHILE	DO-WHILE
Pemeriksaan kondisi dilakukan di <b>awal</b> perulangan	Pemeriksaan kondisi dilakukan di <b>akhir</b> perulangan
Badan perulangan <b> mungkin tidak pernah </b> dieksekusi	Badan perulangan <b> paling sedikit 1 kali </b> dieksekusi

**Gunakan struktur WHILE pada kasus yang mengharuskan terlebih dahulu pemeriksaan kondisi objek tersebut sebelum dimanipulasi.**

**Gunakan struktur DO-WHILE pada kasus yang terlebih dahulu memanipulasi objek, baru kemudian memeriksa kondisi objek tersebut.**



# Latihan : Tentukan Outputnya

## No. 1

```
using namespace std;
int main()
{
    int x =5;
    int y=3;
    for (x=1; x<=5; x++)
    {
        y = y + x;
        cout <<"Nilai X = " << x;
        cout <<"Nilai Y = " << y <<"\n";
    }
    return 0;
}
```



## No. 2

```
using namespace std;
int main()
{
    int x, y, z;
    x = 5;
    y = 8;
    for (x=1; x<=10; x++)
    {
        y = y + x;
        if (x >= 4)
        {
            x = x + 4;
            cout <<"Nilai Y = " << y <<"\n";
        }
    }
    return 0;
}
```

### No. 3

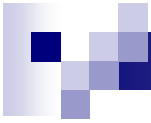
```
using namespace std;
int main()
{
    int num;
    int sum;
    cin >> sum;
    num = sum;
    while(num != -1)
    {
        cin >> num;
        sum = sum + 2 * num;
    }
    cout << "Sum = " << sum << endl;
    return 0;
}
```

**Jika inputnya adalah : 3 4 6 7 2 -1, tentukan output ?**

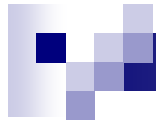


## No. 4

```
using namespace std;
int main()
{
    int num1, num2;
    int temp = 0;
    cout << "Masukan 2(dua) bilangan :";
    cin >> num1 >> num2 ;
    cout << endl ;
    while(num1 % 3 != num2 % 5)
    {
        temp = num1 + num2;
        num1 = num2;
        num2 = temp;
        cout << temp << " ";
    }
    cout << endl;
    return 0;
}
```



- a. Tentukan outputnya jika inputnya 10 17 ?
- b. Tentukan outputnya jika inputnya -4 11 ?
- c. Tentukan outputnya jika inputnya 2 29 ?



# Jawaban

1. Nilai  $X = 1$  Nilai  $Y = 4$

Nilai  $X = 2$  Nilai  $Y = 6$

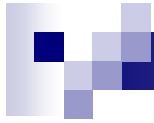
Nilai  $X = 3$  Nilai  $Y = 9$

Nilai  $X = 4$  Nilai  $Y = 13$

Nilai  $X = 5$  Nilai  $Y = 18$

2. Nilai  $Y = 18$

Nilai  $Y = 27$



4.a. Tentukan outputnya jika inputnya 10 17 ? 27

4.b. Tentukan outputnya jika inputnya -4 11 ? 7

4.c. Tentukan outputnya jika inputnya 2 29 ? 31 60 91 151



