

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 12**  
**“TIPE DATA & VARIABEL”**



**DISUSUN OLEH:**  
**MUHAMMAD ZAKY MUBAROK**  
**103112400073 S1**  
**IF-12-01**  
**DOSEN:**  
**Yohani Setiya Rafika Nur, M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**

## DASAR TEORI

### Dasar Teori While-Loop

#### Paradigma Perulangan

Perulangan adalah struktur kontrol yang memungkinkan suatu instruksi dilakukan berulang kali. Ini sangat berguna untuk menghindari penulisan instruksi yang sama berkali-kali. Pada kasus tertentu, kita mungkin tidak tahu berapa kali perulangan harus dilakukan di awal, ini disebut perulangan dengan kondisi.

#### Karakteristik While-Loop

While-loop adalah struktur kontrol yang menjalankan aksi berulang kali selama kondisi bernilai benar (true).

#### Struktur Umum While-Loop

Struktur umum while-loop adalah:

1. **Kondisi:**
  - Kondisi ini harus menghasilkan nilai boolean (true atau false).
  - Perulangan akan terjadi selama kondisi bernilai true.
2. **Aksi:**
  - Ini adalah instruksi yang dieksekusi berulang kali selama kondisi bernilai true.
  - Salah satu instruksi harus bisa mengubah kondisi dari true menjadi false untuk menghentikan perulangan.

#### Penulisan While-Loop

Dalam banyak bahasa pemrograman, kata kunci yang digunakan adalah while. Namun, dalam bahasa Go, kata kunci yang digunakan adalah for.

Contoh Sederhana While-Loop (Dalam Go Menggunakan for)

Misalnya, kita ingin mencetak angka dari 1 hingga 5:

```
go
package main

import (
    "fmt"
)

func main() {
    angka := 1
    for angka <= 5 { // Kondisi
        fmt.Println(angka) // Aksi
        angka++ // Mengubah kondisi
    }
}
```

#### Contoh Dalam Kehidupan Nyata

- Menulis teks selama tinta pena masih ada:
  - Kondisi: Tinta pena masih ada.
  - Aksi: Menulis teks.

- Makan suap demi suap selama masih lapar:
  - Kondisi: Masih lapar.
  - Aksi: Makan suap.

**Pentingnya Kondisi yang Menghentikan**

Penting untuk memastikan bahwa ada mekanisme dalam aksi yang mengubah kondisi menjadi false, agar perulangan bisa berhenti. Jika tidak, program akan berjalan tanpa henti (infinite loop).

**While-Loop di Go Menggunakan for**

Dalam bahasa Go, while-loop ditulis menggunakan for:

```
go
angka := 1
for angka <= 5 { // Kondisi
    fmt.Println(angka) // Aksi
    angka++ // Mengubah kondisi
}
```

Meskipun menggunakan kata kunci for, struktur dan logikanya sama seperti while-loop di bahasa pemrograman lain.

## CONTOH SOAL

### 1. Latihan1

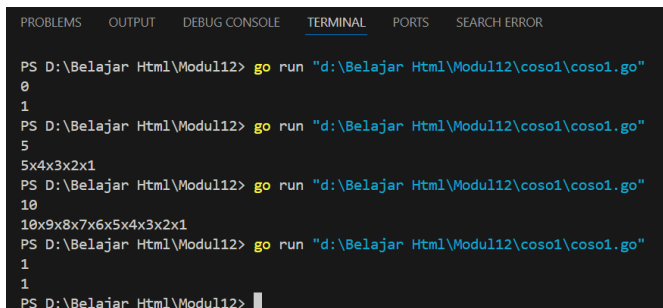
Source Code:

```
package main

import "fmt"

func main() {
    var n, j int
    fmt.Scan(&n)
    j = n
    for j > 1 {
        fmt.Print(j, "x")
        j = j - 1
    }
    fmt.Println(1)
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\coso1\coso1.go"
0
1
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\coso1\coso1.go"
5
5x4x3x2x1
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\coso1\coso1.go"
10
10x9x8x7x6x5x4x3x2x1
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\coso1\coso1.go"
1
1
PS D:\Belajar Html\Modul12> 
```

### Deskripsi Program:

#### 1. Deklarasi Variabel:

- o **n dan j** adalah variabel yang akan digunakan untuk menyimpan nilai input dan nilai sementara untuk perulangan.

#### 2. Membaca Input:

- o Program membaca bilangan bulat **n** yang diinputkan menggunakan **fmt.Scan(&n)**.

