

BRANCHING & LOOPING

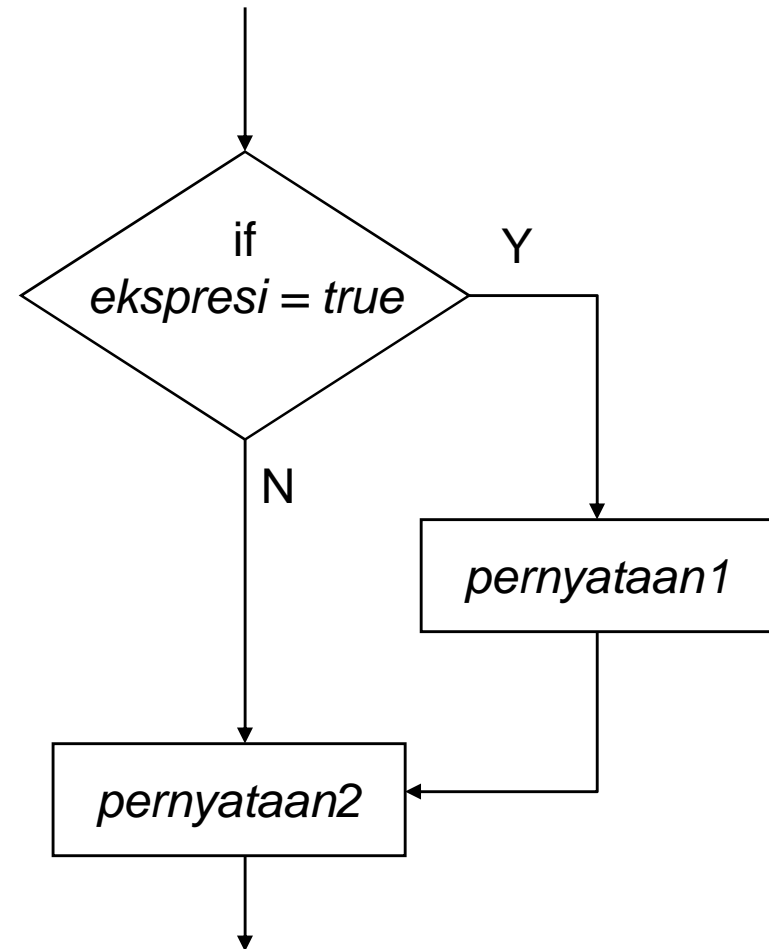
Percabangan & Perulangan

Percabangan

- if
- if-else
- else-if
- switch

if

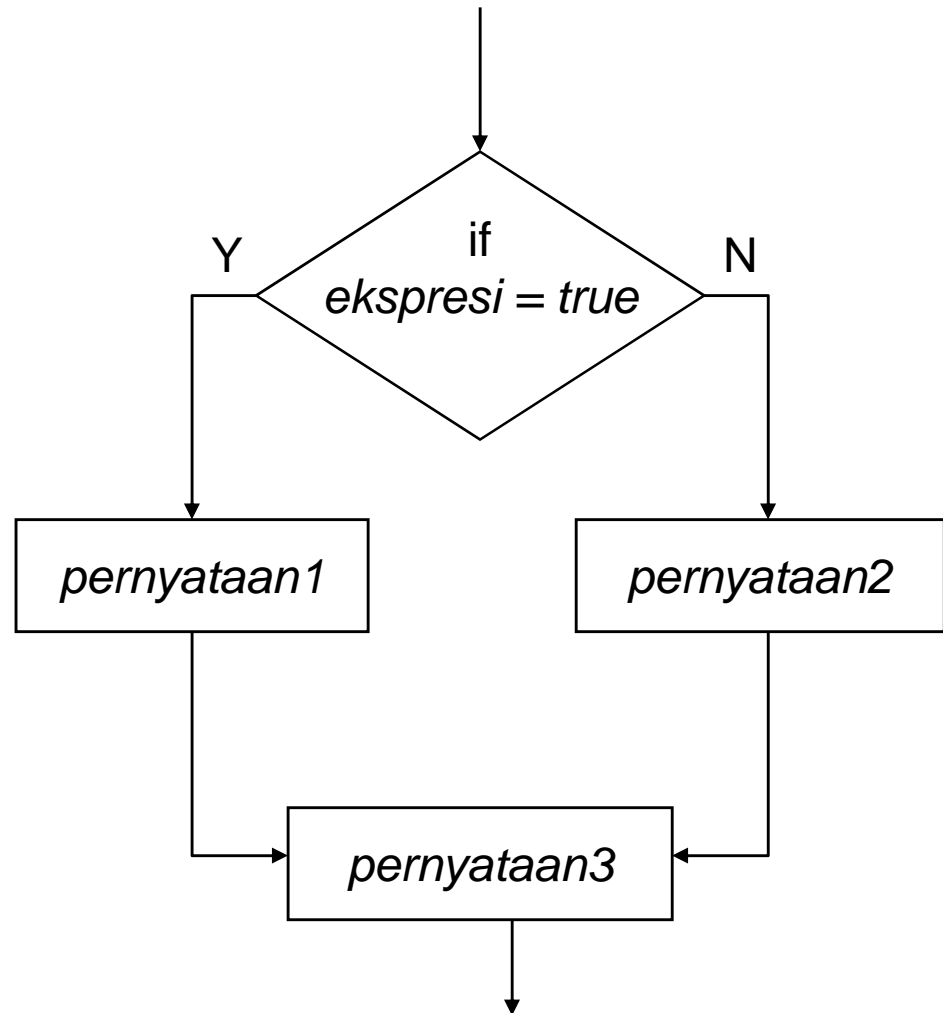
```
if (ekspresi boolean) {  
    pernyataan1;  
}  
pernyataan2;
```



```
public class If {  
    public static void main(String args[]) {  
        int bilangan=-1;  
        if (bilangan<0)  
            System.out.println("Bilangan adalah negatif");  
    }  
}
```

if-else

```
if (ekspresi boolean) {  
    pernyataan1;  
} else {  
    pernyataan2;  
}  
pernyataan3;
```



If - else

- `if ()` statement takes a boolean expression, not a numeric value.
- You cannot convert or cast boolean types and numeric types.

