

Perulangan 1

Tujuan

Di akhir pertemuan, mahasiswa diharapkan mampu :

- Memahami algoritma perulangan bagian 1 (for, while, do-while)
- Memberikan contoh sederhana perulangan
- Menggambarkan permasalahan studi kasus perulangan bagian 1 dengan menggunakan flowchart

Outline

- For
- While
- Do-While

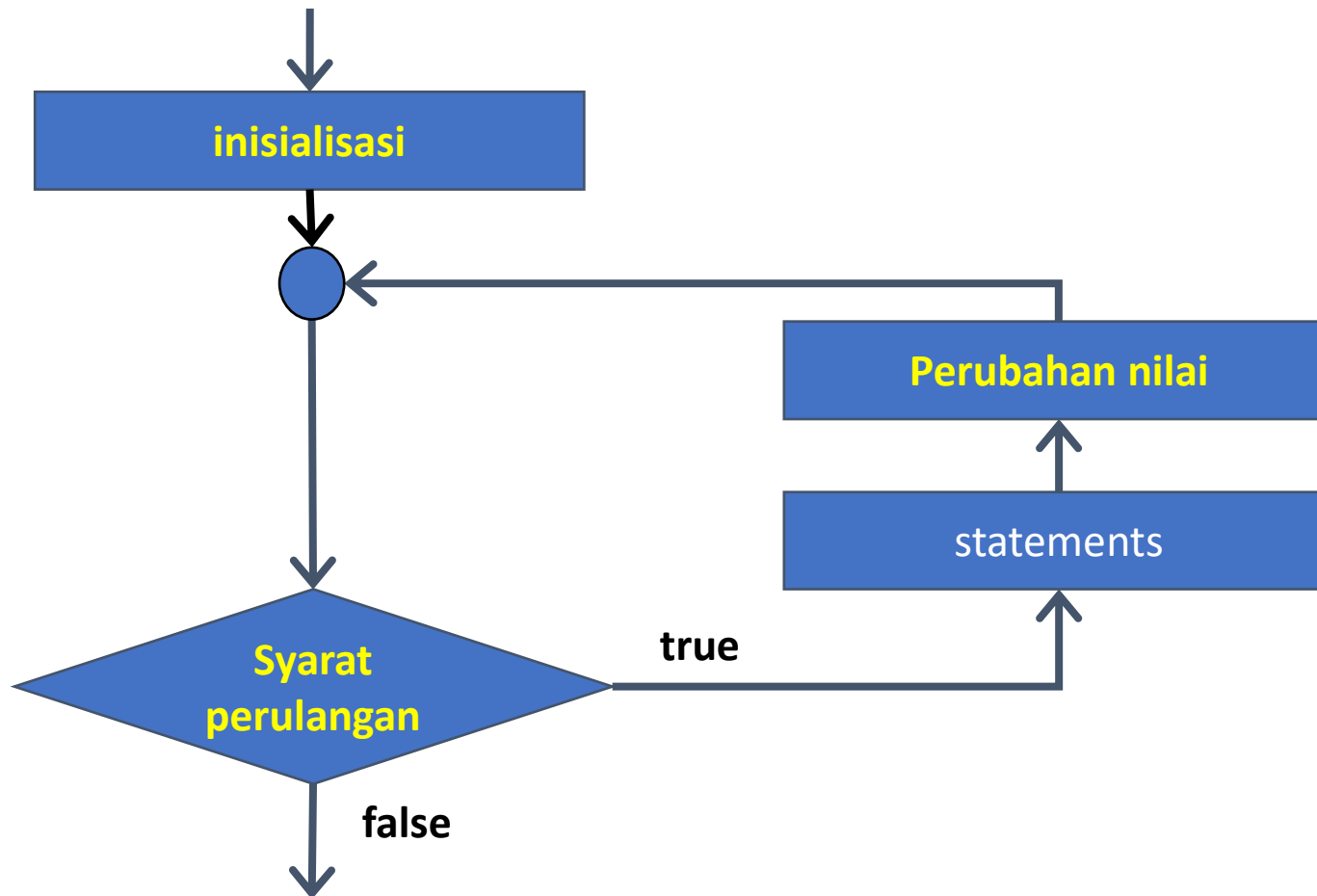
Definisi

- Perintah perulangan/loop statement adalah perintah untuk mengulang satu atau lebih statement sebanyak beberapa kali.
- Loop statement digunakan agar kita tidak perlu menuliskan satu/sekumpulan statement berulang-ulang.
 - Dengan begitu maka kesalahan pengetikan bisa dikurangi
- Contoh:
 - Hitung nilai $n!$ (n faktorial)
 - Jika $n=1000$, sangat tidak efisien bila kita menuliskan kode $1*2*3*4*.....*1000$

Definisi

- Dalam bahasa Java, ada 3 macam perintah perulangan yang umum digunakan yaitu:
 - Perintah for()
 - Perintah while()
 - Perintah do while()

Alur Perulangan FOR





Perulangan menggunakan FOR

- Umumnya digunakan pada pengulangan yang jumlah iterasinya sudah pasti atau sudah diketahui sebelumnya.

- Konstruksi for

```
for ([exp1]; [exp2]; [exp3]) statement;
```

atau:

```
for ([exp1]; [exp2]; [exp3]) {  
    statement1;  
    statement2;  
    .....  
}
```

- **exp1** : adalah ekspresi untuk inisialisasi,
- **exp2** : adalah ekspresi conditional (batas pengulangan)
- **exp3** : adalah ekspresi increment atau decrement
- exp1 exp3 adalah sifatnya optional (boleh ada boleh tidak ada).

Perulangan menggunakan FOR

- **Insialisasi nilai** adalah tempat dimana kita akan memberikan nilai awal pada variabel counter (variabel yang digunakan untuk menghitung jumlah perulangan).
- **Syarat perulangan** adalah syarat yang harus dipenuhi agar perulangan tetap dilakukan.
- **Perubahan nilai** adalah perubahan yang akan dilakukan pada tiap putaran untuk menjamin bahwa perulangan tersebut tidak akan berlangsung terus menerus.



