

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 1
“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 Paradigma Percabangan

Pengertian

Paradigma percabangan memungkinkan program untuk membuat keputusan berdasarkan kondisi tertentu, mirip dengan memilih jalan di persimpangan.

Struktur if-else if

1. **Kondisi:** Ekspresi yang mengevaluasi menjadi true atau false.
2. **Aksi:** Instruksi yang dijalankan jika kondisi bernilai true.
3. **Aksi Lain:** Instruksi yang dijalankan jika kondisi bernilai false. **Contoh**

Implementasi go package main

```
import "fmt"
```

```
func main() {    var umur int
```

```
    fmt.Print("Masukkan umur Anda: ")
```

```
    fmt.Scan(&umur)
```

```
    if umur < 13 {        fmt.Println("Anda  
adalah anak-anak")
```

```
    } else if umur < 20 {
```

```
        fmt.Println("Anda adalah remaja")
```

```
    } else if umur < 60 {
```

```
        fmt.Println("Anda adalah dewasa")
```

```
    } else {        fmt.Println("Anda  
adalah lansia")
```

```
    }
```

```
}
```

Program ini menentukan kategori usia berdasarkan input pengguna dengan menggunakan struktur if-else if.

Manfaat

- **Pengambilan Keputusan:** Memungkinkan respons program yang sesuai dengan kondisi.
- **Fleksibilitas:** Mengontrol alur program secara dinamis.

CONTOH SOAL

1. Latihan1

Source Code:

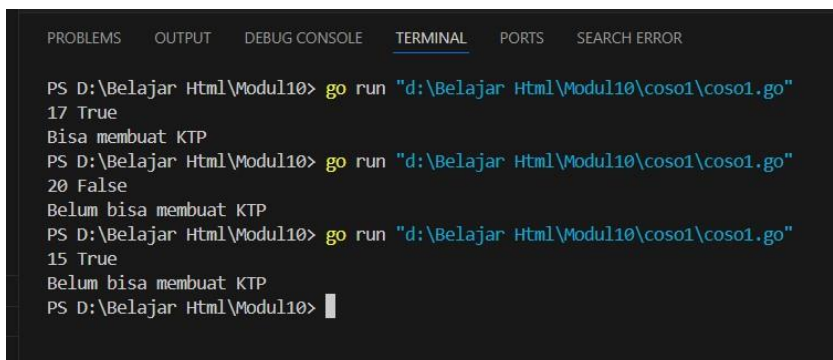
```
package main

import "fmt"

func main() {    var usia
int    var kk bool
fmt.Scan(&usia, &kk)
if usia >= 17 && kk {
    fmt.Print("Bisa membuat KTP")

} else {    fmt.Print("Belum bisa
membuat KTP")
}
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\coso1\coso1.go"
17 True
Bisa membuat KTP
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\coso1\coso1.go"
20 False
Belum bisa membuat KTP
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\coso1\coso1.go"
15 True
Belum bisa membuat KTP
PS D:\Belajar Html\Modul10> █
```

Deskripsi Program:

1. Deklarasi Variabel:

- **var usia int:** Mendeklarasikan variabel usia bertipe integer untuk menyimpan nilai usia yang akan diinput.
- **var kk bool:** Mendeklarasikan variabel kk bertipe boolean untuk menyimpan nilai status Kartu Keluarga (KK) yang diinput.

2. Input:

- **fmt.Scan(&usia, &kk):** Membaca dua input dan menyimpannya dalam variabel usia dan kk. variabel kk sebenarnya tidak digunakan dalam logika program selanjutnya.

3. Logika if-else:

- **if usia >= 17:** Memeriksa apakah usia yang diinput lebih besar atau sama dengan 17.
 - **Jika kondisi ini benar (usia >= 17),** maka program akan mengeksekusi perintah **fmt.Println("Bisa membuat KTP")**, yang mencetak teks **"Bisa membuat KTP"**.
 - **Jika kondisi ini salah (usia < 17),** maka program akan mengeksekusi perintah **fmt.Println("Belum bisa membuat KTP")**, yang mencetak teks **"Belum bisa membuat KTP"**.

