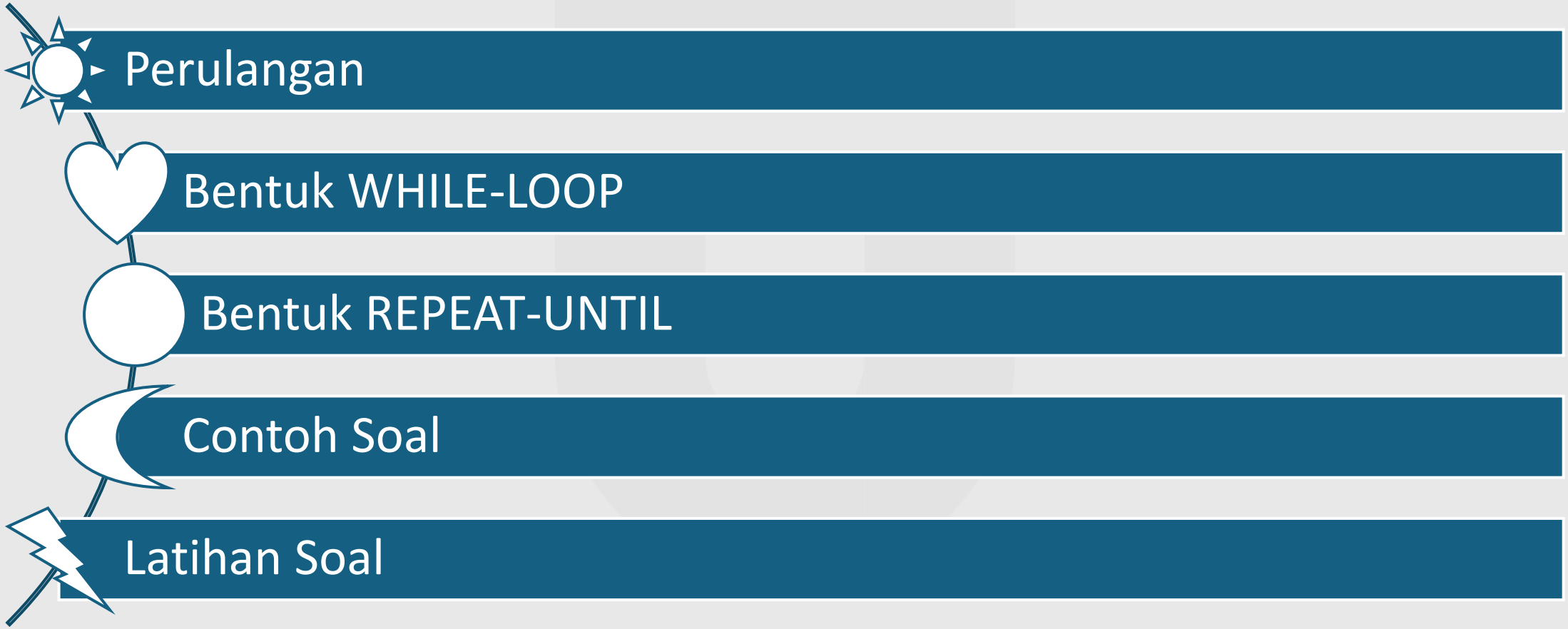
Bachelor of Informatics
School of Computing
Telkom University

Perulangan berdasarkan Kondisi

ALGORITMA & PEMROGRAMAN 1 (CAK1BAB3)

Pertemuan 11 & 12 - Prodi S1 Informatika

Outline


Perulangan
Bentuk WHILE-LOOP
Bentuk REPEAT-UNTIL
Contoh Soal
Latihan Soal

Perulangan (Overview)

Tugas utama komputer adalah melakukan suatu instruksi/proses secara berulang dan terus-menerus tanpa adanya perbedaan.

Hal ini berbeda dengan manusia yang bila melakukan hal sama secara berulang bisa melakukan kesalahan.

