

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 13 REPEAT-UNTIL



Disusun oleh:

MUHAMMAD ADDARU QUTHNI

109082500034 S1IF-13-02

Asisten Praktikum

Adithana dharma putra

Alfin Ilham Berlianto

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 banyak_kata int

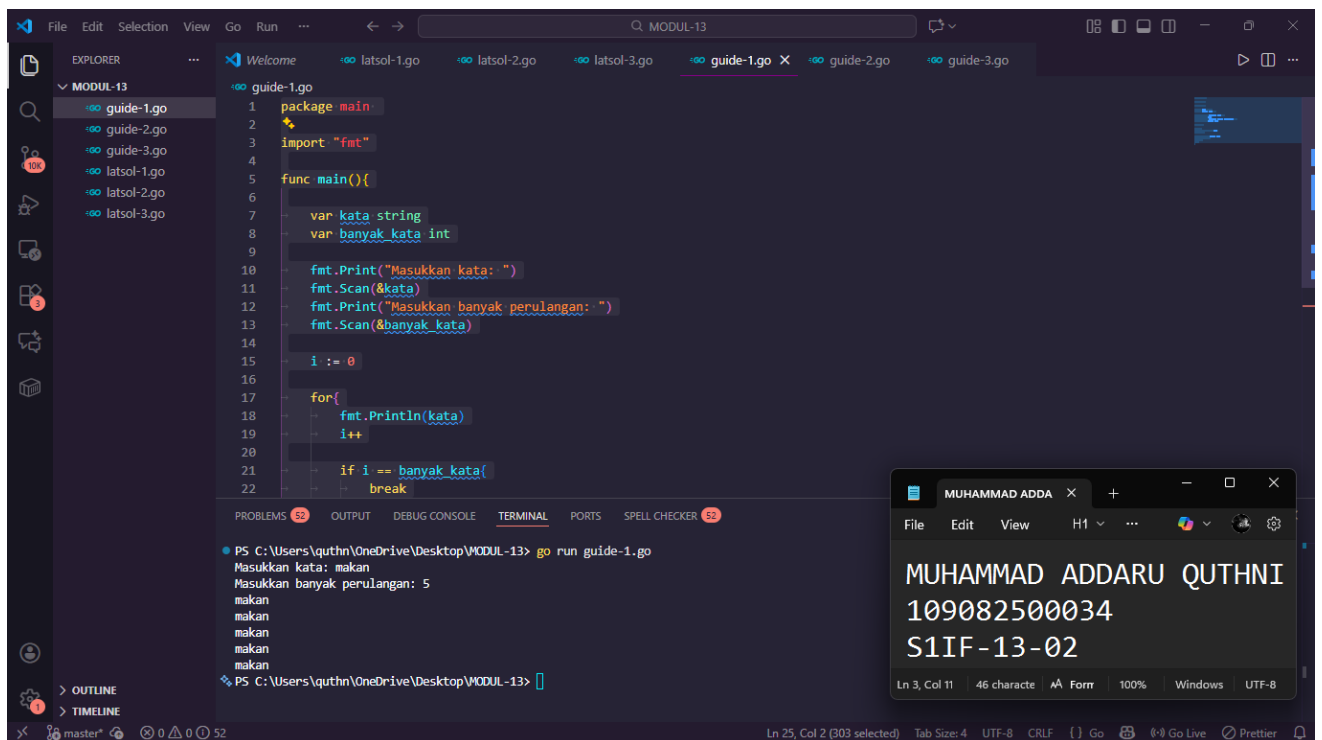
    fmt.Print("Masukkan kata: ")
    fmt.Scan(&kata)
    fmt.Print("Masukkan banyak perulangan: ")
    fmt.Scan(&banyak_kata)

    i := 0

    for{
        fmt.Println(kata)
        i++

        if i == banyak_kata{
            break
        }
    }
}
```

Screenshoot program



Deskripsi program

- Pertama minta user masukan kata (misal "halo")
- Terus minta berapa kali mau diulang (misal 3 kali)
- Nah di bagian loopingnya pake cara yang agak unik, dia bikin infinite loop (for{}) tanpa kondisi awal
- Di dalam loop, dia cetak kata terus tambahin counter i
- Pas counter i sama dengan jumlah pengulangan yang diminta, dia pake break buat keluar dari loop

2. Guided 2

Source Code

