

# PRAKTIKUM 11

---

## Looping : **break**, **Continue**, **Nested loop**

---

### A. TUJUAN PEMBELAJARAN

1. Menjelaskan penggunaan pernyataan *break*
2. Menjelaskan penggunaan pernyataan *continue*
3. Menjelaskan penggunaan pernyataan *goto*
4. Menjelaskan penggunaan *exit()* untuk menghentikan eksekusi program
5. Mempraktekkan perulangan di dalam perulangan.
6. Memecahkan persoalan dengan perulangan didalam perulangan

### B. DASAR TEORI

Untuk keluar dari perulangan/loop dapat dilakukan setelah syarat perulangan dipenuhi. Akan tetapi ada cara lain untuk keluar dari perulangan walaupun syarat belum terpenuhi, yaitu dengan menggunakan pernyataan *break*. Pernyataan *break* sesungguhnya telah diperkenalkan pada pernyataan *switch*. Pernyataan ini berfungsi untuk keluar dari *loop for*, *do-while* dan *while*. Sedangkan pada *switch* yaitu untuk menuju ke akhir (keluar dari) struktur *switch*. Pernyataan *break* digunakan untuk keluar dari satu blok

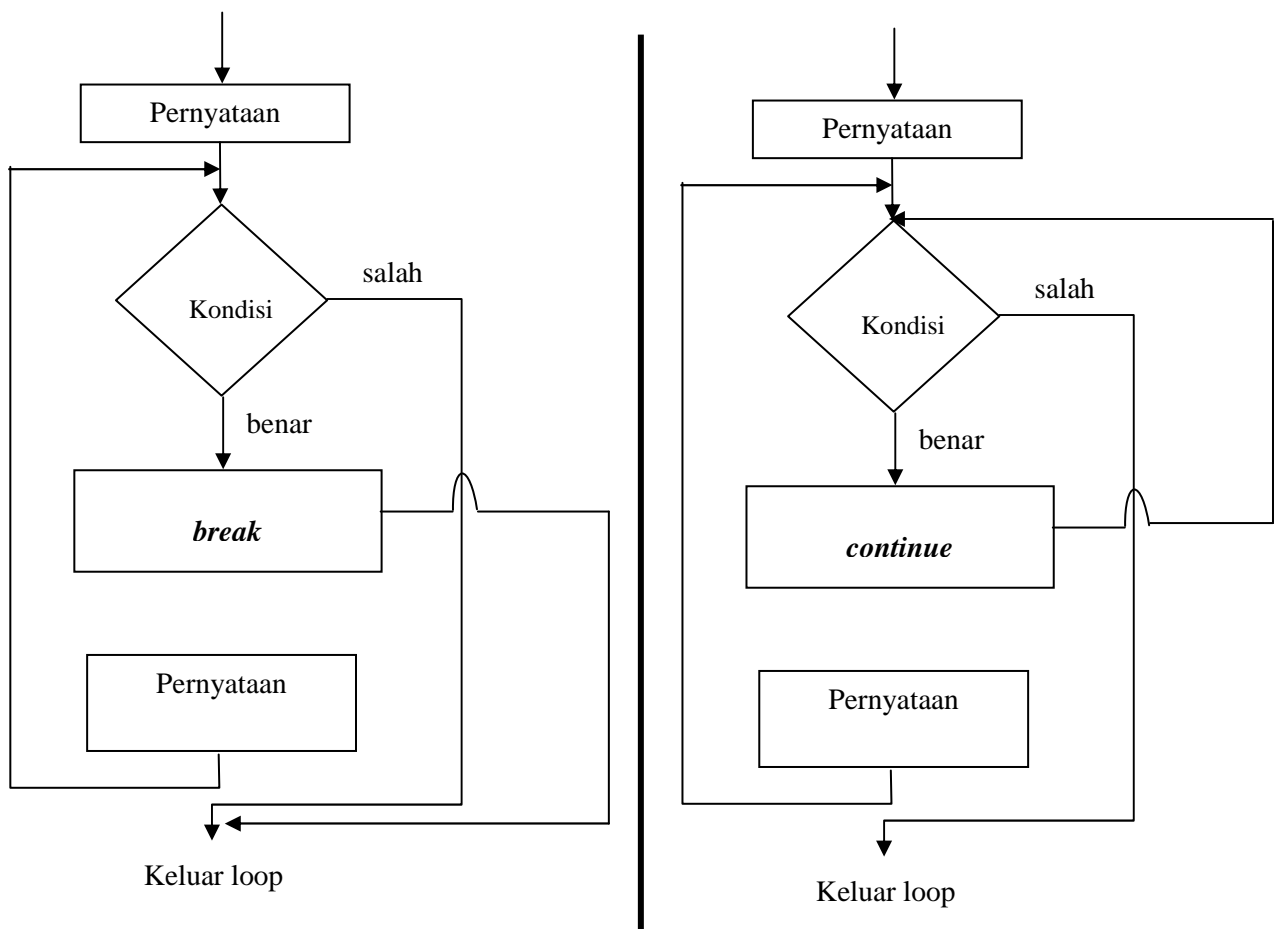
Selain pernyataan *break* ada juga pernyataan yang berhubungan dengan perulangan yaitu pernyataan *continue*. Pernyataan *continue* digunakan untuk mengarahkan eksekusi ke iterasi (proses) berikutnya pada *loop* yang sama. Pada *do-while* dan *while*, pernyataan *continue* menyebabkan eksekusi menuju ke kondisi pengujian pengulangan. Pada *loop for*, pernyataan *continue* menyebabkan bagian penarik variabel pengendali *loop* dikerjakan dan kondisi untuk keluar dari *loop for* diuji kembali. Ilustrasi flowchart yang menunjukkan perbedaan *break* dan *continue* ditunjukkan pada gambar 11.1.