Fungsi Program:

Program ini digunakan untuk menentukan apakah seseorang yang diinputkan usianya memenuhi syarat untuk membuat KTP (Kartu Tanda Penduduk) berdasarkan usia mereka. Program ini akan mencetak "Bisa membuat KTP" jika usia >= 17, dan "Belum bisa membuat KTP" jika usia < 17.

Contoh:

- **Jika pengguna memasukkan usia 18 dan membawa kk, program akan mencetak "Bisa membuat KTP".**
- **Jika pengguna memasukkan usia 18 dan tidak membawa kk, program akan mencetak "Belum bisa membuat KTP".**

CONTOH SOAL

2. Latihan2

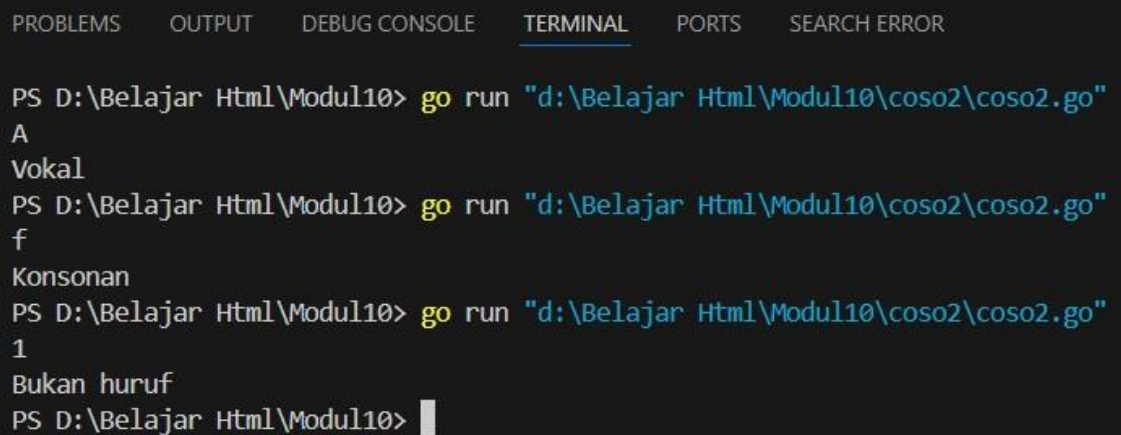
Source Code:

```
package main

import "fmt"

func main() {
    var x rune
    var huruf, vKecil, vBesar bool
    fmt.Scanf("%c", &x)
    huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
    vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o'
    vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'
    if huruf && (vKecil || vBesar) {      fmt.Println("Vokal")
    } else if huruf && !(vKecil || vBesar) {
    fmt.Println("Konsonan")
    } else {
    fmt.Println("Bukan huruf")
    }
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\coso2\coso2.go"
A
Vokal
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\coso2\coso2.go"
f
Konsonan
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\coso2\coso2.go"
1
Bukan huruf
PS D:\Belajar Html\Modul10> █
```

Deskripsi Program:

1. Deklarasi Variabel:

- **var x rune:** Mendeklarasikan variabel x bertipe rune untuk menyimpan karakter yang diinput.
- **var huruf, vKecil, vBesar bool:** Mendeklarasikan variabel boolean huruf, vKecil, dan vBesar untuk menyimpan hasil pemeriksaan karakter.

2. Input :

- **fmt.Scanf("%c", &x):** Membaca satu karakter dari input dan menyimpannya dalam variabel x.

