

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 13
REPEAT - UNTIL**



Disusun oleh:

ELSA DWI RIZQIYANTI

109082500090

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 jumlah int

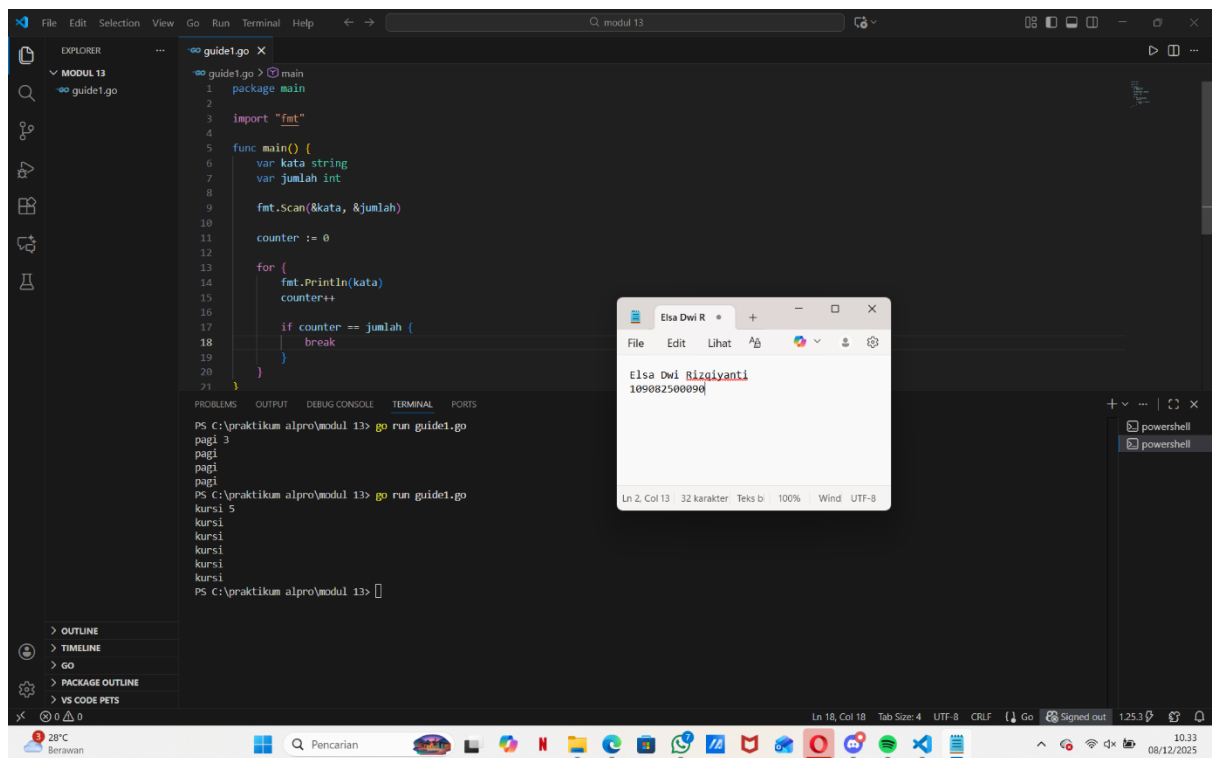
    fmt.Scan(&kata, &jumlah)

    counter := 0

    for {
        fmt.Println(kata)
        counter++

        if counter == jumlah {
            break
        }
    }
}
```

Screenshoot program



Deskripsi program

Isi Perulangan

fmt.Println(kata)

counter++

fmt.Println(kata) mencetak isi variabel kata ke layar.

counter++ menambah nilai counter sebanyak 1 setiap kali loop berjalan.

Kondisi Berhenti (Until)

if counter == jumlah {

break

}

Jika counter sudah sama dengan jumlah, maka:

break menghentikan perulangan.

Artinya, kata akan dicetak sebanyak jumlah kali.

Guided 2

Source Code

```
package main

import "fmt"

