BAB 5 PERINTAH PERULANGAN DAN ALGORITMA PERULANGAN

Kuliah algoritma dan pemrograman

Pendahuluan

Saat membuat suatu program setiap instruksi bisa dimulai dari yang pertama sampai dengan instruksi terakhir,

kemudian setiap instruksi yang dikerjakan juga membutuhkan suatu pilihan berdasarkan kondisi syarat tertentu

kemudian juga instruksi yang bersifat mengulang sesuai dengan kondisi yang di definisikan.

Perintah Apa Yang Diulang?

Struktur perulangan digunakan untuk mengulang sekumpulan perintah sesuai dengan kondisi yang diberikan.

Proses perulangan biasanya digunakan untuk mengulang proses pemasukan data, mengulang proses perhitungan dan mengulang untuk proses penampilan hasil pengolahan data.

StrUktUr Perulangan:

Secara Umum Struktur Perulangan dibagi menjadi 4 bagian:

- **Inisialisasi,** yaitu aksi yang dilakukan sebelum perulangan dilakukan pertama kali.
- **Kondisi perulangan,** yaitu suatu ekspresi boolean yang harus dipenuhi untuk melakukan proses perulangan.
- **Badan perulangan,** yaitu satu atau lebih instruksi yang akan di ulang.
- **Terminasi,** yaitu aksi yang mengakibatkan perulangan dihentikan.

Instruksi/Notasi Algoritmik

- Saat Melakukan proses perulangan seperti halnya struktur selection atau percabangan yang membutuhkan instruksi atau notasi Algoritmik kondisi untuk memproses.
- Instruksi untuk setiap bahasa pemrograman pada prinsipnya sama yaitu melakukan proses perulangan sesuai dengan kondisi yang ditentukan.
- Perbedaannya adalah tergantung dari struktur program (syntax) dan perintahnya.

Intruksi For()

- Digunakan untuk menghasilkan pengulangan sejumlah kali yang dispesifikasikan.
- Jumlah pengulangan diketahui atau dapat ditentukan sebelum eksekusi.

SINTAK DAN CONTOH

Sintak :

for(nilai awal;kondisi;(terminasi)increment/decrement)

{statement yang diulang}

Contoh :

```
for(i=1;i<=3;i++)
  {
    System.out.println("ulang ke-"+i);
  }
```

INSTRUKSI WHILE

- Pernyataan **while adalah** pernyataan yang berguna untuk memproses suatu pernyataan atau pernyataan beberapa kali jika kondisi benar.
- Pernyataan atau aksi akan di ulang jika kondisi bernilai benar dan jika salah maka keluar dari blok perulangan (loop)

Sintak dan contoh

Sintak :

nilai awal

While(kondisi)

{

statement yang diulang

terminasi;

}

Contoh while

```
i=1; // nilai awal
while(i<=3) //kondisi
{
System.out.println("cetak ke-"+i); //bag ulang
i++; //terminasi
}
```

Do {} While ()

- Perulangan akan dilakukan minimal 1x terlebih dahulu, kemudian baru dilakukan pengecekan terhadap kondisi, jika kondisi **benar** maka perulangan masih akan tetap dilakukan.
- Perulangan dengan do{}while() akan dilakukan sampai kondisi **salah**
- Beda Minimal perulangan yang lain yang bernilai 0 Minimal do while minimal perulangan adalah 1

Sintak do while

Nilai awal (jika diperlukan)

Do

{

badan perulangan

terminasi (bisa sekaligus nilai awal)

}

while (kondisi)

Contoh code

```
i=1;//nilai awal
do
  {
   System.out.println("cetak ke-"+i);//bag yg diulang
   i++;//terminasi
  }
while(i<=3); //kondisi
```