3. Logika Pemeriksaan Karakter:

- **huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z'):** Memeriksa apakah x adalah huruf alfabet (baik huruf kecil maupun huruf besar). Jika x berada dalam rentang 'a' sampai 'z' atau 'A' sampai 'Z', variabel huruf akan bernilai true.
- **vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o':** Memeriksa apakah x adalah salah satu vokal huruf kecil (a, i, u, e, o). Jika ya, variabel vKecil akan bernilai true.
- **vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O':** Memeriksa apakah x adalah salah satu vokal huruf besar (A, I, U, E, O). Jika ya, variabel vBesar akan bernilai true.

4. Logika if-else:

- **if huruf && (vKecil || vBesar):** Memeriksa apakah x adalah huruf dan apakah x adalah vokal (baik huruf kecil maupun besar).
 - **Jika kondisi ini benar,** program akan mengeksekusi perintah **fmt.Println("Vokal"),** yang mencetak teks "Vokal".
- **else if huruf && !(vKecil || vBesar):** Memeriksa apakah x adalah huruf dan bukan vokal (berarti konsonan).
 - **Jika kondisi ini benar,** program akan mengeksekusi perintah **fmt.Println("Konsonan"),** yang mencetak teks "Konsonan".
- **else:** Jika x bukan huruf, program akan mengeksekusi perintah **fmt.Println("Bukan huruf"),** yang mencetak teks "Bukan huruf".

Fungsi Program:

Program ini digunakan untuk menentukan karakter input apakah itu vokal, konsonan, atau bukan huruf. Program memeriksa apakah karakter x yang diinput adalah huruf

dan kemudian menentukan apakah itu vokal atau konsonan. Jika x bukan huruf, program akan mencetak "Bukan huruf".

Contoh:

- Jika memasukkan a, program akan mencetak "Vokal".
- Jika memasukkan b, program akan mencetak "Konsonan".
- Jika memasukkan 1, program akan mencetak "Bukan huruf".

CONTOH SOAL

3. Latihan3 Source

Code:

```
package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3, d4
    int    var teks string
    fmt.Print("Bilangan:")
    fmt.Scan(&bilangan)    d4 =
    bilangan % 10    d3 = (bilangan
    % 100) / 10    d2 = (bilangan %
    1000) / 100    d1 = bilangan /
    1000

    if d1 < d2 && d2 < d3 && d3 < d4 {
    teks = "Terurut Membesar"

    }else if d1 > d2 && d2 > d3 && d3 > d4 {
    teks = "Terurut Mengecil"

    }else {    teks =
    "Tidak Terurut"
    }
    fmt.Println("Digit pada bilangan", bilangan, teks) }
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\coso3\coso3.go"
Bilangan:2489
Digit pada bilangan 2489 Terurut Membesar
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\coso3\coso3.go"
Bilangan:3861
Digit pada bilangan 3861 Tidak Terurut
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\coso3\coso3.go"
Bilangan:9651
Digit pada bilangan 9651 Terurut Mengecil
PS D:\Belajar Html\Modul10> █
```

Deskripsi Program:

1. Deklarasi Variabel:

- **var bilangan, d1, d2, d3, d4 int:** Mendeklarasikan variabel bilangan, d1, d2, d3, dan d4 bertipe integer.
- **var teks string:** Mendeklarasikan variabel teks bertipe string untuk menyimpan hasil yang akan dicetak.

2. Input :

- **fmt.Print("Bilangan:");** Menampilkan pesan untuk meminta input.
- **fmt.Scan(&bilangan);** Membaca input dan menyimpannya dalam variabel bilangan.

3. Ekstraksi Digit:

- **d4 = bilangan % 10:** Mengambil digit terakhir dari bilangan.
- **d3 = (bilangan % 100) / 10:** Mengambil digit kedua dari belakang dengan membagi hasil sisa bagi bilangan % 100 dengan 10.
- **d2 = (bilangan % 1000) / 100:** Mengambil digit ketiga dari belakang dengan membagi hasil sisa bagi bilangan % 1000 dengan 100.
- **d1 = bilangan / 1000:** Mengambil digit keempat dari belakang (ribuan) dengan membagi bilangan dengan 1000.

