

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 10

ELSE – IF



Disusun oleh:

Didi Hermawanto

109082500088

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

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 usia int

    var kk bool

    fmt.Scan(&usia, &kk)

    if usia >= 17 && kk {

        fmt.Println("Bisa Membuat KTP")

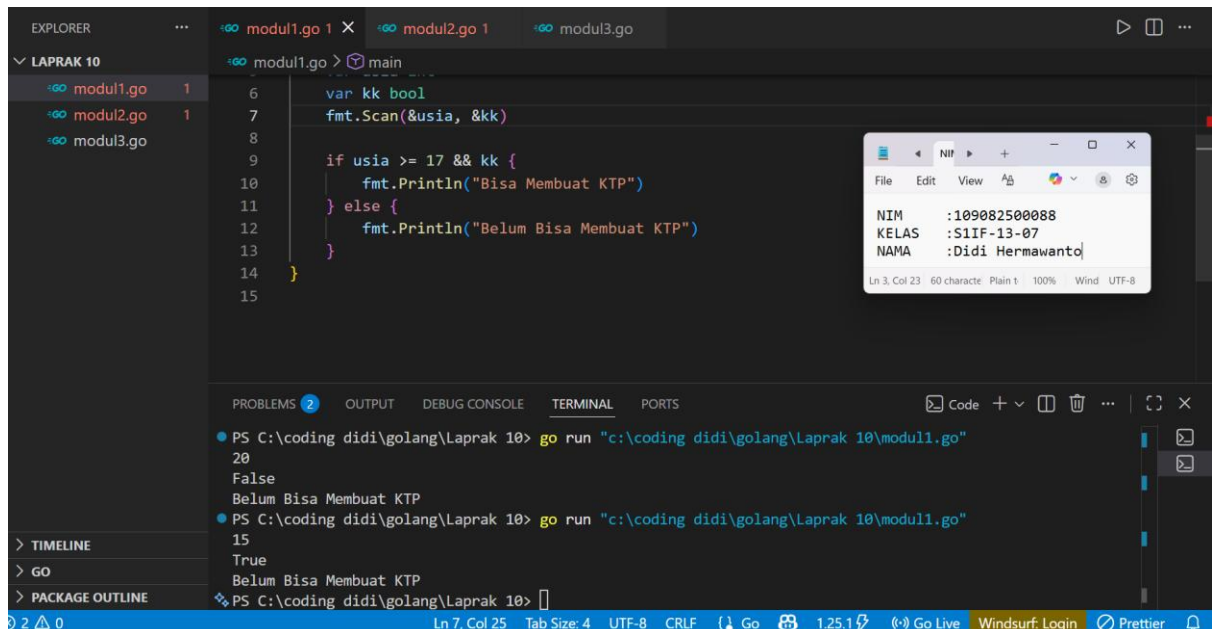
    } else {

        fmt.Println("Belum Bisa Membuat KTP")

    }

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menentukan apakah seseorang sudah memenuhi syarat untuk membuat KTP berdasarkan dua data: usia dan kepemilikan Kartu Keluarga (KK).

Program pertama-tama mendeklarasikan variabel usia dengan tipe integer dan kk bertipe boolean. Lalu program membaca input dari pengguna melalui `fmt.Scan(&usia, &kk)` sehingga data diketik dalam satu baris, misalnya: 18 true.

Bagian pengecekan dilakukan dengan kondisi `if usia >= 17 && kk`. Tanda `>= 17` berarti usia minimal 17, dan `&& kk` memastikan nilai boolean KK harus bernilai true. Jika kedua syarat ini terpenuhi, program menampilkan "Bisa Membuat KTP". Jika tidak memenuhi, blok else akan menampilkan "Belum Bisa Membuat KTP".

