

Bab 5. Looping 1 (Pengulangan Proses)

Konsep Pemrograman
Politeknik Elektronika Negeri Surabaya
2006

Overview

- Pendahuluan
- Perulangan `for`
- Perulangan `while`
- Perulangan `do while`

Pendahuluan

- Mengulang suatu proses merupakan tindakan yang banyak dijumpai dalam pemrograman.
- Pada semua bahasa pemrograman, perulangan proses ditangani dengan suatu mekanisme yang disebut *loop*.
- Dengan menggunakan *loop*, suatu proses yang berulang misalnya menampilkan tulisan yang sama seratus kali pada layar dapat diimplementasikan dengan kode program yang pendek.

Pendahuluan

- Dalam pemrograman C, perulangan diimplementasikan dengan salah satu bentuk pernyataan sbb :
 - pernyataan *for*
 - pernyataan *while*
 - pernyataan *do-while*

Pemilihan Bentuk Perulangan

- Jika bisa ditentukan HOW MANY TIMES-nya
→ sebaiknya gunakan *for*
- Jika tidak bisa ditentukan HOW MANY TIMES-nya, tapi diketahui kondisi untuk menghentikan loop-nya
→ sebaiknya gunakan *while* atau *do while*
 - *while* → pengecekan kondisi dilakukan di AWAL
→ ada kemungkinan body loop tidak akan pernah dilakukan sama sekali (0 kali)
 - *do while* → pengecekan kondisi dilakukan di AKHIR
→ minimal body loop akan dieksekusi satu kali (*at least ONCE*)

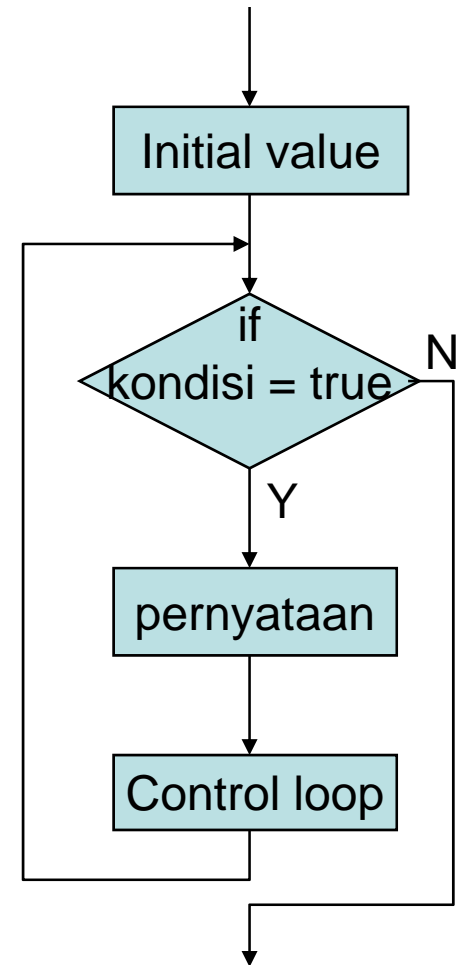
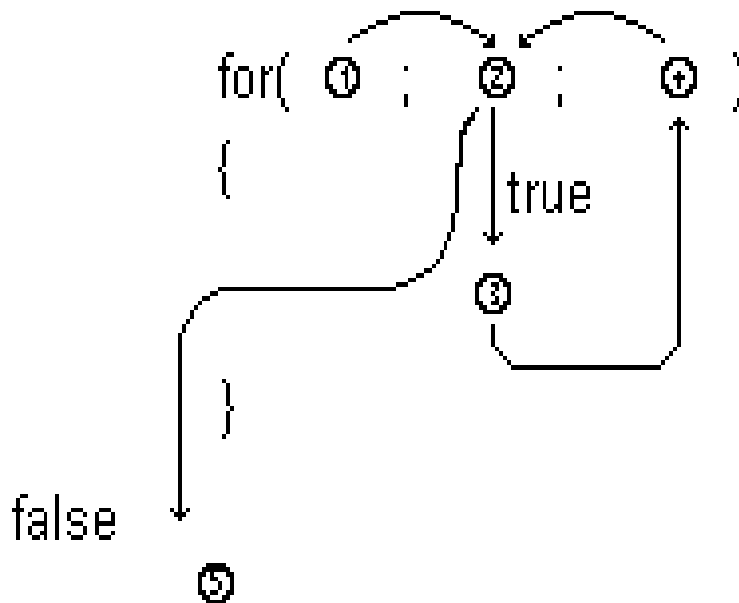
Perulangan *for*