```
package main

import "fmt"

func main(){

    var bilangan int

    var continueLoop bool

    for continueLoop = true; continueLoop;{
```

```

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

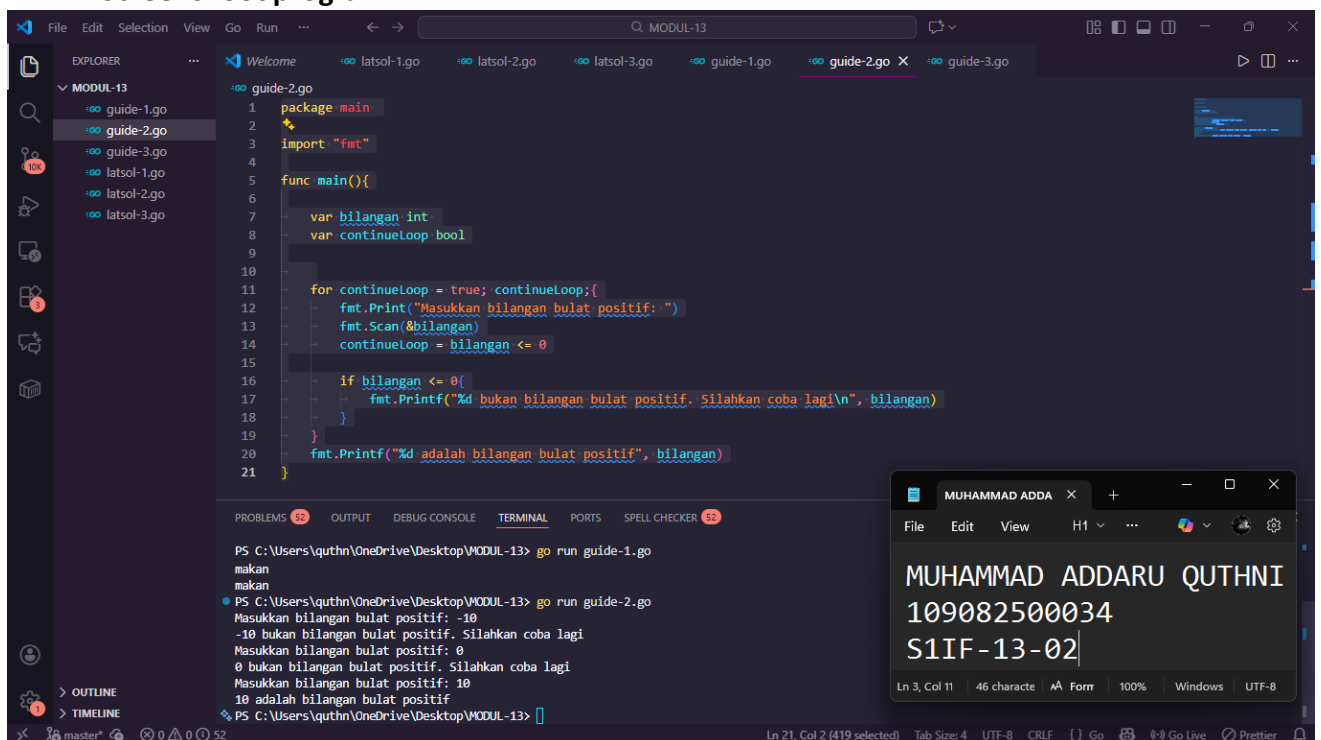
    continueLoop = bilangan <= 0

    if bilangan <= 0{
        fmt.Printf("%d bukan bilangan bulat positif.
        Silahkan coba lagi\n", bilangan)
    }

