

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 13
REPEAT-UNTIL



Disusun oleh:

Johanson Leeroy

109082500017

S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {

    var kata string

    var jumlah, counter int

    var kondisi bool

    fmt.Scan(&kata, &jumlah)

    for kondisi = false; !kondisi; {

        fmt.Println(kata)

        counter++

        kondisi = counter == jumlah

    }

}
```

Screenshot program

The screenshot shows a code editor interface with a terminal window. The code editor displays a Go file named 'Guided1.go' with the following content:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var kata string
7     var jumlah, counter int
8     var kondisi bool
9
10    fmt.Scan(&kata, &jumlah)
11
12    for kondisi = false; !kondisi; {
13        fmt.Println(kata)
14        counter++
15        kondisi = counter == jumlah
16    }
17 }
18

```

The terminal window shows the execution of the program with the command `go run "d:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Guided1.go"`. The output is:

```

PS D:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Guided1.go"
pagi 3
pagi
pagi
pagi
PS D:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Guided1.go"
kursi 5
kursi
kursi
kursi
kursi
kursi
PS D:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> []

```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel kata dengan tipe data string
3. Mendeklarasikan variabel jumlah, counter dengan tipe data integer/int
4. Mendeklarasikan variabel kondisi dengan tipe data boolean
5. User menginputkan string dan bilangan bulat/integer ke dalam variabel kata dan jumlah secara berturut-turut
6. Membuat perulangan repeat-until dengan kondisi, kondisi = false dan negasi dari kondisi yang artinya kondisi = true
7. Maka program akan mengeluarkan output string pada variabel kata dan menambah 1 nilai pada variabel counter / counter ++
8. Memberikan kondisi pada variabel kondisi yaitu, kondisi = counter == jumlah
9. Ketika variabel kondisi bernilai true, maka kondisi perulangan tersebut akan bernilai true lalu dinegasikan, kondisi = false sehingga program berhenti.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var bil int
    var kondisi bool
    bil = 0
    for kondisi = false; !kondisi; {
        fmt.Scan(&bil)
        kondisi = bil > 0
    }
    fmt.Println(bil, "adalah bilangan bulat positif")
}
```

Screenshoot program

```
@@ Guided2.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var bil int
7      var kondisi bool
8      bil = 0
9      for kondisi = false; !kondisi; {
10          fmt.Scan(&bil)
11          kondisi = bil > 0
12      }
13      fmt.Println(bil, "adalah bilangan bulat positif")
14  }
15
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Guided2.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
PS D:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Guided2.go"
17
17 adalah bilangan bulat positif
PS D:\.new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13>
```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel bil dengan tipe data integer/int
3. Mendeklarasikan variabel kondisi dengan tipe data boolean
4. User menginputkan bilangan bulat/integer ke dalam variabel bil
5. Membuat perulangan repeat-until dengan kondisi, kondisi = false dan negasi dari kondisi yang artinya kondisi = true
6. Maka program akan meminta user menginputkan kembali bilangan bulat/integer ke dalam variabel bil
7. Memberikan kondisi pada variabel kondisi yaitu, kondisi = bil > 0. Jadi program akan terus meminta input dari user sampai bilangan bulat yang diinputkan merupakan bilangan bulat positif
8. Ketika variabel kondisi bernilai true, maka kondisi perulangan tersebut akan bernilai true lalu dinegasikan, kondisi = false sehingga program berhenti.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var x, y int
    var kondisi bool
    fmt.Scan(&x, &y)
    for kondisi = false; !kondisi; {
        x -= y
        fmt.Println(x)
        kondisi = x <= 0
    }
    fmt.Print(x == 0)
}
```

Screenshot program

```

1 Guided3.go > ...
2 package main
3 import "fmt"
4
5 func main() {
6     var x, y int
7     var kondisi bool
8     fmt.Scan(&x, &y)
9     for kondisi = false; !kondisi; {
10         x -= y
11         fmt.Println(x)
12         kondisi = x <= 0
13     }
14     fmt.Print(x == 0)
15 }
16

