

# Pemrograman Berorientasi Objek

Struktur Kontrol

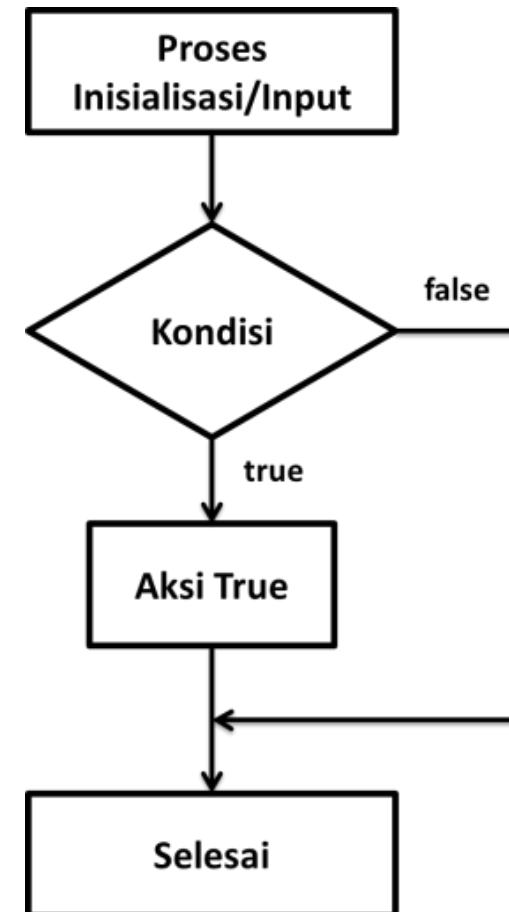
# Bahasan

- Percabangan
  - ✓ Statemen if satu kasus
  - ✓ Statemen if dua kasus
  - ✓ Statemen if tiga kasus atau lebih
  - ✓ Seleksi menggunakan switch
- Pengulangan
  - ✓ Pengulangan menggunakan for
  - ✓ Pengulangan menggunakan while
  - ✓ Pengulangan menggunakan do – while
  - ✓ Pengulangan bersarang

# Statemen if Satu Kasus

hanya ada blok aksi *true* yang dikerjakan

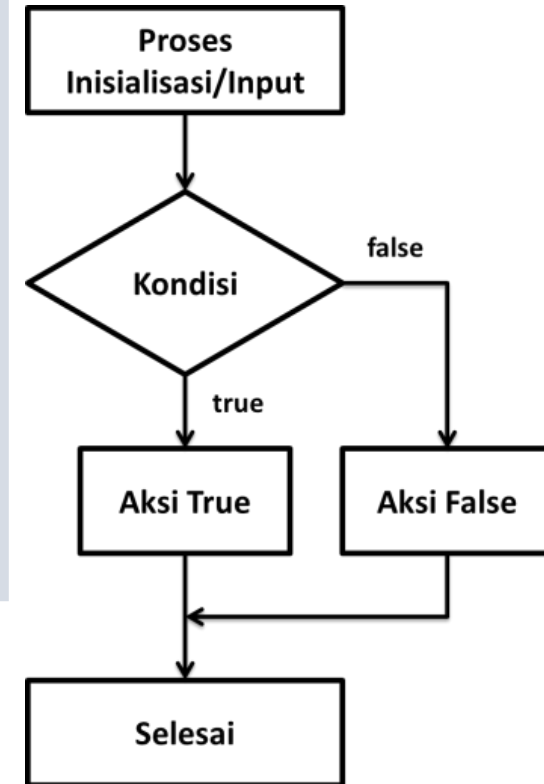
```
// Apabila hanya memiliki 1 aksi
if(kondisi)
    Aksi; // Ini adalah blok AksiTrue
// Apabila memiliki 2 atau lebih aksi
if(kondisi){
    Aksi1; // -----
    Aksi2; // Ini adalah blok AksiTrue
    AksiN; // -----
}
```