    fmt.Printf("%d adalah bilangan bulat positif",
    bilangan)
}

```

Screenshoot program



Deskripsi program

Pertama-tama program bakal minta user ngisi angka dengan tulisan "Masukkan bilangan bulat positif: ". Nah di sini program pake perulangan yang agak unik, dia pake variabel continueLoop yang awalnya di-set true biar loop-nya jalan terus.

Pas user masukin angka, program langsung ngecek tuh angka. Kalau angkanya minus atau nol, dia bakal marah-marah dikit: "-5 bukan bilangan bulat positif. Silahkan coba lagi" (misalnya). Terus karena masih salah, loop-nya bakal jalan terus dan minta input lagi.

Yang menarik di sini, logikanya dibalik dibanding for biasa. Biasanya kita bikin kondisi "jalan selama benar", ini malah "lanjutkan looping selama masih salah inputnya". Jadi variabel `continueLoop` bakal tetap `true` selama user masukan bilangan negatif atau nol. Begitu user akhirnya nurut dan masukan bilangan positif (misal 5), baru deh loopingnya berhenti karena `continueLoop` jadi `false`. Program pun lega dan bilang *"5 adalah bilangan bulat positif"*.

pesan "coba lagi" dan kembali meminta input hingga kata sandi yang benar dimasukkan

3. Guided 3 Source

Code

```
package main

import "fmt"

func main(){

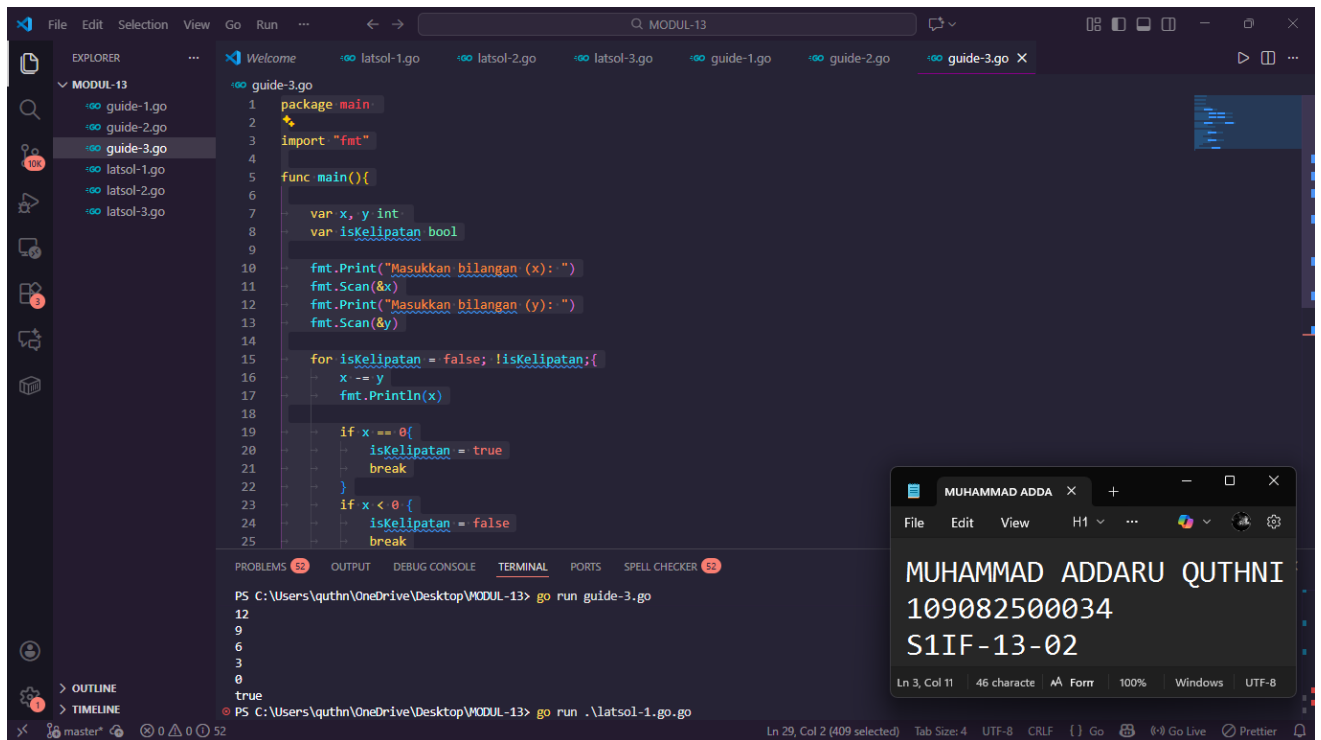
    var x, y int
    var isKelipatan bool

