

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 12  
WHILE-LOOP**



**Disusun oleh:**

**Dharma Chandra Viriya**

**109082500052**

**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 - Menampilkan Deret Bilangan Faktorial

#### Source Code

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("Masukan n: ")
    fmt.Scan(&n)

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

    for n > 0 {
        fmt.Print(n, " ")

        if n != 1 {
            fmt.Print("x" + " ")
        }

        n--
    }

    fmt.Println()
}
```

## Screenshoot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7
8     fmt.Print("Masukan n: ")
9     fmt.Scan(&n)
10
11     if n == 0 {
12         fmt.Println(1)
13         return
14     }
15
16     for n > 0 {
17         fmt.Print(n, " ")
18
19         if n != 1 {
20             fmt.Print("x" + " ")
21         }
22         n--
23     }
24     fmt.Println()
25 }
```

Terminal Output:

```
iancelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Guided1/main.go
Masukan n: 0
1
iancelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Guided1/main.go
Masukan n: 5
5 x 4 x 3 x 2 x 1
iancelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Guided1/main.go
Masukan n: 10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
iancelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$
```

## Deskripsi program

Kode program di atas berfungsi menampilkan deretan angka menurun dari input pengguna dengan pemisah "x". Program menggunakan package main dan library fmt untuk menangani input dan output. Di dalam function main, variabel n dideklarasikan untuk menampung nilai yang dimasukkan melalui fmt.Scan. Program terlebih dahulu memeriksa apakah n bernilai 0. Jika ya, program langsung mencetak angka 1 dan berhenti. Jika tidak, program menjalankan perulangan for selama  $n > 0$ . Pada setiap iterasi, program mencetak nilai n dan, jika n bukan 1, program menambahkan tanda "x" sebagai pemisah. Nilai n kemudian dikurangi satu hingga loop selesai. Setelah perulangan berakhir, program menampilkan baris baru untuk merapikan output.

## 2. Guided 2 - Program Untuk Login Suatu Aplikasi

```

package main

import "fmt"

func main() {

    const validToken string = "12345abcde"

    var token string

    fmt.Print("Masukan token: ")

    fmt.Scan(&token)

    for token != validToken {

        fmt.Println("Gagal Login Token Anda Salah")

        fmt.Print("Masukan token: ")

        fmt.Scan(&token)

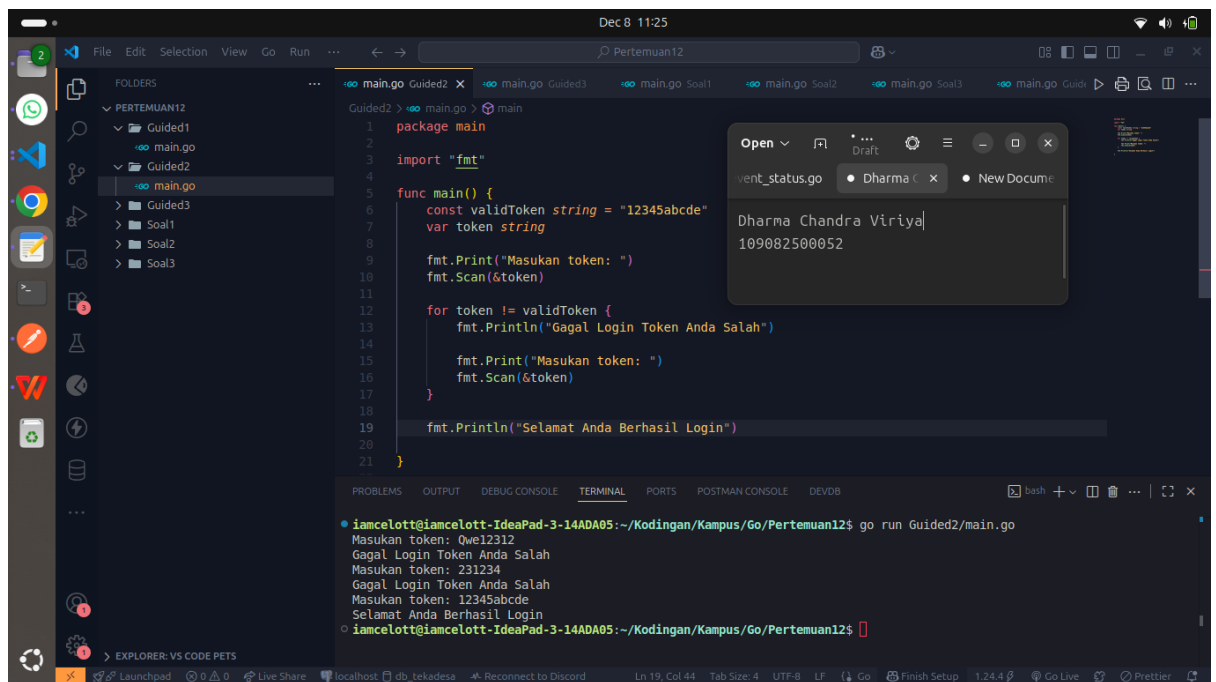
    }

    fmt.Println("Selamat Anda Berhasil Login")

}

