TIB26 – ALGORITMA PEMROGRAMAN

Predicted Loop

Pertemuan 15, 16

Sub-CPMK

- Mahasiswa mampu menggunakan predicted loop pada algoritma (C3, A3)

Materi

1. Pengertian Predicted Loop
2. Pernyataan FOR
3. Variabel Pencacah
4. Nested Loop

Perhatian

- Tidak disarankan copy-paste kode program dari presentasi ini, karena ada beberapa symbol yang dianggap sebagai Unicode oleh editor yang anda gunakan, sehingga akan dianggap sebagai symbol yang salah oleh compiler, sebaiknya diketik ulang saja

1. Pengertian Predicted Loop

1.1 Pengulangan

- Salah satu struktur dasar algoritma adalah pengulangan / repetition / loop
- Pengulangan adalah mekanisme algoritma yang akan mengatur pengulangan bagian dari algoritma
- Memerlukan kondisi tertentu untuk melakukan pengulangan, sehingga tidak terjadi infinite loop

1.2 Jenis pengulangan

- Ada dua jenis pengulangan
- Predicted loop : pengulangan yang sudah diprediksi jumlah pengulangan
- Unpredicted loop : pengulangan yang jumlah pengulangannya tergantung kondisi

1.3 Predicted Loop

- Adalah pengulangan yang secara struktur sudah diperdiksi jumlah pengulangannya
- Predicted loop pada pemrograman dilakukan dengan perintah for

2. Pernyataan FOR

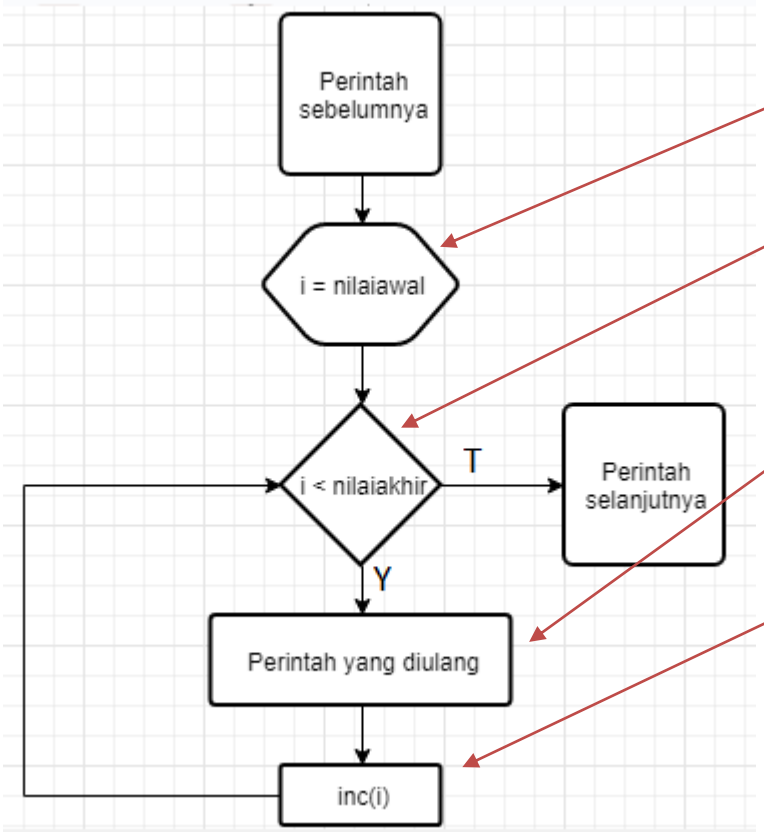
2.1 Penyataan for

- Pernyataan for merupakan pengulangan terprediksi
- Disebut terprediksi karena pada for sudah terdapat nilai awal dan nilai akhir serta counter/pencacah untuk melakukan pengulangan
- Syntax:

```
for(variabelpencacah=nilaiawal, kondisi, pencacahan)
```

- Kondisi berupa batas akhir variable pencacah, dapat berisi operator perbandingan ==, !=, >, <, >=, <=
- Pencacahan berupa penambahan atau pengurangan suatu nilai variable bertipe kardinal

