

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL [No. MODUL]

[NAMA MODUL]



Disusun oleh:

HAFIZD SAMA'I SYAMSI

109082500183

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

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 word string
    var repetitions int
    fmt.Scan(&word, &repetitions)
    counter := 0
    for done := false; !done; {
        fmt.Println(word)
        counter++
        done = (counter >= repetitions)
    }
}
```

Screenshot program

```
VS Code Welcome | -GO guided1.go 2 X | -GO guided2.go 1 | -GO gui

- GO guided1.go > main
1 package main
2 import "fmt"
3 func main() {
4     var word string
5     var repetitions int
6     fmt.Scan(&word, &repetitions)
7     counter := 0
8     for done := false; !done; {
9         fmt.Println(word)
10        counter++
11        done = (counter >= repetitions)
12    }
13 }
14
15
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\go\praktek13> go run guided1.go
pagi 3
pagi
pagi
pagi
PS D:\go\praktek13> go run guided1.go
kursi 5
kursi
kursi
kursi
kursi
kursi
PS D:\go\praktek13>
```

Deskripsi program

Program ini dibuat untuk mencetak sebuah kata ke layar secara berulang sesuai jumlah yang dimasukkan oleh pengguna. Pada awal program, pengguna diminta memasukkan satu kata dan satu bilangan bulat yang menyatakan berapa kali kata tersebut akan ditampilkan. Program kemudian menggunakan sebuah variabel

penghitung untuk melacak jumlah pengulangan yang telah dilakukan. Proses pencetakan dilakukan di dalam perulangan for yang dikendalikan oleh kondisi boolean. Setiap kali perulangan berjalan, kata akan ditampilkan satu kali dan penghitung akan bertambah. Ketika jumlah pengulangan yang diinginkan sudah tercapai, kondisi perulangan berubah sehingga proses berhenti. Dengan demikian, program memastikan kata dicetak tepat sebanyak jumlah yang diminta pengguna.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var n int

    for {
        fmt.Print("Masukkan bilangan bulat positif: ")
        fmt.Scan(&n)

        if n > 0 {
            fmt.Printf("%d adalah bilangan bulat
positif\n", n)
            break
        } else {
            fmt.Println("Input bukan bilangan bulat
positif, coba lagi.")
        }
    }
}
```

Screenshot program

```
-GO guided1.go 2 -GO guided2.go 1 X -GO guided3.go 1
- GO guided2.go > ...
3 func main() {
5     var n int
6
7     for {
8         fmt.Print("Masukkan bilangan bulat positif: ")
9         fmt.Scan(&n)
10
11         if n > 0 {
12             fmt.Printf("%d adalah bilangan bulat positif\n", n)
13             break
14         } else {
15             fmt.Println("Input bukan bilangan bulat positif, coba lagi.")
16         }
17     }
18 }
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Masukkan bilangan bulat positif: -2
Input bukan bilangan bulat positif, coba lagi.
Masukkan bilangan bulat positif: -1
Input bukan bilangan bulat positif, coba lagi.
Masukkan bilangan bulat positif: 0
Input bukan bilangan bulat positif, coba lagi.
Masukkan bilangan bulat positif: 5
5 adalah bilangan bulat positif
PS D:\go\praktek13> go run guided2.go
Masukkan bilangan bulat positif: 17
17 adalah bilangan bulat positif
PS D:\go\praktek13>
```

Deskripsi program

Program ini dibuat untuk memvalidasi input bilangan bulat positif dari pengguna. Program menggunakan perulangan for tanpa kondisi sehingga proses input akan terus berulang. Di setiap iterasi, pengguna diminta memasukkan sebuah bilangan bulat. Nilai yang dimasukkan kemudian dicek dengan kondisi $n > 0$. Jika nilai tersebut memenuhi syarat sebagai bilangan bulat positif, program menampilkan pesan yang menyatakan bahwa input valid lalu menghentikan perulangan menggunakan break. Sebaliknya, jika nilai yang dimasukkan nol atau negatif, program menampilkan pesan kesalahan dan kembali meminta input. Dengan mekanisme ini, program memastikan hanya bilangan bulat positif yang dapat diterima sebelum program selesai dijalankan.

