

Struktur Kontrol: Perulangan

(dengan While-Loop & Repeat-Until)

Fakultas Informatika, Universitas Telkom

Outline



Bentuk While-Loop



Bentuk Repeat-Until



Contoh Soal



Latihan Soal

Berdasarkan Kondisi

❖ While-loop

- a. Aksi akan dieksekusi* selama kondisi bernilai TRUE.

```
while <kondisi perulangan>  
    <aksi>  
end
```

- b. Kondisi di sini adalah syarat terjadinya perulangan

❖ Repeat-until

- a. Aksi akan dieksekusi* selama kondisi bernilai FALSE.

```
repeat  
    <aksi>  
until <kondisi berhenti>
```

- b. Kondisi di sini adalah syarat perulangan dihentikan

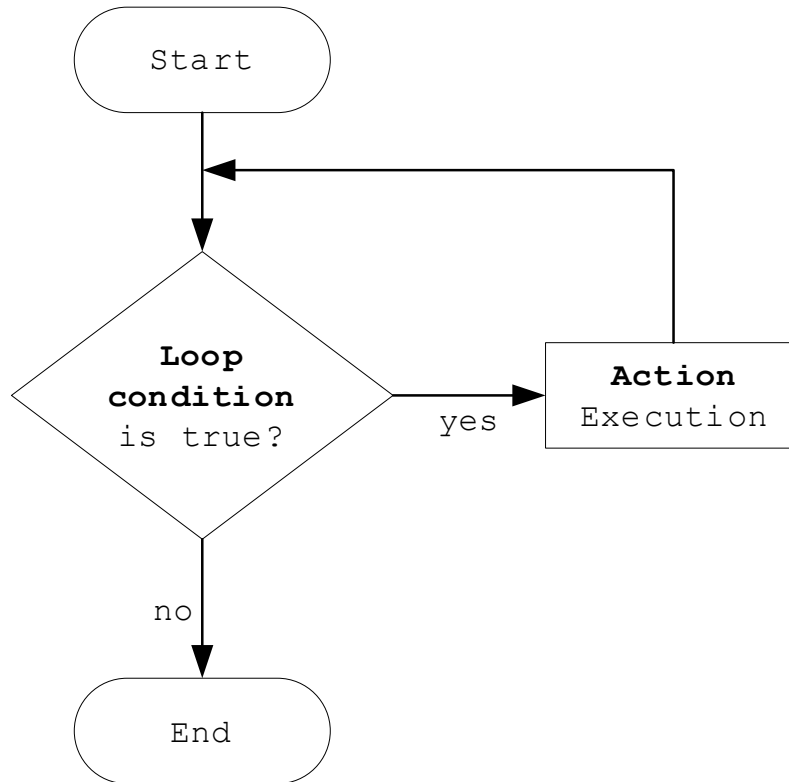
kondisi pada while dan repeat selalu komplemen

kondisi perulangan == not kondisi berhenti

* berulang-ulang

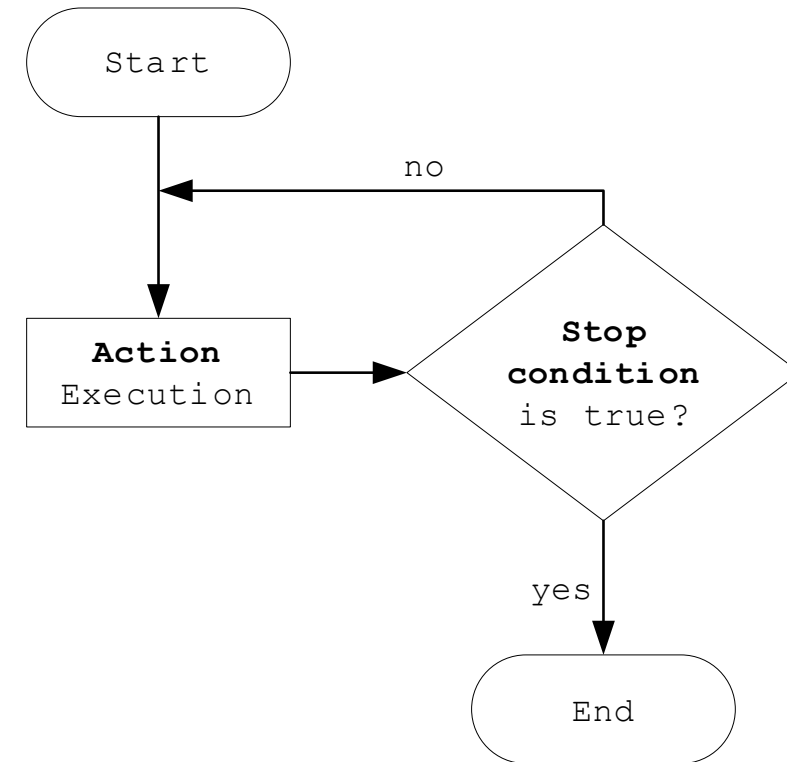
While - Loop

while <kondisi perulangan>
 <aksi>
end



Repeat - Until

repeat
 <aksi>
until <kondisi berhenti>



Contoh berdasarkan Kondisi

while-loop (repetition condition)

```
hungry ← true  
while hungry do  
    print("eat")  
    print("still hungry? ")  
    input(hungry)  
endwhile
```

