

**IF34348 - PEMROGRAMAN LANJUT**

**PERULANGAN**

**05**

Oleh : Andri Heryandi, M.T.

# JENIS-JENIS PERULANGAN

IF34348 - Pemrograman Lanjut

- Perulangan FOR
- Perulangan DO WHILE
- Perulangan WHILE



Oleh : Andri Heryandi, M.T.

# PERULANGAN FOR

IF34348 - Pemrograman Lanjut

- Perulangan for adalah perulangan yang dilakukan ketika banyaknya perulangan telah ditentukan/diketahui.



Oleh : Andri Heryandi, M.T.

# PERULANGAN FOR

IF34348 - Pemrograman Lanjut

## ■ Sintak perulangan for adalah

```
for(inisialisasi; ekspresi boolean; statement update)
{
    badan-perulangan;
}
```

## ■ Keterangan :

- Bagian inisialisasi adalah bagian yang hanya dilakukan di awal perulangan dan hanya akan dieksekusi 1 kali. Bagian ini biasanya digunakan untuk inisialisasi counter. Bagian ini boleh tidak diisi.
- Bagian Ekspresi Boolean harus berisi ekspresi yang bernilai TRUE atau FALSE. Jika bagian ini bernilai TRUE, maka badan-perulangan akan diulang. Jika bagian ini bernilai FALSE maka perulangan tidak dilakukan.
- Bagian Statement Update akan dieksekusi setelah statement yang akan diulang telah dieksekusi. Bagian ini biasanya digunakan untuk menambah/mengurangi (mengubah) counter perulangan.
- Bagian badan-perulangan adalah block-code yang akan diulang. Jika lebih dari 1 statement, maka harus diapit dengan { }.



# CONTOH PERULANGAN FOR

IF34348 - Pemrograman Lanjut

```
public class TestFor {  
    public static void main(String[] args) {  
        int i;  
        for(i=1;i<=10;i++){  
            System.out.println("Perulangan ke-"+i);  
        }  
    }  
}
```

```
Perulangan ke-1  
Perulangan ke-2  
Perulangan ke-3  
Perulangan ke-4  
Perulangan ke-5  
Perulangan ke-6  
Perulangan ke-7  
Perulangan ke-8  
Perulangan ke-9  
Perulangan ke-10
```



Oleh : Andri Heryandi, M.T.

# CONTOH PERULANGAN FOR

IF34348 - Pemrograman Lanjut

```
public class TestFor {  
    public static void main(String[] args) {  
        int i;  
        for(i=10;i>=1;i--){  
            System.out.println("Perulangan ke-"+i);  
        }  
    }  
}
```

```
Perulangan ke-10  
Perulangan ke-9  
Perulangan ke-8  
Perulangan ke-7  
Perulangan ke-6  
Perulangan ke-5  
Perulangan ke-4  
Perulangan ke-3  
Perulangan ke-2  
Perulangan ke-1
```



Oleh : Andri Heryandi, M.T.

# PERULANGAN WHILE

IF34348 - Pemrograman Lanjut

- Perulangan while akan mengulang bagian badan perulangan selama kondisi perulangan (ekspresi boolean) bernilai true. Ketika kondisi perulangan bernilai false, maka eksekusi program akan melanjutkan ke statement setelah while.
- Jika badan perulangan terdiri dari 2 pernyataan atau lebih, maka apitlah badan perulangan dengan { }.
- Sintak perulangan while adalah :

```
while(kondisi-perulangan)
{
    badan-perulangan;
}
```



Oleh : Andri Heryandi, M.T.

# CONTOH PERULANGAN WHILE

IF34348 - Pemrograman Lanjut

```
public class TestWhile {  
    public static void main(String[] args) {  
        int i;  
        i=1;  
        while (i<=10) {  
            System.out.println("Perulangan ke-"+i);  
            i++;  
        }  
    }  
}
```

```
Perulangan ke-1  
Perulangan ke-2  
Perulangan ke-3  
Perulangan ke-4  
Perulangan ke-5  
Perulangan ke-6  
Perulangan ke-7  
Perulangan ke-8  
Perulangan ke-9  
Perulangan ke-10
```



Oleh : Andri Heryandi, M.T.



# PERULANGAN DO WHILE

IF34348 - Pemrograman Lanjut

- Perulangan do while akan mengulang bagian badan perulangan selama kondisi perulangan (ekspresi boolean) bernilai true.
- Perbedaan do while dengan while terletak pada posisi pemeriksaan kondisi perulangan. Kondisi perulangan pada do while diletakkan setelah badan perulangan. Dengan ini di**PASTI**kan badan perulangan minimal dieksekusi 1 kali.
- Sintak perulangan while adalah :

```
do {  
    badan-perulangan;  
} while (kondisi-perulangan)
```



# CONTOH PERULANGAN DO WHILE

IF34348 - Pemrograman Lanjut

```
public class TestDoWhile {  
    public static void main(String[] args) {  
        int i;  
        i=1;  
        do{  
            System.out.println("Perulangan ke-"+i);  
            i++;  
        }while(i<=10);  
    }  
}
```

```
Perulangan ke-1  
Perulangan ke-2  
Perulangan ke-3  
Perulangan ke-4  
Perulangan ke-5  
Perulangan ke-6  
Perulangan ke-7  
Perulangan ke-8  
Perulangan ke-9  
Perulangan ke-10
```



Oleh : Andri Heryandi, M.T.