3. Guided 3

Source Code

```
package main

import "fmt"

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

Screenshot program

The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for 'Welcome', 'guided1.go', 'guided2.go', and 'guided3.go'. The 'guided3.go' tab is active, displaying the following Go code:

```
go guided3.go > main
3 func main() {
4     var x int
5     var y int
6     var selesai bool
7     fmt.Scan(&x, &y)
8     for selesai = false; !selesai; {
9         x = x - y
10        fmt.Println(x)
11        selesai = x <= 0
12    }
13    fmt.Println(x == 0)
14 }
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS (4), OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), and PORTS.

The terminal window displays the execution of the program:

```
-1
false
PS D:\go\praktek13> go run guided3.go
5
2
3
false
PS D:\go\praktek13> go run guided3.go
15
3
12
9
6
3
0
true
PS D:\go\praktek13> go run guided3.go
25
5
20
15
10
```

Deskripsi program

Program ini melakukan pengurangan berulang antara dua bilangan bulat yang dimasukkan pengguna. Nilai x akan terus dikurangi dengan y di dalam perulangan. Setiap hasil pengurangan langsung ditampilkan ke layar sehingga pengguna bisa melihat perubahan nilai x secara bertahap. Perulangan berhenti ketika nilai x sudah lebih kecil atau sama dengan nol. Setelah perulangan selesai, program menampilkan

hasil berupa nilai boolean yang menunjukkan apakah nilai akhir x tepat sama dengan nol atau tidak.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var n int

    fmt.Println("Masukan biangan")
    fmt.Scan(&n)

    jumlahdigit := 0

    for {
        jumlahdigit++
        n = n / 10

        if n == 0 {
            break
        }
    }

    fmt.Println(jumlahdigit)
}
```

Screenshoot program

The screenshot shows a terminal window with the following content:

```
PS D:\go\praktek13> go run soal1.go
Masukan biangan 5
1
PS D:\go\praktek13> go run soal1.go
Masukan biangan 234
3
PS D:\go\praktek13> go run soal1.go
GetFileAttributesEx soal1.go: The system cannot find the file specified.
PS D:\go\praktek13> go run soal1.go
Masukan biangan 78787
5
PS D:\go\praktek13> go run soal1.go
Masukan biangan 1894256
7
PS D:\go\praktek13>
```

On the right side of the terminal, there is a small text editor window displaying student information:

```
NAMA :HAFIZD SAMA'I SYAMSI
KELAS : S1IF-13-07
NIM : 109082500183
```

Deskripsi program

Program ini digunakan untuk menghitung jumlah digit dari sebuah bilangan bulat yang dimasukkan oleh pengguna. Setelah pengguna memasukkan angka, program menyiapkan sebuah variabel penghitung untuk mencatat banyaknya digit. Proses perhitungan dilakukan dengan membagi bilangan tersebut dengan 10 secara berulang. Setiap kali pembagian dilakukan, satu digit dianggap sudah dihitung dan nilai penghitung bertambah satu. Perulangan dihentikan ketika hasil pembagian sudah menjadi nol, yang menandakan tidak ada digit tersisa. Di akhir program, jumlah digit dari bilangan tersebut ditampilkan ke layar.

2. Tugas 2

Source code

```
package main

import (
    "fmt"
    "math"
```

```
)  
func main() {  
    var input float64  
    fmt.Print("Masukan bilangan desimal")  
    fmt.Scan(&input)  
  
    batasAtas := math.Ceil(input)  
  
    for i := input + 0.1; i <= batasAtas; i += 0.1 {  
        if i == float64(int64(i)) {  
            fmt.Printf("%.0f\n", i)  
        } else {  
            fmt.Printf("%.1f\n", i)  
        }  
    }  
}
```

Screenshot program

The screenshot shows a code editor with a Go file named `soal2.go`. The code prints floating-point numbers from a user input to two decimal places. The terminal below shows the execution of the program and its output.