repeat-until (stop condition)

```
repeat  
    print("eat")  
    print("still hungry? ")  
    input(hungry)  
until (not hungry)
```

kondisi perulangan == not kondisi berhenti

kondisi perulangan == not (not hungry)

kondisi perulangan == hungry



Contoh #1. Genap

Buatlah program yang hanya menerima masukan berupa bilangan genap dan menghitung total penjumlahannya. Bilangan genap adalah bilangan yang habis dibagi dua.

Masukan terdiri dari beberapa bilangan bulat positif yang dipisahkan oleh spasi. Masukan akan berakhir apabila bilangan yang diberikan adalah bilangan ganjil.

Keluaran terdiri sebuah bilangan bulat yang menyatakan total penjumlahan bilangan genap pada masukan.

Contoh Masukan dan Keluaran

No	Masukan	Keluaran
1	12 8 30 5	50
2	1	0

Analisis Masalah

- Input: integer x_1, x_2, x_3, \dots dst. (perulangan selama $x_i == \text{genap}$)
- Proses: $\text{total} = x_1 + x_2 + x_3, \dots$ dst (syarat $x_i == \text{genap}$)
- Output: total

Jenis Perulangan



- Repeat-until

kondisi berhenti: $x_i == \text{ganjil}$



- While-loop

kondisi perulangan: $x_i == \text{genap}$



- For-loop

jumlah iterasi tidak dapat diketahui

Solusi dengan Pseudocode

```
program Genap_A
{program dengan menggunakan while}
kamus
  xi, total : integer
algoritma
  total  $\leftarrow$  0
  input(xi)
  while x mod 2 == 0 do
    total  $\leftarrow$  total + x
    input(xi)
  endwhile
  output(total)
endprogram
```

```
program Genap_B
{program dengan menggunakan repeat}
kamus
  xi, total : integer
algoritma
  total  $\leftarrow$  0
  input(xi)
  repeat
    total  $\leftarrow$  total + x
    input(xi)
  until x mod 2 != 0
  output(total)
endprogram
```

Pertanyaan:

1. Algoritma manakah yang lebih tepat?
2. Apakah program Genap_A sudah benar untuk semua kemungkinan bilangan pada input?
3. Apakah program Genap_B sudah benar untuk semua kemungkinan bilangan pada input?
4. Kesimpulan apa yang bisa diambil?

Solusi dengan GoLang

```
package main
/* program dengan menggunakan while-loop */
import "fmt"
func main(){
    var xi, total int
    total = 0
    fmt.Scan(&xi)
    for x mod 2 == 0 {
        total = total + x
        fmt.Scan(xi)
    }
    fmt.Println(total)
}
```

Kata kunci untuk semua tipe perulangan di dalam Bahasa GoLang adalah for saja

while-loop:

```
for <looping condition> {
    <action>
}
```

Contoh #2. Barisan Bilangan

Buatlah algoritma untuk menampilkan barisan bilangan bulat positif dari 1 sampai N.

Masukan terdiri dari sebuah bilangan bulat positif N.

Keluaran terdiri dari barisan bilangan dari 1 hingga N.

Contoh Masukan dan Keluaran

No	Masukan	Keluaran
1	12	1 2 3 4 5 6 7 8 9 10 11 12
2	1	1

Analisis Masalah

- Input: integer N , $N > 0$
- Proses: perulangan dari 1 sampai N
- Output: 1 2 3 4 ... N

Jenis Perulangan

- Repeat-until
kondisi berhenti: iterasi $== N$
- While-loop
kondisi perulangan: iterasi $!= N$
- For-loop*
terdapat N iterasi

Solusi dengan Pseudocode

```
program barisanA
{program dengan menggunakan repeat}
kamus
  N, iterasi : integer
algoritma
  input(N)
  iterasi  $\leftarrow$  0
  repeat
    iterasi  $\leftarrow$  iterasi + 1
    output(iterasi)
  until iterasi == N
endprogram
```

```
program barisanB
{program dengan menggunakan while}
kamus
  N, iterasi : integer
algoritma
  input(N)
  iterasi  $\leftarrow$  0
  while iterasi != N do
    iterasi  $\leftarrow$  iterasi + 1
    output(iterasi)
  endwhile
endprogram
```