Algoritma Perulangan

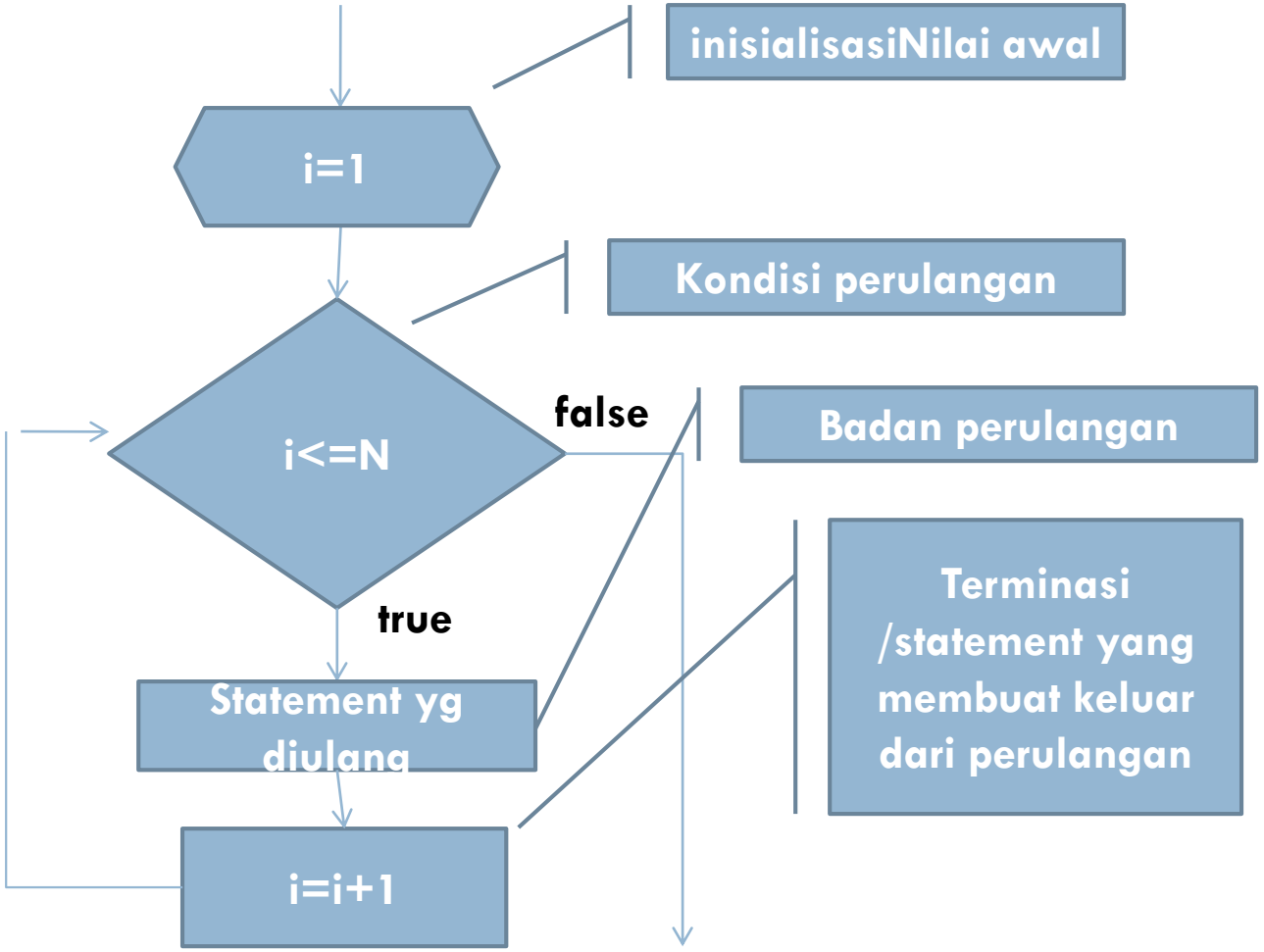
Pseducode for

For(nilaiawal;kondisi;increment/
decrement){statement}