2. Guided 2

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")
    }
}
```

Screenshoot program

```
1 package main
2 import "fmt"
3
4 func main() {
5     var x rune
6     var huruf, vKecil, vBesar bool
7     fmt.Scanf("%c", &x)
8
9     huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
10    vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x
11    vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'
12
13    if huruf && (vKecil || vBesar) {
14        fmt.Println("vokal")
15    } else if huruf {
16        fmt.Println("konsonan")
17    } else {
18        fmt.Println("bukan huruf")
19    }
20 }
```

Terminal Output:

```
PS C:\coding didi\golang\Laparak 10> go run "c:\coding didi\golang\Laparak 10\modul2.go"
i
vokal
PS C:\coding didi\golang\Laparak 10> go run "c:\coding didi\golang\Laparak 10\modul2.go"
x
konsonan
PS C:\coding didi\golang\Laparak 10> go run "c:\coding didi\golang\Laparak 10\modul2.go"
@
bukan huruf
```

Deskripsi program

Program ini dibuat untuk mengidentifikasi satu karakter yang dimasukkan pengguna, apakah termasuk huruf vokal, huruf konsonan, atau bukan huruf sama sekali. Input dibaca menggunakan `fmt.Scanf("%c", &x)` agar hanya satu karakter yang diproses. Variabel `huruf` digunakan untuk mengecek apakah karakter berada dalam rentang huruf alfabet, baik huruf kecil ('a' sampai 'z') maupun huruf besar ('A' sampai 'Z'). Selanjutnya, karakter dibandingkan dengan daftar huruf vokal kecil dan vokal besar melalui variabel `vKecil` dan `vBesar`. Jika karakter termasuk huruf dan juga vokal, program menampilkan "vokal". Jika termasuk huruf tetapi tidak masuk daftar vokal, program menampilkan "konsonan". Namun jika karakter yang dimasukkan bukan huruf, program akan menampilkan "bukan huruf".

3. Guided 3 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)
    }

```

Screenshoot program

```

modul3.go > ...
1 package main
2 import "fmt"
3 func main() {
4     var bilangan, d1, d2, d3, d4 int
5     var teks string
6
7     fmt.Print("Bilangan: ")
8     fmt.Scan(&bilangan)
9
10    d4 = bilangan % 10
11    d3 = (bilangan % 100) / 10
12    d2 = (bilangan % 1000) / 100
13    d1 = bilangan / 1000

```

Terminal Output:

```

PS C:\coding didi\golang\Laprak 10> go run "c:\coding didi\golang\Laprak 10\modul3.go"
Bilangan: 2489
Digit pada bilangan 2489 terurut membesar
PS C:\coding didi\golang\Laprak 10> go run "c:\coding didi\golang\Laprak 10\modul3.go"
Bilangan: 3861
Digit pada bilangan 3861 tidak terurut
PS C:\coding didi\golang\Laprak 10> go run "c:\coding didi\golang\Laprak 10\modul3.go"
Bilangan: 9651
Digit pada bilangan 9651 terurut mengecil

```

Deskripsi program

Program ini digunakan untuk mengecek apakah empat digit angka yang diinput tersusun secara membesar, mengecil, atau acak. Program membaca input melalui

fmt.Scan(&bilangan) lalu memecah bilangan tersebut menjadi empat digit terpisah: d4 = bilangan % 10 mengambil digit paling belakang, d3 = (bilangan % 100) / 10 mengambil digit ketiga, d2 = (bilangan % 1000) / 100 mengambil digit kedua, dan d1 = bilangan / 1000 mengambil digit pertama. Setelah setiap digit diperoleh, program melakukan pengecekan berurutan. Jika kondisi d1 < d2 && d2 < d3 && d3 < d4 terpenuhi, program menandainya sebagai “terurut membesar”. Jika d1 > d2 && d2 > d3 && d3 > d4, maka hasilnya “terurut mengecil”. Selain dua kondisi itu, program menyimpulkan bahwa digit tidak memiliki urutan tertentu dan menampilkan “tidak terurut”. Hasil tersebut kemudian dicetak melalui fmt.Println("Digit pada bilangan", bilangan, teks).