4. Logika if-else untuk Pemeriksaan Urutan:

- **if d1 < d2 && d2 < d3 && d3 < d4:** Memeriksa apakah digit-digit d1, d2, d3, dan d4 terurut membesar.

- Jika kondisi ini terpenuhi, teks diubah menjadi "Terurut Membesar".
- else if `d1 > d2 && d2 > d3 && d3 > d4`: Memeriksa apakah digit-digit d1, d2, d3, dan d4 terurut mengecil.
 - Jika kondisi ini terpenuhi, teks diubah menjadi "Terurut Mengecil".
- else: Jika kondisi lain tidak terpenuhi, teks diubah menjadi "Tidak Terurut".

5. Output:

- `fmt.Println("Digit pada bilangan", bilangan, teks)`: Mencetak teks yang berisi digit dari bilangan dan hasil pemeriksaan urutan digit.

Fungsi Program:

Program ini digunakan untuk memeriksa urutan digit-digit dari sebuah bilangan yang diinput oleh pengguna. Program menentukan apakah digit-digit tersebut terurut membesar, terurut mengecil, atau tidak terurut, dan mencetak hasil pemeriksaannya.

Contoh:

- Jika memasukkan 1234, program akan mencetak "Digit pada bilangan 1234 Terurut Membesar".
- Jika memasukkan 4321, program akan mencetak "Digit pada bilangan 4321 Terurut Mengecil".
- Jika memasukkan 1324, program akan mencetak "Digit pada bilangan 1324 Tidak Terurut".

SOAL LATIHAN

Statement perulangan

1.

Source Code:

```
package main

import "fmt"

func main() {

    var berat, beratKg, sisaGram ,biayaKg, biayaSisaGram, totalBiaya int

    fmt.Print("Masukkan berat parsel (gram): ")
    fmt.Scan(&berat)

    beratKg = berat / 1000
    sisaGram = berat % 1000

    biayaKg = beratKg * 10000

    if sisaGram > 0 {      if
    beratKg > 10 {
    biayaSisaGram = 0      } else
    if sisaGram >= 500 {
        biayaSisaGram = sisaGram * 5
    } else {
        biayaSisaGram = sisaGram * 15
    }
    }

    totalBiaya = biayaKg + biayaSisaGram

    fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, sisaGram)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg, biayaSisaGram)
    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\latsol1\latsol1.go"
Masukkan berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\latsol1\latsol1.go"
Masukkan berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\Belajar Html\Modul10> 
```

Deskripsikan Program :

1. Deklarasi Variabel:

- **var berat, beratKg, sisaGram, biayaKg, biayaSisaGram, totalBiaya int:** Variabel-variabel untuk menyimpan input berat parcel dan hasil perhitungan biaya pengiriman.

2. Input :

- **fmt.Print("Masukkan berat parcel (gram): ")**: Menampilkan pesan untuk meminta input dari pengguna.
- **fmt.Scan(&berat)**: Membaca input berat parcel dalam gram dan menyimpannya dalam variabel berat.

3. Menghitung Berat dalam kg dan Gram:

- **beratKg = berat / 1000**: Menghitung total berat dalam kilogram.
- **sisaGram = berat % 1000**: Menghitung sisa berat dalam gram.

4. Menghitung Biaya Pengiriman per Kilogram:

- **biayaKg = beratKg * 10000**: Menghitung biaya pengiriman berdasarkan berat dalam kilogram.

5. Menghitung Biaya Pengiriman Berdasarkan Sisa Gram:

- **if sisaGram > 0**: Memeriksa apakah ada sisa berat dalam gram.
 - **if beratKg > 10**: Jika total berat lebih dari 10 kg, biaya sisa gram adalah 0.
 - **else if sisaGram >= 500**: Jika sisa gram >= 500, biaya tambahan adalah Rp. 5 per gram.
 - **else**: Jika sisa gram < 500, biaya tambahan adalah Rp. 15 per gram.