Perulangan FOR

- Perintah for() biasa digunakan untuk melakukan perulangan sebanyak jumlah yang telah diketahui.
- Contoh: Saya suka pemrograman sebanyak 10 kali

Output:

```
Saya Suka Pemrograman
Saya Suka Pemrograman
Saya Suka Pemrograman
Saya Suka Pemrograman
Saya Suka Pemrograman
Saya Suka Pemrograman
Saya Suka Pemrograman
Saya Suka Pemrograman
Saya Suka Pemrograman
Saya Suka Pemrograman
```

Perulangan akan dilakukan selama $i \leq 10$

```
int i;
for (i=1; i<=10; i++) System.out.println("Saya Suka Pemrograman");
```

Sebelum dilakukan perulangan, variabel i diberi nilai 1

Pada tiap putaran, variabel i akan ditambah dengan 1

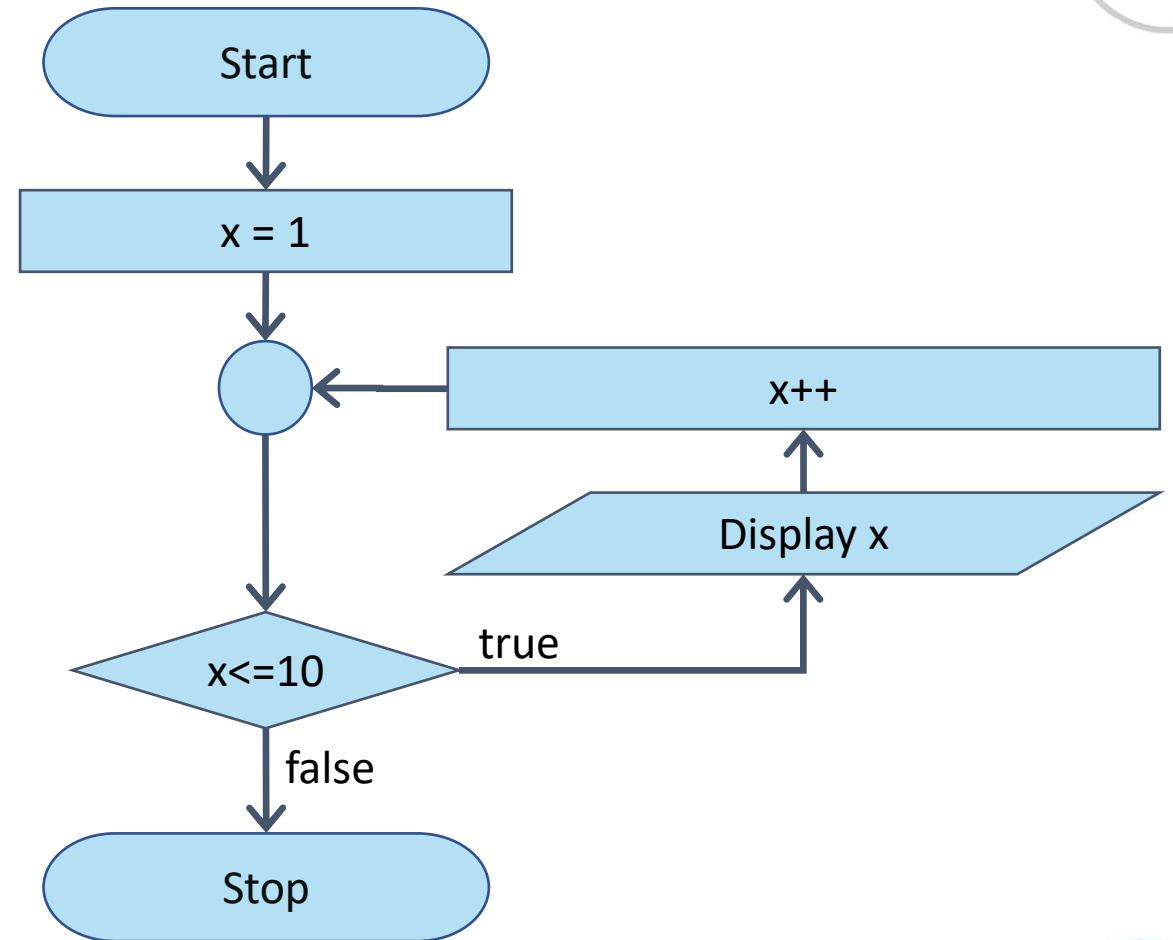
Contoh perulangan FOR

Buatlah program untuk mencetak ke layar angka 1 sampai 10.

```
int x;  
for (x=1; x<=10; x++)  
System.out.printf("%d\n", x);
```

Output:

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



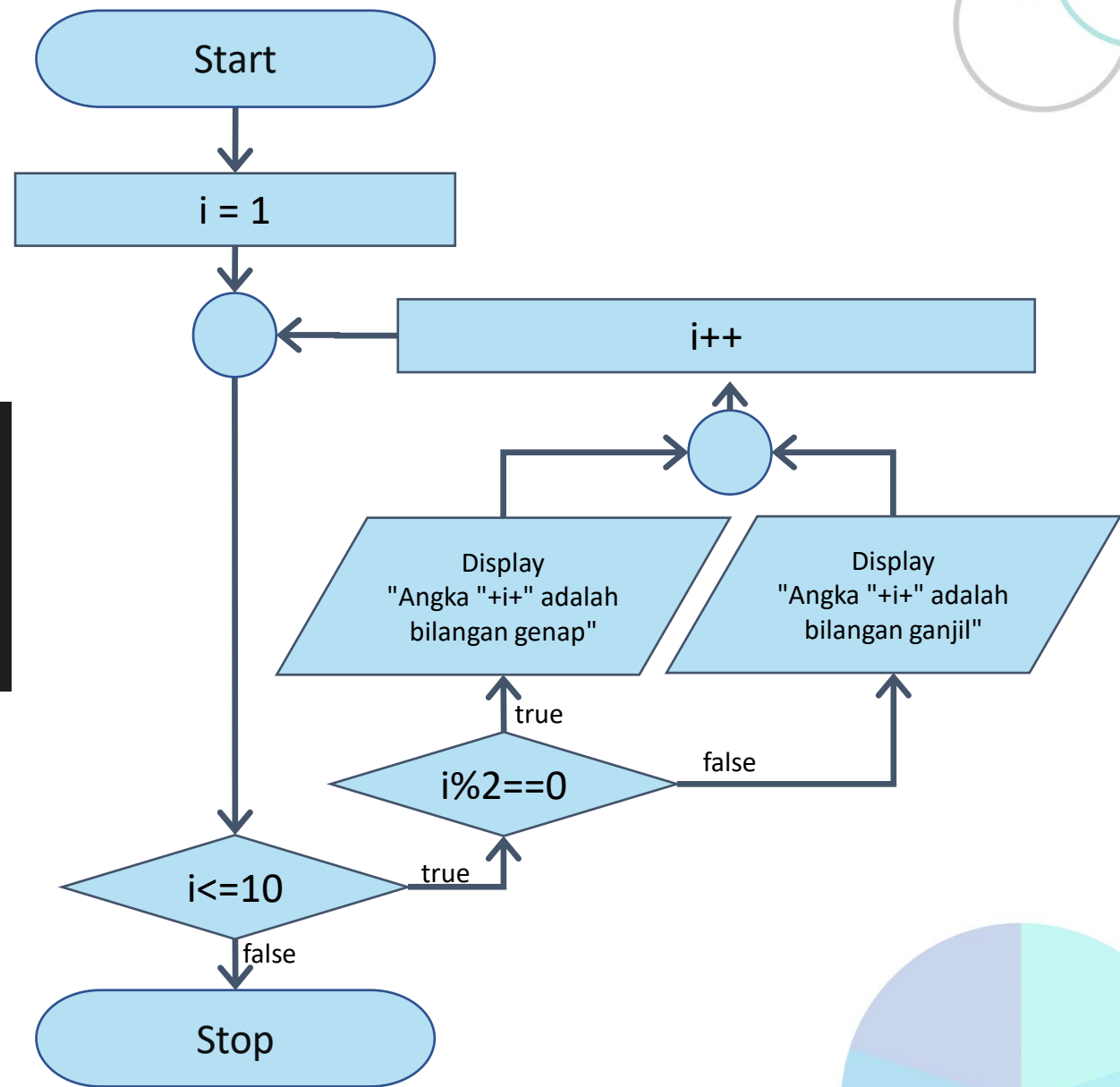
Contoh perulangan FOR

- Buatlah program untuk mencetak keterangan bilangan genap dan ganjil dari 1 sampai 10.

```
int i;  
for(i=1;i<=10;i++){  
    if(i%2==0)  
        System.out.println("Angka "+i+" adalah bilangan genap");  
    else  
        System.out.println("Angka "+i+" adalah bilangan ganjil");  
}
```

Output:

```
Angka 1 adalah bilangan ganjil  
Angka 2 adalah bilangan genap  
Angka 3 adalah bilangan ganjil  
Angka 4 adalah bilangan genap  
Angka 5 adalah bilangan ganjil  
Angka 6 adalah bilangan genap  
Angka 7 adalah bilangan ganjil  
Angka 8 adalah bilangan genap  
Angka 9 adalah bilangan ganjil  
Angka 10 adalah bilangan genap
```





Variasi FOR

- exp1 dan exp3 boleh terdiri dari beberapa ekspresi yang dipisahkan dengan koma.
- Contoh:

```
int i,j;  
for(i=1, j=30; i<j; i++, j--)  
    System.out.printf("%04d -- %04d\n",i,j);
```

- Output:

```
0001 -- 0030  
0002 -- 0029  
0003 -- 0028  
0004 -- 0027  
0005 -- 0026  
0006 -- 0025  
0007 -- 0024  
0008 -- 0023  
0009 -- 0022  
0010 -- 0021  
0011 -- 0020  
0012 -- 0019  
0013 -- 0018  
0014 -- 0017  
0015 -- 0016
```



Variasi FOR

- exp2 pada for dapat diisi dengan sebuah variabel boolean

```
Scanner input = new Scanner(System.in);
int angka, i;
boolean stop=false;

for (i=0; !stop; i++){
    System.out.print("Masukkan angka : ");
    angka = input.nextInt();
    System.out.println("Angka yang anda masukkan: "+angka);
    if (angka == 0)
        stop = true;
}
System.out.println("Selesai.");
```

- Output:

```
Masukkan angka : 16
Angka yang anda masukkan: 16
Masukkan angka : 10
Angka yang anda masukkan: 10
Masukkan angka : 2020
Angka yang anda masukkan: 2020
Masukkan angka : 0
Angka yang anda masukkan: 0
Selesai.
```

Variasi FOR

- exp1 dan exp3 pada for dapat dikosongi, sesuai dengan kebutuhan

```
Scanner input = new Scanner(System.in);
int angka, i;
boolean stop=false;

for (; !stop;){
    System.out.print("Masukkan angka : ");
    angka = input.nextInt();
    System.out.println("Angka yang anda masukkan: "+angka);
    if (angka == 0)
        stop = true;
}
System.out.println("Selesai.");
```

- Output:

```
Masukkan angka : 10
Angka yang anda masukkan: 10
Masukkan angka : 25
Angka yang anda masukkan: 25
Masukkan angka : 0
Angka yang anda masukkan: 0
Selesai.
```