- If you have:

```
if (x) // x is int
```

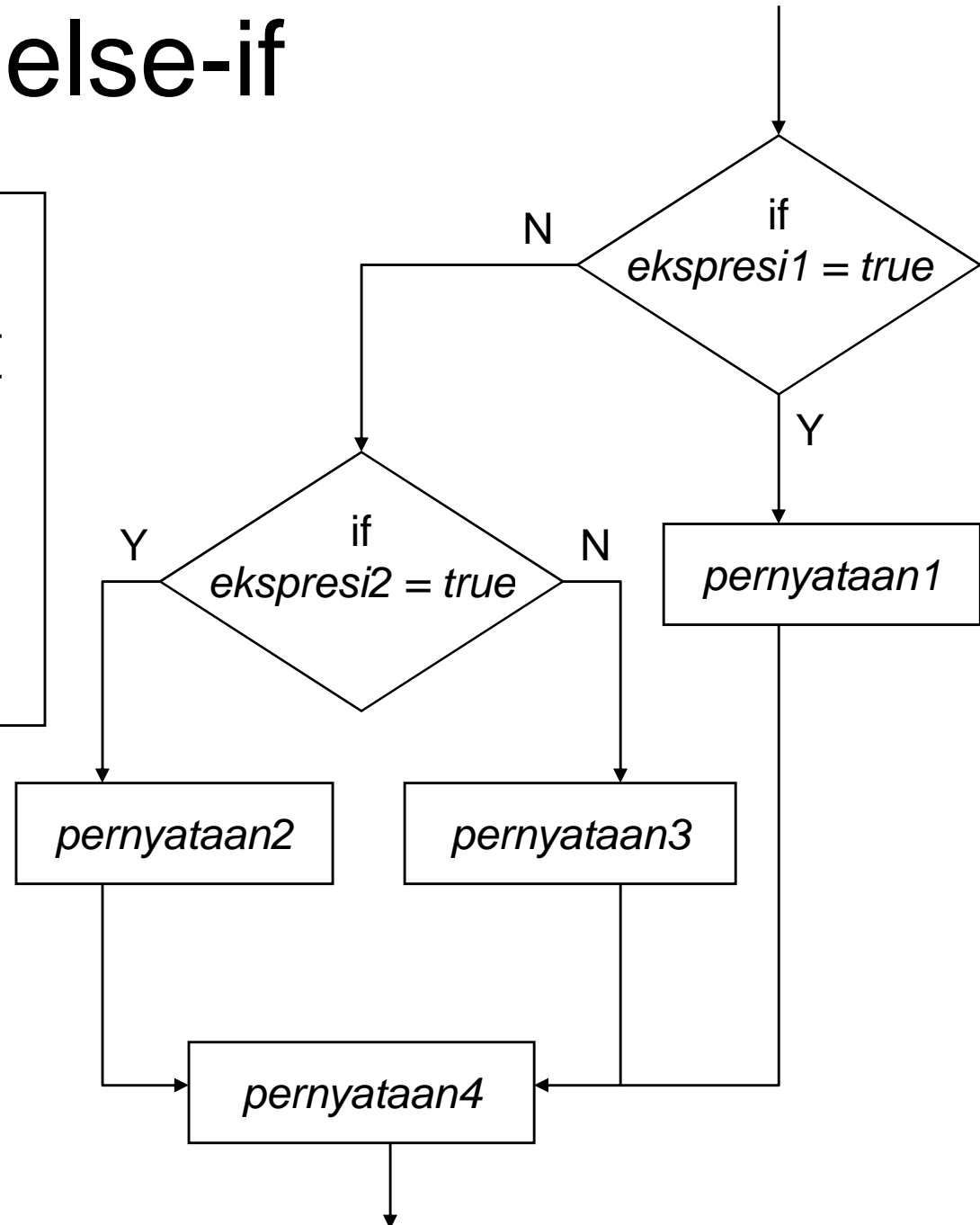
use

```
if (x != 0)
```

```
public class IfElse {  
    public static void main(String args[]) {  
        int bilangan=-1;  
        if (bilangan<0)  
            System.out.println("Bilangan adalah negatif");  
        else  
            System.out.println("Bilangan adalah positif");  
    }  
}
```