2.2 Contoh Pola flowchart pernyataan for

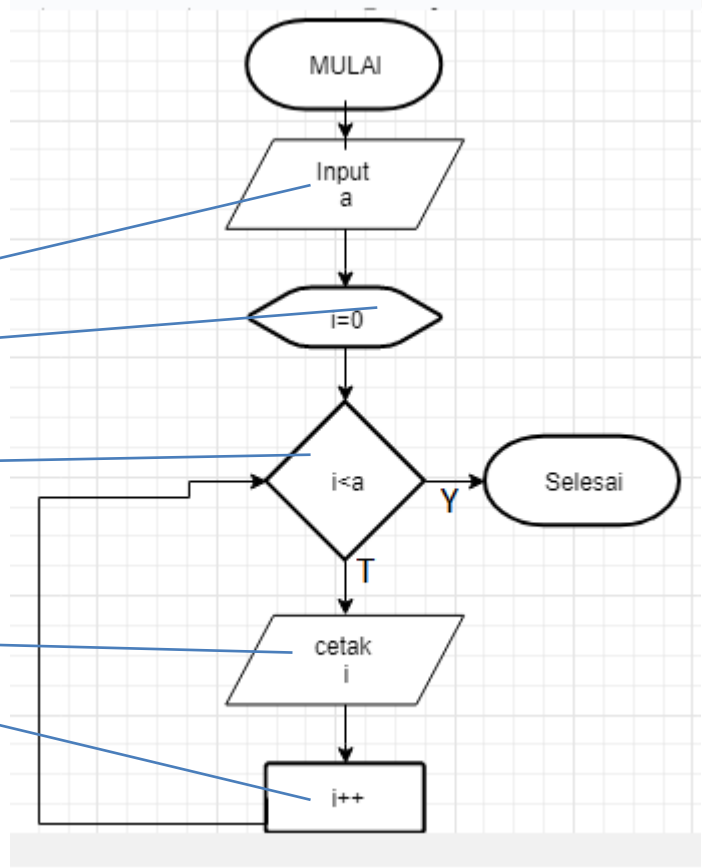


- Inisialisasi nilai awal
- Kondisi pengulangan
- Blok perintah-perintah yang diulang
- Pencacahan, dapat berupa inc ataupun decrement, dapat juga berupa penambahan/pengurangan nilai variable dengan jumlah tertentu

2.3 Contoh Pengulangan

```
#include <stdio.h>

int main(void)
{
   int a, i;
   printf("Jumlah Pengulangan: "); scanf("%d", a);
   for(int i=0;i<a;i++)
   {
      printf("%d \n", i);
   }
}
```



3. Variabel Pencacah

3.1 Variabel Pencacah

- Variabel pencacah bertipe cardinal / bulat
- Dapat berupa integer / enumerasi / character

3.2 Inisialisasi

- Pada bagian inisialisasi, variable pencacah selalu diisi dengan nilai awal. Dapat dimulai dari 0, bilangan negative ataupun bilangan positif
- Contoh: i = 0;

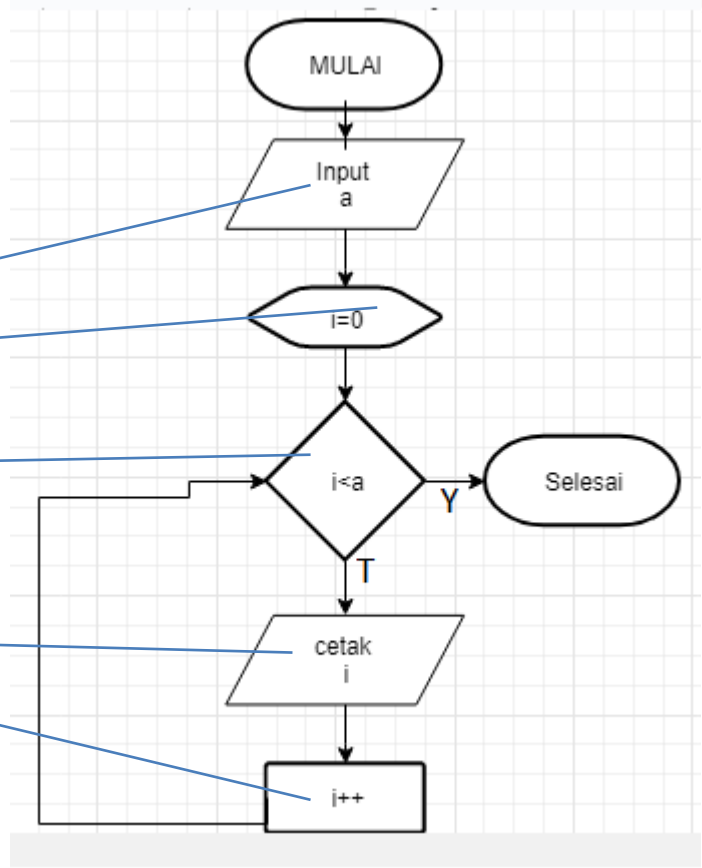
3.3 Kondisi

- Pada bagian kondisi, variabel pencacah dibandingkan dengan nilai akhir
- Operator pembanding dapat berupa ==, !=, <, <=, >, >=, disarankan menggunakan >, <, >=, <= karena merupakan batasan
- Untuk pencacahan maju, maka nilai pembanding dari kondisi harus lebih besar daripada nilaiawal pencacah, serta menggunakan operator pembanding < atau <=
- Untuk pencacahan mundur, maka nilai pembanding dari kondisi harus lebih kecil daripada nilaiawal pencacah, serta menggunakan operator pembanding > atau >=

3.4 Contoh Pengulangan

```
#include <stdio.h>

int main(void)
{
   int a, i;
   printf("Jumlah Pengulangan: "); scanf("%d", a);
   for(int i=0;i<a;i++)
   {
      printf("%d \n", i);
   }
}
```