    fmt.Print("Masukkan bilangan (x): ")
    fmt.Scan(&x)
    fmt.Print("Masukkan bilangan (y): ")
    fmt.Scan(&y)

    for isKelipatan = false; !isKelipatan;{
        x -= y
        fmt.Println(x)

        if x == 0{
            isKelipatan = true
            break
        }
        if x < 0 {
            isKelipatan = false
            break
        }
    }
    fmt.Println(isKelipatan)
}
```

Screenshoot program



Deskripsi program

- Pertama minta input dua bilangan x dan y
- Terus dia mulai looping dan mengurangi x dengan y berulang-ulang
- Setiap hasil pengurangan ditampilkan
- Kalau hasilnya pas nol, berarti x itu kelipatan dari y (true)
- Kalau hasilnya minus, berarti bukan kelipatan (false)
- Program berhenti pas ketemu jawabannya

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var num, count int

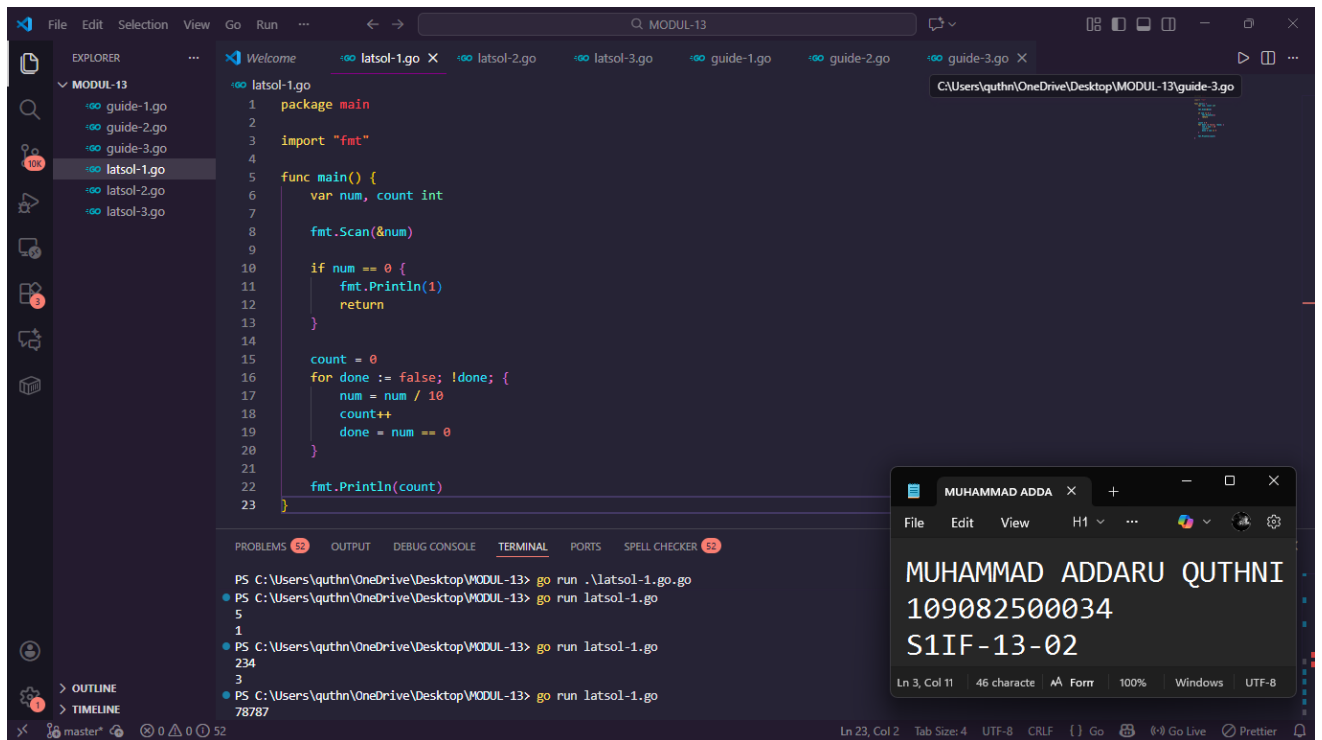
    fmt.Scan(&num)

    if num == 0 {
        fmt.Println(1)
        return
    }

    count = 0
    for done := false; !done; {
        num = num / 10
        count++
        done = num == 0
    }

    fmt.Println(count)
}
```