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var beratGram int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&beratGram)

    kg := beratGram / 1000
    sisa := beratGram % 1000
    biayaKg := kg * 10000
    var biayaSisa int

    if kg > 10 {
        biayaSisa = 0
    } else {
        if sisa >= 500 {
            biayaSisa = sisa * 5
        }
    }
}
```

```

    } else {

        biayaSisa = sisa * 15

    }

}

total := biayaKg + biayaSisa

fmt.Printf("Detail berat: %d kg + %d gr\n", kg, sisa)

fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg,
biayaSisa)

fmt.Printf("Total biaya: Rp. %d\n", total)

}

```

Screenshoot program

```

// main
11  sisa := beratGram % 1000
12  biayaKg := kg * 10000
13  var biayaSisa int
14
15  if kg > 10 {
16      biayaSisa = 0
17  } else {
18      if sisa >= 500 {
19          biayaSisa = sisa * 5
20      } else {
21          biayaSisa = sisa * 15
22      }
23  }
24

```

PROBLEMS 6 **OUTPUT** **DEBUG CONSOLE** **TERMINAL** **PORTS**

```

PS C:\coding didi\golang\Laprak 10> go run "c:\coding didi\golang\Laprak 10\Soal1.go"
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500

PS C:\coding didi\golang\Laprak 10> go run "c:\coding didi\golang\Laprak 10\Soal1.go"
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750

```

NIM :109082500088
 KELAS :S1IF-13-07
 NAMA :Didi Hermawanto

Ln 3, Col 23 60 character Plain t 100% Wind UTF-8

6 0 Ln 26, Col 5 Tab Size: 4 UTF-8 CRLF Go 1.25.1 Go Live Windsurf Login Prettier

Deskripsi program :

Program ini menghitung biaya pengiriman parcel berdasarkan berat yang dimasukkan dalam gram. Setelah berat dibaca melalui `fmt.Scan`, program mengubahnya menjadi satuan kilogram dengan `kg := beratGram / 1000` dan menghitung sisa gram melalui `sisa := beratGram % 1000`. Biaya utama dihitung dari `biayaKg := kg * 10000`, sementara variabel `biayaSisa` menentukan tambahan biaya berdasarkan sisa gram. Jika total berat lebih dari 10 kg, sisa gram digratiskan sehingga `biayaSisa = 0`. Jika tidak, sisa yang ≥ 500 gram dikenakan tarif Rp5 per gram, dan sisa di bawah 500 gram dikenakan Rp15 per gram. Total biaya kemudian dijumlahkan dengan `total := biayaKg + biayaSisa`, lalu

seluruh detail hasil perhitungan ditampilkan menggunakan `fmt.Printf`. Alur program dimulai dari input, memisahkan berat menjadi kg dan gram, menghitung biaya dasar dan tambahan dengan percabangan, menjumlahkan total, lalu menampilkan hasilnya.

2. Tugas 2

Source code :

“ini merupakan Perbaikan Program Yang benarnya” yang berkaitan dengan soal yang C

```
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)
}

```

Screenshoot program

“ini merupakan hasil uji dengan menggunakan program diatas yang benarnya, sekaligus berkaitan dengan soal yang C

The screenshot shows a Go IDE with the following components:

- EXPLORER:** Lists files including modul1.go, modul2.go, modul3.go, Soal1.go, soal2.go, and soal3.go.
- Source Code (soal2.go):**

