



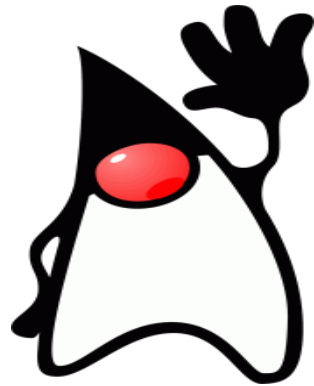
# **LPBO PBO**

## **PERTEMUAN 6**

Dosen :

Bella Hardiyana S. Kom, M. Kom

---



# **BAB V**

## **PERULANGAN**

# Konsep Pengulangan

- **Inisialisasi** adalah pemberian nilai awal kepada suatu variabel yang akan digunakan sebagai *counter* (pencacah) sebelum masuk ke blok pengulangan.
- **Kondisi** adalah syarat yang harus dipenuhi agar pengulangan bisa berlangsung.
- **Counter** adalah memberikan perubahan nilai pada variabel yang menjadi pencacah nilai.

# Pengulangan Menggunakan for

Pengulangan for digunakan jika pengulangan bersifat pasti.

```
// Apabila hanya memiliki 1 aksi
for(inisialisasi; kondisi; counter)
    Aksi; // Ini adalah blok Aksi yang akan diulang

// Apabila memiliki 2 atau lebih aksi
for(inisialisasi; kondisi; counter){
    Aksi1; // -----
    Aksi2; // Ini adalah blok Aksi yang akan diulang
    AksiN; // -----
}
```

# Pengulangan Menggunakan for (lanj.)

```
public class ContohLoopFor1 {  
    public static void main(String[] args){  
        int i;  
        for(i=1; i<=10; i++)  
            System.out.println("Halo-Halo Bandung");  
    }  
}
```

## TAMPILAN OUTPUT

```
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung
```

## OUTPUT

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

*counter increment*  
**(i++ atau i=i+1 atau i+=1)**

```
public class ContohLoopFor2 {  
    public static void main(String[] args){  
        int i;  
        for(i=1; i<=10; i++)  
            System.out.println(i);  
    }  
}
```

# Pengulangan Menggunakan for (lanj.)

```
public class ContohLoopFor1 {  
    public static void main(String[] args){  
        int i;  
        for(i=10; i>=1; i--)  
            System.out.println("Halo-Halo Bandung");  
    }  
}
```

## TAMPILAN OUTPUT

```
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung  
Halo-Halo Bandung
```

## OUTPUT

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

*counter decrement*  
**(i-- atau i=i-1 atau i-=1)**

```
public class ContohLoopFor2 {  
    public static void main(String[] args){  
        int i;  
        for(i=10; i>=10; i--)  
            System.out.println(i);  
    }  
}
```

# Pengulangan Menggunakan for (lanj.)

## Catatan Penting

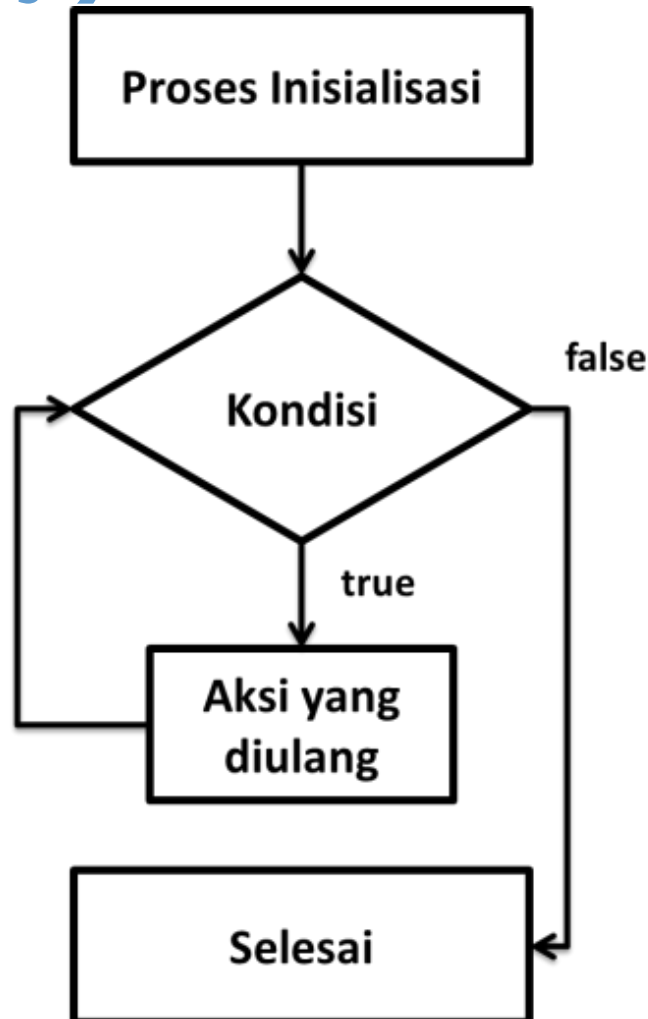
1. Nama variabel tidak harus selalu **i** ataupun **x**
2. Ciri dari pengulangan **counter increment** menggunakan perbandingan (**<=** atau **<**)
3. Ciri dari pengulangan **counter decrement** menggunakan perbandingan (**>=** atau **>**)
4. Semua poin **berlaku** juga untuk pengulangan **while** dan **do-while**