else-if

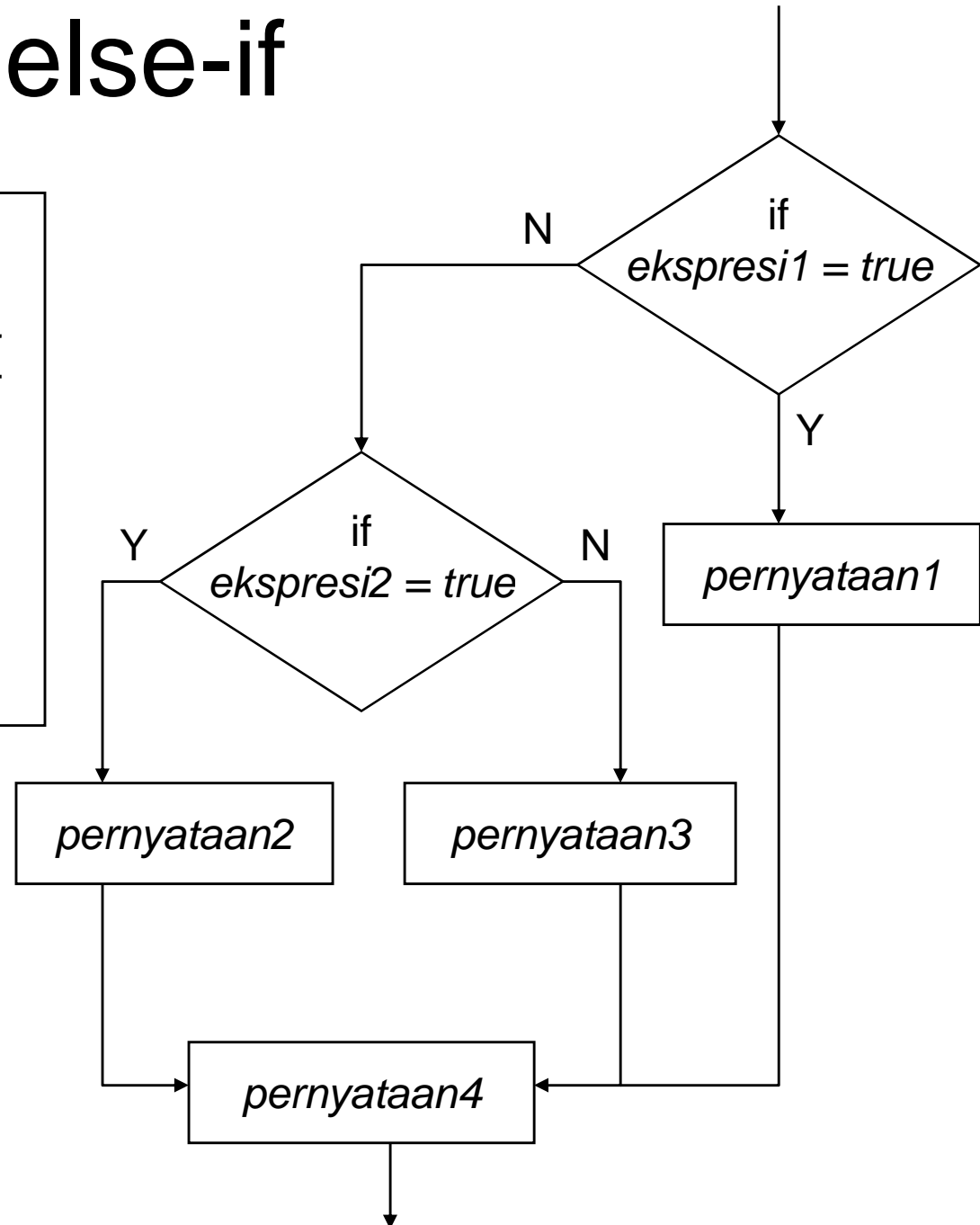
```
if (ekspresi1 boolean) {  
  pernyataan1;  
} else if (ekspresi2 boolean) {  
  pernyataan2;  
} else {  
  pernyataan3;  
}  
pernyataan4;
```




```
public class If {  
    public static void main(String args[]) {  
        int bilangan=-1;  
        if (bilangan<0)  
            System.out.println("Bilangan adalah negatif");  
    }  
}
```