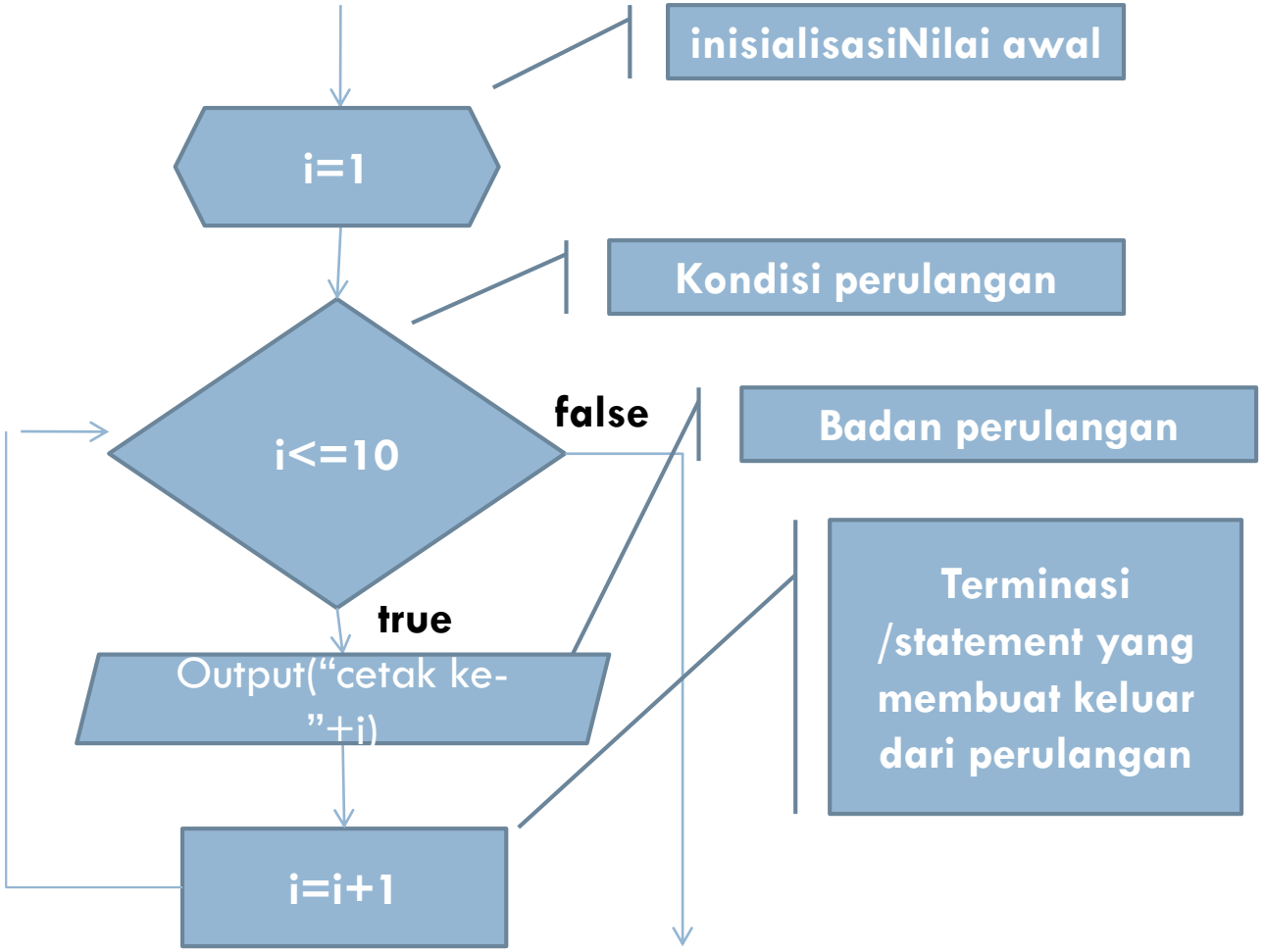
Contoh mencetak String 10 kali:

```
For( i=1;i<=10;i++ )
 {output("ulang ke-"+i);}
```

