

# Pemrograman Java

Yoannita



- **Pengulangan(i) (FOR)**
- **Loncatan**

"The whole art of teaching is only the art of awakening the natural curiosity of young minds for the purpose of satisfying it afterwards."  
(Anatole France)

# Pengulangan: For

- Pernyataan ini akan melaksanakan satu blok pernyataan secara berulang-ulang sampai batas tertentu yang ditentukan

```
for (inisialisasi ; terminasi ; step) {  
    statement1;  
    statement2;  
}
```

- ***Inisialisasi*** – inisialisasi dari variabel loop.
- ***Terminasi**, atau juga **LoopCondition*** - membandingkan variabel loop pada nilai batas tertentu, merupakan kondisi yang harus dipenuhi agar loop dapat terjadi
- ***Step*** - melakukan update pada variabel loop (dapat berupa *increment* atau *decrement*).

# Pengulangan: For

- Flowchart →

- Contoh perbedaan pengulangan for pada pascal dan java :

**Pascal:**

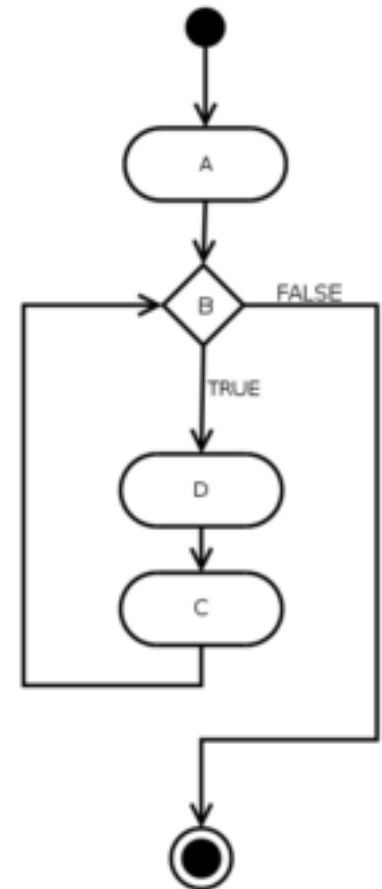
```
int p;  
for p :=1 to 5 do  
  begin  
    write(p);  
  end;
```

**Java :**

```
int p;  
for (p = 1; p <=5; p++)  
{  
  System.out.print(p);  
}
```

Hasil :  
12345

for(A;B;C)  
D;



# Pengulangan: For

## Contoh Program (i)

```
class ulangFor{
    public static void main (String [ ] args) {
        int a;
        for (a = 0; a < 10; a++) {
            System.out.println (a);
        }
        //atau
        for (int a = 0; a < 10; a++) {
            System.out.println (a);
        }
    }
}
```



0  
1  
2  
3  
4  
5  
6  
7  
8  
9

# Pengulangan: For

## Contoh Program (i)

```
class ulangFor{
    public static void main (String [ ] args) {
        for (char x = 'a'; x <= 'e'; x++)
        {
            System.out.print(x);
        }
    }
}
```



abcde

# Loncatan

## **BREAK**

- digunakan untuk menghentikan eksekusi dari suatu blok dan keluar dari blok yang sedang dieksekusi tersebut.
- sering digunakan dalam blok perulangan (`while`, `do...while`) dan blok kondisi (`switch...case`).

## **CONTINUE**

- sering digunakan dalam perulangan, berfungsi untuk melanjutkan perulangan tanpa mengeksekusi baris-baris pernyataan setelah pernyataan `continue`. Pemakaian pernyataan `continue` biasanya bersamaan dengan pernyataan kondisi (`if`)

# Contoh Penggunaan Break

```
// nama file : break1.java
class break1{
    public static void main(String[] args){
        char na = 'A';
        switch (na){
            case 'A' : System.out.println("Memuaskan");
                break;
            case 'B' : System.out.println("Baik");
                break;
            case 'D' : System.out.println("Cukup");
                break;
            case 'E' : System.out.println("Jelek");
                break;
            default :
                System.out.println("Nilai tidak ada");
        }
    }
}
```

```
// nama file : break2.java
class break2 {
    public static void main(String[] args){
        for (int a=5; a>0; a--){
            System.out.println("a = " + a );
            if (a==3)
                break;
        }
    }
}
```

berikut hasil dari program di atas.

a = 5

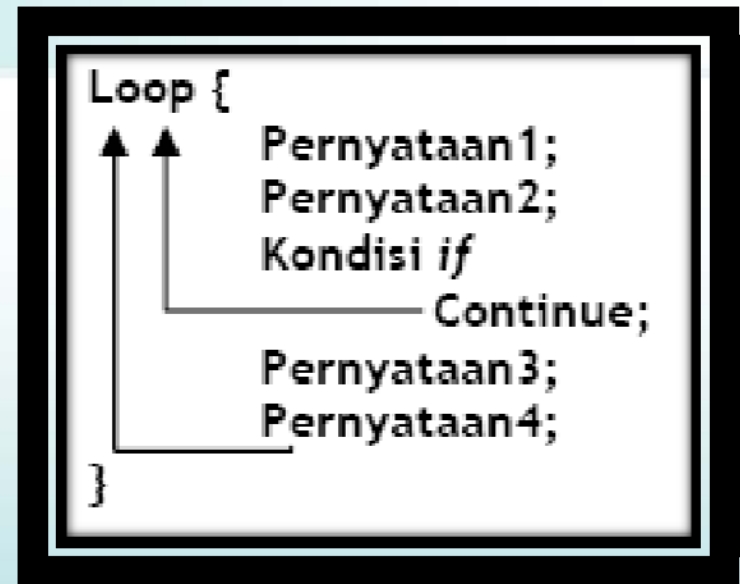
a = 4

a = 3

# Contoh Penggunaan Continue

```
// nama file : continue1.java
```

```
class continue1{  
    public static void main(String[] args) {  
        for (int a=0 ; a < 10 ; a++){  
            if (a % 3==0) {  
                continue;  
            }  
            System.out.println("a = " + a );  
        }  
    }  
}
```



berikut hasil dari program

a = 1

a = 2

a = 4

**a = 5**

**a = 7**

a = 8