# Statemen if Dua Kasus

```
// Apabila hanya memiliki 1 aksi
if(kondisi)
    AksiTrue;
else
    AksiFalse;

// Apabila memiliki 2 atau lebih aksi
if(kondisi){
    AksiTrue; // Gunakan kurung kurawal jika blok aksiTrue
    ...      // memiliki lebih dari satu aksi
}else{
    AksiFalse; // Gunakan kurung kurawal jika blok aksiFalse
    ...       // memiliki lebih dari satu aksi
}
```



# Statemen if Tiga Kasus atau Lebih

## if empat kasus

### if tiga kasus

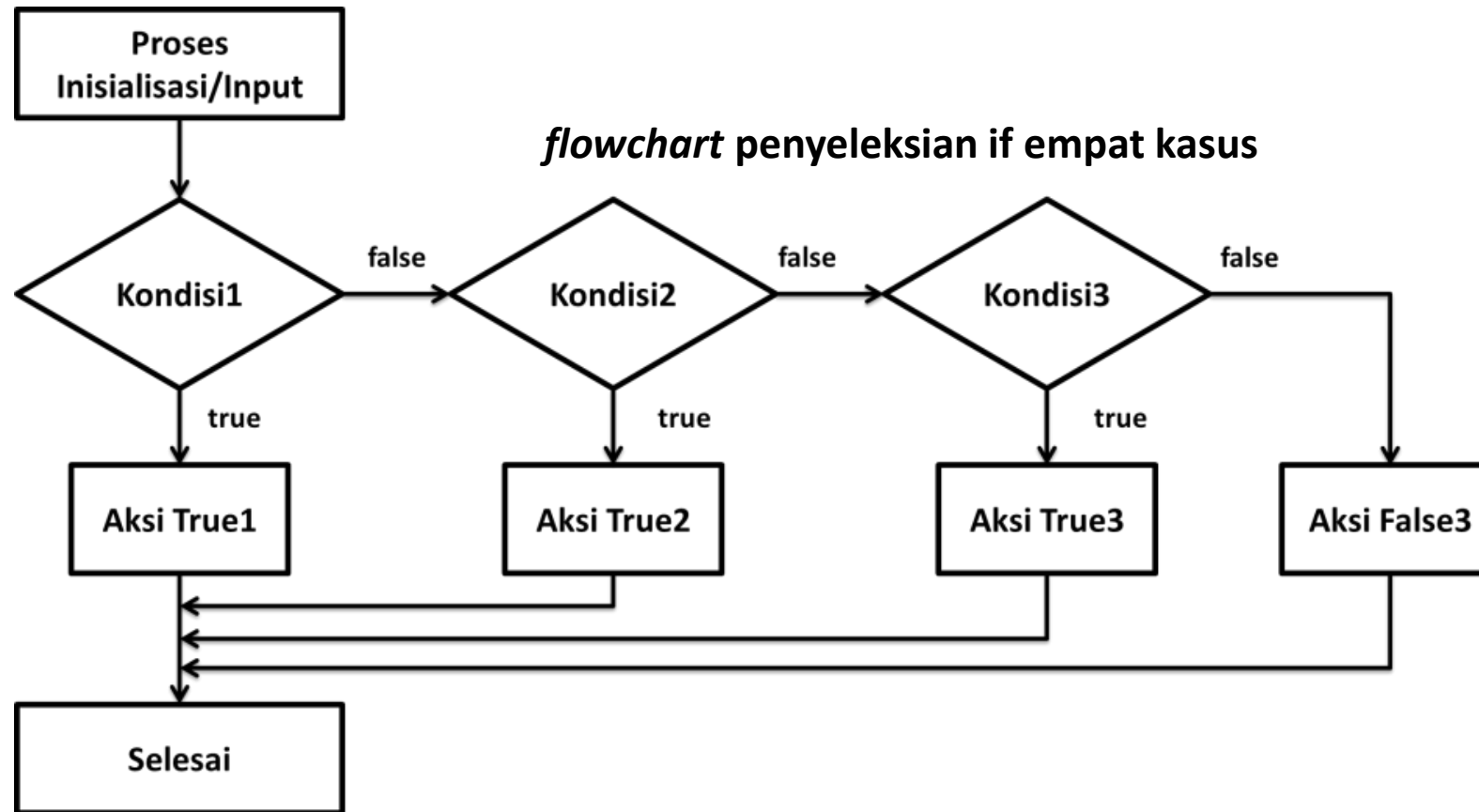
```
// Apabila hanya memiliki 1 aksi
if(kondisi1)
    AksiTrue1;
else if(kondisi2)
    AksiTrue2;
else
    AksiFalse2;

// Apabila memiliki 2 atau lebih aksi
if(kondisi1){
    AksiTrue1;
    ...
}else if(kondisi2){
    AksiTrue2;
    ...
}else{
    AksiFalse2;
    ...
}
```

```
// Apabila hanya memiliki 1 aksi
if(kondisi1)
    AksiTrue1;
else if(kondisi2)
    AksiTrue2;
else if(kondisi3)
    AksiTrue3;
else
    AksiFalse3;

// Apabila memiliki 2 atau lebih aksi
if(kondisi1){
    AksiTrue1;
    ...
}else if(kondisi2){
    AksiTrue2;
    ...
}else if(kondisi3){
    AksiTrue3;
    ...
}else {
    AksiFalse3;
    ...
}
```