Contoh Flowchart for



Flowchart cetak string 10 kali



Pseducode while (){}

Nilai awal

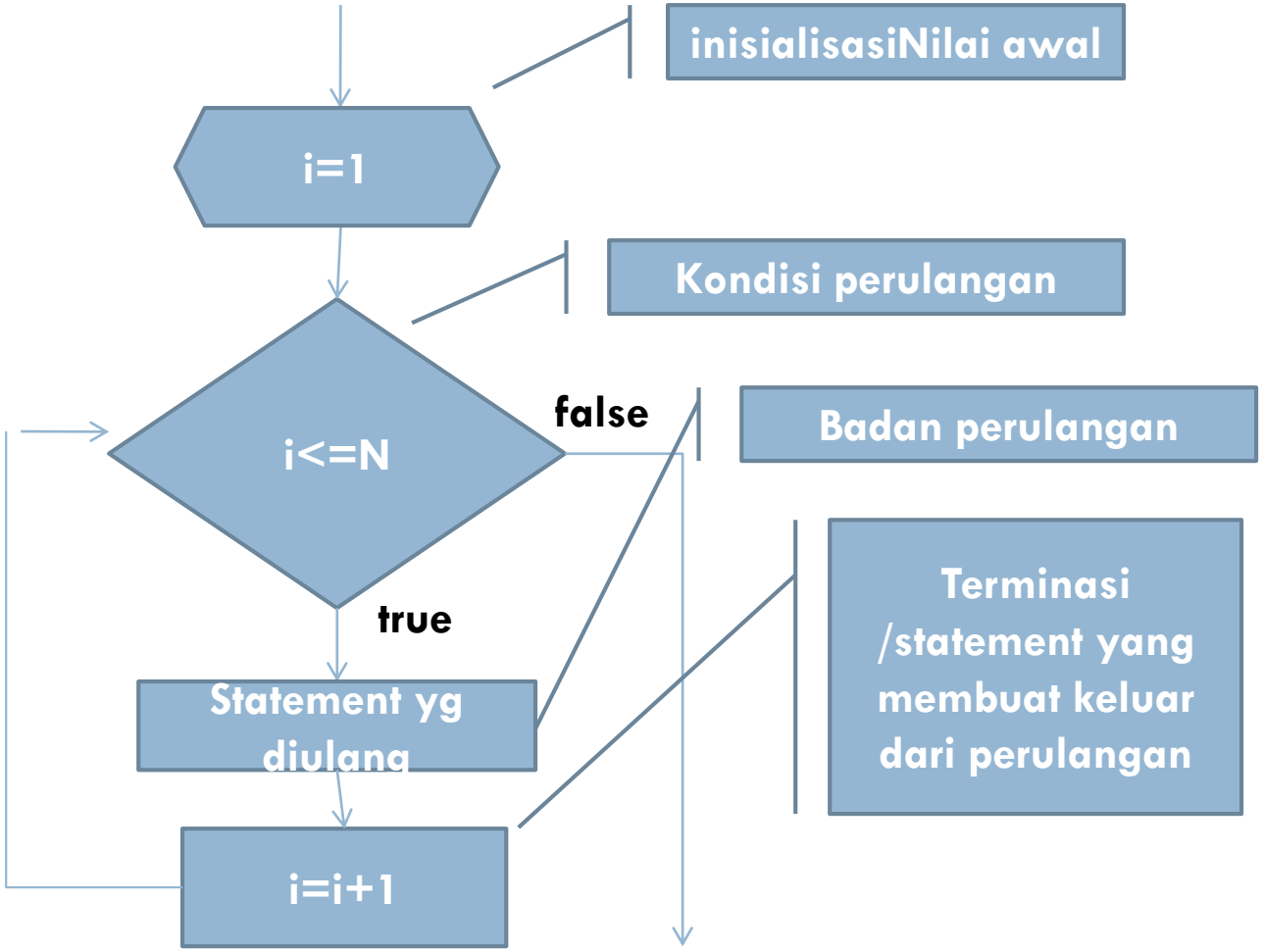
While (kondisi)