```

## Screenshoot program



## Deskripsi program

Kode program di atas berfungsi memvalidasi token login yang dimasukkan pengguna. Program menggunakan package main dan library fmt untuk menangani input dan output. Di dalam function main, sebuah konstanta bernama validToken didefinisikan sebagai token yang benar, dan variabel token disiapkan untuk menampung input pengguna. Program meminta pengguna memasukkan token melalui fmt.Scan. Jika token yang dimasukkan tidak sama dengan validToken, program masuk ke dalam perulangan for yang terus menampilkan pesan kesalahan dan meminta input ulang. Perulangan hanya berhenti ketika pengguna memasukkan token yang sesuai. Setelah token cocok dengan validToken, program menampilkan pesan bahwa login berhasil.

### **3. Guided 3 - Program Mencetak Bilangan Pertama Dalam Deret Fibonacci**

#### **Source Code**

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("Masukan n: ")
    fmt.Scan(&n)

    i := 2
    a := 0
    b := 1

    fmt.Print(a, " ")
    fmt.Print(b, " ")

    for i < n {
        temp := a + b
        fmt.Print(temp, " ")

        a = b
        b = temp

        i++
    }
    fmt.Println()
}
```

**Screenshoot program**

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7
8     fmt.Print("Masukan n: ")
9     fmt.Scan(&n)
10
11     i := 2
12     a := 0
13     b := 1
14
15     fmt.Print(a, " ")
16     fmt.Print(b, " ")
17
18     for i < n {
19         temp := a + b
20         fmt.Print(temp, " ")
21     }
```

Terminal output:

```
iancelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Guided3/main.go
Masukan n: 5
0 1 1 2 3
iancelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Guided3/main.go
Masukan n: 2
0 1
iancelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Guided3/main.go
Masukan n: 10
0 1 1 2 3 5 8 13 21 34
```

## Deskripsi program

Kode program di atas berfungsi menampilkan deret Fibonacci sebanyak  $n$  angka. Program menggunakan package main dan library fmt untuk menangani input dan output. Di dalam function main, variabel  $n$  dideklarasikan untuk menampung jumlah angka deret yang diminta pengguna melalui `fmt.Scan`. Program menginisialisasi tiga variabel:  $i$  sebagai penghitung mulai dari 2, serta  $a$  dan  $b$  sebagai dua angka pertama deret Fibonacci, yaitu 0 dan 1. Kedua angka awal tersebut langsung dicetak sebagai output awal program. Selanjutnya, program menjalankan perulangan `for` selama  $i < n$ . Pada setiap iterasi, program menghitung angka berikutnya dengan menjumlahkan  $a$  dan  $b$ , lalu mencetak hasilnya. Nilai  $a$  dan  $b$  kemudian diperbarui untuk mempersiapkan perhitungan angka berikutnya, dan  $i$  ditambah satu. Setelah perulangan selesai, program mencetak baris baru untuk merapikan output.

## TUGAS

### 1. Tugas 1 - Program Counter Gagal Percobaan Login

#### Source code

```
package main
```

```
import "fmt"

func main() {
    var usr, pwd string

    counter := 0
    for usr != "Admin" && pwd != "Admin" {
        fmt.Print("Masukkan username: ")
        fmt.Scanln(&usr)
        fmt.Print("Masukkan password: ")
        fmt.Scanln(&pwd)
        fmt.Println("Username atau Password Yang  
Anda Masukkan Salah")
        counter++
    }

    fmt.Printf("%d percobaan gagal login\n", counter-1)
}
```

**Screenshoot program**



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var usr, pwd string
7
8     counter := 0
9     for usr != "Admin" && pwd != "Admin" {
10         fmt.Println("Masukkan username: ")
11         fmt.Scanln(&usr)
12         fmt.Println("Masukkan password: ")
13         fmt.Scanln(&pwd)
14         fmt.Println("Username atau Password Yang Anda Masukkan Salah")
15         counter++
16     }
17
18     fmt.Printf("%d percobaan gagal login\n", counter-1)
19 }
```