# Pengulangan Menggunakan **while**

Pengulangan ini akan mengecek kondisi terlebih dahulu. Jika kondisi bernilai *true*, maka melakukan aksi berupa pengulangan.

```
// Bentuk umum pengulangan while terdiri dari beberapa sintaks
// sehingga wajib diapit oleh kurung kurawal.
inisialisasi;
while(kondisi) {
    Aksi1; // -----
    Aksi2; // Ini adalah blok Aksi yang akan diulang
    AksiN; // -----
    counter;
}
```

# Pengulangan Menggunakan while (lanj.)



# Pengulangan Menggunakan while (lanj.)

*counter increment*

**(i++ atau i=i+1 atau i+=1)**

```
public class ContohLoopWhile2 {  
    public static void main(String[] args){  
        int i;  
        i=1;  
        while(i<=10){  
            System.out.println(i);  
            i++;  
        }  
    }  
}
```

**OUTPUT**

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

*counter decrement*

**(i-- atau i=i-1 atau i-=1)**

```
public class ContohLoopWhile4 {  
    public static void main(String[] args){  
        int i;  
        i=10;  
        while(i>=1){  
            System.out.println(i);  
            i--;  
        }  
    }  
}
```

**OUTPUT**

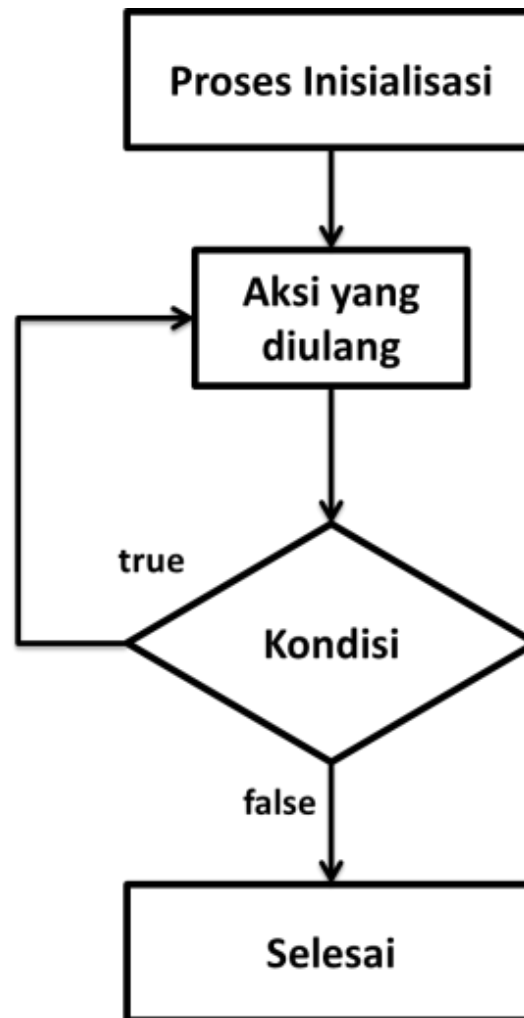
10  
9  
8  
7  
6  
5  
4  
3  
2  
1

# Pengulangan Menggunakan do-while

Melakukan aksi terlebih dahulu, setelah itu baru melakukan pengecekan kondisi

```
// Bentuk umum pengulangan do-while terdiri dari beberapa
// sintaks, sehingga wajib diapit oleh kurung kurawal.
inisialisasi;
do{
    Aksi1; // -----
    Aksi2; // Ini adalah blok Aksi yang akan diulang
    AksiN; // -----
    counter;
}while(kondisi) ;
```