```
VS Code Welcome
File Explorer
Search
Terminal
File Edit View Insert Run Terminal Help

∞ soal2.go > main
∞ guided1.go 4 ∞ guided2.go 1 ∞ guided3.go 1 ∞ soal1.go 1 ×

∞ soal2.go > main
3 import "fmt"
4 import "math"
5
6 }
7 func main() {
8     var input float64
9     fmt.Println("Masukan bilangan desimal")
10    fmt.Scan(&input)
11
12    batasAtas := math.Ceil(input)
13
14    for i := input + 0.1; i <= batasAtas; i += 0.1 {
15        if i == float64(int64(i)) {
16            fmt.Printf("%.0f\n", i)
17        } else {
18            fmt.Printf("%.1f\n", i)
19        }
20    }
21 }
22

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Masukan bilangan desimal 0.2
0.3
0.4
0.5
0.8
0.9
1.0
PS D:\go\praktek13> go run soal2.go
Masukan bilangan desimal 2.7
2.8
2.9
PS D:\go\praktek13> go run soal2.go
Masukan bilangan desimal 2.7
2.8

+ v ...
File Edit View Insert Run Terminal Help
NAMA : HAFIZD SAMA'I SYAMSI
KELAS : STIF-13-07
NIM : 109082500183

Ln 3, Col 20 65 character Plain t 100% Wind UTF-8
```

Deskripsi program

Program ini digunakan untuk menampilkan deretan bilangan desimal yang dimulai sedikit di atas nilai input hingga mencapai bilangan bulat terdekat di atasnya. Pengguna memasukkan sebuah bilangan desimal, lalu program menentukan batas atas dengan membulatkan nilai tersebut ke atas menggunakan fungsi math.Ceil. Setelah itu, program melakukan perulangan dengan kenaikan 0,1 dimulai dari nilai input ditambah 0,1. Setiap nilai hasil perulangan dicek: jika nilainya merupakan bilangan bulat (tidak memiliki angka di belakang koma), maka ditampilkan tanpa desimal. Jika masih berupa bilangan desimal, maka ditampilkan dengan satu angka di belakang koma. Proses ini berlanjut sampai nilai perulangan mencapai batas atas. Dengan cara ini, program menampilkan urutan angka secara bertahap sekaligus membedakan tampilan antara bilangan bulat dan bilangan desimal.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var target int
```

```

fmt.Scan(&target)

total := 0
donatur := 0

for total < target {
    var donasi int
    fmt.Scan(&donasi)

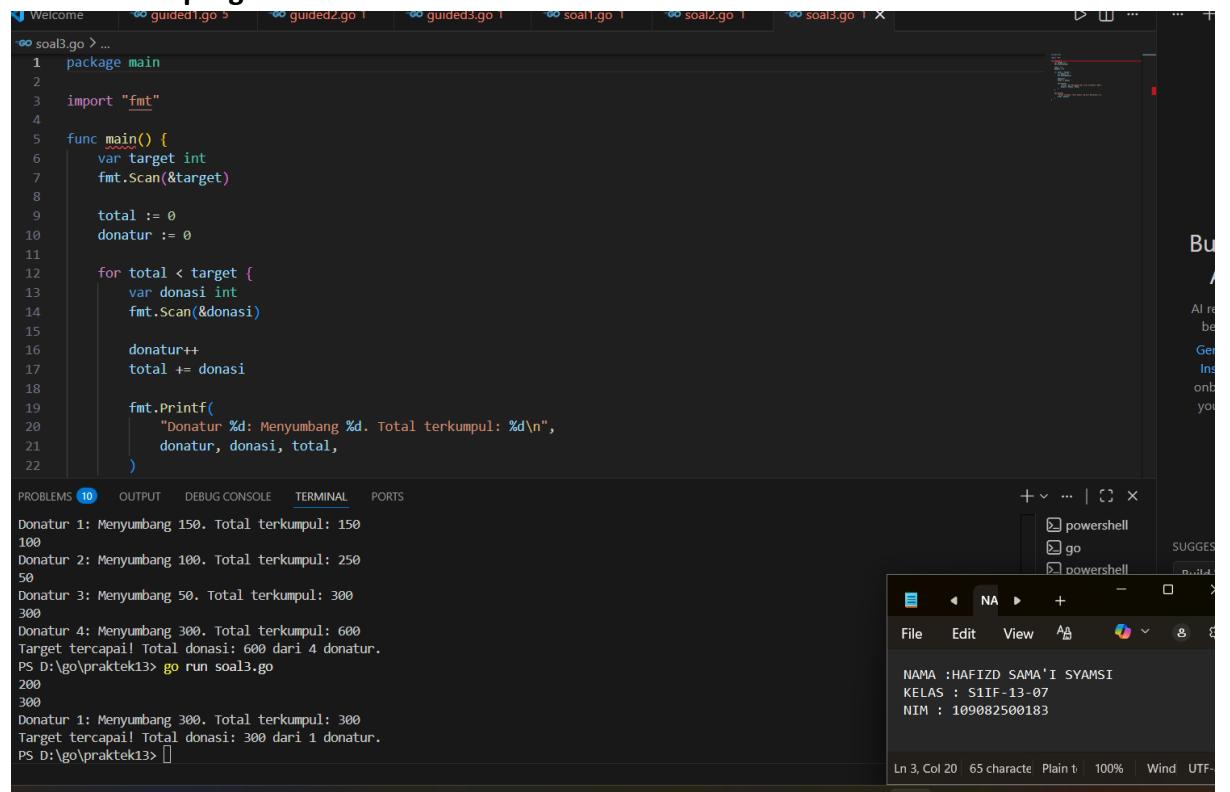
    donatur++
    total += donasi

    fmt.Printf(
        "Donatur %d: Menyumbang %d. Total terkumpul:
        %d\n",
        donatur, donasi, total,
    )
}

fmt.Printf(
    "Target tercapai! Total donasi: %d dari %d
    donatur.\n",
    total, donatur,
)
}