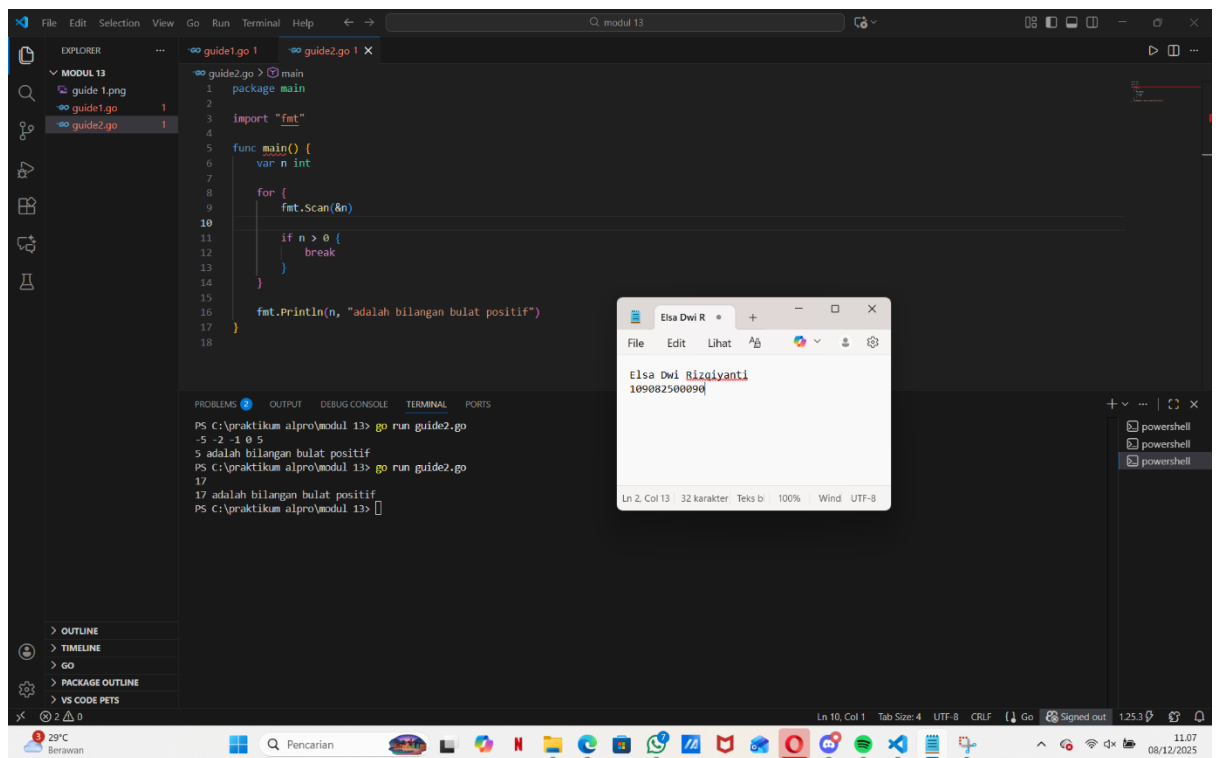
func main() {
    var n int

    for {
        fmt.Scan(&n)

        if n > 0 {
            break
        }
    }

    fmt.Println(n, "adalah bilangan bulat positif")
}
```

Screenshoot program



Deskripsi program

Kondisi Berhenti (Until)

```
if n > 0 {  
    break  
}
```

Program memeriksa nilai n:

Jika $n > 0$ (bilangan bulat positif),

maka perulangan dihentikan dengan break.

Jika $n \leq 0$, perulangan berlanjut dan program meminta input ulang.

Output Hasil