# Statemen if Tiga Kasus atau Lebih (lanj.)



# Seleksi Menggunakan switch

```
switch (ekspresi){ //variabel yang akan dicocokkan
    case nilai1: aksiCase1 //aksi case 1
                break;
    case nilai2: aksiCase2 //aksi case 2
                break;
    ...
    case nilaiN: aksiCaseN //aksi case N
                break;
    default: aksiDefault //bersifat opsional
            //dikerjakan jika tidak ada case yang cocok
}
```

- **Ekspresi** diisi variabel **bilangan bulat** atau **karakter** (char).
- Nilai dalam case diisi nilai bilangan bulat atau karakter
- tidak perlu menuliskan kurung kurawal ('{}') meskipun aksi dalam blok suatu case memiliki banyak statemen
- *Keyword* **default** bersifat opsional dan akan dikerjakan ketika tidak ada satupun case dengan nilai yang cocok
- Gunakan *keyword* **break** untuk mengakhiri aksi dari blok case.

# Seleksi Menggunakan switch (lanj.)

```
public class ContohSwitch1 {  
    public static void main(String[] args){  
        int hari=0;  
        switch(hari){  
            case 1: System.out.println("Hari Senin");  
                    break;  
            case 2: System.out.println("Hari Selasa");  
                    break;  
            case 3: System.out.println("Hari Rabu");  
                    break;  
            case 4: System.out.println("Hari Kamis");  
                    break;  
            case 5: System.out.println("Hari Jum'at");  
                    break;  
            case 6: System.out.println("Hari Sabtu");  
                    break;  
            case 7: System.out.println("Hari Minggu");  
                    break;  
            default: System.out.println("Input Salah");  
        }  
    }  
}
```