### **3. Inisialisasi Variabel:**

- **j diinisialisasi dengan nilai n.**

### **4. Perulangan for:**

- **Perulangan dijalankan selama nilai j lebih besar dari 1.**
- **Di dalam perulangan, program mencetak nilai j diikuti oleh tanda "x".**
- **Nilai j dikurangi satu setiap kali iterasi ( $j = j - 1$ ).**

### **5. Mencetak Angka Terakhir:**

- **Setelah perulangan selesai, program mencetak angka 1 dan menambahkan baris baru (`fmt.Println(1)`).**

### **Contoh Penggunaan:**

**Misalkan pengguna memasukkan nilai 5. Maka program akan menghasilkan output berikut:**

**5x4x3x2x1**

## CONTOH SOAL

### 2. Latihan2

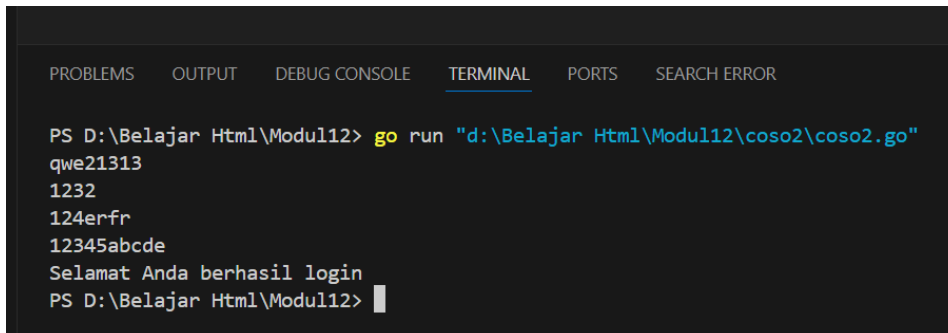
Source Code:

```
package main

import "fmt"

func main() {
    var token string
    fmt.Scan(&token)
    for token != "12345abcde" {
        fmt.Scan(&token)
    }
    fmt.Println("Selamat Anda berhasil login")
}
```

Output:



```
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\coso2\coso2.go"
qwe21313
1232
124erfr
12345abcde
Selamat Anda berhasil login
PS D:\Belajar Html\Modul12>
```

### Deskripsi Program:

#### 1. Deklarasi Variabel:

- token adalah variabel yang akan digunakan untuk menyimpan nilai input token.

#### 2. Membaca Input Awal:

- Program membaca token yang diinputkan oleh pengguna menggunakan `fmt.Scan(&token)`.

### **3. Perulangan for:**

- **Perulangan dijalankan selama token yang diinputkan tidak sama dengan "12345abcde".**
- **Jika token tidak sesuai, program kembali meminta input token (fmt.Scan(&token)).**

### **4. Mencetak Pesan Sukses:**

- **Setelah token yang benar dimasukkan (yakni "12345abcde"), program keluar dari perulangan dan mencetak pesan "Selamat Anda berhasil login" (fmt.Println("Selamat Anda berhasil login")).**

### **Contoh Penggunaan:**

- **Pengguna memasukkan token: abcdef**
- **(Program meminta input lagi)**
- **Pengguna memasukkan token: 12345abcde**

**Selamat Anda berhasil login**

## CONTOH SOAL

### 3. Latihan3

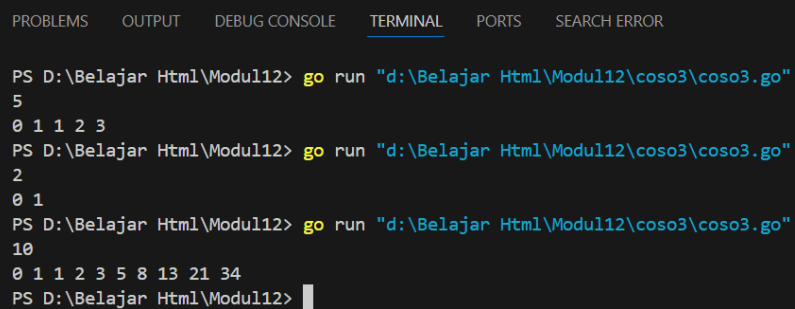
Source Code:

```
package main

import "fmt"

func main() {
    var n, s1, s2, j, temp int
    fmt.Scan(&n)
    s1 = 0
    s2 = 1
    j = 0
    for j < n {
        fmt.Print(s1, " ")
        temp = s1 + s2
        s1 = s2
        s2 = temp
        j = j + 1
    }
}
```

Output:



The screenshot shows a terminal window with the following content:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\coso3\coso3.go"
5
0 1 1 2 3
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\coso3\coso3.go"
2
0 1
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\coso3\coso3.go"
10
0 1 1 2 3 5 8 13 21 34
PS D:\Belajar Html\Modul12> 
```



## Deskripsi Program:

### 1. Deklarasi Variabel:

- **n:** Variabel untuk menyimpan jumlah elemen deret Fibonacci yang akan dicetak.
- **s1:** Variabel untuk menyimpan elemen pertama deret Fibonacci, diinisialisasi dengan 0.
- **s2:** Variabel untuk menyimpan elemen kedua deret Fibonacci, diinisialisasi dengan 1.
- **j:** Variabel penghitung iterasi, diinisialisasi dengan 0.
- **temp:** Variabel sementara untuk menyimpan hasil penjumlahan elemen Fibonacci.

### 2. Membaca Input:

- Program membaca nilai **n** yang diinputkan oleh pengguna menggunakan `fmt.Scan(&n)`.

### 3. Inisialisasi Variabel:

- **s1** diinisialisasi dengan nilai 0.
- **s2** diinisialisasi dengan nilai 1.
- **j** diinisialisasi dengan nilai 0.

### 4. Perulangan for:

- Perulangan dijalankan sebanyak **n** kali.
- Di dalam perulangan:
  - Program mencetak nilai **s1** diikuti oleh spasi.
  - **temp** diisi dengan nilai penjumlahan **s1** dan **s2**.
  - **s1** diisi dengan nilai **s2**.
  - **s2** diisi dengan nilai **temp**.
  - **j** ditambah 1 pada setiap iterasi.

## Contoh Penggunaan:

Misalkan pengguna menginputkan nilai 5. Maka program akan menghasilkan output berikut:

0 1 1 2 3

## Penjelasan:

- Iterasi 1: **s1 = 0, s2 = 1** -> Output: 0
- Iterasi 2: **s1 = 1, s2 = 1** -> Output: 1

- **Iterasi 3:  $s1 = 1$ ,  $s2 = 2$  -> Output: 1**
- **Iterasi 4:  $s1 = 2$ ,  $s2 = 3$  -> Output: 2**
- **Iterasi 5:  $s1 = 3$ ,  $s2 = 5$  -> Output: 3**

## SOAL LATIHAN

### Statement perulangan

1.


#### Source Code:

```
package main

import "fmt"

func main() {
    var username, password string
    fmt.Scan(&username, &password)
    percobaan := 0
    for username != "Admin" && password != "Admin" {
        fmt.Scan(&username, &password)
        percobaan++
    }
    fmt.Println(percobaan, "Percobaan gagal login")
}
```

#### Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\latsol1\latsol1.go"
1w12e 12r23
2r ffg g
2e1e f4f
Admin Admin
3 Percobaan gagal login
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\latsol1\latsol1.go"
Admin Admin
0 Percobaan gagal login
PS D:\Belajar Html\Modul12> 
```

## **Deskripsikan Program :**

### **1. Deklarasi Variabel:**

- username dan password adalah variabel yang akan digunakan untuk menyimpan nilai input dari pengguna.
- percobaan adalah variabel yang digunakan untuk menghitung jumlah percobaan login yang gagal.

### **2. Membaca Input Awal:**

- Program membaca username dan password yang diinputkan oleh pengguna menggunakan `fmt.Scan(&username, &password)`.

### **3. Perulangan for:**

- Perulangan dijalankan selama username yang diinputkan tidak sama dengan "Admin" dan password yang diinputkan juga tidak sama dengan "Admin".
- Setiap kali username dan password yang diinputkan salah, program kembali meminta input baru dari pengguna (`fmt.Scan(&username, &password)`).
- Variabel percobaan ditambah satu setiap kali percobaan login gagal.

### **4. Mencetak Pesan Jumlah Percobaan Gagal:**

- Setelah username dan password yang benar dimasukkan, program keluar dari perulangan dan mencetak jumlah percobaan gagal login dengan pesan "Percobaan gagal login".

### **Contoh Penggunaan:**

- Pengguna memasukkan username: user, password: 1234
- (Program meminta input lagi)
- Pengguna memasukkan username: Admin, password: pass
- (Program meminta input lagi)
- Pengguna memasukkan username: Admin, password: Admin

**2 Percobaan gagal login**

## SOAL LATIHAN

### 2. Latihan2 Source

Code:

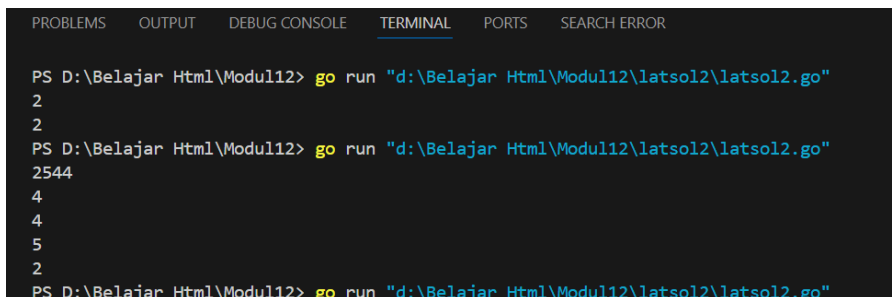
```
package main

import (
    "fmt"
)

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

    for bilangan > 0 {
        digit := bilangan % 10
        fmt.Println(digit)
        bilangan /= 10
    }
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\latsol2\latsol2.go"
2
2
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\latsol2\latsol2.go"
2544
4
4
4
5
2
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\latsol2\latsol2.go"
```

### Deskripsi Program:

#### 1. Deklarasi Variabel:

- **bilangan:** Variabel untuk menyimpan bilangan bulat yang diinputkan oleh pengguna.

#### 2. Membaca Input:

- Program membaca nilai bilangan yang diinputkan oleh pengguna menggunakan `fmt.Scan(&bilangan)`.

### 3. Perulangan for:

- Perulangan dijalankan selama bilangan lebih besar dari 0.
- Di dalam perulangan:
  - `digit := bilangan % 10`: Mengambil digit paling kanan dari bilangan dengan menggunakan operasi modulus.
  - `fmt.Println(digit)`: Mencetak digit yang telah diambil.
  - `bilangan /= 10`: Menghapus digit paling kanan dari bilangan dengan membagi bilangan dengan 10 dan membuang bagian desimalnya.

#### Contoh Penggunaan:

Misalkan pengguna menginputkan nilai 12345. Maka program akan menghasilkan output berikut:

5

4

3

2

1

#### Penjelasan:

- Iterasi 1: `bilangan = 12345`, `digit = 5` -> Output: 5
- Iterasi 2: `bilangan = 1234`, `digit = 4` -> Output: 4
- Iterasi 3: `bilangan = 123`, `digit = 3` -> Output: 3
- Iterasi 4: `bilangan = 12`, `digit = 2` -> Output: 2
- Iterasi 5: `bilangan = 1`, `digit = 1` -> Output: 1

## SOAL LATIHAN

### Statement perulangan

#### 3. Latihan3

#### Source Code:

```
package main

import (
    "fmt"
)

func main() {
    var x, y int


    fmt.Scan(&x, &y)

    hasilBagi := 0

    for x >= y {
        x -= y
        hasilBagi++
    }

    fmt.Println(hasilBagi)
}
```

#### Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\latsol3\latsol3.go"
5 2
2
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\latsol3\latsol3.go"
10 7
1
PS D:\Belajar Html\Modul12> go run "d:\Belajar Html\Modul12\latsol3\latsol3.go"
120
4
30
PS D:\Belajar Html\Modul12> 
```

## **Deskripsi Program:**

### **1. Deklarasi Variabel:**

- **x dan y:** Variabel untuk menyimpan dua bilangan bulat yang diinputkan oleh pengguna.
- **hasilBagi:** Variabel untuk menyimpan hasil bagi dari pembagian x oleh y.

### **2. Membaca Input:**

- Program membaca nilai x dan y yang diinputkan oleh pengguna menggunakan `fmt.Scan(&x, &y)`.

### **3. Inisialisasi Variabel:**

- **hasilBagi** diinisialisasi dengan nilai 0.

### **4. Perulangan for:**

- Perulangan dijalankan selama x lebih besar atau sama dengan y.
- Di dalam perulangan:
  - **x -= y:** Mengurangi nilai x dengan y.
  - **hasilBagi++:** Menambah nilai hasilBagi sebesar 1 pada setiap iterasi.

### **5. Mencetak Hasil Bagi:**

- Setelah perulangan selesai, program mencetak nilai hasilBagi yang merupakan hasil pembagian bilangan bulat dari x oleh y.

## **Contoh Penggunaan:**

Misalkan pengguna menginputkan nilai 10 untuk x dan 3 untuk y. Maka program akan menghasilkan output berikut:

**3**

## **Penjelasan:**

- Iterasi 1: x = 10, y = 3, x menjadi 7, hasilBagi = 1
- Iterasi 2: x = 7, y = 3, x menjadi 4, hasilBagi = 2
- Iterasi 3: x = 4, y = 3, x menjadi 1, hasilBagi = 3
- Perulangan berhenti karena x tidak lagi lebih besar atau sama dengan y.



## ***DAFTAR PUSTAKA***

## ***MODUL PRAKTIKUM 12 - WHILE-LOOP***