```

Screenshoot program



```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var target int
7     fmt.Scan(&target)
8
9     total := 0
10    donatur := 0
11
12    for total < target {
13        var donasi int
14        fmt.Scan(&donasi)
15
16        donatur++
17        total += donasi
18
19        fmt.Printf(
20            "Donatur %d: Menyumbang %d. Total terkumpul: %d\n",
21            donatur, donasi, total,
22        )
23    }
24
25    fmt.Printf(
26        "Target tercapai! Total donasi: %d dari %d
27        donatur.\n",
28        total, donatur,
29    )
30}

```

The screenshot shows the code in a code editor. Below the code, the terminal output shows four runs of the program with different target values (150, 100, 50, 300) and the final successful run with target 300.

```

Donatur 1: Menyumbang 150. Total terkumpul: 150
100
Donatur 2: Menyumbang 100. Total terkumpul: 250
50
Donatur 3: Menyumbang 50. Total terkumpul: 300
300
Donatur 4: Menyumbang 300. Total terkumpul: 600
Target tercapai! Total donasi: 600 dari 4 donatur.
PS D:\go\praktek13> go run soal3.go
200
300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.
PS D:\go\praktek13> []

```

The terminal also displays the user's profile information:

```

NAMA :HAFIZD SAMA'I SYAMSI
KELAS : S11F-13-07
NIM : 109082500183

```

Deskripsi program

Program ini mensimulasikan proses pengumpulan donasi hingga mencapai target tertentu. Di awal, pengguna memasukkan nilai target donasi yang ingin dicapai. Program kemudian mencatat total donasi yang terkumpul dan jumlah donatur yang sudah menyumbang.

Selama total donasi masih lebih kecil dari target, program akan terus meminta input berupa jumlah donasi dari setiap donatur. Setiap kali ada donasi masuk, program menambah jumlah donatur, menambahkan nilai donasi ke total, lalu langsung menampilkan informasi donatur ke berapa, besar sumbangan yang diberikan, dan total dana yang sudah terkumpul saat itu.

Ketika total donasi sudah mencapai atau melebihi target, perulangan dihentikan. Di akhir program, ditampilkan pesan bahwa target telah tercapai, beserta total donasi akhir dan jumlah donatur yang berpartisipasi.