Setiap baris instruksi dieksekusi satu persatu

Satu atau lebih instruksi bisa dieksekusi berulang kali

Perulangan (Overview)

Tanpa Perulangan

```
i = 1
output(“informatika”)
i = i + 1
output(i)
output(“hello world”)
```

```
informatika
2
hello world
```

Dengan Perulangan

```
i = 1
<instruksi perulangan>
    output(“informatika”)
    i = i + 1
    output(i)
<batas instruksi perulangan>
output(“hello world”)
```

```
informatika
2
informatika
3
informatika
4
...
...
hello world
```

Perulangan (Overview)

Syarat Perulangan:

- Perulangan **harus berhenti**
- Apabila perulangan **tidak pernah berhenti**, maka algoritmanya salah (proses lama ≠ tidak pernah berhenti)
- Pemrogram **harus mengetahui** perulangan akan berhenti/tidak sebelum program dijalankan

Jenis Instruksi Perulangan:

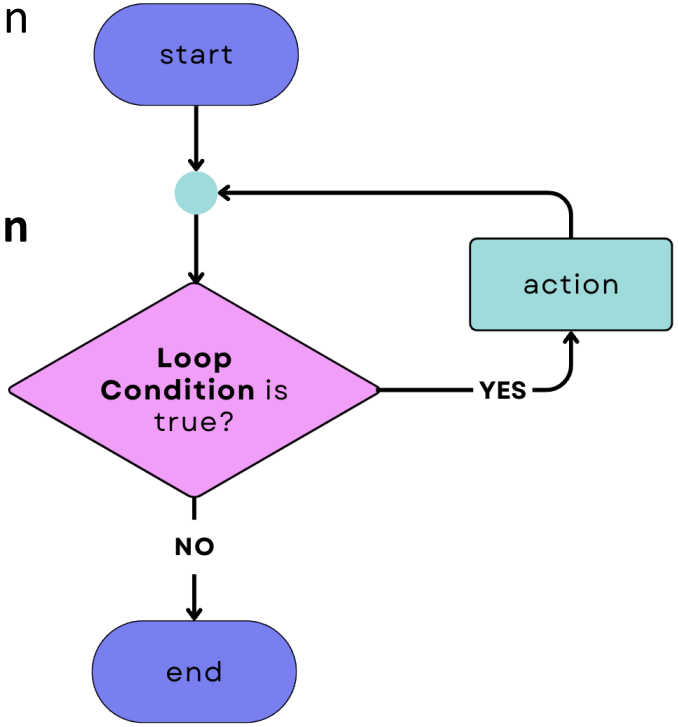
- Berdasarkan jumlah iterasi
- Berdasarkan kondisi (kapan harus diulangi/berhenti)

Bentuk Perulangan

- While-Loop
- Repeat-Until

Bentuk While-Loop

- Secara struktur penulisannya mirip seperti If-then, yang mana **<action>** akan **dieksekusi secara berulang SELAMA <condition>** bernilai **true**.
- **Jumlah iterasi tidak dapat ditentukan**, karena bergantung dengan **perubahan nilai** pada **<condition>**.



	Pseudocode	GoLang
1	`while <condition> do`	`for <condition> {`
2	`    <action>`	`    <action>`
3	`endwhile`	`}`

- Pada bahasa Go, keyword yang digunakan adalah **"for"** dengan struktur penulisan while-loop.

Bentuk While-Loop

- Sebagai contoh:

	Pseudocode	GoLang
1	hungry = **true**	hungry = **true**
2	**while** hungry **do**	**for** hungry {
3	**output**("eat")	fmt.**Println**("eat")
4	**output**("still hungry?")	fmt.**Println**("still hungry?")
5	**input**(hungry)	fmt.**Scan**(&hungry)
6	**endwhile**	}

- Perulangan akan berhenti apabila variabel **hungry** pada baris ke-5 bernilai **false**

Contoh Soal #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

Jawaban Soal #1 Genap


```
program Genap
kamus
    xi, total : integer
algoritma
    total ← 0
    input(xi)
    while xi mod 2 == 0 do
        total ← total + xi
        input(xi)
    endwhile
    output(total)
endprogram
```

Input: integer $x_1, x_2, x_3, \ldots\ dst.$ (perulangan selama x_i == genap)

Proses: total = $x_1 + x_2 + x_3, \ldots\ dst$,(syarat x_i == genap)

