Fakultas Informatika
School of Computing
Telkom University

CCH1A4 / Dasar Algoritma & Pemrogramanan

Yuliant Sibaroni M.T, Abdurahman Baizal M.Kom

KK Modeling and Computational Experiment

Pengulangan

Pendahuluan

- Salah satu proses yang hampir selalu ada dalam pemrograman adalah pengulangan/*looping*.
- Pengulangan adalah suatu proses dimana komputer akan mengeksekusi satu atau lebih aksi(statemen) berulang kali menurut aturan tertentu.
- Salah satu kelebihan komputer dibandingkan manusia adalah kemampuan komputer untuk melakukan pengulangan aksi/proses dengan performa yang sama
- Beberapa perhitungan yang memerlukan pengulangan antara lain:
  - Menghitung integral tentu
  - Penjumlahan vektor/matriks
  - Perkalian matriks
  - Perhitungan nilai-nilai statistik
  - dll

Pengulangan

Struktur Pengulangan

Struktur pengulangan secara umum terdiri dari atas dua bagian :

- **Kondisi/ Syarat pengulangan**, yaitu berupa *ekspresi Boolean* yang harus dipenuhi untuk melaksanakan kondisi pengulangan. Kondisi ini mengakibatkan suatu kondisi pengulangan akan berhenti pada saat kondisi Boolean tersebut terpenuhi.
- **Badan (*body*) pengulangan**, yaitu sebuah/beberapa aksi (bagian algoritma) yang harus diulang selama kondisi yang ditentukan untuk pengulangan tersebut masih dipenuhi.

Pengulangan

Bentuk-bentuk Pengulangan

Di dalam kuliah ini, bentuk- bentuk pengulangan yang dipelajari adalah :

- **For-to-do**
- **While Do**
- **Repeat Until**

Untuk banyak pengulangan yang bersifat **pasti/fixed**, digunakan **for-to-do**, sedangkan untuk pengulangan berdasarkan **kondisi berhenti/pengulangan**, menggunakan **while do** atau **repeat until**

For – to - do

Definisi

Digunakan untuk pengulangan yang mempunyai jumlah pengulangan yang telah dipastikan sebelumnya.

Diperlukan variabel pencacah dengan nilai awal dan nilai akhir tertentu.

Variabel pencacah ini secara otomatis akan bertambah 1 untuk setiap pengulangan.

Notasi

```
For var_pencacah ← nilaiAwal to nilaiAkhir do
     Aksi1
     Aksi2
     ...
```

For – to - do

Penjelasan

- Aksi dilakukan sebanyak N kali, dimana N = (nilaiAkhir- nilaiAwal)+1.

  Sebagai contoh :

  ```
  For i← 2 to 10 do
      output('halo')
  ```

  Ini berarti

  - nilaiAwal = 2
  - nilaiAkhir = 10
  - Maka N = 9, sehingga `output('halo')` dilakukan 9x

- Variabel_pencacah harus suatu *type* yang terdefinisi suksesor dan predesesornya

- Setelah pelaksanaan pengulangan selesai, harga yang tersimpan pada *var_pencacah* tidak terdefinisi : jika hendak dipakai, harus didefinisikan kembali.

For – to - do

Contoh 5.1

Berikut adalah program dengan menggunakan for to do

```
Program ForToDo
Kamus
 i,N:integer
Algoritma
 Input(N)
 For i←1 to N do
    Output('looping ke –', i)
```

Misal N=4, Tabel Tracingnya sbb :

i	output
1	Looping ke- 1
2	Looping ke- 2
3	Looping ke- 3
4	Looping ke- 4

For – to - do

Contoh 5.2

Berikut adalah program untuk menampilkan semua bilangan dari a sampai b

```
Program Cetak_a_ke_b
Kamus
 a,b,i:integer
Algoritma
 Input(a)
 Input(b)
 For i←a to b do
    Output(i)
```

Kalau yang ditampilkan bilangan genap saja, apa yang perlu ditambahkan?

For – to - do

Contoh 5.3

Diberikan program untuk menghitung dan menampilkan jumlah bilangan dari a sampai b

```
Program Jumlah_a_ke_b
Kamus
 sum,a,b,i:integer
Algoritma
 Input(a)
 Input(b)
 sum ← 0
 For i←a to b do
    sum ← sum+i
 Output(sum)
```

Jika nilai yang diinputkan untuk a = 3 dan b = 6,
maka hasil eksekusi program adalah 18.
{ 18= 3+4+5+6}