```
i@iarnelott@iarnelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Soal1/main.go
Masukkan username: User123
Masukkan password: user123
Username atau Password Yang Anda Masukkan Salah
Masukkan username: User
Masukkan password: admin
Username atau Password Yang Anda Masukkan Salah
Masukkan username: Admin
Masukkan password: Admin
Username atau Password Yang Anda Masukkan Salah
2 percobaan gagal login
```

## Deskripsi program

Kode program di atas berfungsi memvalidasi login berdasarkan username dan password. Program menggunakan package main dan library fmt untuk menangani input dan output. Di dalam function main, dua variabel bertipe string, yaitu usr dan pwd, disiapkan untuk menampung input pengguna, serta counter digunakan untuk menghitung jumlah percobaan. Program menjalankan perulangan for yang terus meminta pengguna memasukkan username dan password selama keduanya tidak bernilai "Admin". Pada setiap percobaan, program menampilkan pesan bahwa data login salah dan menambah nilai counter. Setelah username dan password sesuai dengan nilai yang ditentukan, perulangan berhenti dan program menampilkan jumlah percobaan login yang gagal dengan mencetak counter-1, karena percobaan terakhir adalah percobaan yang berhasil.

## 2. Tugas 2 - Program Mencacah Digit Bilangan Positif Dari Digit Terakhir

### Source code

```
package main

import "fmt"

func main() {
    var num int
    var ok bool = false

    fmt.Print("Masukkan: ")
    fmt.Scanln(&num)

    for !ok {
        tmp := num % 10
        fmt.Println(tmp)
        num /= 10
        if num == 0 {
            ok = true
        }
    }
}
```

**Screenshoot program**

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var num int
7     var ok bool = false
8
9     fmt.Print("Masukkan: ")
10    fmt.Scanln(&num)
11
12    for lok {
13        tmp := num % 10
14        fmt.Println(tmp)
15        num /= 10
16        if num == 0 {
17            ok = true
18        }
19    }
20 }
```

Terminal output:

```
iancelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Soal2/main.go
Masukkan: 2345
5
4
3
2

iancelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Soal2/main.go
Masukkan: 123
3
2
1
```

## Deskripsi program

Kode program di atas berfungsi menampilkan setiap digit dari sebuah angka mulai dari digit terakhir (satuan) hingga digit pertama. Program menggunakan package main dan library fmt untuk menangani input dan output. Di dalam function main, variabel num disiapkan untuk menampung angka dari pengguna, dan variabel ok digunakan sebagai penanda untuk menghentikan perulangan. Setelah menerima input angka melalui fmt.Scanln, program menjalankan perulangan for selama ok bernilai false. Pada setiap iterasi, program mengambil digit terakhir dengan operasi  $\text{num} \% 10$ , lalu mencetaknya. Setelah itu, nilai num dibagi 10 untuk menghilangkan digit yang sudah diproses. Jika hasil pembagian membuat num menjadi 0, variabel ok diubah menjadi true sehingga perulangan berhenti. Dengan logika ini, program mencetak seluruh digit angka dari belakang ke depan.