## TAMPILAN OUTPUT

**Contoh hari = 1**

Hari Senin

**Contoh hari = 2**

Hari Selasa

**Contoh hari = 3**

Hari Rabu

**Contoh hari = 4**

Hari Kamis

**Contoh hari = 5**

Hari Jum'at

**Contoh hari = 6**

Hari Sabtu

**Contoh hari = 7**

Hari Minggu

**Contoh hari = 0**

Input Salah



# 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.)

## Catatan Penting

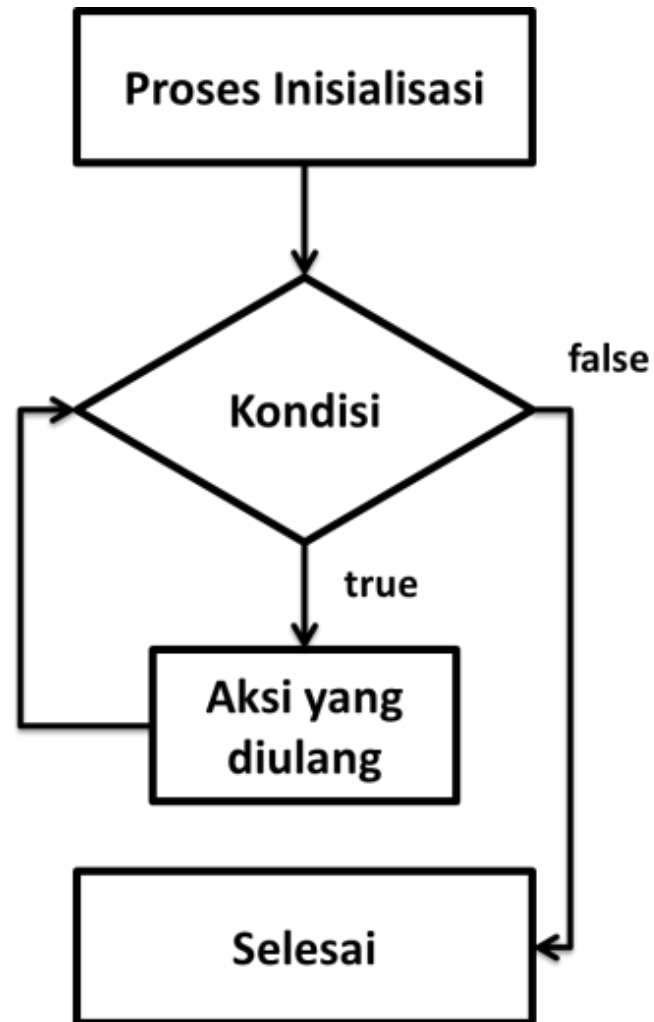
1. Nama variabel tidak harus selalu **i** ataupun **x**
2. Ciri dari pengulangan ***counter increment*** menggunakan perbandingan ( **$\leq$**  atau  **$<$** )
3. Ciri dari pengulangan ***counter decrement*** menggunakan perbandingan ( **$\geq$**  atau  **$>$** )
4. Semua poin **berlaku** juga untuk 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.)

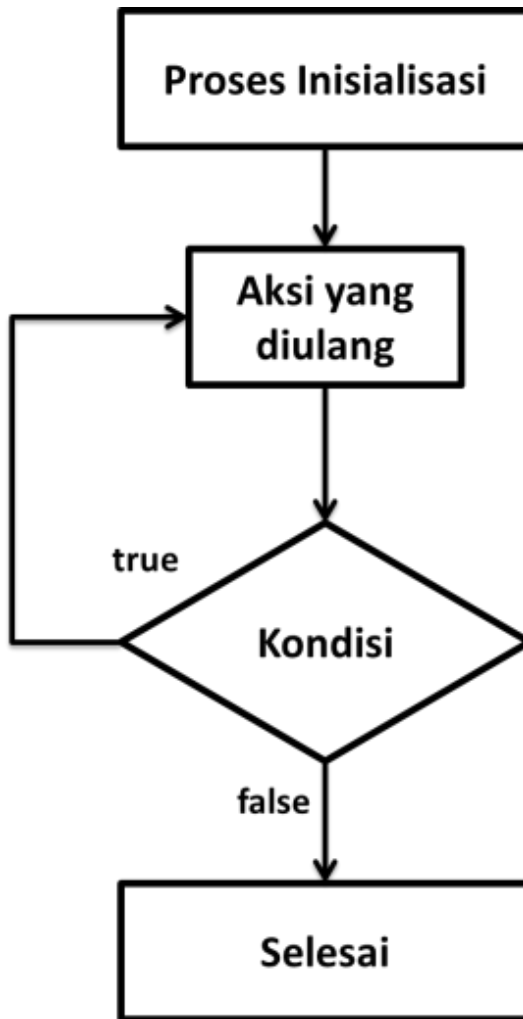


# Pengulangan Menggunakan do-while

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.)



# 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*



# 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

# Pengulangan Bersarang

```
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
```

# Pengulangan Bersarang (lanj.)

```
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
*
* *
* * *
* * * *
```

# Pengulangan Bersarang (lanj.)

```
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 Bersarang

```
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 Bersarang (lanj.)

```
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 continue* dan *break* (lanj.)

```
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 continue* dan *break* (lanj.)

```
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
```

# Terima Kasih