# PERNYATAAN BREAK

IF34348 - Pemrograman Lanjut

- Pernyataan BREAK digunakan jika anda ingin keluar dari perulangan.
- BREAK ditulis di dalam badan perulangan
- Jika BREAK digunakan dalam perulangan-dalam-perulangan, maka BREAK akan berlaku untuk perulangan paling dalam.



# CONTOH PENGGUNAAN BREAK

IF34348 - Pemrograman Lanjut

```
public class TestBreak {  
    public static void main(String[] args) {  
        int i;  
        for(i=1;i<=20;i++){  
            System.out.println(i);  
            if(i==10)  
                break;  
        }  
    }  
}
```

Perulangan di atas digunakan untuk menampilkan angka dari 1 sampai 20. Tetapi di dalam perulangan ada sebuah kondisi dimana perulangan akan berhenti ketika `i == 10`.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10



Oleh : Andri Heryandi, M.T.

# CONTOH PENGGUNAAN BREAK

IF34348 - Pemrograman Lanjut

```
public class TestBreak2 {  
    public static void main(String[] args) {  
        int i,j;  
        for(i=1;i<=5;i++){  
            for(j=1;j<=50;j++){  
                System.out.println("I : "+i+", "+  
                                   "J : "+j);  
  
                if(j==2)  
                    break;  
            }  
        }  
    }  
}
```

Walau pun perulangan J akan mengulang ketika J =1 sampai 50, tapi di dalam perulangan ada sebuah kondisi dimana ketika J==2 maka perulangan J akan dihentikan (break)

```
I : 1,J : 1  
I : 1,J : 2  
I : 2,J : 1  
I : 2,J : 2  
I : 3,J : 1  
I : 3,J : 2  
I : 4,J : 1  
I : 4,J : 2  
I : 5,J : 1  
I : 5,J : 2
```



# PERNYATAAN CONTINUE

IF34348 - Pemrograman Lanjut

- Perintah **CONTINUE** digunakan di dalam badan perulangan untuk mengabaikan perintah dibawahnya dan melakukan perulangan berikutnya.
- Contoh kasus :
  - Tampilkan angka 1 sampai dengan 10, tapi jangan tampilkan yang genap.



# CONTOH PENGGUNAAN CONTINUE

IF34348 - Pemrograman Lanjut

```
public class TestContinue {  
    public static void main(String[] args) {  
        int i,j;  
        for(i=1;i<=10;i++){  
            if(i%2==0)  
                continue;  
            System.out.println(i);  
        }  
    }  
}
```

Perulangan di atas akan menampilkan angka 1 sampai 10. Tetapi jika 1 kelipatan 2 maka perulangan dilanjutkan tanpa mengeksekusi perintah di bawahnya

1  
3  
5  
7  
9



# LATIHAN 1

IF34348 - Pemrograman Lanjut

## ■ Perhitungan Saldo Tabungan

### ■ Input

- Saldo Awal : Rp. 1.000.000
- Bunga/Bulan (%) : 10
- Lama (bulan) : 5

### ■ Output

|                     |   |     |           |
|---------------------|---|-----|-----------|
| Saldo di bulan ke-1 | : | Rp. | 1.100.000 |
| Saldo di bulan ke-2 | : | Rp. | 1.210.000 |
| Saldo di bulan ke-3 | : | Rp. | 1.331.000 |
| Saldo di bulan ke-4 | : | Rp. | 1.464.100 |
| Saldo di bulan ke-5 | : | Rp. | 1.610.510 |





# LATIHAN 2

IF34348 - Pemrograman Lanjut

■ Perhitungan lama tabungan sampai mencapai saldo tertentu.

■ Input

- Saldo Awal : Rp. 1.000.000
- Bunga/Bulan (%) : 10
- Saldo target : Rp. 2.000.000

■ Output

|                     |   |     |           |
|---------------------|---|-----|-----------|
| Saldo di bulan ke-1 | : | Rp. | 1.100.000 |
| Saldo di bulan ke-2 | : | Rp. | 1.210.000 |
| Saldo di bulan ke-3 | : | Rp. | 1.331.000 |
| Saldo di bulan ke-4 | : | Rp. | 1.464.100 |
| Saldo di bulan ke-5 | : | Rp. | 1.610.510 |
| Saldo di bulan ke-6 | : | Rp. | 1.771.561 |
| Saldo di bulan ke-7 | : | Rp. | 1.948.717 |
| Saldo di bulan ke-8 | : | Rp. | 2.143.589 |