Selain dua pernyataan diatas masih ada pernyataan yang berhubungan dengan loop yaitu goto. Pernyataan goto merupakan intruksi untuk mengarahkan eksekusi ke pernyataan yang diawali dengan suatu label. Label sendiri berupa suatu pengenalan (identifier) yang diikuti dengan tanda titik dua (:)

Contoh pemakaian goto:

```
goto cetak;
```

Mengisyaratkan agar eksekusi dilanjutkan ke pernyataan yang diawali dengan label **cetak** : . Pernyataan ini sebaiknya tidak digunakan.



Gambar 11. 1 Perbedaan flowchart untuk Perintan break dan continue

Untuk menghentikan program tanpa harus menyelesaikan sampai akhir pernyataan, yaitu melalui pemanggilan fungsi `exit()`. Hal ini biasa dilakukan, jika di dalam suatu eksekusi terdapat suatu kondisi yang tak dikehendaki. Prototipe dari fungsi `exit()` didefinisikan pada file `stdlib.h`, yang memiliki deklarasi sebagai berikut :

```
void exit(int status);
```

status biasanya diberi nilai 0 untuk pernghentian program secara normal.

### Loop di dalam Loop

Dalam suatu loop bisa terkandung loop yang lain, sehingga jika loop tersebut diulangi n kali dan loop yang didalamnya dijuga dilakukan m kali maka pernyataan yang berada didalam nested loop sebanyak n x n. Loop yang terletak di dalam loop biasa disebut dengan loop di dalam loop (nested loop). Salah satu contoh nested loop misalnya pada permasalahan untuk membuat tabel perkalian:

	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	2	4	6	8	10	12	14	16
3	3	6	9	12	15	18	21	24
4	4	8	12	16	20	24	28	32
5	5	10	15	20	25	30	35	40
6	6	12	18	24	30	36	42	48
7	7	14	21	28	35	42	49	56
8	8	16	24	32	40	48	56	64

Implementasi dalam program selengkapnya adalah sebagai berikut :

```
for (baris = 1; baris <= MAKS; baris++) {
    for (kolom = 1; kolom <= MAKS; kolom++) {
        hasil_kali = baris * kolom;
        printf ("%2d", hasil_kali);
    }
    printf("\n");    /* pindah baris */
}
```

Bagian yang terletak dalam bingkai di depan dapat dapat diperoleh melalui

```
for (baris = 1; baris <= MAKS; baris++)
{
    hasil_kali = baris * kolom;
    printf ("%2d", hasil_kali);
}
```

dengan MAKS didefinisikan bernilai 8. Bagian loop yang terdalam :

```

for (kolom = 1; kolom <= MAKS; kolom++) {
    hasil_kali = baris * kolom;
    printf ("%2d", hasil_kali);
}

```

digunakan untuk mencetak suatu deret hasil perkalian dalam satu baris. Untuk berpindah ke baris berikutnya, pernyataan yang digunakan yaitu

```
printf("\n");
```

Adapun pencetakan untuk semua baris dikendalikan melalui

```
for (baris = 1; baris <= MAKS; baris++)
```

Pernyataan di atas mempunyai arti “dari baris ke-1 sampai dengan baris ke-MAKS”.

### C. TUGAS PENDAHULUAN

Buatlah desain flowchart untuk setiap soal dalam percobaan

### D. PERCOBAAN

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.

Contoh Input : 10

Output : 1 5 7

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  $< 100$

Contoh Input : 20

Output : 1 3 5 9 13 15 17 19

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*)

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  
.....  
1 2 3 4 5 ... n
```

} n kali

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

input: n

output:

```
1  
2 2  
3 3 3  
.....  
n n n n n ... n
```

} n kali

7. Pada akhir setiap 4 buah program diatas tambahkan tanya “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)

## E. LAPORAN RESMI

1. Untuk setiap listing program dari percobaan-percobaan di atas, ambil *capture* outputnya.
2. Tuliskan kesimpulan dari percobaan yang telah anda lakukan.
3. Dengan menggunakan pernyataan *nested loop*, buatlah program berikut:

input: n

output:

2 3 5 7 11.... Bilangan prima ke n