# Pengulangan Menggunakan do-while (lanj.)



# Pengulangan Menggunakan do-do-while (lanj.)

*counter increment*

**(i++ atau i=i+1 atau i+=1)**

```
public class ContohLoopDoWhile2 {  
    public static void main(String[] args){  
        int i;  
        i=1;  
        do{  
            System.out.println(i);  
            i++;  
        }while(i<=10);  
    }  
}
```

OUTPUT

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

*counter decrement*

**(i-- atau i=i-1 atau i-=1)**

```
public class ContohLoopDoWhile4 {  
    public static void main(String[] args){  
        int i;  
        i=10;  
        do{  
            System.out.println(i);  
            i--;  
        }while(i>=1);  
    }  
}
```

OUTPUT

10  
9  
8  
7  
6  
5  
4  
3  
2  
1

# Perbedaan Kondisi

## Menggunakan for, while dan do-while

- Pengulangan **for** dan **while**
  - **mengecek kondisi** → **melakukan aksi**
- Pengulangan **do-while**
  - **melakukan aksi** → **mengecek kondisi**
  - Melakukan aksi **minimal satu** kali, meskipun kondisi awal langsung bernilai *false*

# Perbedaan Kondisi Menggunakan for, while dan do-while

**for :**

```
public class ContohLoopBedaFor {  
    public static void main(String[] args){  
        int i;  
        for(i=11; i<=10; i++)  
            System.out.println("Halo-Halo Bandung");  
    }  
}
```

**TAMPILAN OUTPUT**

**do – while :**

```
public class ContohLoopBedaDoWhile {  
    public static void main(String[] args){  
        int i;  
        i=11;  
        do{  
            System.out.println("Halo-Halo Bandung");  
            i++;  
        }while(i<=10);  
    }  
}
```

**while :**

```
public class ContohLoopBedaWhile {  
    public static void main(String[] args){  
        int i;  
        i=11;  
        while(i<=10){  
            System.out.println("Halo-Halo Bandung");  
            i++;  
        }  
    }  
}
```

**TAMPILAN OUTPUT**

Halo-Halo Bandung

# Seleksi didalam Pengulangan

**bilangan ganjil diantara angka 1 s/d 10 :**

```
public class ContohIfDalamFor {  
    public static void main(String[] args){  
        int x;  
        for(x=1; x<=10; x++){  
            if(x%2!=0)  
                System.out.print(x+" ");  
        }  
    }  
}
```

**TAMPILAN OUTPUT**

1 3 5 7 9

# Seleksi didalam Pengulangan

**bilangan bulat positif atau negatif**

```
import java.util.Scanner;

public class ContohIfDalamDoWhile {
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int bil=0;
        do{
            System.out.print("Input Angka : ");
            bil = input.nextInt();
            if(bil>0)
                System.out.println("Bilangan Positif");
            else if(bil<0)
                System.out.println("Bilangan Negatif");
            else
                System.out.println("Menutup Aplikasi");
        }while (bil!=0) ;
    }
}
```

## TAMPILAN OUTPUT

```
Input Angka : 5 [enter]
Bilangan Positif
Input Angka : 1 [enter]
Bilangan Positif
Input Angka : -7 [enter]
Bilangan Negatif
Input Angka : 0 [enter]
Menutup Aplikasi
```

```
import java.util.Scanner;

public class ContohNestedLoop1 {
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        System.out.print("Input jumlah rangkaian : ");
        int x = input.nextInt();
        int a;
        int b;
        for(a=1; a<=x; a++){
            for(b=1; b<=3; b++){
                System.out.print(b+" ");
                System.out.println(a);
            }
        }
    }
}
```

#### TAMPILAN OUTPUT

```
Input jumlah rangkaian : 5 [enter]
1 2 3 1
1 2 3 2
1 2 3 3
1 2 3 4
1 2 3 5
```

```
import java.util.Scanner;

public class ContohNestedLoop2 {
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        System.out.print("Input panjang segitiga : ");
        int x = input.nextInt();
        int a;
        int b;
        a=1;
        while (a<=x) {
            for(b=1; b<=a; b++)
                System.out.print("* ");
            System.out.println();
            a++;
        }
    }
}
```