3.5 Pencacahan

- Pencacahan akan menambah atau mengurangi nilai variable pencacah.
- Nilai pertambahan atau pengurangan dapat bernilai 1 atau lebih
- Nilai pertambahan atau pengurangan mempengaruhi jumla pengulangan
- Untuk pertambahan dengan nilai satu, dapat mempergunakan operator increment, misal i++
- Untuk pengurangan dengan nilai satu, dapat mempergunakan operator decrement, missal i--

3.6 Melacak Predicted Loop

- Untuk melacak predicted Loop seperti for dapat kita lakukan dengan cara membuat table dengan field-field berisi informasi nomor iterasi, nilai variable yang dilacak, nilai kondisi
- Contoh dari slide 12, misalkan a diinput 5, maka dapat kita buat tabelnya sebagai berikut:

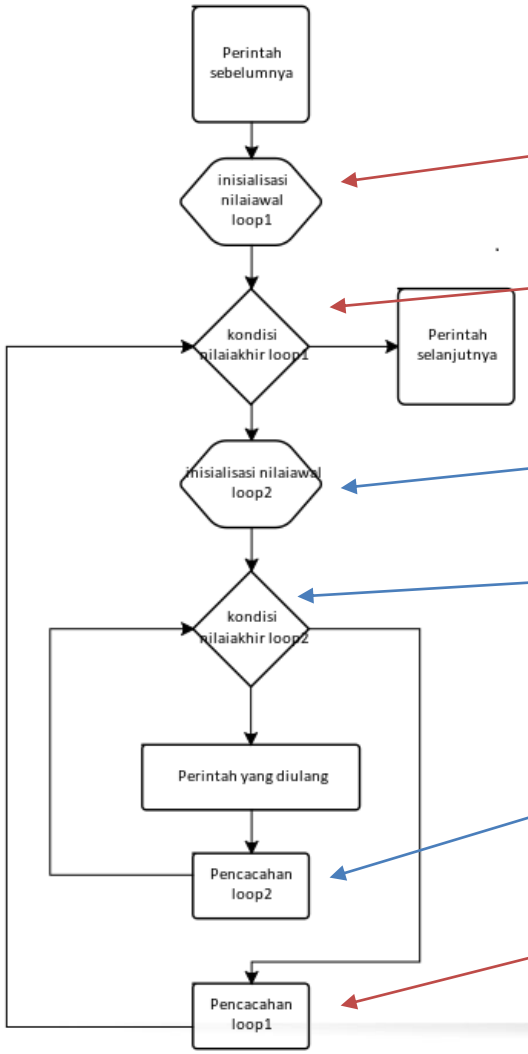
Iterasi ke	i	i<5	Loop?	Cetak i
0	0	true	Ya	0
1	1	true	Ya	1
2	2	true	ya	2
3	3	true	ya	3
4	4	true	ya	4
5	5	false	tidak	

4. Nested Loop

4.1 Nested Loop

- Nested Loop adalah pengulangan di dalam pengulangan

4.2 Pola Nested Loop



- Inisialisasi loop1
- Kondisi loop1
- Inisialisasi loop2
- Kondisi loop2
- Pencacahan loop2
- Pencacahan loop1

Nested Loop

Loop utama

4.3 Contoh nested Loop

```
#include <stdio.h>

int main(void)
{
  int a, i, j;
  printf("Jumlah Pengulangan: ");
  scanf("%d", a);
  for(int i=0;i<a;i++)
  {
    printf("\n Loop ke: %d",i);
    for(int j=0;j<a;j++)
    {
      printf("%d", j);
    }
  }
}
```

```
C:\Data\BCC55\tempat mengkompile>contohnestedfor
Jumlah Pengulangan: 5
Loop ke 0 [ 0  1  2  3  4 ]
Loop ke 1 [ 0  1  2  3  4 ]
Loop ke 2 [ 0  1  2  3  4 ]
Loop ke 3 [ 0  1  2  3  4 ]
Loop ke 4 [ 0  1  2  3  4 ]
```

4.4 Contoh Alternatif

- Pada beberapa compiler contoh pada slide sebelumnya tidak dapat membaca nilai integer dengan baik, dapat menggunakan contoh berikut ini:

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>

int main(void)
{
  int a, i, j;
  printf("Jumlah Pengulangan: ");
  cin>>a;
  for(i=0;i<a;i++)
  {
    cout<<"Loop ke "<<i<<" [";
    for(j=0;j<5;j++)
    {
      printf(" %d ", j);
    }
    cout<<"]"<<endl;
  }
}
```

Ringkasan

- Salah satu struktur dasar algoritma adalah pengulangan / repetition / loop
- Pengulangan adalah mekanisme algoritma yang akan mengatur pengulangan bagian dari algoritma
- Memerlukan kondisi tertentu untuk melakukan pengulangan, sehingga tidak terjadi infinite loop
- Predicted loop : pengulangan yang sudah diprediksi jumlah pengulangan
- Unpredicted loop : pengulangan yang jumlah pengulangannya tergantung kondisi

Terimakasih

TUHAN Memberkati Anda

Teady Matius Surya Mulyana (tmulyana@bundamulia.ac.id)