

Perulangan - C++

Bahan Kuliah SD3105 Algoritma dan Pemrograman

Sevi Nurafni

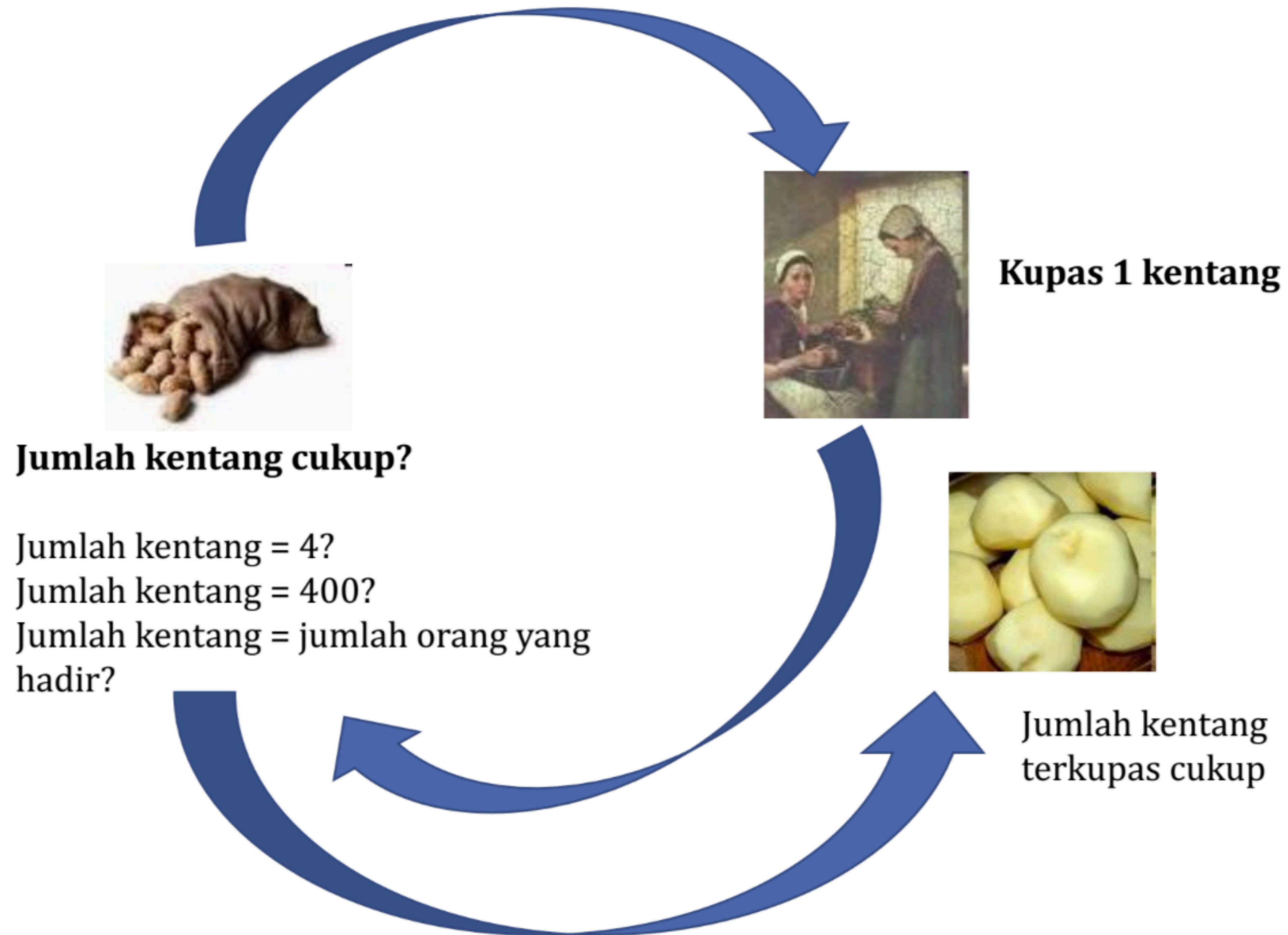
Fakultas Sains dan Teknologi

Universitas Koperasi Indonesia 2024

Menyiapkan kentang untuk makan malam



- Asumsi: jumlah kentang tersedia tidak terbatas
- Pada suatu hari Ibu hanya mengupas kentang hanya 4 buah karena hanya anggota keluarga saja yang makan malam
- Pada hari yang lain, Ibu mengundang mahasiswa SD3105 sejumlah 400 orang untuk makan malam di rumahnya sehingga ibu mengupas 400 kentang untuk semua orang
- Hari yang lain, ibu tidak tahu berapa jumlah orang yang akan makan malam
 - Setiap selesai mengupas 1 kentang, dicek apakah jumlah cukup atau tidak