Screenshoot program



Deskripsi program

- Baca bilangan dari input
- Jika bilangan 0, langsung output 1 karena 0 memiliki 1 digit
- Gunakan repeat-until untuk membagi bilangan dengan 10 secara berulang
- Hitung berapa kali pembagian dilakukan hingga bilangan menjadi 0
- Tampilkan hasil hitungan sebagai jumlah digit

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var num float64
    fmt.Scan(&num)

    current := int(num * 10)
    target := ((current / 10) + 1) * 10

    first := true
    for done := false; !done; {
        current += 1

        if !first {
            fmt.Print(" ")
        }

        if current % 10 == 0 {
            fmt.Print(current / 10)
        } else {
            fmt.Printf("%d.%d", current/10, current%10)
        }

        first = false
        done = current >= target
    }
}
```

```
}  
  
}
```

Screenshoot program

The screenshot shows a Go program in VS Code and its execution output in a terminal window.

Go Program (latsol-2.go):

```
4  
5 func main() {  
6     var num float64  
7     fmt.Scan(&num)  
8     ...  
9     current := int(num * 10)  
10    target := ((current / 10) + 1) * 10  
11    ...  
12    first := true  
13    for done := false; !done; {  
14        current += 1  
15        ...  
16        if !first {  
17            fmt.Print(".")  
18        }  
19        ...  
20        if current % 10 == 0 {  
21            fmt.Print(current / 10)  
22        } else {  
23            fmt.Printf("%d.%d", current/10, current%10)  
24        }  
25        ...  
26        first = false  
27        done = current >= target  
28    }
```

Terminal Output:

```
PS C:\Users\quthn\OneDrive\Desktop\MODUL-13> go run latsol-2.go  
0.2  
0.3 0.4 0.5 0.6 0.7 0.8 0.9 1  
PS C:\Users\quthn\OneDrive\Desktop\MODUL-13> go run latsol-2.go  
2.7  
2.8 2.9 3
```

Terminal Window (MUHAMMAD ADDA):

```
MUHAMMAD ADDARU QUTHNI  
109082500034  
S1IF-13-02
```

- Baca
bilangan
desimal dari
input
- Konversi ke
bilangan
bulat
persepuluhan untuk
menghindari
masalah
presisi
- Hitung
target
sebagai
pembulatan
ke atas dari
bilangan
awal
- Gunakan
repeat-until
untuk
menambahkan 1
(mewakili
0.1) terus
menerus
- Format
output
sesuai
contoh,
bilangan
bulat
ditampilkan
tanpa
desimal
- Hentikan
perulangan
ketika
mencapai
atau
melebihi
target

3. Tugas 3 Source code

```
package main

import "fmt"

func main() {
    var target, donation, total, donorCount int

    fmt.Scan(&target)

    total = 0
    donorCount = 0

    for done := false; !done; {
        fmt.Scan(&donation)
        donorCount++
        total += donation

        fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n",
            donorCount, donation, total)

        done = total >= target
    }

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

Screenshoot program

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

PS C:\Users\quthn\OneDrive\Desktop\MODUL-13> go run latsol-3.go

300
100
Donatur 1: Menyumbang 100. Total terkumpul: 100
50
Donatur 2: Menyumbang 50. Total terkumpul: 150
200
Donatur 3: Menyumbang 200. Total terkumpul: 350
Target tercapai! Total donasi: 350 dari 3 donatur.

MUHAMMAD ADDARU QUTHNI
109082500034
S1IF-13-02

- Baca target donasi dari input pertama
- Gunakan repeat-until untuk terus menerima input donasi
- Hitung total donasi dan jumlah donatur
- Tampilkan detail setiap donasi dan total terkumpul
- Hentikan perulangan ketika total donasi mencapai atau melebihi target
- Tampilkan pesan akhir dengan total donasi dan jumlah donatur