Solusi dengan Go

```
package main
/* salah satu solusi menggunakan repeat */
import "fmt"
func main(){
    var N, iterasi int
    var berhenti bool
    fmt.Scan(&N)
    iterasi = 0; berhenti = false
    for !berhenti {
        iterasi = iterasi + 1
        fmt.Print(iterasi, " ")
        berhenti = iterasi == N
    }
}
```

```
package main
/* program dengan menggunakan while */
import "fmt"
func main(){
    var N, iterasi int
    fmt.Scan(&N)
    iterasi = 0
    for iterasi != N {
        iterasi = iterasi + 1
        fmt.Print(iterasi, " ")
    }
}
```

Kata kunci perulangan untuk semua jenis perulangan pada GoLang adalah for.

repeat-until:

stop = false

for not stop {

 <action>

 stop = <stop condition>

}

Contoh 3. Integer Division

Buatlah sebuah algoritma ModDiv yang berisi cara untuk memperoleh hasil modulo dan integer division.

Masukan berupa dua buah bilangan bulat non-negative x dan y .

Keluaran berupa dua buah bilangan yang menyatakan hasil operasi modulo dan integer division dari x dan y . Perhatikan contoh yang diberikan:

Dilarang menggunakan operator mod dan div, contoh $x \bmod y$ atau $x \div y$ dilarang.

Gunakan operasi pengurangan dan perulangan

Contoh Masukan dan Keluaran

No	Masukan	Keluaran
1	12 3	$12 \bmod 3 = 0$ $12 \div 3 = 4$
2	1 7	$1 \bmod 7 = 1$ $1 \div 7 = 0$
3	0 4	$0 \bmod 4 = 0$ $0 \div 4 = 0$
4	10 6	$10 \bmod 6 = 4$ $10 \div 6 = 1$

Analisis Masalah

Masukan: integer x dan y , x dan $y \geq 0$

Proses: nilai x dikurangi y secara terus menerus (nilai x selalu diupdate dengan hasil pengurangan) hingga x tidak bisa dikurangi. Banyaknya pengurangan adalah hasil div, sedangkan nilai x yang tidak bisa dikurangi adalah hasil mod. Contoh $x = 20$ dan $y = 6$

$$\rightarrow 20 - 6 = 14$$

$$\rightarrow 14 - 6 = 8$$

$$\rightarrow 8 - 6 = 2$$

Operasi pengurangan dilakukan 3x, sedangkan nilai x terakhir adalah 2
Hasil $20 \text{ div } 6$ adalah 3, sedangkan hasil $20 \text{ mod } 6$ adalah 2.

Keluaran: hasil $x \text{ mod } y$ dan hasil $x \text{ div } y$

Solusi dengan Pseudocode dan GoLang

program IntDiv

kamus

x,xi,y,i : **integer**

algoritma

input(x,y)

i \leftarrow 0

xi \leftarrow x

while xi >= y **do**

xi \leftarrow xi - y

i \leftarrow i + 1

endwhile

output(x,"mod",y,"=",xi)

output(x,"div",y,"=",i)

endprogram

```
package main
```

```
import "fmt"
```

```
func main(){
```

```
    var x,xi,y,i int
```

```
    fmt.Scan(&x,&y)
```

```
    i = 0
```

```
    xi = x
```

```
    for xi >= y {
```

```
        xi = xi - y
```

```
        i = i + 1
```

```
    }
```

```
    fmt.Println(x,"mod",y,"=",xi)
```

```
    fmt.Println(x,"div",y,"=",i)
```

```
}
```


**KEEP
CALM
AND
STUDY
HARD**

Sesi Latihan 😊



MOTIVATIONAL
PICTURE QUOTES.COM

Image <https://id.pinterest.com/pin/498703358729526087/>

Soal 1. Login

Sebuah program digunakan untuk menghitung berapa banyak seorang user melakukan gagal login karena salah input username dan password.

Masukan terdiri dari username dan password. Apabila username dan password salah, maka lakukan proses input kembali sampai username dan password benar. Asumsi username dan password yang benar adalah “admin” dan “admin”

Keluaran terdiri dari berapa banyak user gagal melakukan login, dan sebuah pesan “Login berhasil”.

Contoh Masukan dan Keluaran

No	Masukan	Keluaran
1	User 123 user 123 user user admin 5678 admin admin! admin admin	5 Login berhasil
2	admin admin	0 Login berhasil
3	admin Admin admin123 admin123 admin admin	2 Login berhasil

Soal 2. Dompot

Buatlah algoritma untuk menghitung saldo uang yang ada dalam dompet seorang mahasiswa di akhir bulan.