Menulis 1 s.d. 3

- Tuliskan program yang menuliskan angka 1 s.d. 3 dan selanjutnya 1+2+3 ke layar
- Contoh keluaran:

```
1  
2  
3  
6
```

```
#ALGORITMA  
print(1)  
print(2)  
print(3)  
print(1+2+3)
```



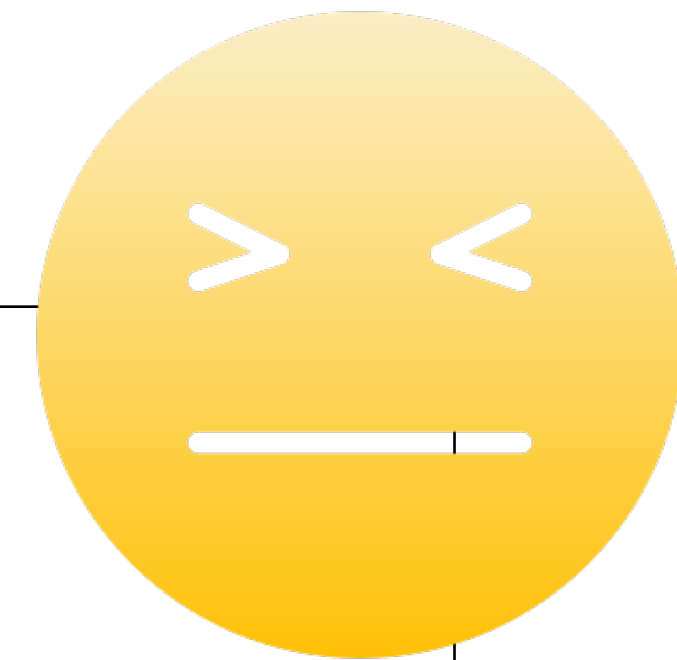
Menulis 1 s.d. 3

- Tuliskan program yang menuliskan angka 1 s.d. 10 dan selanjutnya $1+2+3 + \dots + 10$ ke layar
- Contoh keluaran:

```
1
2
3
4
5
6
.
.
10
55
```

```
#ALGORITMA
print(1)
print(2)
print(3)
print(4)
print(5)

...
print(10)
print(1+2+3+4+5+6+..+10)
```



Menulis 1 s.d. 100

- Tuliskan program yang menuliskan angka 1 s.d. 100 dan selanjutnya 1+2+3 +... +100 ke layar
- Contoh keluaran:

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

```
#ALGORITMA  
print(1)  
print(2)  
print(3)  
print(4)  
print(5)  
  
...  
print(100)  
print(1+2+3+4+5+6+..+100)
```



Bagaimana kalau...

- Anda diminta menulis dan menjumlahkan..

1 s.d 1000??

1 s.d. 10000??

1 s.d. 1000000??

...



Pengulangan: Latar Belakang

- Melakukan suatu instruksi, bahkan aksi, secara berulang-ulang
- Komputer: memiliki performansi yang sama
- Manusia: punya kecenderungan untuk melakukan kesalahan (karena letih atau bosan)



Pengulangan (Looping)



- Elemen:
 - **Kondisi pengulangan:** ekspresi logik
 - **Badan pengulangan:** aksi yang diulang
- Jenis-jenis notasi pengulangan di Python:
 - Berdasarkan kondisi mengulang di akhir: **do-while**
 - Berdasarkan pencacah: **for**
 - Berdasarkan kondisi mengulang di awal: **while**

Contoh-1

- Tuliskan program yang menerima masukan sebuah integer misalnya N dan menuliskan angka 1, 2, 3, ..., N dan menuliskan $1+2+3+\dots+N$ ke layar.
- Asumsikan $N > 0$
- Contoh:

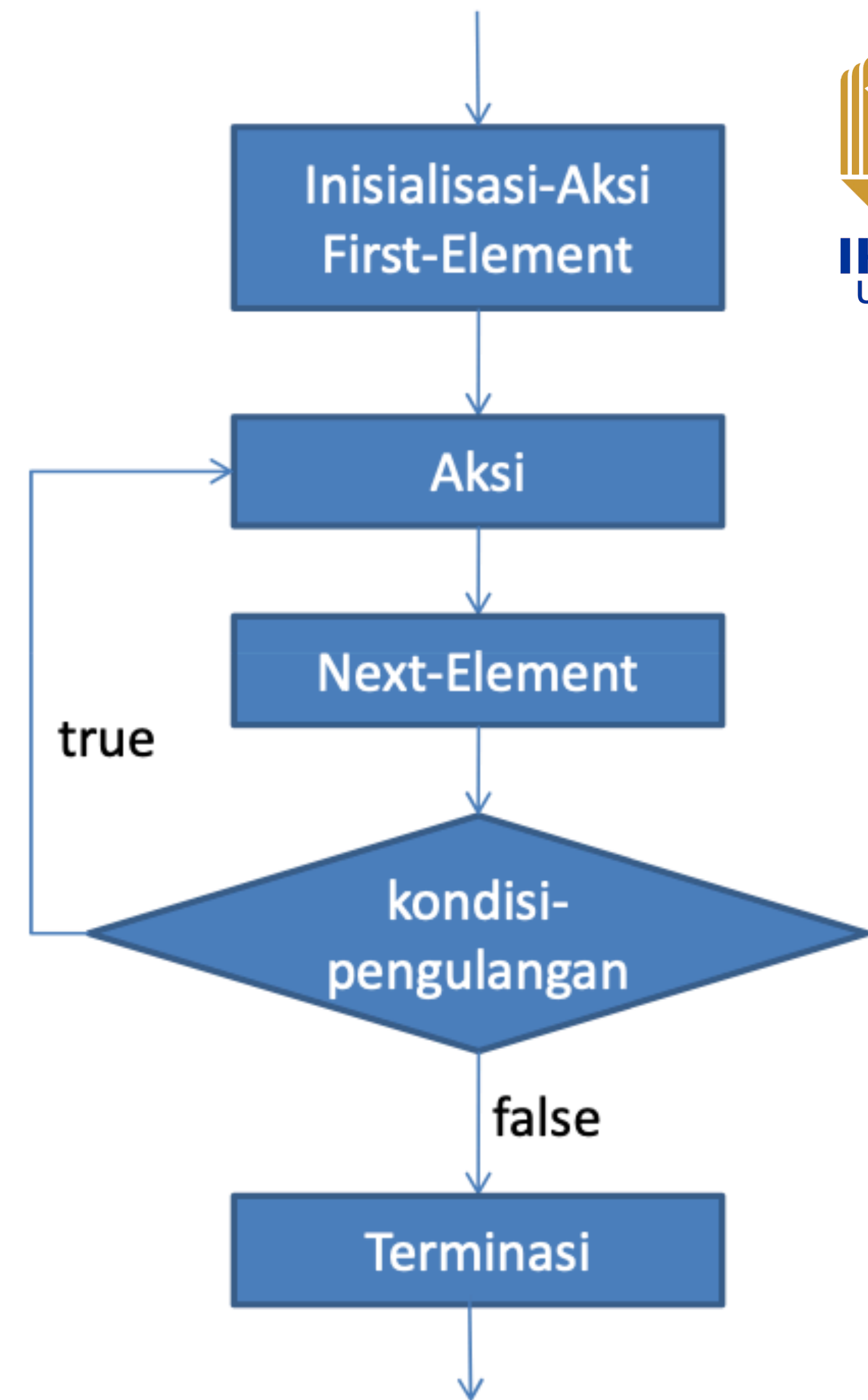
$N = 1$
Tampilan di layar:
1
1

$N = 5$
Tampilan di layar:
1
2
3
4
5
15

$N = 10$
Tampilan di layar:
1
2
3
4
5
6
7
8
9
10
55

Do-while

```
Inisialisasi-aksi  
First-Element  
do {  
    Aksi  
    Next-Element  
} while (kondisi-pengulangan);  
Terminasi
```



do-while

- Pengulangan dikendalikan oleh elemen pengulangan yang diinisialisasi sebagai `First-Element` dan diubah nilainya dalam badan pengulangan menjadi `Next-Element`
- Aksi minimal akan dilakukan satu kali karena pada waktu eksekusi pengulangan yang pertama tidak dilakukan test terhadap kondisi-pengulangan
- Aksi akan dihentikan jika kondisi-pengulangan tidak dipenuhi (berharga `false`), akan diulang jika kondisi-pengulangan tercapai
- Test terhadap kondisi pengulangan dilakukan setelah Aksi dilaksanakan
- Pengulangan berpotensi mengalami “**kebocoran**”, jika ada kemungkinan bahwa seharusnya Aksi tidak pernah boleh dilakukan untuk kasus tertentu