```
fmt.Println(n, "adalah bilangan bulat positif")
```

Setelah keluar dari perulangan, program menampilkan pesan bahwa nilai n adalah bilangan bulat positif.

Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var x, y int

    fmt.Scan(&x, &y)

    sisa := x

    for {

        sisa = sisa - y

        fmt.Println(sisa)

        if sisa <= 0 {

            break

        }

    }

    if sisa == 0 {

        fmt.Println(true)

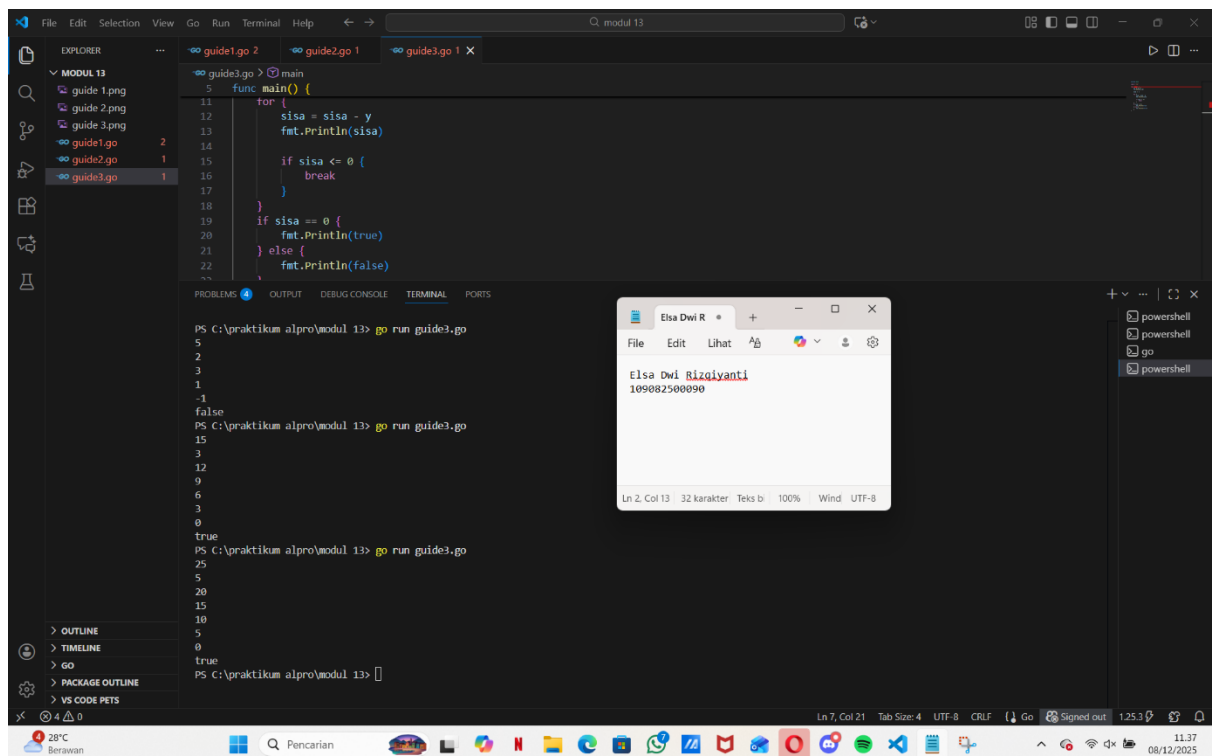
    } else {

        fmt.Println(false)

    }

}
```

Screenshoot program



Deskripsi program

Inisialisasi Variabel sisa

sisa := x

Variabel sisa menyimpan nilai awal x.

Digunakan untuk menampung hasil pengurangan berulang.

Proses Pengurangan dan Output

sisa = sisa - y

fmt.Println(sisa)

Setiap iterasi, nilai sisa dikurangi y.

Hasil sementara langsung dicetak ke layar.

Kondisi Berhenti

if sisa <= 0 {

break

}

Perulangan dihentikan jika:

sisa kurang dari atau sama dengan 0

Artinya, pengurangan berhenti ketika nilai sudah habis atau negatif.

Pemeriksaan Hasil Akhir

if sisa == 0 {

```
        fmt.Println(true)
    } else {
        fmt.Println(false)
    }
```

Jika sisa == 0:

Program mencetak true

Artinya x habis dibagi y (pengurangan tepat)

Jika sisa \neq 0:

Program mencetak false

Artinya x tidak habis dibagi y

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var n int

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

        if n > 0 {
            break
        }

        fmt.Println("Input harus bilangan positif!")
    }

    count := 0
    tmp := n

    for tmp > 0 {
        tmp /= 10
    }
}
```

```

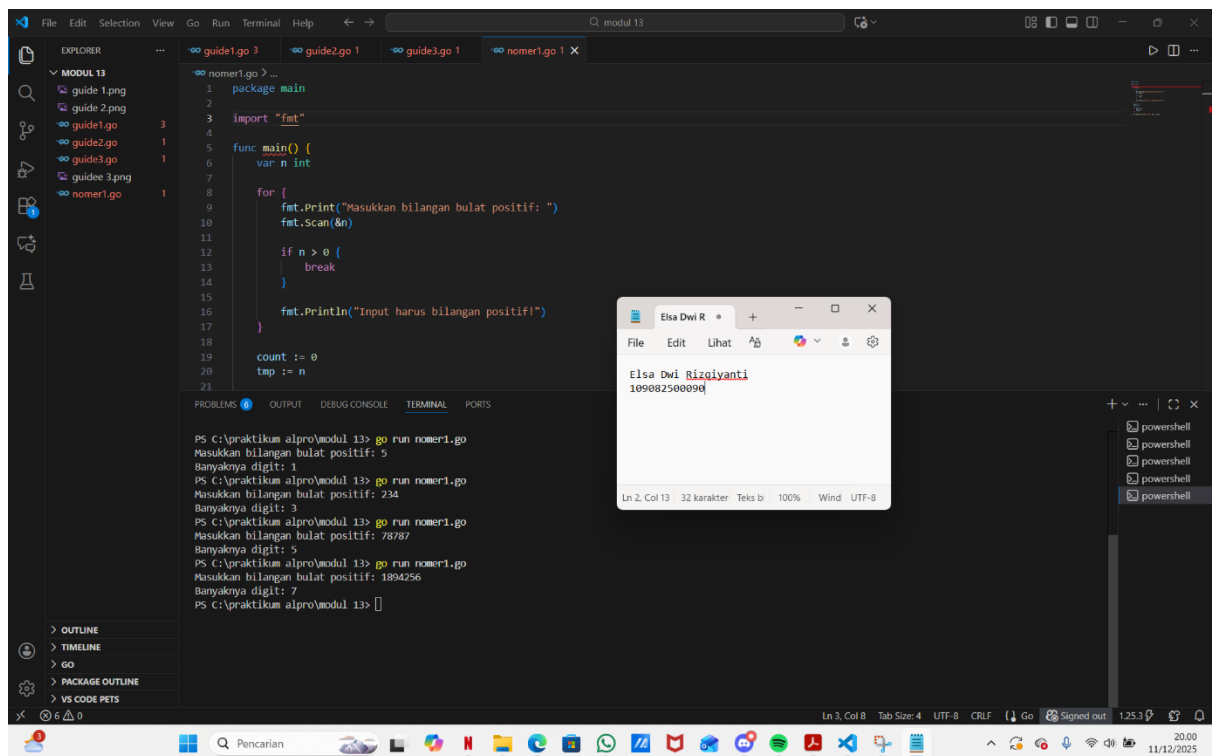
        count++

    }

    fmt.Printf("Banyaknya digit: %d\n", count)
}

```

Screenshoot program



Deskripsi program

Perulangan Validasi Input (Repeat–Until):

for {

```

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

```

```

    if n > 0 {
        break
    }

```

```

    fmt.Println("Input harus bilangan positif!")

```

}

Program meminta input terus-menerus sampai pengguna memasukkan bilangan positif.

Jika input $n \leq 0$:

Program menampilkan pesan kesalahan

Mengulang input kembali

Jika $n > 0$:

Perulangan dihentikan dengan break

Inisialisasi Variabel Penghitung:

count := 0

tmp := n

count digunakan untuk menghitung jumlah digit.

tmp adalah variabel sementara agar nilai n tidak berubah

Perulangan Menghitung Digit:

```
for tmp > 0 {  
    tmp /= 10  
    count++  
}
```

Setiap iterasi:

tmp dibagi 10 (menghapus satu digit dari belakang)

count bertambah 1

Proses berhenti saat tmp menjadi 0

Jumlah iterasi = jumlah digit bilangan

Menampilkan Hasil:

```
fmt.Printf("Banyaknya digit: %d\n", count)
```

Program menampilkan jumlah digit dari bilangan yang dimasukkan.

2. Tugas 2

Source code

```
package main  
  
import "fmt"  
  
func main() {  
    var x float64  
    fmt.Scan(&x)  
  