Perulangan While

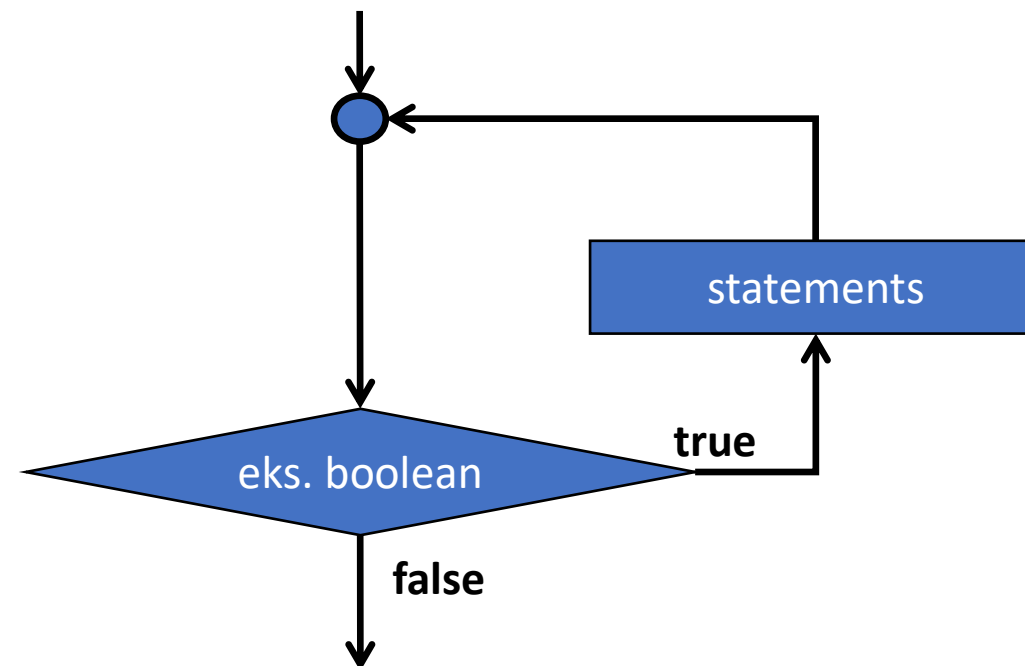
- Sintaks :

```
while (syarat perulangan)  
statement; //perintah yang akan diulang
```

atau :

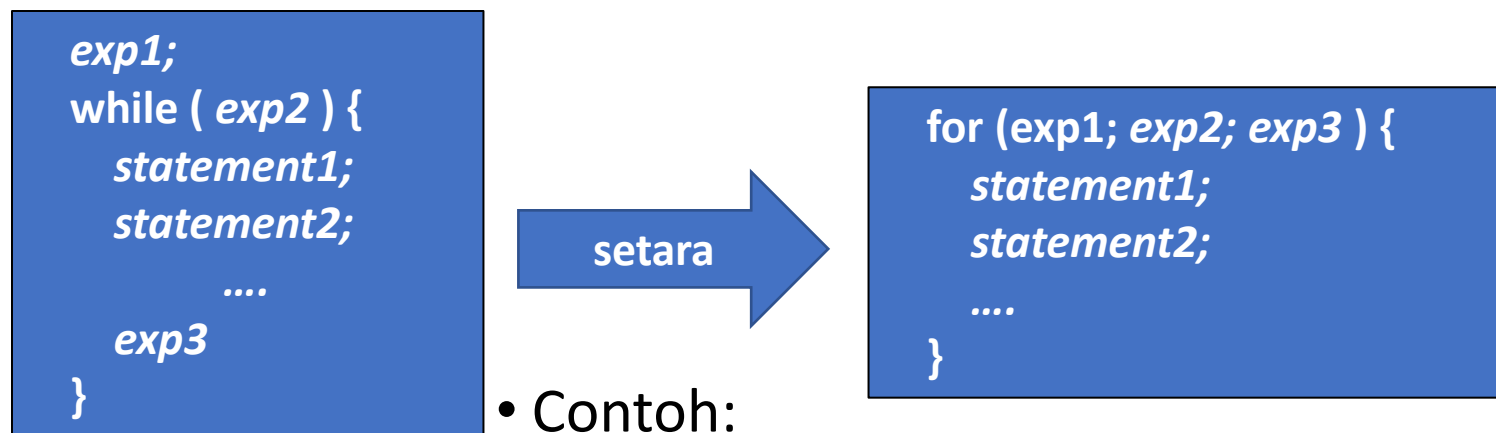
```
while(syarat perulangan) {  
    statement1;  
    statement2;  
    ...  
}
```

- Syarat perulangan adalah syarat yang harus dipenuhi agar perulangan tetap dilakukan
- Perulangan while akan terus dijalankan selama syarat perulangan bernilai TRUE

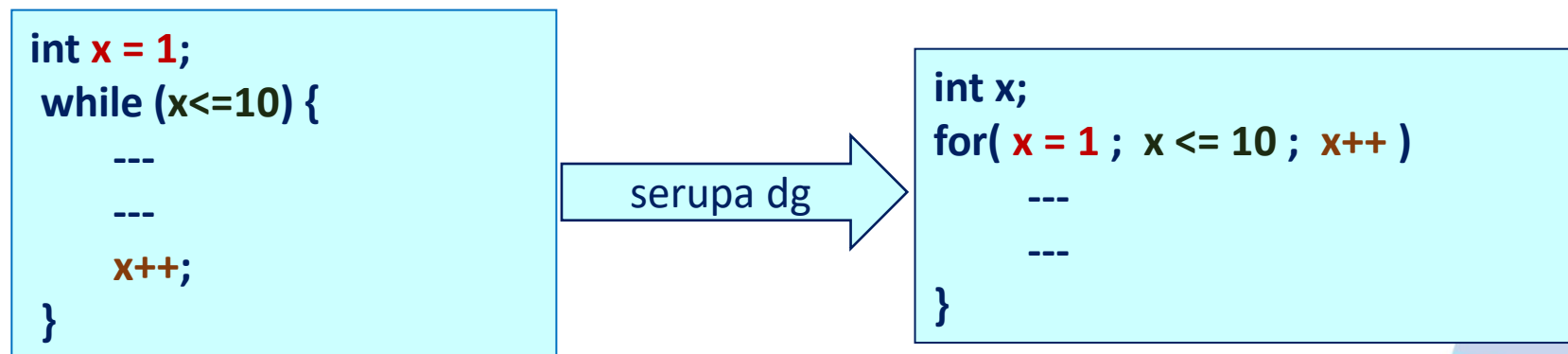


For vs While

- Kesetaraan for dengan while sbb:



- Contoh:



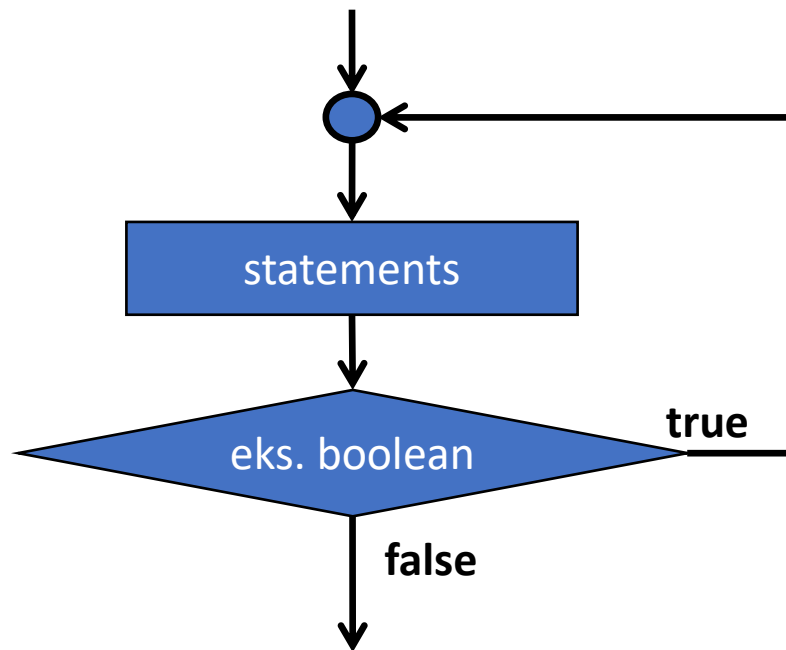
Perulangan Do-While

- Pada prinsipnya, perintah do-while() sama dengan perintah while().
- Perintah do-while() akan mengulang statement miliknya selama syarat pengulangannya terpenuhi.
- Hanya saja, perintah do-while() menjalankan statementnya terlebih dahulu, setelah itu baru memeriksa syaratnya.
 - Sedangkan perintah while() memeriksa syarat terlebih dahulu.
- Oleh karena itu, perintah do-while() akan menjalankan statementnya sebanyak satu kali, meskipun syarat pengulangan tidak terpenuhi.

Perulangan Do-While

- Sintaks :

```
do  
Statement;  
while (eks.boolean);  
atau :  
do {  
    statement1;  
    statement2;  
    ...  
} while(eks.boolean)
```



- Selama exp.boolean bernilai true maka statements dieksekusi lagi.
- Pengetesan exp.boolean dilakukan setelah meng-eksekusi statements.

Contoh Perulangan Do - While

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

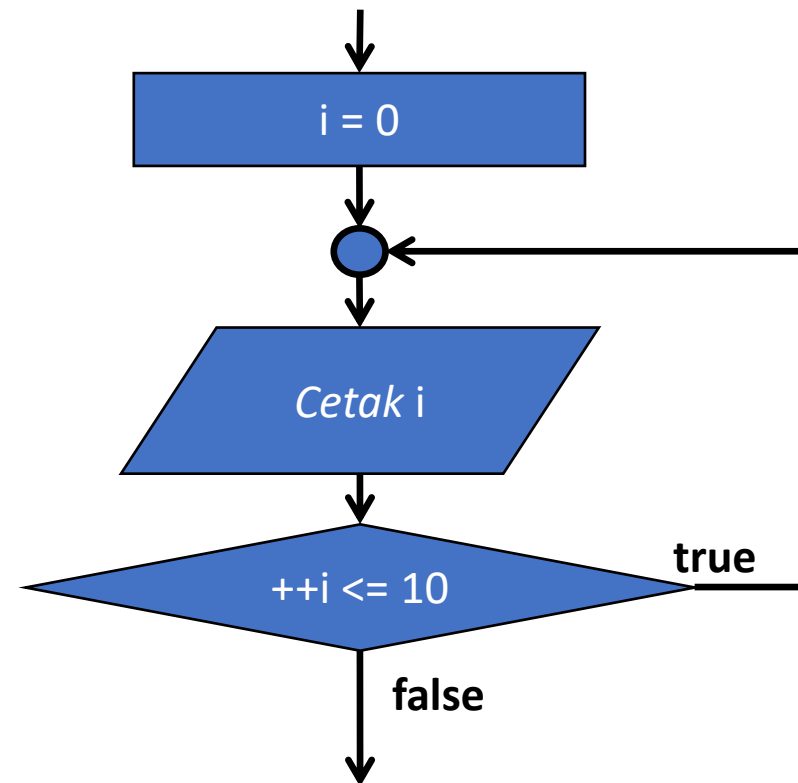
output

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
looping selesai
```

```
int i = 11;  
do {  
    System.out.println(i);  
} while(++i <= 10);  
System.out.println("looping selesai");
```

output

```
11  
looping selesai
```



While vs Do-While

- Pada perulangan **while**, statement atau blok statement mungkin tidak akan pernah dilaksanakan, bila nilai ekspresi boolean bernilai false, karena operasi pengulangan diawali dengan mengeksekusi ekspresi boolean terlebih dahulu.
- Pada perulangan **do-while** statement atau blok statement pasti dikerjakan paling sedikit satu kali, karena ekspresi boolean baru diuji pada akhir blok pengulangan.

Menghentikan Perulangan

- Beberapa cara untuk menghentikan pengulangan untuk program interaktif, di antaranya dapat dilakukan dengan menambah:
 - **Sentinel** atau Pembatas dengan kode khusus.
 - **Pertanyaan**, misalkan: “Apakah pengulangan akan dilanjutkan?”.

Cara Sentinel

- Contoh : Cara 'sentinel' pada konstruksi do-while dengan memakai nilai 0 pada variabel panjang dan variabel lebar.

Output:

```
PROGRAM MENGHITUNG LUAS PERSEGI PANJANG

Panjang [0=selesai] : 20
Lebar [0=selesai] : 13
Luas = 20 x 13 = 260

Panjang [0=selesai] : 4
Lebar [0=selesai] : 0
Luas = 4 x 0 = 0

=====selesai=====
```

```
int p, l, luas;
Scanner input=new Scanner(System.in);
System.out.println("PROGRAM MENGHITUNG LUAS PERSEGI PANJANG\n");
do{
    System.out.print("Panjang [0=selesai] : ");
    p=input.nextInt();
    System.out.print("Lebar [0=selesai] : ");
    l=input.nextInt();
    luas=p*l;
    System.out.printf("Luas = %d x %d = %d\n\n",p,l,luas);
}while((p!=0)&&(l!=0));
System.out.print("=====selesai=====");
```

Cara Pertanyaan

- Contoh : Cara 'Pertanyaan' pada konstruksi While

Output:

```
PROGRAM MENGHITUNG LUAS PERSEGI PANJANG

Panjang : 14
Lebar : 10
Luas = 14 x 10 = 140

Apakah anda ingin menghitung kembali? <Y/T>
y
Panjang : 22
Lebar : 9
Luas = 22 x 9 = 198

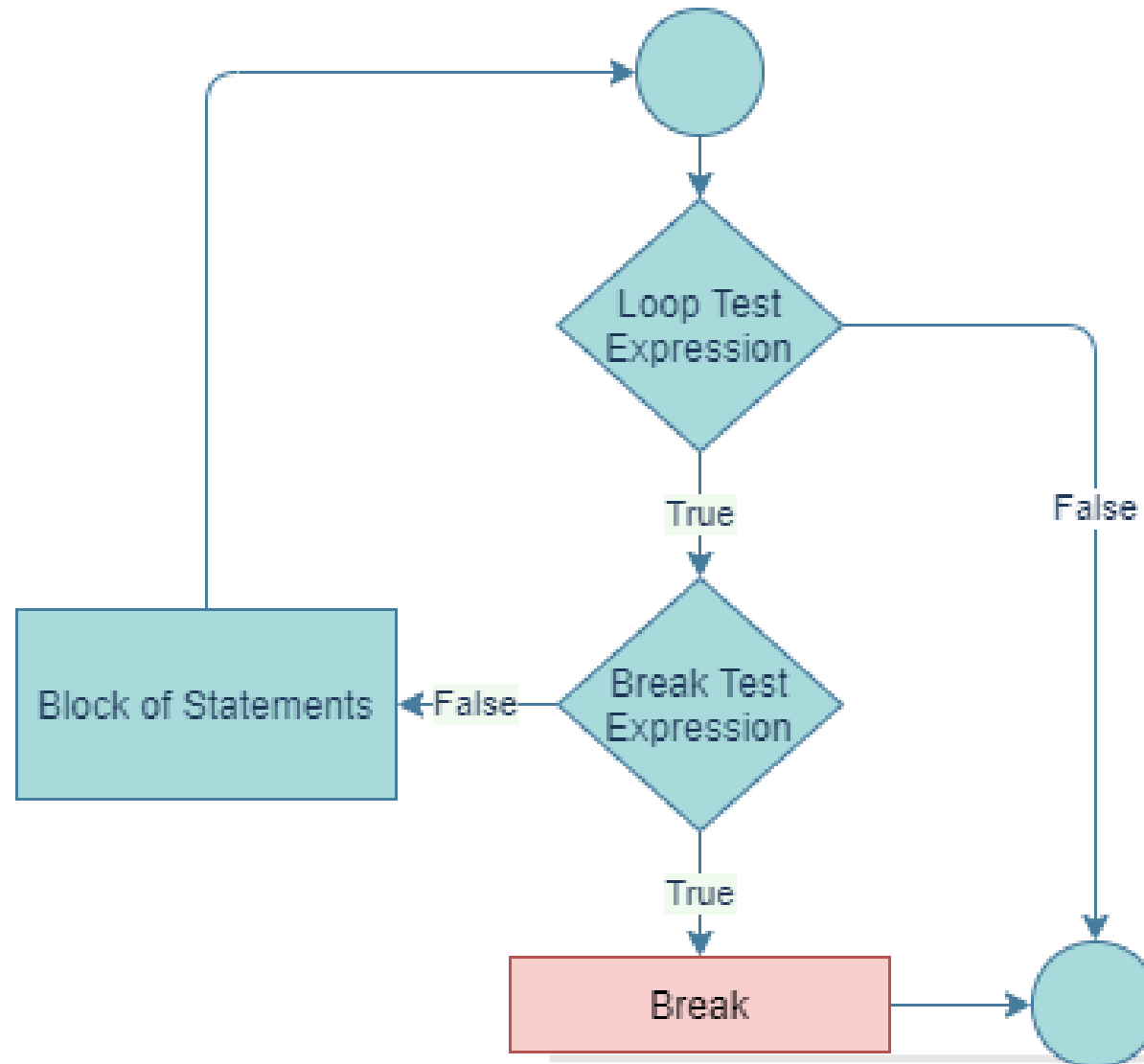
Apakah anda ingin menghitung kembali? <Y/T>
t
=====selesai=====
```