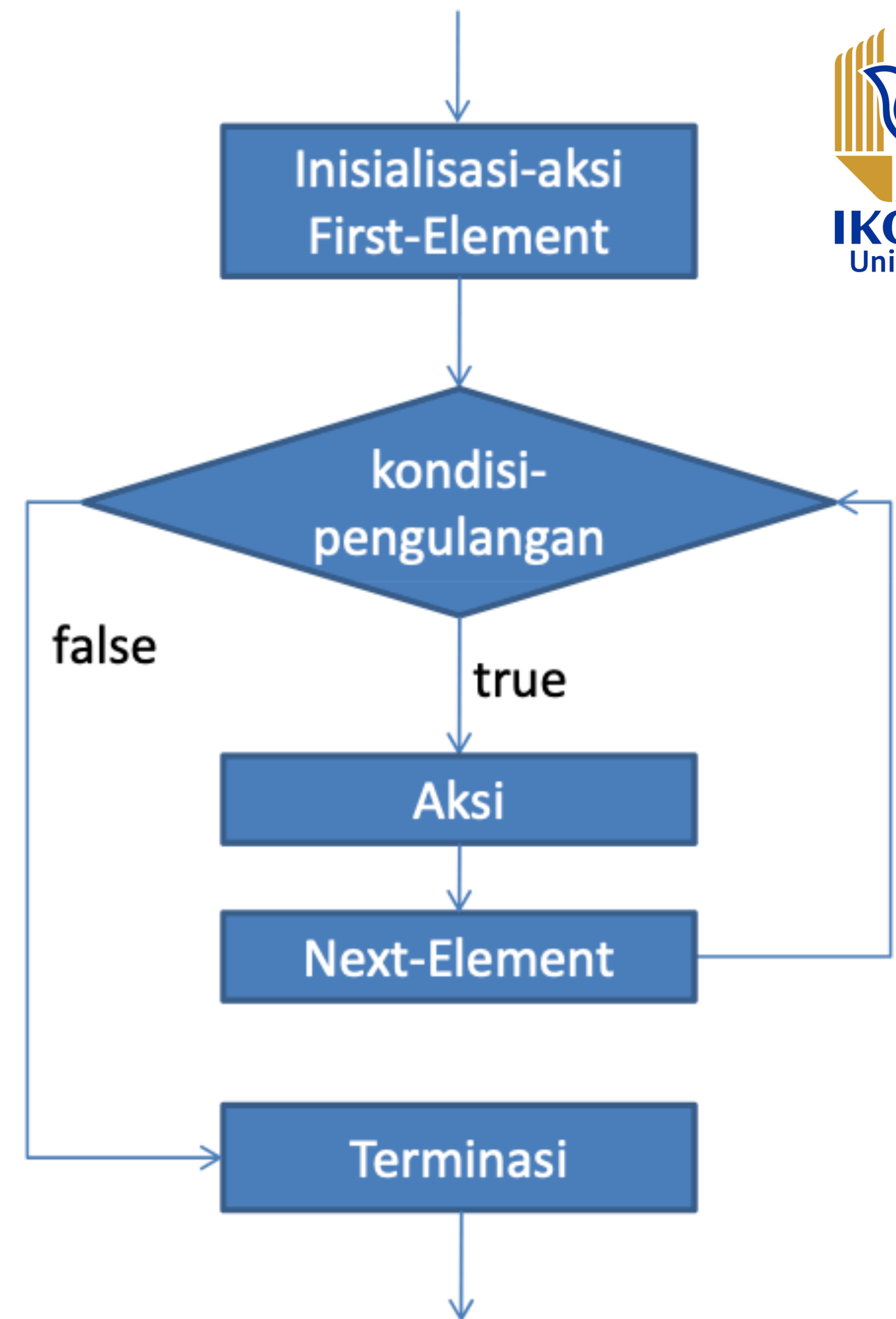
Contoh-1: do-while

```
# Program JumlahAngka
# Menghitung 1+2+3+...+N. Asumsi N > 0

#include <iostream>
using namespace std;
int main () {
// KAMUS
    int N, i, sum;
// ALGORITMA
    cin >> N;
    sum = 0                //Inisialisasi-aksi
    i = 1;                 //First-Element
    do {
        cout << i << endl; //Aksi
        sum = sum + i;      //Aksi
        i = i + 1;          //Next-Element
    } while (i <= N);       //Kondisi Pengulangan
    cout << sum << endl;    //Terminasi
    return 0;
}
```