{statement (terdapat statement yang membuat perulangan berhenti)}

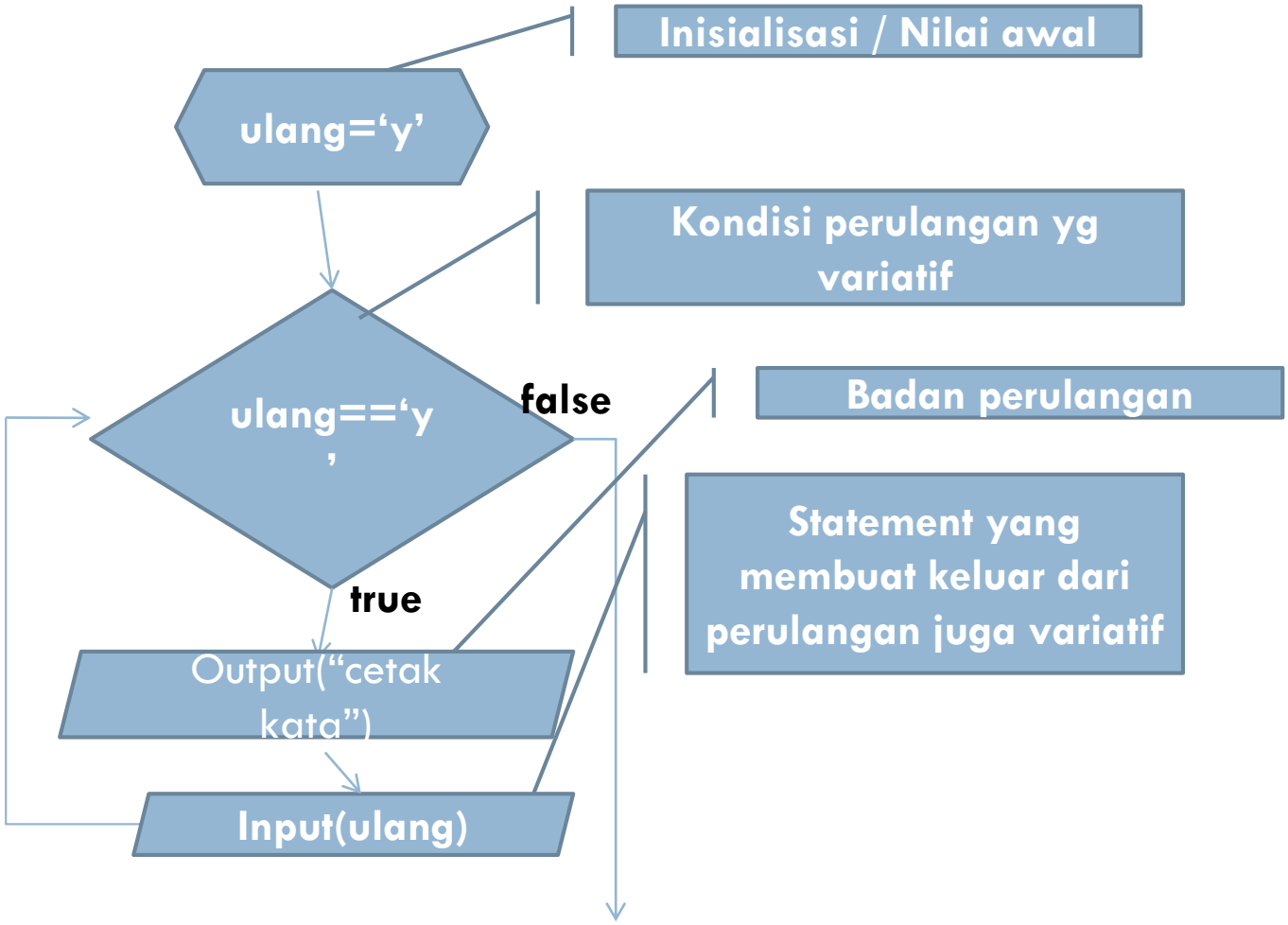
Contoh pseducode cetak bilangan 1 sampai 10

```
x=1;
While(x<=10){
output(x);
x++;}
```

Contoh Flowchart while (sama dengan for hanya kondisinya dan terminator dapat lebih bervariatif)