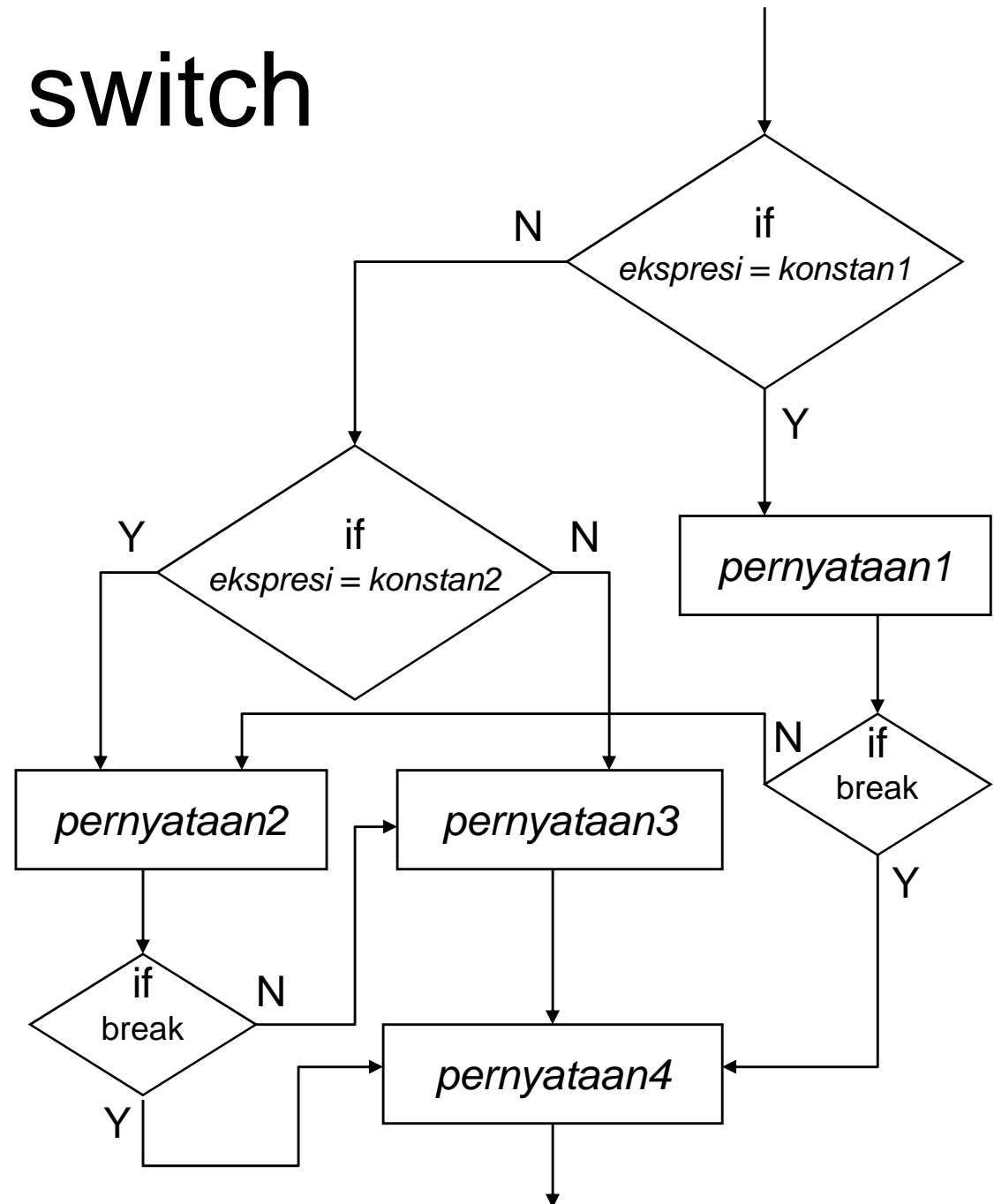
else-if

```
if (ekspresi1 boolean) {  
    pernyataan1;  
} else if (ekspresi2 boolean) {  
    pernyataan2;  
} else {  
    pernyataan3;  
}  
pernyataan4;
```



switch

```
switch (ekspresi) {  
  case konstan1 :  
    pernyataan1;  
    break;  
  case konstan2:  
    pernyataan2;  
    break;  
  default :  
    pernyataan3;  
}  
pernyataan4;
```



switch

- Dalam statemen switch, ekspresi harus kompatibel dengan tipe int
- Promosi akan terjadi untuk tipe byte, short, or char
- Floating point, long, referensi class ataupun String tidak diperbolehkan.

switch

Aturan-aturan pada pernyataan *switch*

- Urutan dari pernyataan *case* tidak penting
- Klausula *default* boleh diletakkan di awal (secara konvensi diletakkan terakhir)
- Klausula *default* bersifat *optional* → akan dieksekusi HANYA JIKA tidak ada satupun ekspresi yang match dengan *case* yg ada
- Keyword *break* **HARUS** disertakan pada akhir dari setiap pernyataan *case*, digunakan untuk melompat ke akhir dari blok *switch* setelah ketemu satu *case* yg match
- Jika tidak ada statemen *break* di akhir *case*, maka eksekusi akan berlanjut ke *case* berikutnya tanpa mengecek nilai ekspresi *case* tsb

```
public class Switch {
    public static void main(String args[]) {
        int i = 2;
        switch (i) {
            case 1 : i+=3;
                break;
            case 2 : i+=5;
                break;
            default: i+=10;
        }
        System.out.println(i);
    }
}
```

