



Bab 6. Looping 2 (Pengulangan Proses)

Konsep Pemrograman
Politeknik Elektronika Negeri Surabaya
2017



Overview

- Pernyataan `break`
- Pernyataan `continue`
- *Nested loop* (loop di dalam loop yang lain)
- Pernyataan `exit`



Kondisional : *break*

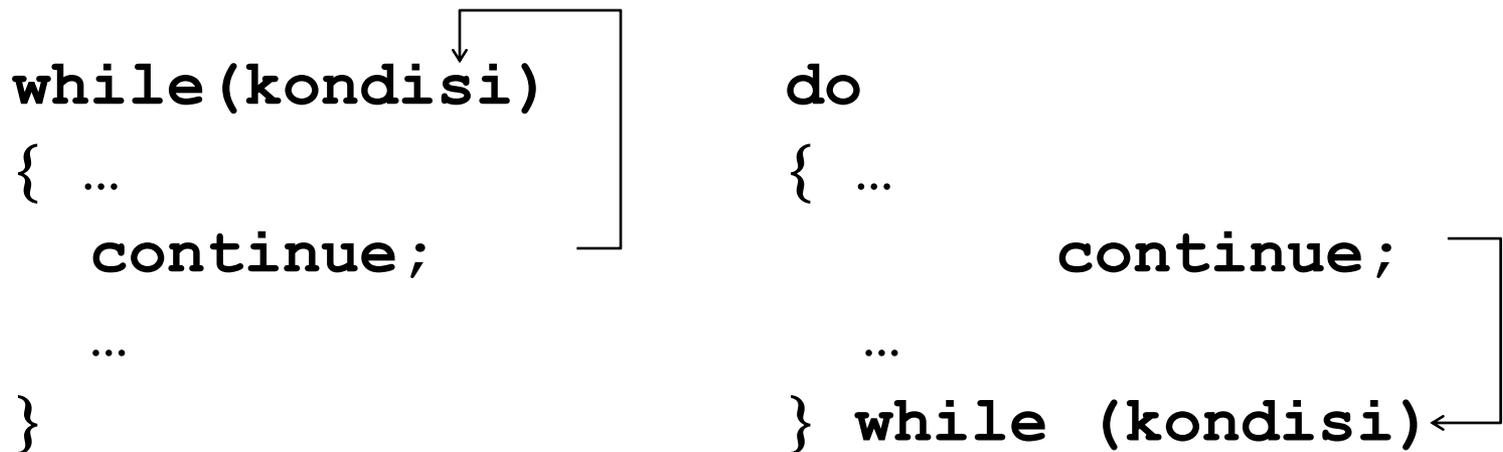
- Pada *switch-case*, *break* digunakan untuk menuju ke akhir (keluar dari) struktur *switch*.
- Dalam looping, pernyataan ini berfungsi untuk keluar secara ‘paksa’ dari *loop for*, *do-while* dan *while*.
- Jika pernyataan *break* berada dalam loop yang bertingkat (*nested loop*), maka pernyataan *break* hanya akan membuat proses keluar dari loop yang bersangkutan (tempat *break* dituliskan), bukan keluar dari semua loop.

Kondisional : *continue*

- Digunakan untuk mengarahkan eksekusi ke iterasi (proses) berikutnya pada *loop* yang sama (*skip the current iteration, continue to the next iteration*)
- Pada do-while dan while, *continue* menyebabkan eksekusi menuju ke kondisi pengujian pengulangan sbb:

```
while (kondisi)
{
    ...
    continue;
    ...
}

do
{
    ...
    continue;
    ...
} while (kondisi)
```





Kondisional : *continue*

- Pada loop for, *continue* menyebabkan bagian *control loop* (ekspresi3) dikerjakan dan kondisi untuk keluar dari loop for (ekspresi2) diuji kembali.

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



Nested loop

- Dalam suatu *loop* bisa terkandung *loop* yang lain.
- *Loop* yang terletak di dalam *loop* biasa disebut dengan *loop* di dalam *loop* (*nested loop*).
- Contoh :

```
for (eksp1; eksp2; eksp3)  
    for (eksp1; eksp2; eksp3)  
        pernyataan;
```



exit () : Menghentikan Eksekusi Program

- Jika di dalam suatu eksekusi terdapat suatu kondisi yang tak dikehendaki, maka eksekusi program dapat dihentikan (secara normal) melalui pemanggilan fungsi *exit*().
- Prototipe dari fungsi *exit*() didefinisikan pada file **stdlib.h**, yang memiliki deklarasi sebagai berikut :

```
void exit(int status) ;
```
- Menurut kebiasaan, nilai nol diberikan pada argumen *exit*() untuk menunjukkan penghentian program yang normal → **exit(0) ;**



Latihan

1. Dengan menggunakan pernyataan break, buatlah program yang dapat menampilkan semua tampilan karakter yang diketikkan dan program berakhir ketika ditekan tombol Enter.
 2. Dengan menggunakan pernyataan continue, buatlah program yang dapat menampilkan bilangan ganjil dari 1 sampai $< n$ (n diinputkan), kecuali bilangan ganjil tersebut kelipatan 3.
 3. Dengan menggunakan pernyataan break dan continue, buatlah program untuk membuat program dengan input n , dan output, bilangan ganjil kecuali kelipatan 7 dan 11 mulai dari 1 sampai $< n$ atau bilangan tersebut < 100
 4. Buatlah program untuk menerima daftar nilai mahasiswa sbb :
 - Input : - Jumlah data (n)
- Nilai ke-1 s/d Nilai ke- n
 - Output : - Nilai minimal
- Nilai maksimal
- Nilai rata-rata (rata-rata = nilai total / jumlah data)
- (Petunjuk : Gunakan loop *for* dan seleksi kondisi dengan *if*)



Latihan

5. Dengan menggunakan pernyataan *nested loop*, buatlah program berikut:

input: n

output:

1 2 3 4 5 ... n

1 2 3 4 5 ... n

1 2 3 4 5 ... n

n kali

.....

1 2 3 4 5 ... n

6. Dengan menggunakan pernyataan *nested loop*, buatlah program berikut:

input: n

output:

1

2 2

3 3 3

n kali

.....

n n n n n ... n



Latihan

7. Dengan menggunakan pernyataan *nested loop*, buatlah program berikut:
input: n
output: 2 3 5 7 11.... Bilangan prima ke n

8. Dengan menggunakan pernyataan *nested loop*, buatlah program berikut:
input: n
output: 0 1 3 6 10 15 21 28 Bilangan ke n

9. Pada akhir setiap 2 buah program diatas tambahkan pertanyaan “apakah anda ingin keluar (y/t)?”, pertanyaan tersebut hanya bisa di jawab dengan huruf ‘y’ (y kecil) dan ‘t’(t kecil). Dan akan keluar dari program setelah dijawab dengan ‘y’ (y kecil)