while

```
Inisialisasi-aksi  
First-Element  
  
while (kondisi-pengulangan) {  
    Aksi  
    Next-Element  
}  
//Kondisi-pengulangan = false  
Terminasi
```



while

- Pengulangan dikendalikan oleh elemen pengulangan yang diinisialisasi sebagai `First-Element` dan diubah nilainya dalam badan pengulangan menjadi `Next-Element`
- Aksi akan dilakukan selama kondisi-pengulangan masih dipenuhi (bernilai `true`)
- Tes terhadap kondisi-pengulangan dilakukan setiap kali sebelum aksi dilaksanakan
- Pengulangan ini berpotensi untuk menimbulkan Aksi “**kosong**” (tidak pernah melakukan apa-apa) karena pada test yang pertama, kondisi-pengulangan tidak dipenuhi (berharga `false`) sehingga langsung ke luar loop

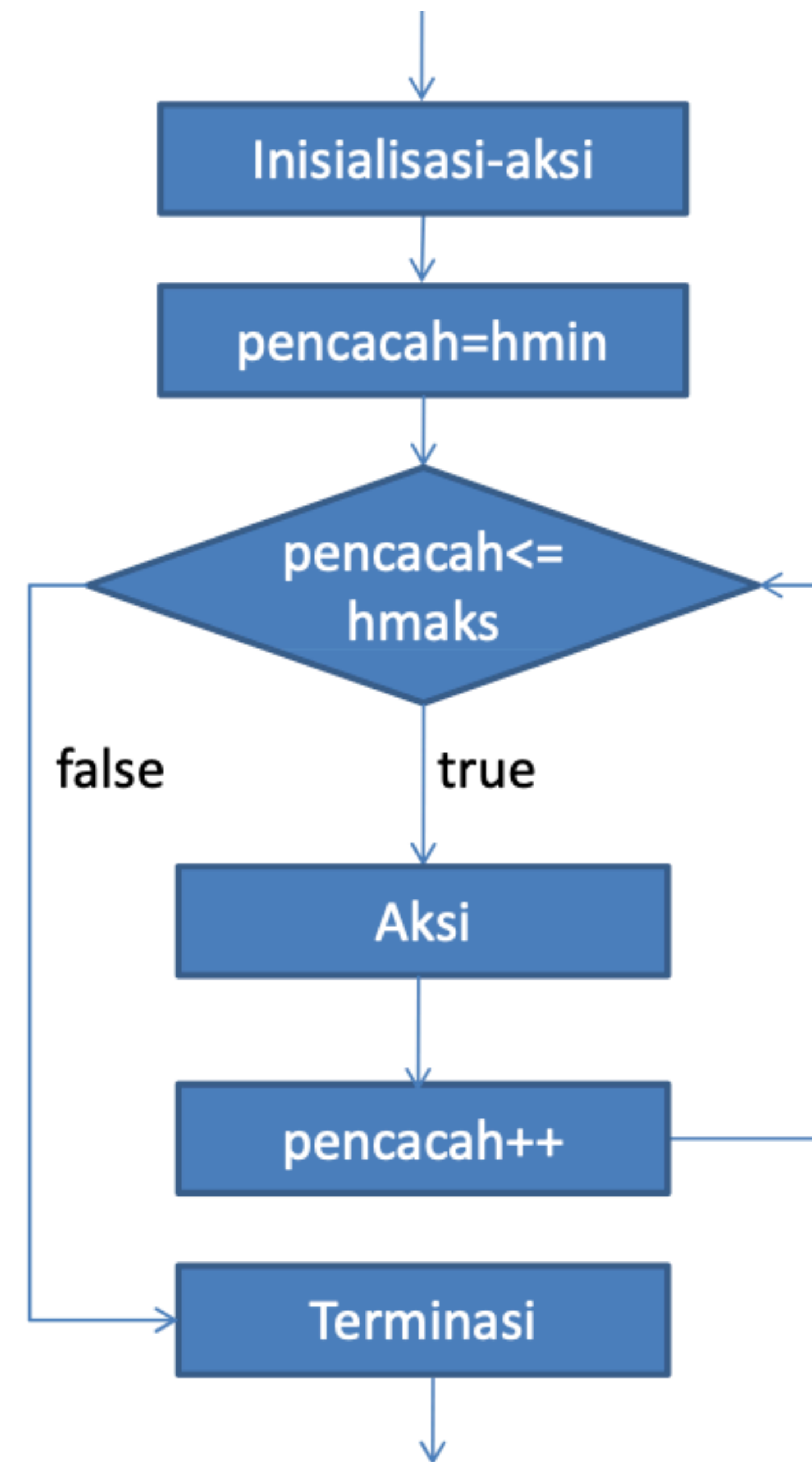
Contoh-1: while

```
# Program JumlahAngka
# Menghitung 1+2+3+...+N. Asumsi N > 0

#include <iostream>
using namespace std;
int main () {
// KAMUS
    int N, i, sum;
// ALGORITMA
    cin >> N;
    sum = 0                //Inisialisasi-aksi
    i = 1;                 //First-Element
    while (i <= N){         //Kondisi Pengulangan
        cout << i << endl; //Aksi
        sum = sum + i;     //Aksi
        i = i + 1;         //Next-Element
    } //i>N
    cout << sum << endl;   //Terminasi
    return 0;
}
```