Flowchart cetak Sring sampai konfirmasi selesai cetak



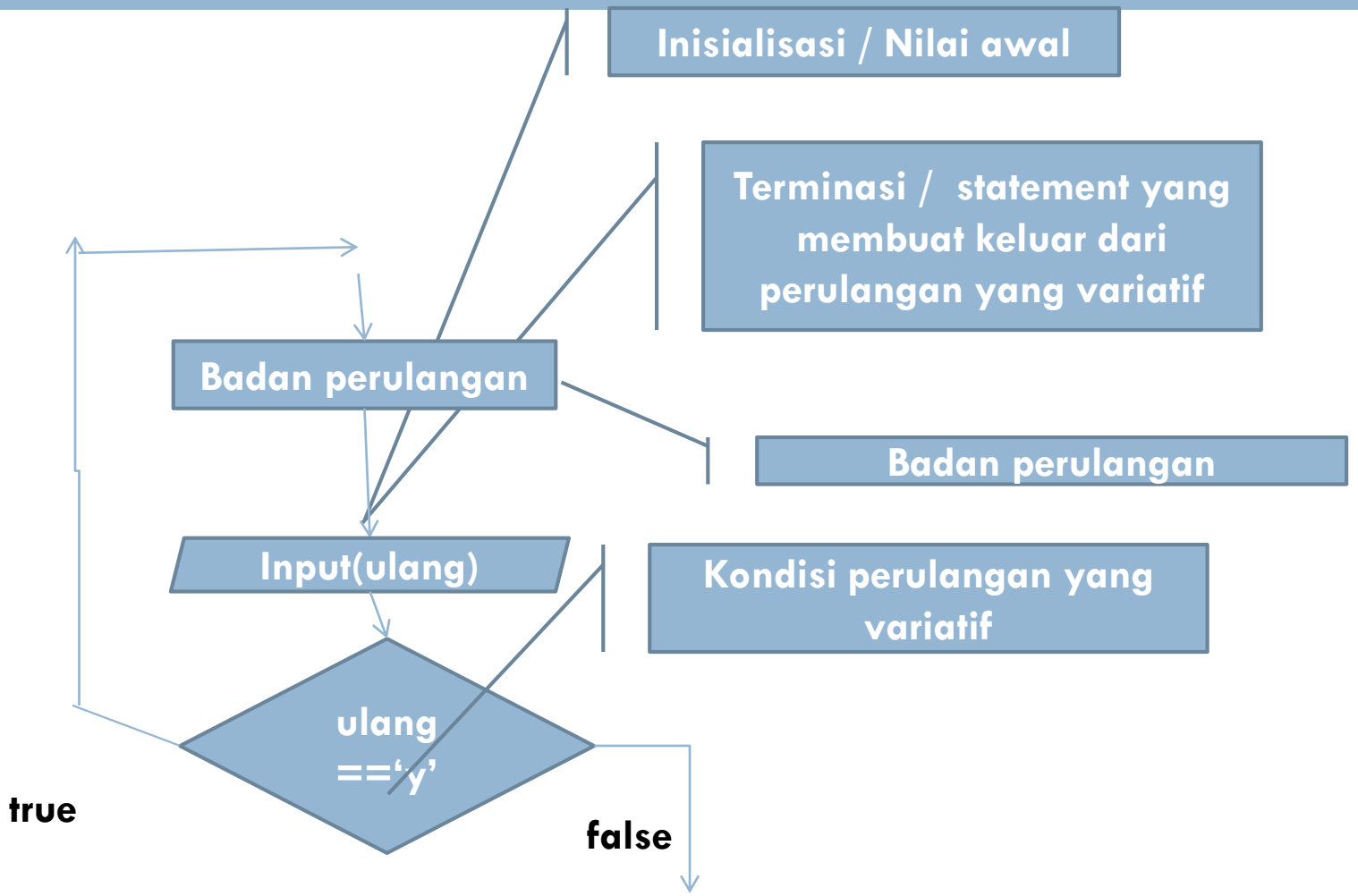
Pseducode do{}while()

Do {Statement}
While (Kondisi)