```

9      fmt.Print("Nilai akhir mata kuliah: ")
10     fmt.Scan(&nam)
11
12     if nam > 80 {
13         nmk = "A"
14     } else if nam > 72.5 {
15         nmk = "AB"
16     } else if nam > 65 {
17         nmk = "B"
18     } else if nam > 57.5 {
19         nmk = "BC"
20     } else if nam > 50 {
21         nmk = "C"
22     } else if nam > 40 {

```
- Terminal:** Shows the execution of the program with three test cases:
 - Input: 93.5, Output: Nilai akhir mata kuliah: 93.5, Nilai mata kuliah: A
 - Input: 70.6, Output: Nilai akhir mata kuliah: 70.6, Nilai mata kuliah: B
 - Input: 49.5, Output: Nilai akhir mata kuliah: 49.5, Nilai mata kuliah: D
- Output Window:** Displays user input: NIM: 109082500088, KELAS: S1IF-13-07, NAMA: Didi Hermawanto.

Deskripsi program :

Program ini digunakan untuk menentukan nilai huruf mata kuliah berdasarkan nilai akhir yang dimasukkan pengguna. Nilai disimpan pada variabel `nam` dengan tipe `float64`, lalu dibaca melalui `fmt.Scan(&nam)`. Setelah nilai diinput, program membandingkan nilai tersebut dengan beberapa rentang menggunakan percabangan `if` bertingkat. Jika nilai lebih dari 80, maka `nmk` diisi "A", kemudian turun ke rentang berikutnya seperti `> 72.5` untuk "AB", `> 65` untuk "B", dan seterusnya sampai akhirnya nilai di bawah atau sama dengan 40 menghasilkan "E". Seluruh hasil disimpan dalam variabel `nmk` dan ditampilkan melalui `fmt.Println("Nilai mata kuliah:", nmk)`. Dengan pembagian rentang tersebut, program bisa mengubah nilai angka menjadi nilai huruf sesuai standar penilaian yang diberikan.

Jawablah pertanyaan-pertanyaan berikut :

a. Jika $\text{nam} = 80.1$, apa keluarannya? Apakah sesuai spesifikasi?

- Input: **80.1**
- Dalam program asli, pengecekan dilakukan berurutan dengan if tanpa else if. Nilai nam mula-mula 80.1, tetapi kemudian variabel nam malah diubah menjadi string, bukan nmk . Selain itu, karena tidak menggunakan else if, kondisi akan terus dicek meskipun sudah terpenuhi sebelumnya.

Urutan yang terjadi:

Kondisi	Hasil
$\text{nam} > 80 \rightarrow \text{benar}$	$\text{nam} = \text{"A"}$
$\text{nam} > 72.5 \rightarrow \text{benar}$	$\text{nam} = \text{"AB"}$
$\text{nam} > 65 \rightarrow \text{benar}$	$\text{nam} = \text{"B"}$
$\text{nam} > 57.5 \rightarrow \text{benar}$	$\text{nam} = \text{"BC"}$
$\text{nam} > 50 \rightarrow \text{benar}$	$\text{nam} = \text{"C"}$
$\text{nam} > 40 \rightarrow \text{benar}$	$\text{nam} = \text{"D"}$

Karena seluruh kondisi benar, nilai terakhir jadi **"D"** dan yang dicetak adalah nmk yang **kosong**, sehingga outputnya:

Nilai mata kuliah:

Program ini **tidak sesuai spesifikasi**, karena nilai 80.1 seharusnya menghasilkan **A**, bukan nilai berlapis dan variabel yang salah.

b. Kesalahan program serta alasannya

Kesalahan

Mengubah nam menjadi string (misal $\text{nam} = \text{"A"}$).

Semua kondisi memakai if terpisah, bukan else if.

Output mencetak nmk padahal nilai yang diubah nam .

Batas interval tidak tepat (harus menggunakan rentang lengkap).

Penjelasan

nam harus tetap bertipe angka; nilai huruf disimpan di variabel nmk .

Akibatnya semua kondisi benar akan dieksekusi bertumpuk.

Program tidak pernah menetapkan nilai ke nmk .

Spec mensyaratkan contoh $65 < \text{NAM} \leq 72.5$.

c. Perbaiki program + hasil uji

Program versi benar

```
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)
}
```

Hasil Pengujian

Input	Output	Sesuai?
93.5	A	✓
70.6	B	✓
49.5	D	✓

3. Tugas 3

Source code

```
package main

import "fmt"
```

```

func main() {
    var b int

    fmt.Print("Bilangan: ")
    fmt.Scan(&b)

    fmt.Print("Faktor: ")