Perulangan

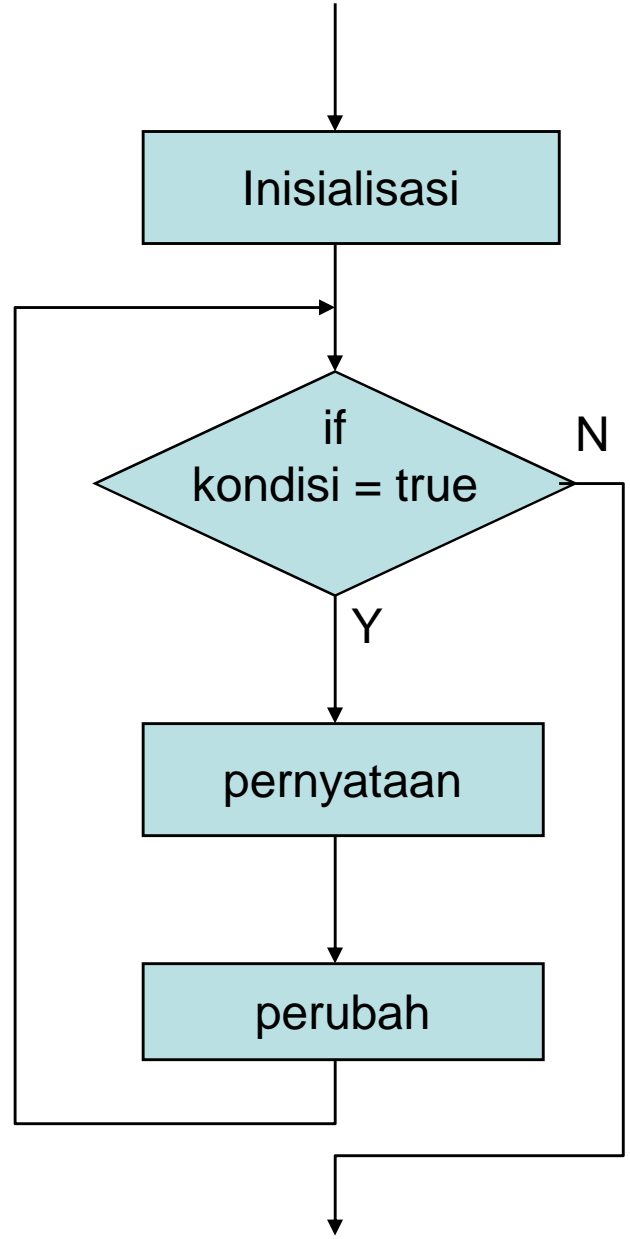
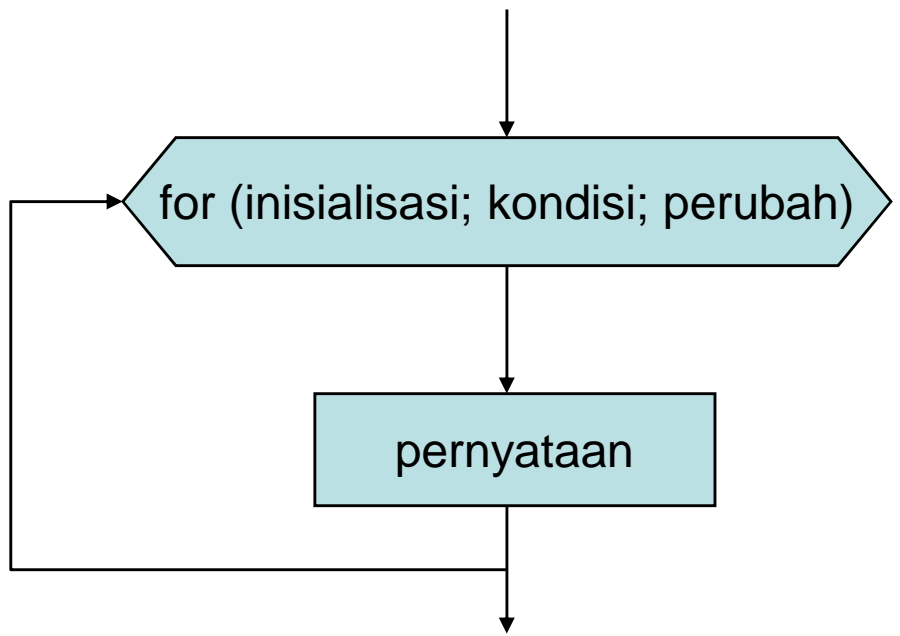
- for
- while
- do-while