Repeat - Until

Definisi

- Bentuk pengulangan **Repeat-Until** digunakan untuk pengulangan yang mempunyai jumlah pengulangan yang tidak dipastikan sebelumnya.
- Aksi1,Aksi2,... akan dilakukan secara berulang-ulang sampai *kondisi_berhenti* terpenuhi (bernilai TRUE).
- Pengulangan AKSI pada bentuk ini akan dilakukan setidaknya/minimal satu kali, karena pengecekan *kondisi_berhenti* dilakukan di akhir

Notasi

```
Repeat
    Aksi1
    Aksi2
    ...
Until (kondisi_berhenti)
```

Repeat - Until

Contoh 5.4

Diberikan program dengan menggunakan repeat until

```
Program RepeatUntil
Kamus
 i,N:integer
Algoritma
 Input(N)
 i ← 1
 Repeat
    Output('looping ke-', i)
    i ← i + 1
 Until(i>N)
```

Repeat - Until

Contoh 5.4

Diberikan program dengan menggunakan repeat until

```
Program RepeatUntil
Kamus
 i,N:integer
Algoritma
 Input(N)
 i ← 1
 Repeat
    Output('looping ke-', i)
    i ← i + 1
 Until(i>N)
```

Harus ada inisialisasi nilai i

Aksi-aksi yang dilakukan
Nilai i harus ditambah secara manual

Kondisi berhenti

Repeat - Until

Contoh 5.4

Diberikan program dengan menggunakan repeat until

```
Program RepeatUntil
Kamus
 i,N:integer
Algoritma
 Input(N)
 i ← 1
 Repeat
    Output('Looping ke-', i)
    i ← i + 1
 Until(i>N)
```

Misal N=4, Tabel Tracingnya sbb :

i	output
1	Looping ke 1
2	Looping ke 2
3	Looping ke 3
4	Looping ke 4

Repeat - Until

Contoh 5.5

Diberikan program untuk menghitung dan menampilkan jumlah bilangan dari a sampai b menggunakan repeat until

```
Program Jumlah_a_ke_b
Kamus
 sum,a,b,i:integer
Algoritma
 Input(a)
 Input(b)
 sum ← 0
 i ← a
 Repeat
    sum ← sum+i
    i ←i+1
 Until(i>b)
 Output(sum)
```

Bandingkan dengan While - do

Contoh 5.6

Diberikan program untuk menghitung dan menampilkan jumlah bilangan dari a sampai b menggunakan repeat until

```
Program Jumlah_a_ke_b
Kamus
  sum,a,b,i:integer
Algoritma
  Input(a)
  Input(b)
  sum ← 0
  i ← a
  while(i>b)do
     sum ← sum+i
     i ←i+1
  Output(sum)
```

Jika a=2 dan b=4, maka nilai sum terakhir berapa?

While-Do

Definisi

- Bentuk pengulangan **While-Do** digunakan untuk pengulangan yang mempunyai jumlah pengulangan yang tidak dipastikan sebelumnya.
- Pengulangan akan terus dilakukan selama kondisi terpenuhi (bernilai TRUE), dan jika kondisi tidak terpenuhi (bernilai FALSE), maka AKSI tidak dilakukan atau pengulangan berhenti
- Jumlah pengulangan ini minimal nol kali, karena pengecekan kondisi dilakukan di awal

Notasi

```
While (kondisi_pengulangan) do
    Aksi1
    Aksi2
    ..........
```

While-Do

Contoh 5.6

Diberikan program dengan menggunakan while do

```
Program WhileDo
Kamus
 i,N:integer
Algoritma
 Input(N)
 i ← 1
 while (i<=N) do
   output('Looping ke ',i)
   i ← i + 1
 {i > N}
```

While-Do
Contoh 5.6

Diberikan program dengan menggunakan while do

```
Program WhileDo
Kamus
 i,N:integer
Algoritma
 Input(N)
 i ← 1
 while (i=<N) do
   output('Looping ke ',i)
   i ← i + 1
 {i > N}
```

Harus ada inisialisasi nilai i

Kondisi Pengulangan

Aksi-aksi yang dilakukan
Nilai i harus ditambah secara manual

Kondisi berhenti

While-Do
Contoh 5.6

Diberikan program dengan menggunakan while do

```
Program WhileDo
Kamus
 i,N:integer
Algoritma
 Input(N)
 i ← 1
 while (i=<N) do
   output('Looping ke ',i)
   i ← i + 1
 {i > N}
```

Misal N=4, Tabel Tracingnya sbb :

i	output
1	Looping ke 1
2	Looping ke 2
3	Looping ke 3
4	Looping ke 4

While-Do

Contoh 5.7

Berikut adalah program mencari nilai maksimum berdasarkan nilai-nilai yang diinputkan user

```
Program cariMax
Kamus
i, bil, max: integer
Algoritma
 max ← -9999
 For i ← 1 to 5 do
   Input (bil)
   If bil > max then
     Max ← bil
 output ('Max=',max)
```