### 3. Tugas 3 - Program Mencari Hasil Integer Division Tanpa Menggunakan Operator Division

#### Source code

```
package main

import "fmt"

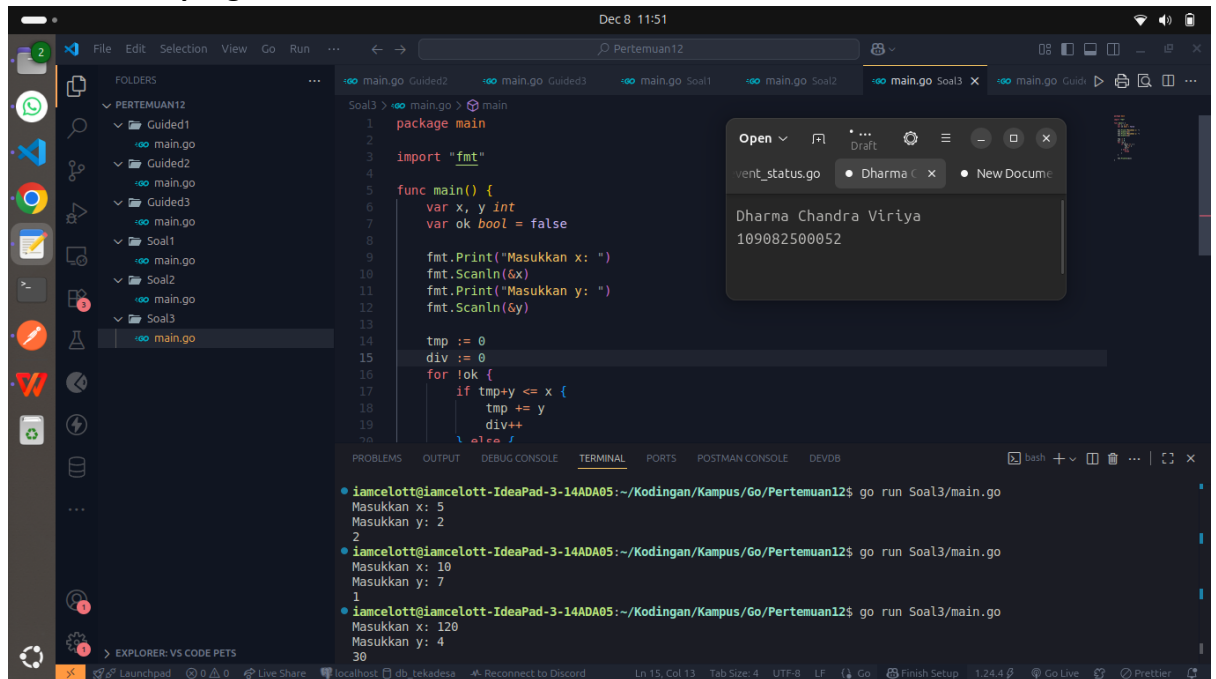
func main() {
    var x, y int
    var ok bool = false

    fmt.Print("Masukkan x: ")
    fmt.Scanln(&x)
    fmt.Print("Masukkan y: ")
    fmt.Scanln(&y)

    tmp := 0
    div := 0
    for !ok {
        if tmp+y <= x {
            tmp += y
            div++
        } else {
            break
        }
    }

    fmt.Println(div)
}
```

## Screenshoot program



The screenshot shows a VS Code editor window with a Go program in the main editor and its execution output in the terminal. The program is a Go file named `main.go` located in the `Soal3` directory. The code defines a `main` function that takes two integers `x` and `y` as input and returns a boolean `ok`. The function uses a `for` loop to calculate the sum of `x` and `y` until the sum is greater than or equal to `x`. The output of the program is shown in the terminal, displaying the input values and the resulting sum.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x, y int
7     var ok bool = false
8
9     fmt.Print("Masukkan x: ")
10    fmt.Scanln(&x)
11    fmt.Print("Masukkan y: ")
12    fmt.Scanln(&y)
13
14    tmp := 0
15    div := 0
16    for lok {
17        if tmp+y <= x {
18            tmp += y
19            div++
20        } else {
21            break
22        }
23    }
24    fmt.Println(div)
```

Terminal output:

```
iancelott@iancelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Soal3/main.go
Masukkan x: 5
Masukkan y: 2
2
iancelott@iancelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Soal3/main.go
Masukkan x: 10
Masukkan y: 7
1
iancelott@iancelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan12$ go run Soal3/main.go
Masukkan x: 120
Masukkan y: 4
30
```

## Deskripsi program

Kode program di atas berfungsi menghitung berapa kali nilai `y` dapat dijumlahkan ke dirinya sendiri tanpa melebihi nilai `x`. Program menggunakan package `main` dan library `fmt` untuk menangani input dan output. Di dalam function `main`, dua variabel `x` dan `y` disiapkan untuk menampung input pengguna, serta variabel `ok` sebagai penanda perulangan. Setelah menerima input, program menginisialisasi dua variabel: `tmp` untuk menampung hasil penjumlahan sementara dan `div` untuk menghitung berapa kali `y` berhasil dijumlahkan. Program kemudian menjalankan perulangan `for` selama `ok` bernilai `false`. Pada setiap iterasi, program memeriksa apakah `tmp + y` masih kurang dari atau sama dengan `x`. Jika iya, `y` ditambahkan ke `tmp` dan `div` ditambah satu. Jika tidak, perulangan dihentikan dengan `break`. Setelah loop selesai, program mencetak nilai `div` sebagai jumlah maksimum `y` yang dapat dijumlahkan tanpa melampaui `x`.