Penentuan Looping

- Bisa ditentukan HOW MANY TIMES-nya → sebaiknya gunakan *for*
- Tidak bisa ditentukan HOW MANY TIMES-nya, tapi diketahui kondisi untuk menghentikan loop-nya
 - sebaiknya gunakan *while()* atau *do-while()*
 - *while* → pengecekan kondisi dilakukan di AWAL
 - ada kemungkinan body loop tidak akan pernah dilakukan sama sekali (0 kali)
 - *do while* → pengecekan kondisi dilakukan di AKHIR
 - minimal body loop akan dieksekusi 1 kali (*at least ONCE*)

for

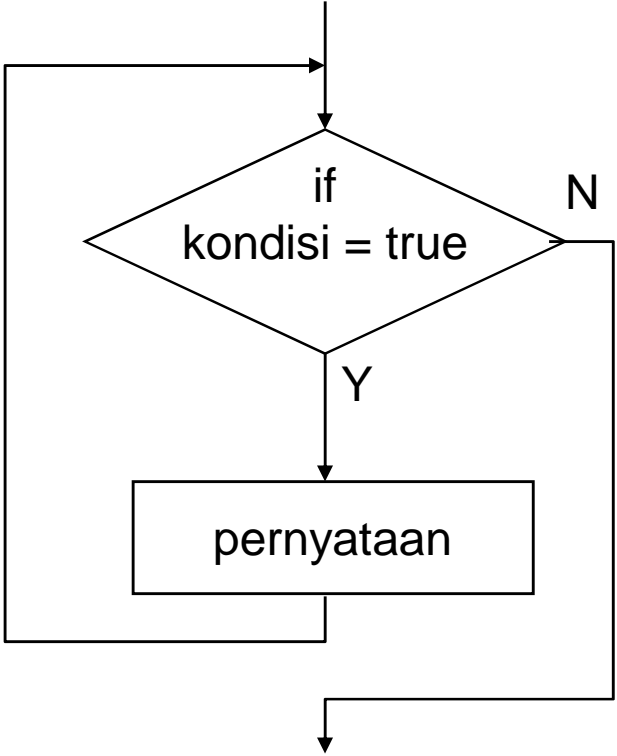
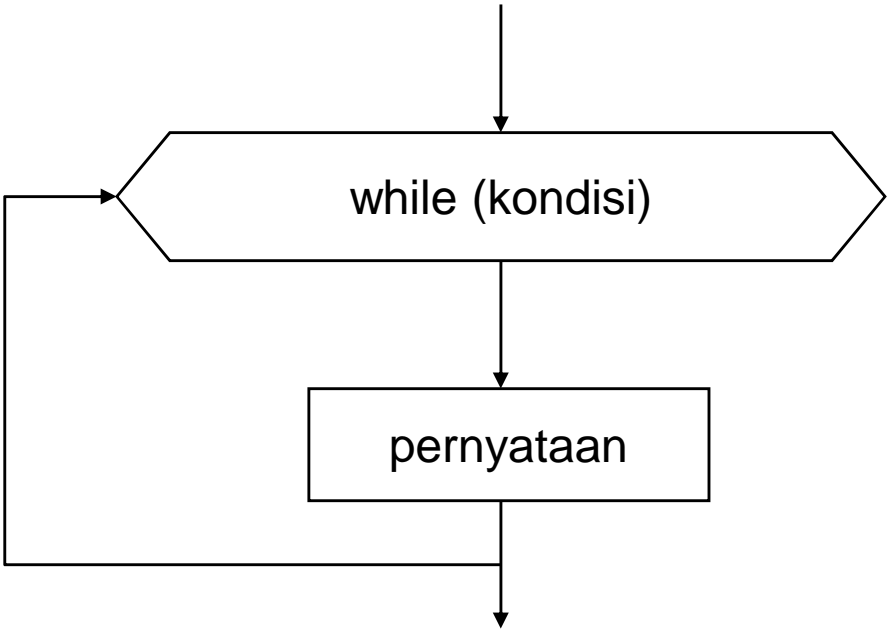
```
for (inisialisasi; kondisi; perubah) {  
    pernyataan;  
}
```



```
for (int i=0; i<10; i++) {  
    System.out.println("Hore !!");  
}
```