Masukan terdiri dari serangkaian bilangan bulat yang menyatakan transaksi keluar masuk dompet. Bilangan tanpa tanda berarti jumlah uang yang masuk ke dalam dompet, bilangan negatif menyatakan uang keluar dari dompet. Masukan berakhir apabila jumlah uang yang diberikan adalah 0 (**nol**)

Keluaran adalah jumlah saldo uang dalam dompet.

Contoh Masukan dan Keluaran

N o	Masukan	Keluaran
1	500000 -150000 -45000 -10000 -100000 0	195000

Soal 3. Digit

Sebuah algoritma digunakan untuk mencacah digit suatu bilangan bulat positif.

Masukan terdiri dari sebuah bilangan bulat positif X.

Keluaran terdiri dari dua baris. Baris pertama adalah nilai setiap digit dari bilangan X dan dipisahkan oleh spasi (digit dimulai dari paling kanan, perhatikan contoh). Baris kedua adalah total penjumlahan setiap digit dari X.

Contoh Masukan dan Keluaran

No	Masukan	Keluaran	Penjelasan
1	1234	4 3 2 1 10	$4 + 3 + 2 + 1 = 10$
2	135798642	2 4 6 8 9 7 5 3 1 45	$2 + 4 + 6 + 8 + 9 + 7 + 5 + 3 + 1 = 45$
3	20	0 2 2	$0 + 2 = 2$

Soal 4. Cangkir Kopi

Buatlah sebuah program untuk menghitung banyak cangkir kopi yang bisa dibuat apabila terdefinisi sejumlah **n** gula dan **m** kopi. Satu cangkir kopi memerlukan sejumlah **x** gula dan **y** kopi.

Masukan berupa empat bilangan bulat yang dipisahkan spasi, **n**, **m**, **x** dan **y**. Di mana nilai $x \leq n$ dan $y \leq m$.

Keluaran berupa bilangan bulat yang menyatakan banyaknya cangkir kopi yang berhasil dibuat.

Catatan: gunakan perulangan

Contoh Masukan dan Keluaran

No	Masukan	Keluaran	Penjelasan
1	5 9 1 3	3	Tersedia 5 gula dan 9 kopi, 1 cangkir memerlukan 1 gula dan 3 kopi, sehingga didapat 3 cangkir.
2	10 12 10 12	1	Tersedia 10 gula dan 12 kopi, 1 cangkir memerlukan 10 gula dan 12 kopi, sehingga didapat 1 cangkir.
3	20 25 4 2	5	Tersedia 20 gula dan 25 kopi, 1 cangkir memerlukan 4 gula dan 2 kopi, sehingga didapat 5 cangkir.

Soal 5. Konsekatif

Apabila didefinisikan sebuah bilangan konsekatif adalah bilangan yang selisih setiap digit yang bersebelahannya adalah satu. Maka buatlah algoritma untuk menentukan suatu bilangan konsekatif atau tidak.

Masukan sebuah bilangan bulat positif X.

Keluaran adalah sebuah nilai Boolean yang menyatakan X adalah konsekatif atau tidak.

Contoh Masukan dan Keluaran

No	Masukan	Keluaran	Penjelasan
1	101010101	true	Selisih setiap digit adalah 1
2	1234321	true	Selisih setiap digit adalah 1
3	888888	false	Selisih 8 dengan 8 adalah 0
4	1234567890	false	Selisih 9 dengan 0 adalah 9

Soal 6. Tanki Air

Sebuah tanki kosong dengan kapasitas T liter akan diisi air menggunakan ember E dengan berbagai ukuran volume ($0 < E \leq N$). Buatlah algoritma untuk mengisi tanki dari asalnya kosong hingga penuh.

Masukan terdiri dari beberapa baris. Baris pertama adalah sebuah bilangan bulat positif T yang menyatakan kapasitas tanki. T baris berikutnya, masing-masing adalah bilangan bulat V yang menyatakan volume air dalam ember E dimasukkan ke dalam tanki.

Keluaran terdiri dari beberapa baris, yang masing-masing baris berisi Boolean yang menyatakan tanki penuh atau tidak setiap kali pengisian tanki.

Contoh Masukan dan Keluaran

No	Masukan	Keluaran	Penjelasan
1	30 5 10 5 5 10	false false false false true	$T = 30$ $5 + 10 + 5 + 5 + 10 = 35$
2	45 20 10 15	false false true	$T = 45$ $20 + 10 + 15 = 45$

Terima Kasih 😊