```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Guided3.go"
5 2
3
1
-1
false
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Guided3.go"
15 3
12
9
6
3
0
true
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Guided3.go"
25 5
20
15
10
5
0
true
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13>

```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel x, y dengan tipe data integer/int
3. Mendeklarasikan variabel kondisi dengan tipe data boolean
4. User menginputkan bilangan bulat/integer ke dalam variabel x dan y
5. Membuat perulangan repeat-until dengan kondisi, kondisi = false dan negasi dari kondisi yang artinya kondisi = true
6. Maka program akan melakukan operasi x -= y, lalu mengeluarkan output nilai x terbaru
7. Memberikan kondisi pada variabel kondisi yaitu, kondisi = x <= 0
8. Ketika variabel kondisi bernilai true, maka kondisi perulangan tersebut akan bernilai true lalu dinegasikan, kondisi = false sehingga program berhenti.
9. Terakhir program akan mengeluarkan output nilai kebenaran dari x == 0. Yang artinya, jika x = 0 maka output merupakan true dan jika x !=0 maka output merupakan false

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var bil, digit int
    var kondisi bool
    fmt.Scan(&bil)

    for kondisi = false; !kondisi; {
        bil /= 10
        digit++
        kondisi = bil == 0
    }
    fmt.Println(digit)
}
```

Screenshot program

```

go Tugas1.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bil, digit int
7     var kondisi bool
8     fmt.Scan(&bil)
9
10    for kondisi = false; !kondisi; {
11        bil /= 10
12        digit++
13        kondisi = bil == 0
14    }
15    fmt.Println(digit)
16 }

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
ugas1.go"
5
1
PS D:\new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\T
ugas1.go"
234
3
PS D:\new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\T
ugas1.go"
78787
5
PS D:\new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\T
ugas1.go"
1894256
7
PS D:\new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13>

```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel bil, digit dengan tipe data integer/int
3. Mendeklarasikan variabel kondisi dengan tipe data boolean
4. User menginputkan bilangan bulat/integer ke dalam variabel bil
5. Membuat perulangan repeat-until dengan kondisi, kondisi = false dan negasi dari kondisi yang artinya kondisi = true
6. Maka program akan melakukan operasi bil /= 10 dan digit++
7. Memberikan kondisi pada variabel kondisi yaitu, kondisi = bil == 0
8. Ketika variabel kondisi bernilai true, maka kondisi perulangan tersebut akan bernilai true lalu dinegasikan, kondisi = false sehingga program berhenti.
9. Output berupa nilai terbaru dari variabel digit

2. Tugas 2

Source code

```

package main

import (
    "fmt"
    "math"
)

```

```
)  
  
func main() {  
  
    var i int  
  
    var bil, bulat float64  
  
    var kondisi bool  
  
    fmt.Scan(&bil)  
  
    bulat = math.Ceil(bil)  
  
    bulat -= bil  
  
    bulat *= 10  
  
    bulat = math.Ceil(bulat)  
  
    var bulatINT = int(bulat)  
  
  
    for kondisi = false; !kondisi; {  
  
        bil += 0.1  
  
        fmt.Printf("%.1f \n", bil)  
  
        i++  
  
        kondisi = i == bulatINT  
  
    }  
  
}
```

Screenshot program

The screenshot shows a Go development environment with the following components:

- Code Editor:** Displays the source code for `Tugas2.go`. The code imports `fmt` and `math`, defines a `main` package, and contains a `main` function that reads a float64 value from the user, rounds it down to the nearest integer, and then prints the integer value followed by its decimal representation.
- Terminal:** Shows the command `go run "d:\new journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Tugas2.go"` being run, followed by the output of the program which prints a series of floating-point numbers starting from 0.2 up to 3.0.