for

```
Inisialisasi-aksi  
First-Element  
  
for (pencacah = hmin;  
      pencacah <= hmaks; pencacah++)  
{  
    Aksi  
}  
  
Terminasi
```



For



- Pencacah harus suatu variable dengan type yang terdefinisi suksesor dan predesesornya, misalnya integer
- Nilai pencacah adalah dari hmin s.d. hmaks
- Aksi akan dilakukan dengan memperhitungkan harga-harga dari pencacahan yang di-“jelajahi”
- Harga pencacahan di-increment melalui proses `pencacah++` (alias `pencacah=pencacah+1`), setiap kali Aksi selesai dilaksanakan
- Jika `pencacah=hmaks`, maka pengulangan berhenti

Contoh-1: for

```
# Program JumlahAngka
# Menghitung 1+2+3+...+N. Asumsi N > 0

#include <iostream>
using namespace std;
int main () {
// KAMUS
    int N, i, sum;
// ALGORITMA
    cin >> N;
    sum = 0                //Inisialisasi-aksi
    for (i =1; i <= N; i++){ //Kondisi Pengulangan
        cout << i << endl;  //Aksi
        sum = sum + i;      //Aksi
    } //i>N
    cout << sum << endl;    //Terminasi
    return 0;
}
```

Soal 1

- Buatlah program yang menerima masukan 10 buah bilangan integer (dari keyboard) dan menuliskan ke layar jumlah total ke-10 integer tersebut.
- Contoh

Masukan	Tampilan di Layar
2 1 0 -9 7 13 2 2 1 -1	18

Soal 2

- Buatlah program yang membaca sejumlah bilangan integer dari keyboard sampai pengguna memasukkan angka -999 (angka -999 tidak termasuk bilangan yang diolah).
- Tuliskan berapa **banyak bilangan** yang dimasukkan, **nilai total**, dan **rata-rata** semua bilangan
- Jika dari masukan pertama sudah menuliskan -999, maka tuliskan pesan “Tidak ada data yang diolah”

No	Input	Output
1	<u>-1</u> <u>12</u> <u>-6</u> <u>10</u> <u>2</u> <u>-999</u>	Banyak bilangan = <u>5</u> Jumlah total = <u>17</u> Rata-rata = <u>3.40</u>
2	<u>-999</u>	<u>Tidak ada data yang diolah</u>

Soal 3



- Buatlah program untuk menghitung mundur dari N , $N - 1$, $N - 2$, ..., 3, 2, 1, 0.
Nilai N diinputkan oleh user.

- Contoh $N=5$

Output: 5

4

3

2

1

**SELAMAT
BELAJAR**