    atas := int(x)  
    if float64(atas) != x {  
        atas++  
    }  
}
```

```

    }

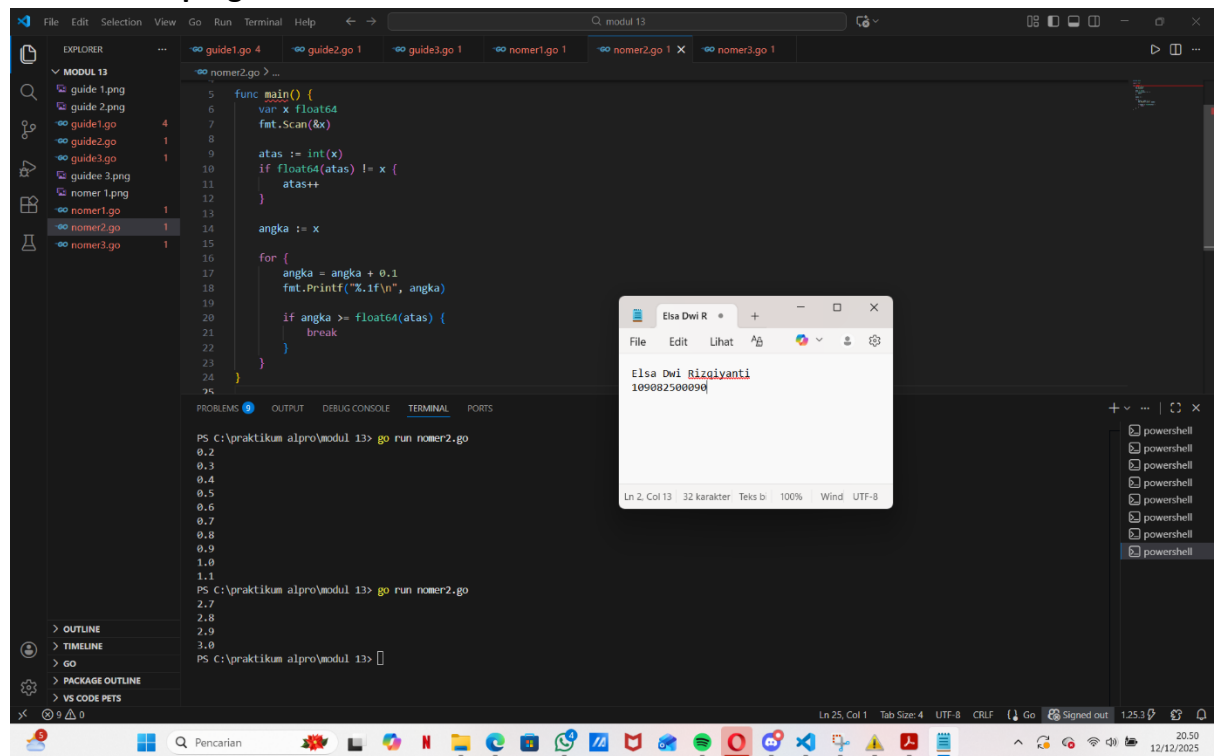
    angka := x

    for {
        angka = angka + 0.1
        fmt.Printf("%.1f\n", angka)

        if angka >= float64(atas) {
            break
        }
    }
}

```

Screenshoot program



Deskripsi program

Menentukan Batas Atas (Ceiling):

```

atas := int(x)
if float64(atas) != x {
    atas++
}

```

Nilai `x` diubah menjadi bilangan bulat dengan `int(x)` (pembulatan ke bawah).

Jika `x` bukan bilangan bulat, maka:

atas ditambah 1

Dengan demikian, atas adalah pembulatan ke atas (ceiling) dari x.

Contoh:

$x = 3.0 \rightarrow \text{atas} = 3$

$x = 3.2 \rightarrow \text{atas} = 4$

Inisialisasi Variabel Perulangan:

angka := x

angka digunakan sebagai nilai awal perulangan.

Dimulai dari nilai input x.

Proses Penambahan dan Output:

angka = angka + 0.1

fmt.Printf("%.1f\n", angka)

Setiap iterasi:

angka ditambah 0.1

Dicetak dengan 1 angka di belakang koma

Kondisi Berhenti:

```
if angka >= float64(atas) {
```

```
    break
```

```
}
```

Perulangan dihentikan jika:

nilai angka sudah mencapai atau melebihi batas atas

Ini merupakan bagian until dari repeat-until.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var target int
    fmt.Scan(&target)

    total := 0
    donatur := 0
```

```

for {
    var donasi int
    fmt.Scan(&donasi)

    donatur++
    total += donasi

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

    if total >= target {
        break
    }
}

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

```

Screenshoot program

The screenshot shows a VS Code editor with a Go program in a file named `nomer3.go`. The program is a loop that prompts the user to enter a donation amount, increments a counter, and adds the amount to a total. It prints the current state after each input and breaks the loop when the total reaches or exceeds a target value of 300.

```

5 func main() {
12     for {
13         var donasi int
14         fmt.Scan(&donasi)
15
16         donatur++
17         total += donasi
18
19         fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n",
20             donatur, donasi, total)
21
22         if total >= target {
23             break
24         }
25     }
26
27     fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n",
28         total, donatur)
29 }
30

```

The terminal output shows the program being run multiple times, demonstrating the loop's behavior as it accumulates donations until the target is reached.

```

PS C:\praktikum alpro\modul 13> go run nomer3.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 C:\praktikum alpro\modul 13> go run nomer3.go
300 150 100 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 C:\praktikum alpro\modul 13> go run nomer3.go
200 300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.
PS C:\praktikum alpro\modul 13>

```

Deskripsi program

Proses Penjumlahan dan Output:

```
donatur++
```

```
total += donasi
```

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

Jumlah donatur bertambah satu setiap ada input donasi.

Nilai donasi ditambahkan ke total.

Program menampilkan:

nomor donatur

jumlah donasi

total donasi sementara

Kondisi Berhenti

```
if total >= target {  
    break  
}
```

Perulangan dihentikan jika total donasi sudah mencapai atau melebihi target.

Ini merupakan bagian until dari konsep repeat–until.

Output Akhir

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

Setelah perulangan berhenti, program menampilkan:

total donasi akhir

jumlah donatur