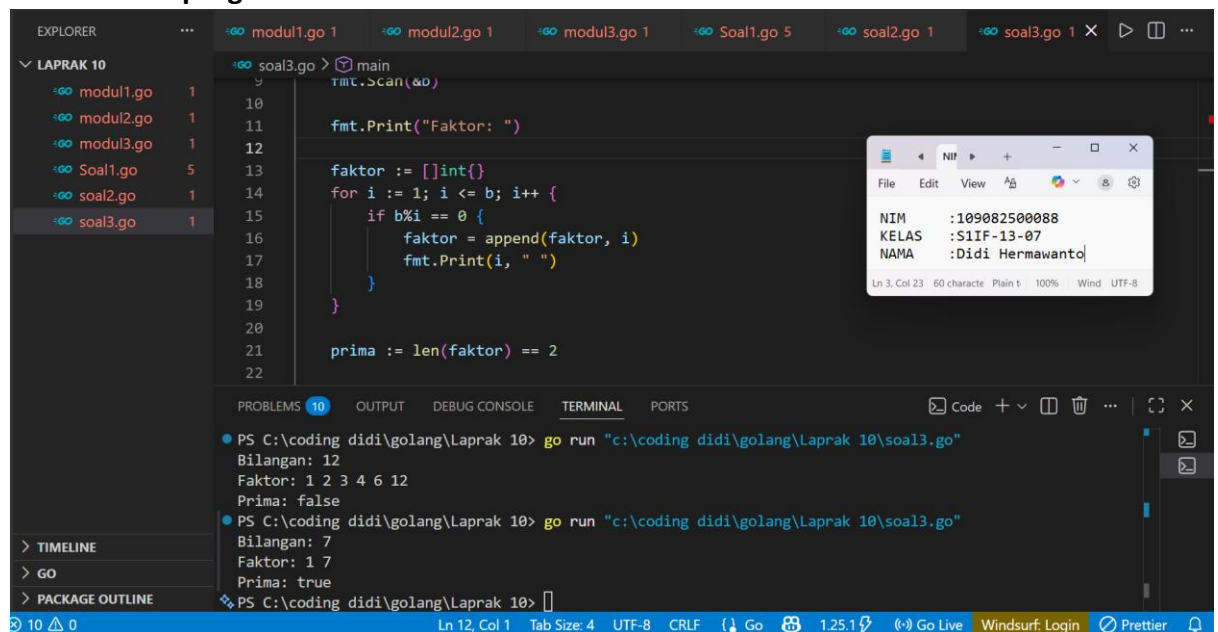
    faktor := []int{}
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            faktor = append(faktor, i)
            fmt.Print(i, " ")
        }
    }

    prima := len(faktor) == 2

    fmt.Println()
    fmt.Println("Prima:", prima)
}

```

Screenshoot program



Deskripsi program :

Program ini dibuat untuk mencari semua faktor dari sebuah bilangan bulat positif yang diinput oleh pengguna, kemudian menentukan apakah bilangan tersebut merupakan bilangan prima atau bukan. Proses dimulai dengan meminta input angka melalui `fmt.Scan()`. Setelah nilai dimasukkan, program melakukan pengecekan faktor dengan perulangan dari angka 1 hingga nilai bilangan tersebut. Setiap angka yang habis

membagi bilangan utama ($b \% i == 0$) dianggap sebagai faktor dan langsung dicetak serta dimasukkan ke dalam slice faktor.

Setelah seluruh angka dicek, program menentukan apakah bilangan tersebut prima dengan cara menghitung jumlah faktor yang ditemukan. Jika jumlah faktor tepat dua, yaitu hanya 1 dan dirinya sendiri, maka bilangan dianggap prima ($\text{len}(\text{faktor}) == 2$), jika lebih dari itu maka bukan bilangan prima. Terakhir, program menampilkan hasil daftar faktor dan status apakah bilangan tersebut prima atau tidak.

Contoh alurnya: jika pengguna memasukkan angka 12, program akan menemukan faktor 1, 2, 3, 4, 6, dan 12 sehingga jumlahnya lebih dari dua dan program menampilkan "Prima: false". Sebaliknya, bila input 7, faktor hanya 1 dan 7 sehingga program menampilkan "Prima: true".