Deskripsi program

1. Program diawali dengan package main, import “fmt” dan “math”, dan func main () {
2. Mendeklarasikan variabel i dengan tipe data integer/int
3. Mendeklarasikan variabel bil, bulat dengan tipe data float64
4. Mendeklarasikan variabel kondisi dengan tipe data boolean
5. User menginputkan bilangan bulat/integer ke dalam variabel bil
6. Memberi nilai pada variabel bulat yang merupakan pembulatan keatas dari bilangan ril yang ada di variabel bil
7. Mengurangi nilai pada variabel bulat dengan bilangan ril pada variabel bil untuk mendapatkan selisih
8. Melakukan operasi pada variabel bulat yaitu, bulat*= 10
9. Kembali membulatkan nilai pada variabel bulat keatas
10. Mengubah tipe data pada variabel bulat dari float64 ke integer di variabel baru bulatINT
11. Membuat perulangan repeat-until dengan kondisi, kondisi = false dan negasi dari kondisi yang artinya kondisi = true
12. Maka program akan melakukan operasi bil+= 0.1, mengeluarkan output berupa update dari nilai pada variabel bil dan melakukan operasi i++
13. Memberikan kondisi pada variabel kondisi yaitu, kondisi = i == bulatINT
14. Ketika variabel kondisi bernilai true, maka kondisi perulangan tersebut akan bernilai true lalu dinegasikan, kondisi = false sehingga program berhenti.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var target, donasi, donatur, total int
    var kondisi bool
    donasi = 0
    fmt.Scan(&target)

    for kondisi = false; !kondisi; {
        fmt.Scan(&donasi)
        total += donasi
        donatur++
        fmt.Print("Donatur ", donatur, " menyumbang ",
donasi, ". Total terkumpul :", total, "\n")
        kondisi = total >= target
    }
    fmt.Println("Target tercapai! Total donasi:", total,
"dari", donatur, "donatur")
}
```

Screenshot program

The screenshot shows a code editor interface with a terminal window below it. The terminal window displays the execution of a Go program named Tugas3.go. The program prompts the user for a target amount (100) and a number of donors (3). It then enters a loop where each donor's contribution is read and added to a total. After three contributions of 100, the total reaches 300, which is printed as the target amount achieved.

```


1 package main
2
3 import "fmt"
4
5 func main() {
6     var target, donasi, donatur, total int
7     var kondisi bool
8     donasi = 0
9     fmt.Scan(&target)
10
11    for kondisi = false; !kondisi; {
12        fmt.Scan(&donasi)
13        total += donasi
14        donatur++
15        fmt.Println("Donatur ", donatur, " menyumbang ", donasi, ". Total terkumpul :", total, "\n")
16        kondisi = total >= target
17    }
18    fmt.Println("Target tercapai! Total donasi:", total, "dari", donatur, "donatur")
19 }


```

```


PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Tugas3.go"
300 100 50 200
Donatur 1 menyumbang 100. Total terkumpul :100
Donatur 2 menyumbang 50. Total terkumpul :150
Donatur 3 menyumbang 200. Total terkumpul :350
Target tercapai! Total donasi: 350 dari 3 donatur
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Tugas3.go"
500 150 100 50 300
Donatur 1 menyumbang 150. Total terkumpul :150
Donatur 2 menyumbang 100. Total terkumpul :250
Donatur 3 menyumbang 50. Total terkumpul :300
Donatur 4 menyumbang 300. Total terkumpul :600
Target tercapai! Total donasi: 600 dari 4 donatur
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13> go run "d:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13\Tugas3.go"
200 300
Donatur 1 menyumbang 300. Total terkumpul :300
Target tercapai! Total donasi: 300 dari 1 donatur
PS D:\.new_journey\SMTR 1\ALGORITMA DAN PEMROGRAMAN\LAPRAK Modul 13>


```

Deskripsi program

1. Program diawali dengan package main, import “fmt”, dan func main () {
2. Mendeklarasikan variabel target, donasi, donatur, total dengan tipe data integer/int
3. Mendeklarasikan variabel kondisi dengan tipe data boolean
4. User menginputkan bilangan bulat/integer ke dalam variabel target
5. Membuat perulangan repeat-until dengan kondisi, kondisi = false dan negasi dari kondisi yang artinya kondisi = true
6. Maka program akan meminta user menginputkan bilangan bulat/integer ke dalam variabel donasi, melakukan operasi total+= donasi dan donatur++
7. Mengeluarkan Output berupa detail donasi yaitu, donatur keberapa, jumlah donasi dan total donasi terbaru
8. Memberikan kondisi pada variabel kondisi yaitu, kondisi = total >= target
9. Ketika variabel kondisi bernilai true, maka kondisi perulangan tersebut akan bernilai true lalu dinegasikan, kondisi = false sehingga program berhenti.
10. Output akhir berupa informasi total donasi dan jumlah donatur secara keseluruhan