```
int p, l, luas;
String s;
Scanner input=new Scanner(System.in);
System.out.println("PROGRAM MENGHITUNG LUAS PERSEGI PANJANG\n");
do{
    System.out.print("Panjang : ");
    p=input.nextInt();
    System.out.print("Lebar : ");
    l=input.nextInt();
    luas=p*l;
    System.out.printf("Luas = %d x %d = %d\n\n",p,l,luas);
    System.out.println("Apakah anda ingin menghitung kembali? <Y/T>");
    s=input.next();
}while((s.charAt(0)=='Y')||(s.charAt(0)=='y'));
System.out.print("=====selesai=====");
```

Statement Break

- break
 - Terkadang suatu program perlu untuk keluar dari perulangan
 - Pernyataan break akan menghentikan paksa perulangan dan kode diluar perulangan akan dieksekusi
 - Digunakan untuk keluar dari perulangan (for, while dan do-while)
 - Digunakan untuk keluar dari switch

Alur Perulangan yang menggunakan break

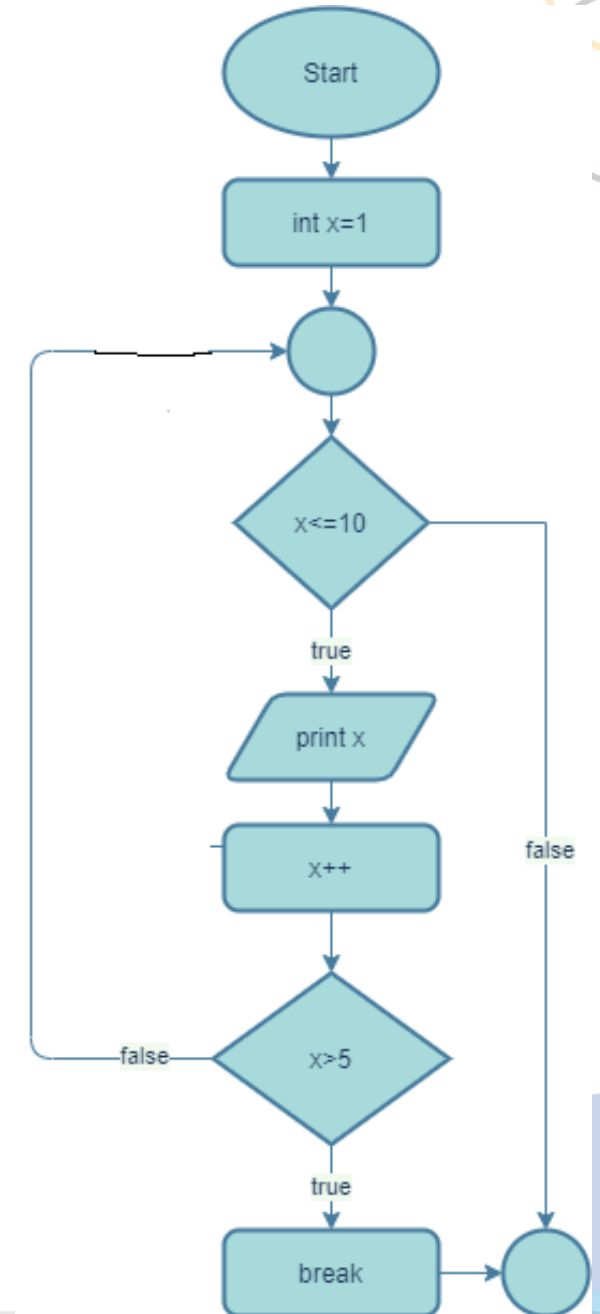


break

- Contoh penggunaan statement break pada loop yang menyebabkan program keluar dari loop tersebut