Output: total

Jenis Perulangan

While-loop

kondisi perulangan: x_i == genap

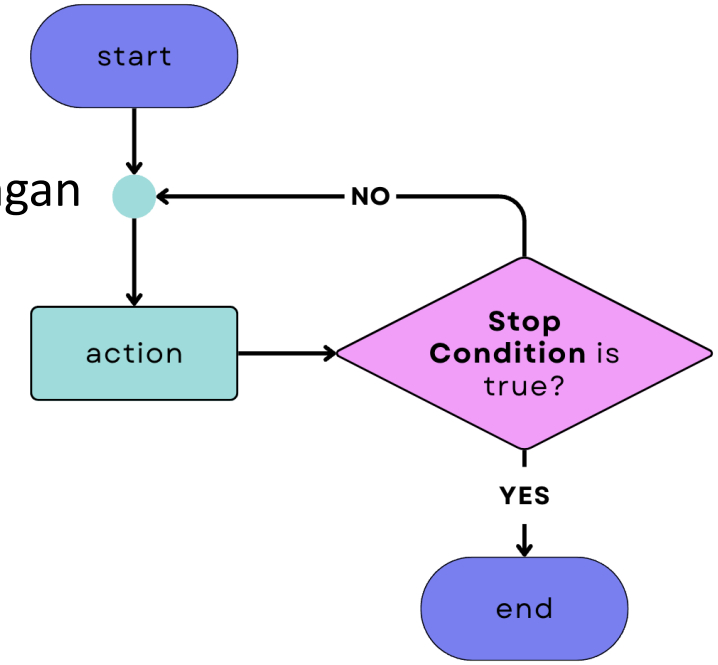
For-loop

jumlah iterasi tidak dapat diketahui

```
package main
import "fmt"
func main(){
    var xi, total int
    total = 0
    fmt.Scan(&xi)
    for xi % 2 == 0 {
        total = total + xi
        fmt.Scan(&xi)
    }
    fmt.Println(total)
}
```

Bentuk Repeat-Until

- Berbeda dengan while-loop, di mana **<action>** akan **dieksekusi secara berulang SAMPAI <condition>** bernilai **true**.
- **Jumlah iterasi tidak dapat ditentukan**, karena bergantung dengan **perubahan nilai** pada **<condition>**.



	Pseudocode	GoLang
1	**repeat**	-
2	<action>	
3	**until** <condition>	

- Pada bahasa Go, tidak ada penulisan secara spesifik untuk **repeat-until**, jadi alternatifnya bisa menggunakan bentuk **while-loop**

Contoh Soal #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

Jawaban Soal #2 Barisan Bilangan

```
program BarisanA
kamus
    N, iterasi : integer
algoritma
    input(N)
    iterasi ← 0
    repeat
        iterasi ← iterasi + 1
        output(iterasi)
    until iterasi == N
endprogram
```

```
program BarisanB
kamus
    N, iterasi : integer
algoritma
    input(N)
    iterasi ← 0
    while iterasi != N do
        iterasi ← iterasi + 1
        output(iterasi)
    endwhile
endprogram
```

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

Contoh Soal #3 Integer Division

Buatlah sebuah algoritma 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:

Gunakan operasi pengurangan dan perulangan untuk menggantikan operator **mod** dan **div**.

Contoh masukan dan keluaran

No	Masukan	Keluaran
1	12 3	12 mod 3 = 0 12 div 3 = 4
2	1 7	1 mod 7 = 1 1 div 7 = 0
3	0 4	0 mod 4 = 0 0 div 4 = 0

Jawaban Soal #3 Integer Division

Input: integer x dan y, x dan $y >= 0$

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 $y = 6$

→ $20 - 6 = 14$

→ $14 - 6 = 8$

→ $8 - 6 = 2$

Operasi pengurangan dilakukan 3x, sedangkan nilai x terakhir adalah 2

Hasil 20 div 6 adalah 3, sedangkan hasil 20 mod 6 adalah 2.

Output: hasil $x \bmod y$ dan hasil $x \operatorname{div} y$.

Jawaban Soal #3 Integer Division

```
program IntDiv
kamus
    x,xi,y,i : integer
algoritma
    input(x,y)
    i = 0
    xi = x
    while xi >= y do
        xi = xi - y
        i = 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)
}
```

Kesimpulan

- **while-loop** dan **repeat-until** bisa digunakan untuk perulangan berdasarkan kondisi ataupun iterasi.
- Kondisi pada **while-loop** adalah **kondisi perulangan**.
- Kondisi pada **repeat-until** adalah **kondisi berhenti**.
- **Kondisi berhenti adalah negasi dari kondisi perulangan**, sehingga bentuk **while-loop** dapat dengan mudah dikonversi ke bentuk **repeat-until**.
- Pada **repeat-until**, ***perulangan pasti dilakukan minimal 1x*** karena pengecekan kondisi dilakukan setelah aksi dilakukan. Sedangkan, pada **while-loop** ***perulangan mungkin tidak terjadi*** karena pengecekan kondisi dilakukan di awal.

Latihan Soal

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 Dompet

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

No	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 teridri 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 <= n dan y <= 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 Konsekutif

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

Masukan sebuah bilangan bulat positif X.

Keluaran adalah sebuah nilai Boolean yang menyatakan X adalah konsekutif 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 Tangki Air

Sebuah tanki kosong dengan kapasitas T liter akan diisi air menggunakan ember E dengan berbagai ukuran volume (0 < E ≤ 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 ☺