while

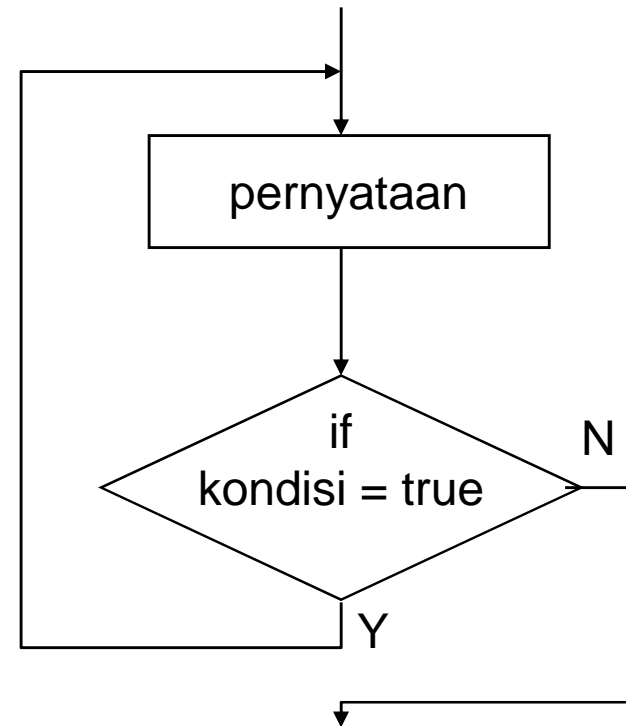
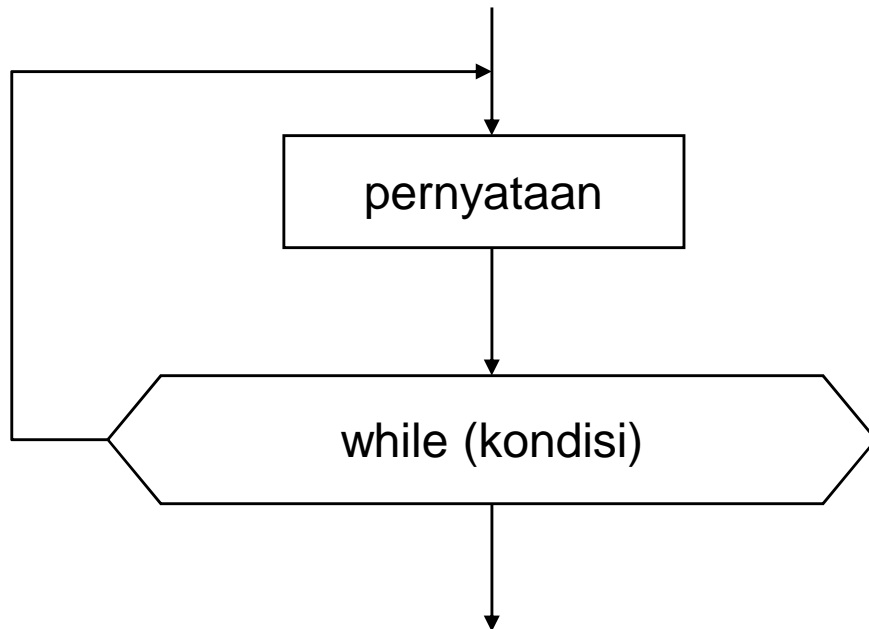
```
while (kondisi) {  
    pernyataan;  
}
```



```
int i=0;
while (i<10) {
    System.out.println("Hore !!");
    i++;
}
```

do-while

```
do {  
    pernyataan;  
} while (kondisi);
```



```
int i=0;
do {
    System.out.println("Hore !!");
    i++;
} while (i<10);
```

Kondisional

break

- Pada *switch-case* digunakan untuk menuju ke akhir (keluar dari) struktur *switch*.
- Dalam looping, pernyataan ini berfungsi untuk keluar dari *loop for*, *do-while* dan *while*.

continue

- digunakan untuk mengarahkan eksekusi ke iterasi (proses) berikutnya pada *loop* yang sama (*skip the current iteration, continue to the next iteration*)