

looping

* For

Digunakan apabila user sudah mengetahui jumlah Perulangan yang ingin dilakukan.

Bentuk umum:

```
for (ekspresi1; ekspresi2; ekspresi3) {
    Statement;
}
```

}

⇒ ketiga ekspresi dipisahkan titik koma

⇒ Apabila Statement hanya 1 baris tidak perlu menggunakan kurawal

ekspresi 1 = Inisialisasi

ekspresi 2 = Kondisi

ekspresi 3 = Pengendali

* While

Pengecekan kondisi dilakukan diawal dan Statement didalam loop akan dijalankan selama kondisi terpenuhi.

Bentuk Umum:

```
while (kondisi) {
```

```
    Statement;
}
```

}

⇒ Untuk menghentikan loop perlu ada Statement yg bisa merubah kondisi

Note: Ada kemungkinan loop tidak dijalankan sama sekali

* Do-While

Pengecekan dilakukan diakhir dan Statement didalam loop akan dijalankan selama kondisi terpenuhi.

Bentuk umum:

```
do {
```

```
    Statement
}
```

```
while (condition)
```

Note: loop akan dijalankan minimal 1 kali

* Nested Loop

Nested loop adalah loop yang memiliki loop lain didalamnya, pada nested loop, loop luar disebut outer loop sedangkan loop yang didalam disebut innerloop.

Contoh:

```
do {
```

```
    Statement;
```

```
    if ( ekspresi ) { Statement; }
```

```
} while ( kondisi );
```

* Break

Dalam looping, Statement ini berfungsi untuk keluar secara paksa dari loop for, do-while, dan while apabila kondisi terpenuhi. Sedangkan dalam switch-case, break digunakan untuk keluar dari struktur switch-case.

Contoh:

Note: Jika break pada nested loop maka hanya keluar dari loop yg bersangkutan

```
for (i=10; i > 30; i++) {
```

```
    if (i % 2 == 0 {
```

```
        printf (" Bilangan Genap ");
```

```
        break break;
```

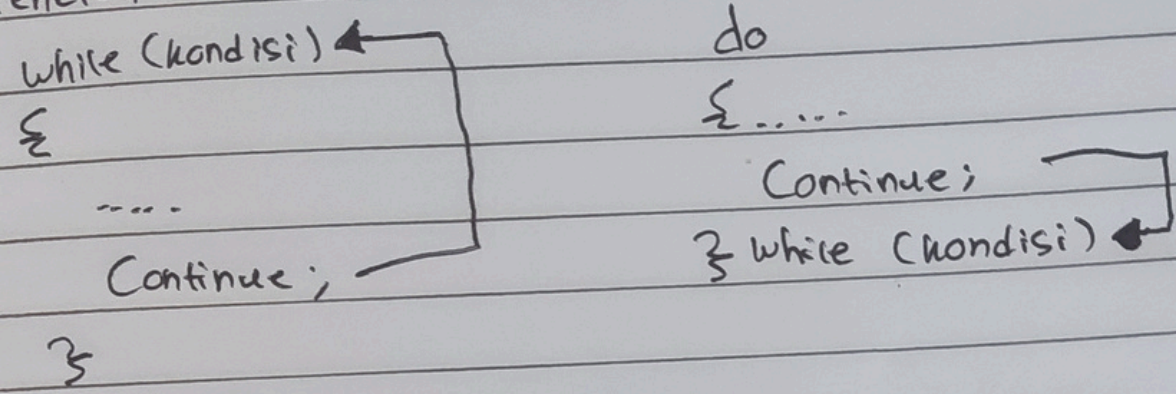
```
    }
```

```
}
```

* Continue

Digunakan untuk mengarahkan eksekusi ke iterasi berikutnya pada loop yang sama. Pada loop do-while dan while, Continue menyebabkan eksekusi menuju pengecekan Continue Condition.

Penerapan:



Contoh:

```
for (i=11; i<=20; i++) {  
    if (i%3 == 0) Continue;  
    .....  
}
```

Note: Jika `continue` berada pada nested loop maka hanya mempengaruhi loop bersangkutan

* Exit

Jika didalam suatu eksekusi terdapat Suatu kondisi yang tidak dikehendaki maka dapat dihentikan dgn normal melalui `exit()`.

Prototipe dari fungsi `exit()` didefinisikan pada file `stdlib.h` yang memiliki deklarasi = `void exit(int status);`

Menurut kebiasaan, nilai nol diberikan pada argumen `exit()` menunjukan penghentian program yang normal
➡ `exit(0)`