Diubah menjadi while - Do:

```
Program cariMax
Kamus
i, bil, max: integer
Algoritma
 max ← -9999
 i ← 1
 while i <= 5 do
   Input (bil)
   If bil > max then
     Max ← bil
   i ← i + 1
 output ('Max=',max)
```

While-Do

Contoh 5.7

Berikut adalah program mencari nilai maksimum berdasarkan nilai-nilai yang diinputkan user

```
Program cariMax
Kamus
i, bil, max: integer
Algoritma
 max ← -9999
 i ← 1
 while i <= 5 do
   Input (bil)
   If bil > max then
     Max ← bil
   i ← i + 1
 output ('Max=',max)
```

Contoh eksekusi :

```
12
20
5
40
4
Max = 40
```

Tabel Tracingnya sbb :

i	bil	max
		-9999
1	12	12
2	20	20
3	5	20
4	40	40
5	4	40

While-Do

Contoh 5.7

Diberikan program mencari nilai maksimum berdasarkan nilai-nilai yang diinputkan user menggunakan While Do

```
Program cariMax2
Kamus
 i, bil, max: integer
Algoritma
 max ← -9999
 i ← 1
 While i<=5 do
  Input(bil)
  If bil > max then
    Max ← bil
  i ← i+1
 output ('Max=',max)
```

While-Do

Contoh 5.8

Diberikan program mencari nilai rata-rata berdasarkan nilai-nilai yang diinputkan user

```
Program cariRata2
Kamus
 i,N,jumlah,nilai:integer
 rata2 : real
Algoritma
 Input(N)
 jumlah ← 0
 for i←1 to N do
  Input(nilai)
  jumlah ← jumlah + nilai
 rata2 ← jumlah/N
 Output(rata2)
```

Misal N=4, dan nilai yang diinputkan sebagai berikut, tabel tracing :

i	nilai	jumlah	rata2
		0	
1	4	4	
2	3	7	
3	6	13	
4	2	15	
			3.5

SOAL LATIHAN

Soal 1

Jika input dari user a = 4 dan b = 5. Hasil keluaran dari program di atas adalah (jika menurut anda program tersebut salah, tunjukkan kesalahannya):

```
Program Inilah
Kamus:
  a,b,c : integer
Algoritma :
  Input(a,b)
  c=a*b
  while c>10 do
     output ('halo')
     c←c-1
  {c=10}
```

SOAL LATIHAN

Soal 2

Jika n = 10, maka hasil eksekusi dari program dibawah adalah :

```
Program ajib
Kamus:
   i,n : integer
Algoritma :
  Input (n)
  i ← n
  While i>=2 do
    Output('halo')
    i ← i-2
    i ← i+1
  {i<n}
```

SOAL LATIHAN

Soal 3

Hasil eksekusi dari program tersebut adalah:

```
Program entah
Kamus:
 x,y : boolean
  i  : integer
Algoritma :
 x←(2+3)mod 2 =0
 y←false
 i←1
 Output(i)
 While x and not(y) do
   i←i+1
   If i = 5 then
     y←true
   Output(i)
```

SOAL LATIHAN

Soal 3

Hasil eksekusi dari program tersebut adalah:

```
Program entah
Kamus:
 x,y : boolean
  i  : integer
Algoritma :
 x←(2+3)mod 2 <> 0
 y←false
 i←1
 Output(i)
 While x and not(y) do
   i←i+1
   If i = 5 then
     y←true
   Output(i)
```

Soal Latihan

Soal 4

Jika nilai yang diinputkan untuk x adalah 6, maka hasil eksekusi program dibawah adalah:

```
Program mumet
Kamus
 x: integer; apahayo: boolean
Algoritma
 Input(x)
 apahayo ← true
 While (x<10) and apahayo do
   Output('hidup PT 1')
```

SOAL LATIHAN

Soal 5

Hasil eksekusi program dibawah adalah:

```
Program bingung
Kamus
 a,b : boolean
   i : integer
Algoritma
 a ← false
 b ← true
 i ← 0
 while (a OR b) do
   i ← i + 2
   output ('Halo')
   if i > 7 then
      b ← false
```

Referensi

- Inggriani Liem, Diktat Kuliah IF223 Algoritma Dan Pemrograman, Jurusan Teknik Informatika Bandung, 1999

Soal

- Buat algoritma menghitung rata-rata bilangan yang dimasukkan sebanyak 7 kali.
- Buat algoritma yang menampilkan menu, berikut:

  [1] masukkan

  [2] keluar
- Buat algoritma menampilkan kata ‘Halo’ jika masukkan ‘Y’, jika masukkan selain ‘Y’, maka keluar program

Fakultas Informatika
School of Computing
Telkom University

THANK YOU