- Bentuk pernyataan ini for
for (ekspresi1; ekspresi2; ekspresi3)
pernyataan;
 - ekspresi1 : *initial value* variabel pengendali *loop*.
 - ekspresi2 : *continue condition*, kondisi yang dipakai untuk melanjutkan *loop*.
 - ekspresi3 : pengatur kenaikan nilai variabel pengendali *loop*.
- Ketiga ekspresi dalam *for* tersebut harus dipisahkan dengan tanda titik koma (;)
- Dalam hal ini pernyataan bisa berupa tunggal maupun jamak.

Diagram Alir & Urutan Proses

Perulangan for

- The following diagram shows the order of processing each part of a *for*



Perulangan *for*

- Jika berbentuk jamak, maka pernyataan tersebut harus diletakkan di antara kurung kurawal buka ({) dan kurung kurawal tutup (})

```
for(ekspresi1; ekspresi2; ekspresi3)
{
    pernyataan;
    pernyataan;
    .
    .
    .
}
```


Perulangan *while*

- *while* menyediakan mekanisme untuk mengulang pernyataan-pernyataan dalam body loop selama kondisi dalam *while* masih bernilai TRUE

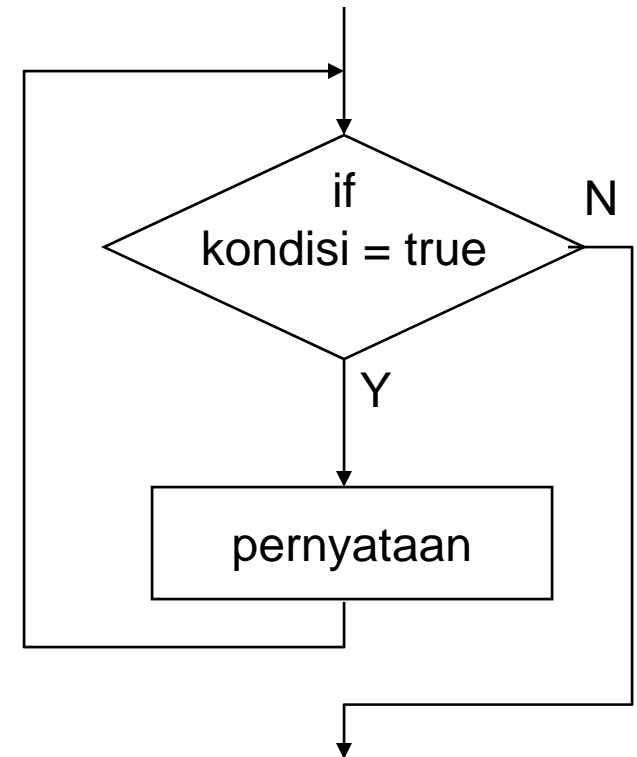
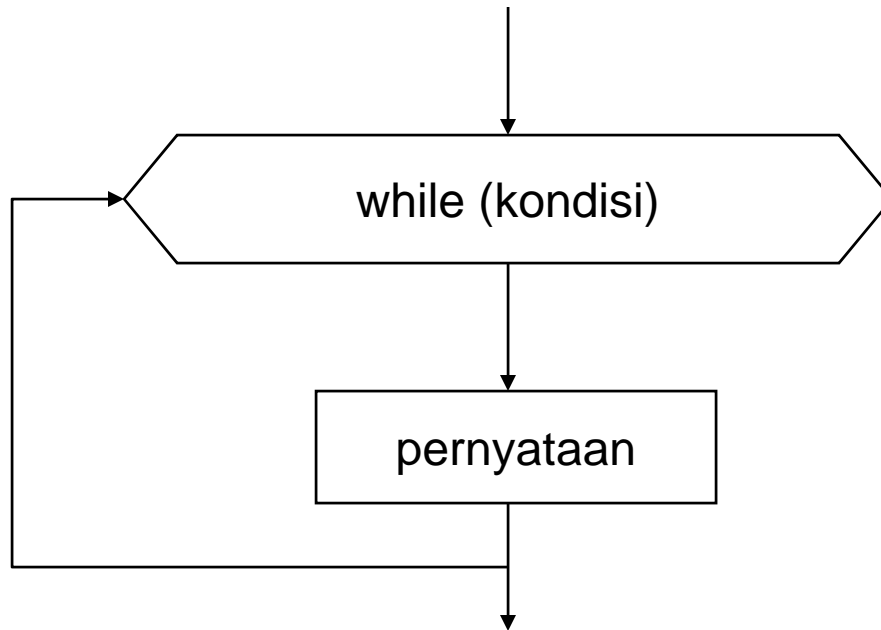
- Formatnya :

```
while(kondisi) {  
    pernyataan;  
}
```