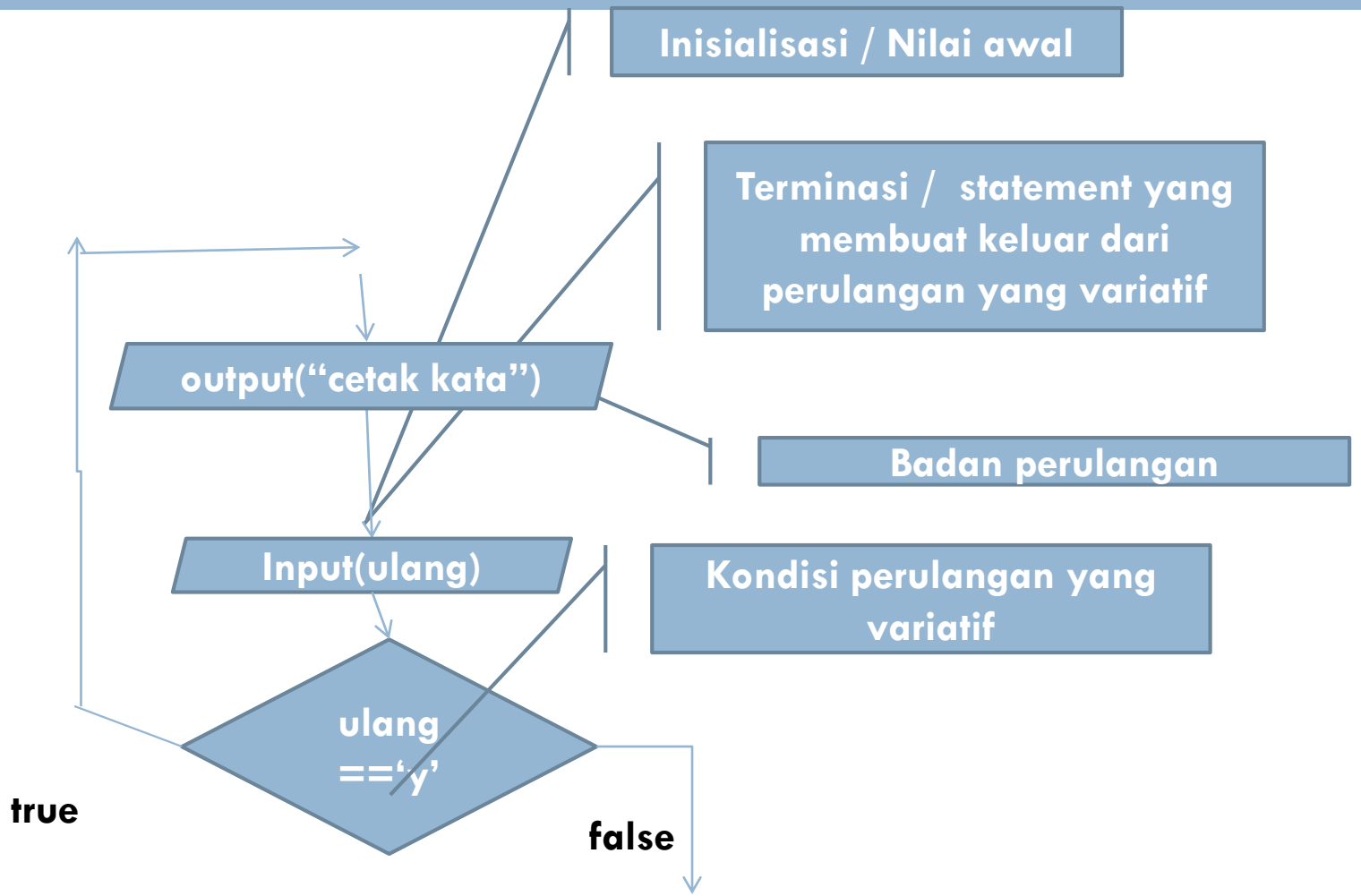
Contoh pseducode cetak angka 1 sampai 10

```
x=1;
Do {
Output(x);
x=x+1;
}
While (x<=10)
```

Flowchart


Inisialisasi / Nilai awal
Terminasi / statement yang membuat keluar dari perulangan yang variatif
Badan perulangan
Badan perulangan
Input(ulang)
Kondisi perulangan yang variatif
ulang
=='y'
true
false