6. Menghitung Total Biaya Pengiriman:

- `totalBiaya = biayaKg + biayaSisaGram`: Menghitung total biaya pengiriman dengan menjumlahkan biaya per kilogram dan biaya sisa gram.

Output:

- `fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, sisaGram)`: Mencetak detail berat parcel.
- `fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg, biayaSisaGram)`: Mencetak detail biaya pengiriman.
- `fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)`: Mencetak total biaya pengiriman.

Fungsi Program:

Program ini digunakan untuk menghitung biaya pengiriman parcel berdasarkan berat yang diinput. Biaya pengiriman dihitung dengan tarif Rp. 10.000 per kg. Jika sisa berat tidak kurang dari 500 gram, tambahan biaya adalah Rp. 5 per gram, namun jika kurang dari 500 gram, tambahan biaya adalah Rp. 15,- per gram. Sisa berat tidak dikenakan biaya jika total berat lebih dari 10 kg

SOAL LATIHAN

2. Latihan2 Source

Code:

```
package main

import "fmt"

func main() {
var nam float64
var nmk string

    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scan(&nam)

    if nam > 80 {
nmk = "A"
    } else if nam > 72.5
{      nmk = "AB"    }
    else if nam > 65 {
nmk = "B"
    } else if nam > 57.5
{      nmk = "BC"    }
    else if nam > 50 {
nmk = "C"    } else if
nam > 40 {      nmk =
"D"    } else {
nmk = "E"
    }

    fmt.Println("Nilai mata kuliah: ", nmk)
}
```

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
latsol2\latsol2.go:12:7: cannot use "AB" (untyped string constant) as float64 value in assignment
latsol2\latsol2.go:15:7: cannot use "B" (untyped string constant) as float64 value in assignment
latsol2\latsol2.go:18:7: cannot use "BC" (untyped string constant) as float64 value in assignment
latsol2\latsol2.go:21:7: cannot use "C" (untyped string constant) as float64 value in assignment
latsol2\latsol2.go:24:7: cannot use "D" (untyped string constant) as float64 value in assignment
latsol2\latsol2.go:26:7: cannot use "E" (untyped string constant) as float64 value in assignment
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 80.1
Nilai mata kuliah: A
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS D:\Belajar Html\Modul10> |
```

Jawaban :

A. Setelah nam diisi dengan 80.1:

- **Kondisi if nam > 80 akan terpenuhi, sehingga program akan mencoba mengubah nam menjadi "A", tetapi ini akan menyebabkan kesalahan karena nam adalah variabel bertipe float64, program akan berakhir dengan nilai "D" untuk nam, tetapi akan ada kesalahan karena perubahan tipe data dari float64 ke string.**

B.

Kesalahan dalam Program

Penetapan Variabel:

mencoba menetapkan nilai string ke variabel float64, yang tidak diperbolehkan dalam Go. Variabel nam (nilai akhir mata kuliah) seharusnya tetap bertipe float64, dan nilai huruf harus disimpan dalam nmk yang bertipe string.

Struktur Kontrol if-else:

Semua kondisi if terpisah sehingga setiap kondisi dievaluasi secara independen. Hal ini menyebabkan bahwa jika nam lebih dari 80, program tetap mengubahnya menjadi "A", dan kemudian bisa mengubah menjadi nilai lain berdasarkan kondisi berikutnya. Penggunaan else if diperlukan untuk memastikan hanya satu blok kode yang dijalankan.

Penetapan Nilai nmk:

Output akhir mencetak variabel nmk, tetapi variabel tersebut tidak pernah diisi dengan nilai apa pun.

Alur Program yang Seharusnya

Program seharusnya berjalan dengan memeriksa nilai nam menggunakan struktur if-else if untuk menghindari evaluasi kondisi yang tidak diperlukan setelah kondisi benar ditemukan. Kemudian menetapkan nilai yang sesuai dengan kondisi else if

C. Sudah ada di output

SOAL LATIHAN

Statement perulangan

3. Latihan3

Source Code:

```
package main

import "fmt"

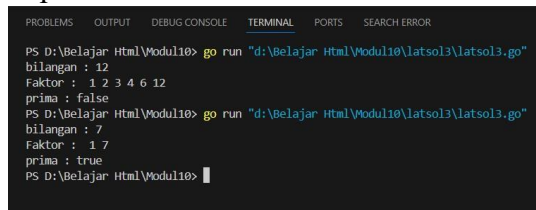
func main() {    var b int
    fmt.Print("bilangan : ")
    fmt.Scan(&b)

    fmt.Print("Faktor : ")
    jumlahFaktor := 0    for
    i := 1; i <= b; i++ {
    if b % i == 0 {
    fmt.Print(" " ,i)
        jumlahFaktor++
    }
    }

    fmt.Println()

    if jumlahFaktor == 2 {
    fmt.Println( "prima: true")
    } else {
        fmt.Println( "prima: false")
    }
    }
```

Output:

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and SEARCH ERROR. The output shows two runs of the program. The first run takes input 12 and outputs factors 1 2 3 4 6 12, and is not prime. The second run takes input 7 and outputs factors 1 7, and is prime.

```
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\latsol3\latsol3.go"
bilangan : 12
Faktor : 1 2 3 4 6 12
prima : false
PS D:\Belajar Html\Modul10> go run "d:\Belajar Html\Modul10\latsol3\latsol3.go"
bilangan : 7
Faktor : 1 7
prima : true
PS D:\Belajar Html\Modul10>
```

Deskripsi Program:

1. Deklarasi Variabel:

- **var b int:** Mendeklarasikan variabel b untuk menyimpan bilangan yang diinput oleh pengguna.

2. Input :

- **fmt.Print("bilangan : "):** Menampilkan pesan untuk meminta input dari pengguna.
- **fmt.Scan(&b):** Membaca input bilangan dari pengguna dan menyimpannya dalam variabel b.

3. Menentukan Faktor dari Bilangan:

- **fmt.Print("Faktor : "):** Menampilkan pesan untuk menunjukkan bahwa faktor-faktor dari bilangan akan ditampilkan.
- **jumlahFaktor := 0:** Mendeklarasikan dan menginisialisasi variabel jumlahFaktor untuk menghitung jumlah faktor dari bilangan b.

4. Menghitung dan Menampilkan Faktor:

- **for i := 1; i <= b; i++:** Melakukan iterasi dari 1 hingga bilangan b.
- **if b % i == 0:** Memeriksa apakah i adalah faktor dari b (dengan memeriksa sisa hasil bagi b dengan i adalah 0).
- **fmt.Print(" ", i):** Menampilkan faktor i.
- **jumlahFaktor++:** Menambah jumlah faktor.

5. Mengecek dan Menampilkan Apakah Bilangan adalah Bilangan Prima:

- **if jumlahFaktor == 2:** Memeriksa apakah jumlah faktor dari bilangan b adalah 2 (jika benar, berarti b adalah bilangan prima).
- **fmt.Println("prima : true"):** Menampilkan pesan bahwa bilangan tersebut adalah bilangan prima.
- **else:** Jika jumlah faktor tidak sama

dengan 2, maka bilangan tersebut bukan bilangan prima. ○
`fmt.Println("prima : false")`: Menampilkan pesan bahwa bilangan tersebut bukan bilangan prima.

Fungsi Program:

Program ini digunakan untuk menentukan faktor-faktor dari sebuah bilangan yang diinput dan juga untuk memeriksa apakah bilangan tersebut adalah bilangan prima atau bukan. Program akan mencetak semua faktor dari bilangan tersebut dan kemudian mencetak apakah bilangan tersebut adalah bilangan prima (true) atau bukan (false).

DAFTAR PUSTAKA

MODUL PRAKTIKUM 10 - ELSE-IF ALGORITMA DAN PEMROGRAMAN 1