```
int x = 1;  
while (x<=10) {  
    System.out.print( x + "\n" );  
    x++;  
    if (x>5) break;  
}
```

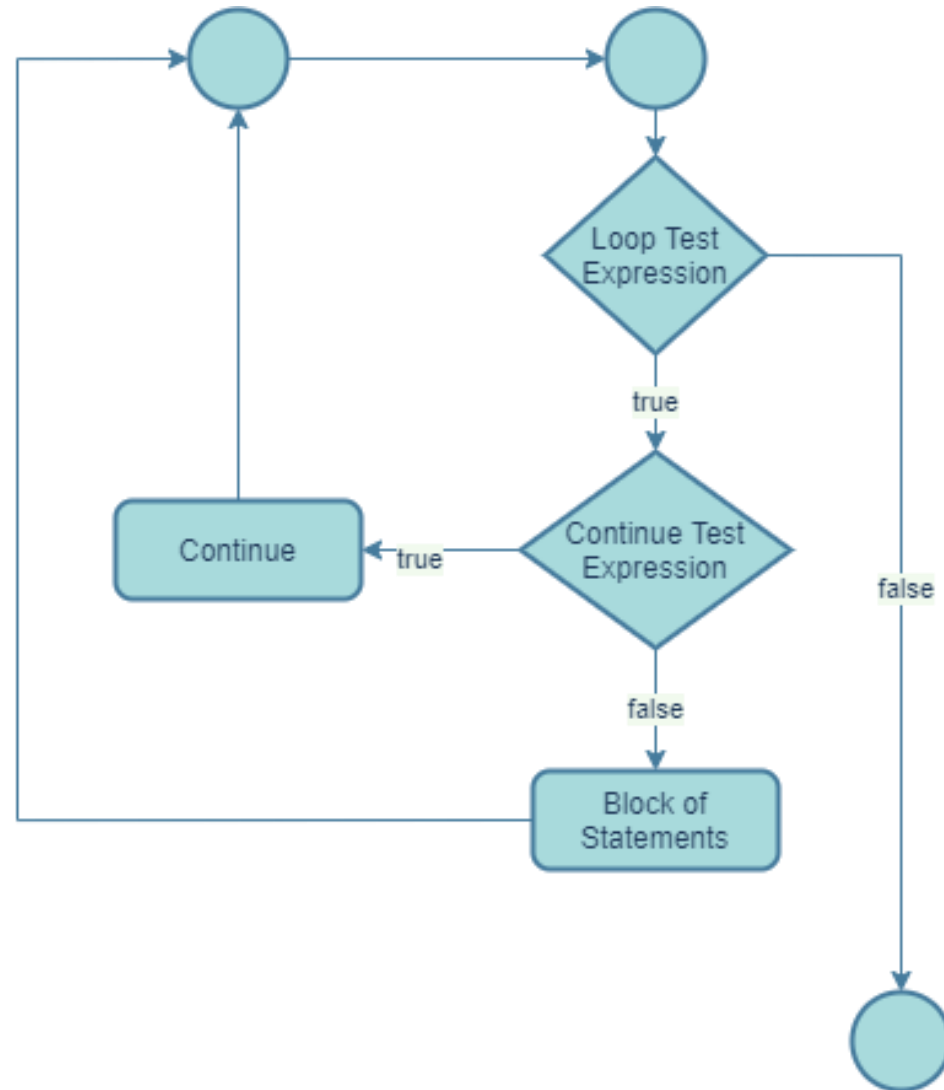
Keluar dari loop



Statement Continue

- continue
 - Bisa jadi program perlu untuk berhenti dari perulangan di tengah dan memulai kembali dari awal
 - menghentikan perulangan yang saat ini terjadi (1 iterasi saja)
 - kemudian melanjutkan perulangan iterasi berikutnya, atau bisa disebut juga untuk 'melewati' 1 perulangan.
 - skip sisa instruksi dalam loop, dan eksekusi loop berjalan ke tahap selanjutnya

Alur Perulangan yang menggunakan continue

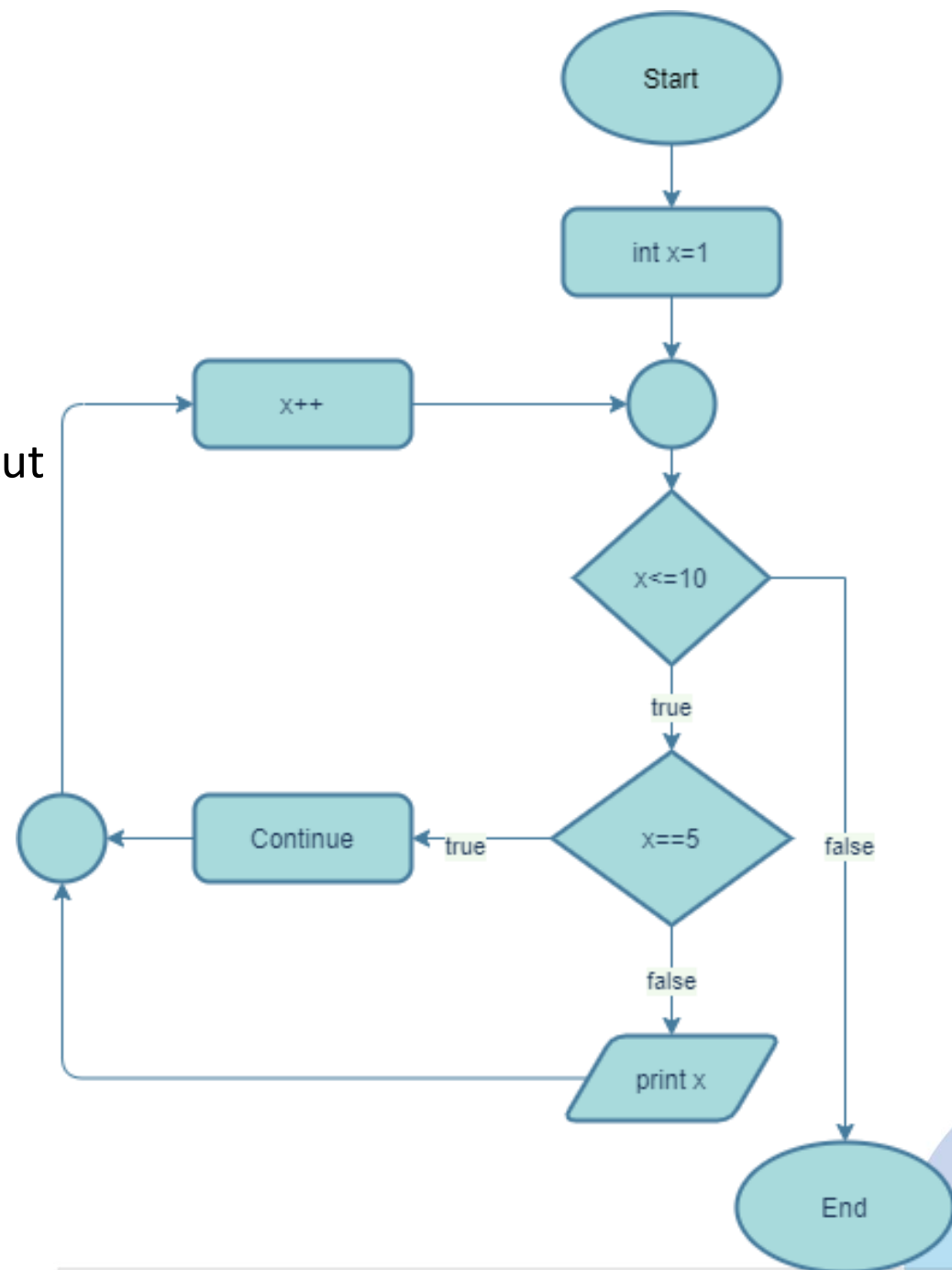


continue

- Contoh:

Buat program dengan output deret angka berikut

Output : 1 2 3 4 6 7 8 9 10





continue

- Contoh:

```
int x;  
for(x=1; x<=10; x++) {  
    if (x == 5) continue;  
    System.out.print(x);  
}
```

Output : 1 2 3 4 6 7 8 9 10

Latihan

Buat flowchart soal-soal di bawah ini dengan menggunakan for, while atau do – while!

1. Menampilkan bilangan ganjil dari 11 s/d 188
2. Menampilkan hasil penjumlahan deret bilangan 1 s/d 30
3. Menghitung hasil pangkat X^Y dengan X dan Y dari inputan pengguna
4. Menampilkan output deret bilangan 2 4 8 16 32 ... 256