- Somewhere within the body of the *while* loop a statement must alter the value of the condition to allow the loop to finish.

Diagram Alir Perulangan *while*

```
while(kondisi) {  
    pernyataan;  
}
```

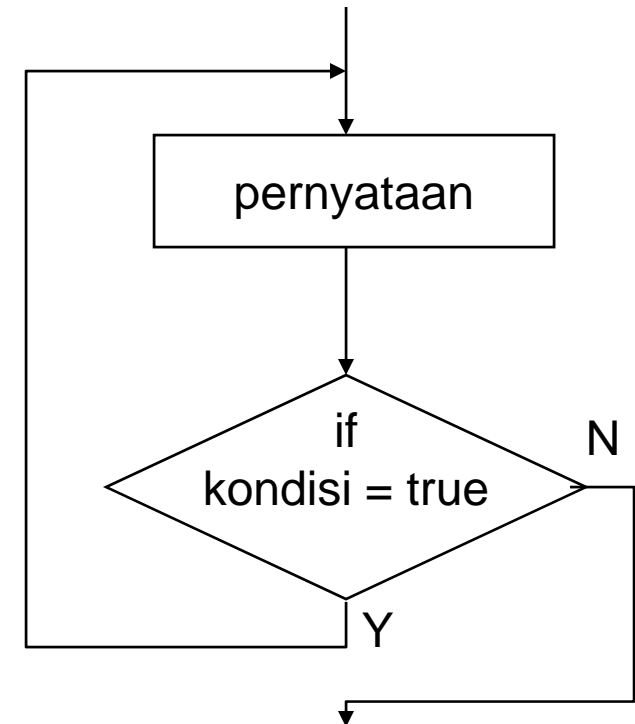
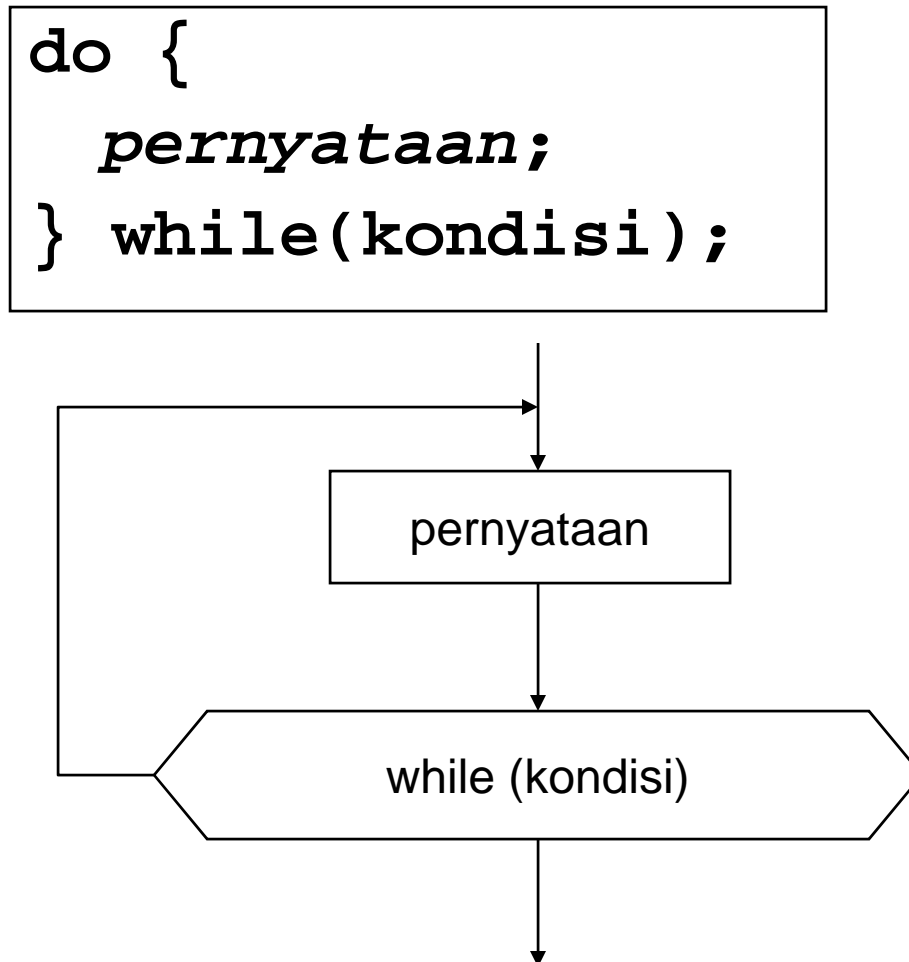