Flowchart cetak String dengan konfirmasi



Contoh pseducode

Menampilkan bilangan ganjil dari 1 sampai 100

1.Mulai

2.int i=1

3.while(i<=100){

a. if(i % 2==1){output(i);}

b. i=i+1

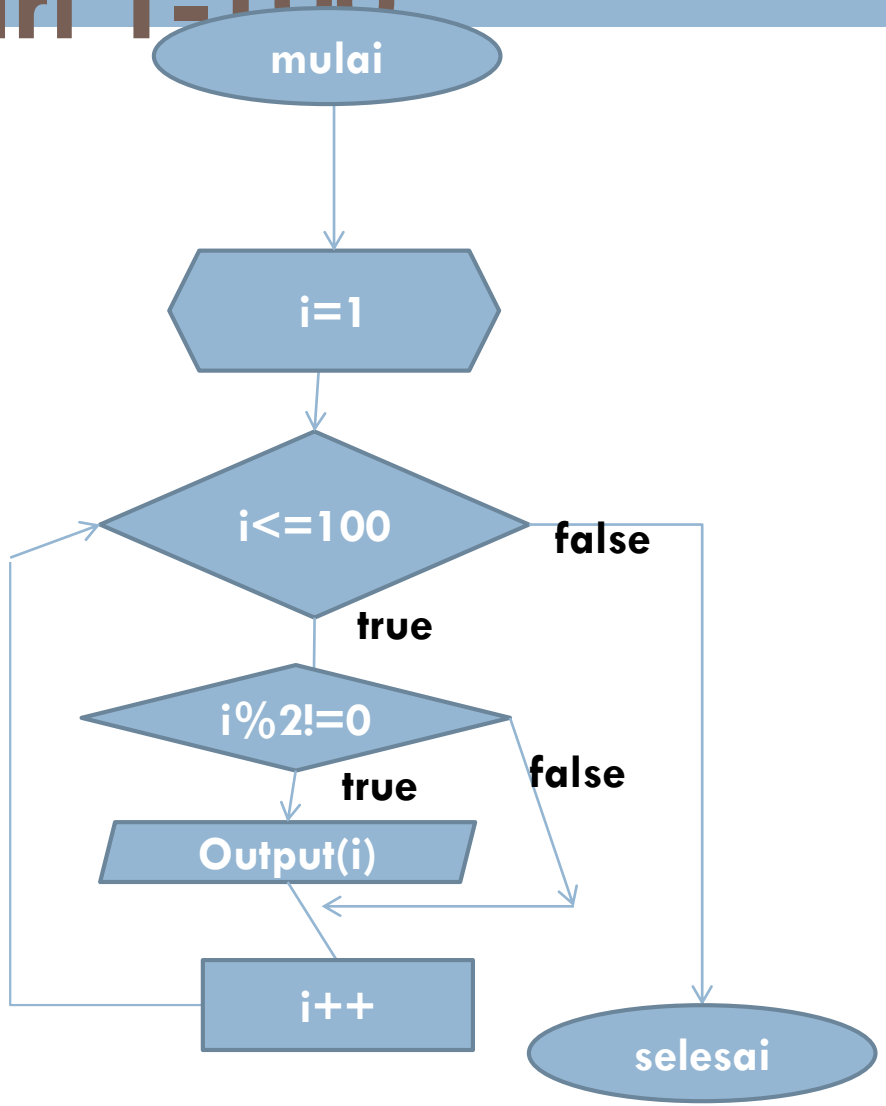
}

4.selesai

Perbedaan flowchart perulangan dengan percabangan

- Perulangan terdapat kembali ke atas sebelum kondisi tetapi percabangan tidak ada
- Perulangan antara true dan false jalurnya tidak bertemu sedang percabangan bertemu
- Perulangan hanya diperlukan statement kodisi true sedangkan percabangan diperlukan keduanya

Flowchart Menampilkan bilangan ganjil dari 1-100



Latihan 1

1. Algortima menampilkan angka diantara 2 nilai yang dimasukan, misal nilai yang di masukan adalah 2 dan 10, maka outputnya adalah : **2 3 4 5 6 7 8 9 10**
2. Algoritma untuk mencetak karakter dari suatu inputan dan akan ber akhir jika nilai dari input bernilai ‘x’ atau ‘X’ (input ‘a’ akan tercetak a dilayar dst sampai menginput ‘x’ atau ‘X’ baru keluar program)
3. Algoritma penjumlahan deret ke N bil asli pertama.(misal n=4 maka hasilnya =1+2+3+4=10)
4. Algoritma menghitung rata-rata dari n bilangan
5. Algoritma menghitung rata-rata dari sejumlah bilangan yang diinput sebelum konfirmasi selesai input
6. Algoritma penjumlahan deret aritmatika (Un=a+(n-1)b), dengan perulangan bukan dengan rumus Sn!
7. Mengkalikan n bilangan yang diinputkan (misal n=3 bilangan :yang diinput 5,3,4 maka hasilnya 5*3*4=60)

Latihan 2
buatlah pseducode dan flowchart

1. Algortima menghitung n!(n faktorial) (tidak menggunakan fungsi faktorial yg tersedia di java)
2. Algoritma menghitung X pangkat Y (tidak menggunakan fungsi pangkat yg tersedia di java)
3. Algoritma menghitung jumlah deret geometri (Un=a.r^(n-1)) dengan perulangan bukan dengan rumus Sn
4. Algoritma Menghitung KPK dari 2 bilangan integer (tidak menggunakan fungsi KPK yg tersedia di java)
5. Algoritma Menghitung FPB dari 2 bilangan integer (tidak menggunakan fungsi FPB yg tersedia di java)