#### TAMPILAN OUTPUT

```
Input panjang segitiga : 4
*
* *
* * *
* * * *
```

```
import java.util.Scanner;

public class ContohNestedLoop3 {
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        System.out.print("Input panjang matriks : ");
        int x = input.nextInt();
        int a;
        int b;
        a=1;
        while (a<=x) {
            b=1;
            do{
                System.out.printf("%d,%d ", a, b);
                b++;
            }while (b<=x) ;
            System.out.println();
            a++;
        }
    }
}
```

#### TAMPILAN OUTPUT

```
Input panjang matriks : 3
1,1 1,2 1,3
2,1 2,2 2,3
3,1 3,2 3,3
```

# Seleksi didalam Pengulangan

```
public class ContohIfDalamNestedLoop1 {  
    public static void main(String[] args){  
        int a;  
        int b;  
        for(a=1; a<=5; a++){  
            for(b=1; b<=5; b++){  
                if(b==1 || a==5 || a==b)  
                    System.out.print("* ");  
                else  
                    System.out.print("  ");  
                System.out.println();  
            }  
        }  
    }  
}
```

## TAMPILAN OUTPUT

```
Input panjang segitiga : 4  
*  
*  *  
*    *  
*      *  
*        *  
*  *  *  *  *
```

# Seleksi didalam Pengulangan

```
public class ContohIfDalamNestedLoop2 {  
    public static void main(String[] args){  
        int a;  
        int b;  
        int maxIndex = 4;  
        int temp=0;  
        int data[] = {5,7,3,2,8};  
        for(a=0; a<=maxIndex-1; a++){  
            for(b=a+1; b<=maxIndex; b++){  
                if(data[a]>data[b]){  
                    temp      = data[a];  
                    data[a] = data[b];  
                    data[b] = temp;  
                }  
            }  
        }  
        for(a=0; a<=maxIndex; a++)  
            System.out.print(data[a]+" ");  
    }  
}
```

**TAMPILAN OUTPUT**

2 3 5 7 8

# Menggunakan *keyword continue* dan *break*

- Keyword **continue** digunakan untuk melewati semua aksi yang ada di bawah blok *continue*.  
Biasanya digunakan ketika menggunakan seleksi di dalam pengulangan.
- Keyword **break** digunakan untuk keluar dari blok seleksi ataupun pengulangan.

# Menggunakan *keyword*

```
import java.util.Scanner;

public class ContohContinue {
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int data[] = {5,7,3,2,8};
        int indeks = 0;
        int maxIndeks = 4;
        System.out.println("Data: 5,7,3,2,8");
        System.out.print("Angka yang dicari : ");
        int cari = input.nextInt();
        for(int i=0; i<=4; i++){
            if(data[i]==cari)
                continue; //melewatkan proses indeks++
            indeks++;
        }
        if(indeks==maxIndeks)
            System.out.println("Angka "+cari+" terdapat disalah satu rangkaian");
        else
            System.out.println("Angka "+cari+" tidak ditemukan");
    }
}
```

## TAMPILAN OUTPUT

```
Data: 5,7,3,2,8
Angka yang dicari : 3 [enter]
Angka 3 terdapat disalah satu
rangkaihan
Data: 5,7,3,2,8
Angka yang dicari : 10 [enter]
Angka 10 tidak ditemukan
```

# Menggunakan *keyword*

```
import java.util.Scanner;

public class ContohBreak {
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int data[] = {5,7,3,2,8};
        int indeks = 0;
        int maxIndeks = 4;
        System.out.println("Data: 5,7,3,2,8");
        System.out.print("Angka yang dicari : ");
        int cari = input.nextInt();
        for(int i=0; i<=maxIndeks; i++){
            if(data[i]==cari)
                break; //jika ketemu, maka keluar pengulangan
            indeks++;
        }
        if(indeks>maxIndeks)
            System.out.println("Angka "+cari+" tidak ditemukan");
        else
            System.out.println("Angka "+cari+" berada pada urutan ke-"+(indeks+1));
    }
}
```

## TAMPILAN OUTPUT

```
Data: 5,7,3,2,8
Angka yang dicari : 2 [enter]
Angka 2 berada pada urutan ke-4
Data: 5,7,3,2,8
Angka yang dicari : 1 [enter]
Angka 1 tidak ditemukan
```