Perulangan *do while*

- Pernyataan *do while* memungkinkan perulangan berlanjut selama kondisi dalam *while* masih bernilai TRUE (non-zero).
- Formatnya :

```
do {  
    pernyataan;  
} while(kondisi);
```
- The loop is executed at least once.

Diagram Alir Perulangan *do while*



Latihan

1. Gunakan loop *for* untuk menampilkan nilai 1 sampai dengan 20 dalam baris-baris yang terpisah.
2. Hitunglah bilangan triangular dari masukan pengguna, yang dibaca dari keyboard dengan menggunakan *scanf()*. Bilangan triangular adalah penjumlahan dari bilangan masukan dengan seluruh bilangan sebelumnya, sehingga bilangan triangular dari 7 adalah : $7 + 6 + 5 + 4 + 3 + 2 + 1$
input : Masukkan sebuah bilangan : 7
output : Bilangan triangular 7 adalah 28
3. Gunakan loop *for* untuk menampilkan seluruh karakter dari A sampai dengan Z dalam baris-baris yang terpisah.
4. Gunakan loop *for* dengan kenaikan variabel negatif untuk menampilkan seluruh karakter dari Z sampai dengan A dalam baris-baris yang terpisah.

Latihan

5. Gunakan loop *for* untuk membuat program sebagai berikut:

input : n

output : 1 3 5 7 ... m (m = bilangan ganjil ke n)

6. Gunakan loop *for* untuk membuat program sebagai berikut:

input : n

output : 1 -2 3 -4 5 -6 7 -8 ... n

7. Gunakan loop *for* untuk membuat program sebagai berikut:

input : n

output : 1*2*3*4*5*... *n (faktorial)

Latihan

8. Gunakan loop *while* untuk membuat program yang dapat mencari total angka yang dimasukkan dengan tampilan sebagai berikut :

Masukkan bilangan ke-1 : 5

Mau memasukkan data lagi [y/t] ? y

Masukkan bilangan ke-2 : 3

Mau memasukkan data lagi [y/t] ? t

Total bilangan = 8

9. Pada program no 8 tambahkan penghitungan rata-rata, maksimum dan minimum dari angka yang dimasukkan.

Latihan

10. Gunakan *while* pada program yang digunakan untuk menghitung banyaknya karakter dari kalimat yang dimasukkan melalui keyboard (termasuk karakter spasi). Untuk mengakhiri pemasukan kalimat, tombol ENTER ('`\n`') harus ditekan

Input : Ketikkan sembarang kalimat, akhiri dengan enter

Output : jumlah karakter = m
jumlah spasi = n