

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 10

ELSE-IF



Disusun oleh:

RAFFI YULIANSYAH

109082500101

S1IF-13-07

Asisten Praktikum

Adithana dharmaputra

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 umur int

    var kk bool

    fmt.Scan(&umur, &kk)

    if umur >= 17 && kk == true {

        fmt.Print("Bisa membuat ktp")

    } else {

        fmt.Print("Belum bisa membuat ktp")

    }

}
```

Screenshoot program

The screenshot shows a Go IDE with a file named `guided1.go`. The code defines a `main` function that declares `umur` as an integer and `kk` as a boolean. It uses `fmt.Scan(&umur, &kk)` to read input. An `if-else` statement checks if `umur` is greater than or equal to 17 and `kk` is true. If both conditions are met, it prints "Bisa membuat ktp"; otherwise, it prints "Belum bisa membuat ktp".

The terminal output shows the execution of the program with the following inputs and outputs:

```
PS D:\Coding\Go> cd .\modul-10\
PS D:\Coding\Go\modul-10> go run .\guided1.go
17 true
Bisa membuat ktp
PS D:\Coding\Go\modul-10> go run .\guided1.go
20 false
Belum bisa membuat ktp
PS D:\Coding\Go\modul-10> go run .\guided1.go
15 true
Belum bisa membuat ktp
PS D:\Coding\Go\modul-10>
```

A Notepad window is also visible, containing the text:

```
109082500101
Raffi Yuliansyah
IF-13-07
```

Deskripsi program

Program tersebut memiliki tujuan menentukan apakah seseorang bisa membuat KTP atau tidak bisa berdasarkan umur dan kepemilikan KK.

Pada awal program akan diinisialisasikan variabel umur bertipe integer untuk menyimpan input umur (tahun) dan variabel `kk` bertipe data boolean untuk mewakili kepemilikan KK (`true` = memiliki dan `false` = tidak memiliki). Kemudian akan diinputkan menggunakan `fmt.Scan(&umur, &kk)` suatu bilangan untuk umur dan value `true/false` untuk `kk`, selanjutnya disimpan di masing masing variabel. Kemudian untuk menentukan dapatkah seseorang membuat KTP menggunakan `if-else`. Kode di dalam `if` dengan kondisi, apabila umur lebih besar sama dengan dari 17 DAN `kk` memiliki value `true`, maka akan ditampilkan "Bisa membuat ktp". Jika kondisi tersebut salah maka bagian `else` akan dijalankan yaitu menampilkan "Tidak bisa membuat ktp".

2. Guided 2

Source Code

```
package main

import (

    "fmt"

)
```

```

func main() {

    var char rune

    fmt.Scanf("%c", &char)

    if char == 'a' || char == 'A' || char == 'i' || char
    == 'I' || char == 'u' || char == 'U' || char == 'e' ||
    char == 'E' || char == 'o' || char == 'O' {

        fmt.Print("vokal")

    } else if (char >= 'A' && char <= 'Z') || (char >=
    'a' && char <= 'z') {

        fmt.Print("konsonan")

    } else {

        fmt.Print("Bukan Huruf")

    }

}

```

Screenshoot program

The screenshot shows a Go IDE with the following components:

- Editor:** Displays the Go code for `main.go`. The code is identical to the one in the first block.
- Terminal:** Shows the execution of the program. The output is:


```

PS D:\Coding\Go\modul-10> go run .\guided2.go
A
vokal
PS D:\Coding\Go\modul-10> go run .\guided2.go
f
konsonan
PS D:\Coding\Go\modul-10> go run .\guided2.go
1
Bukan Huruf
PS D:\Coding\Go\modul-10> go run .\guided2.go
$
Bukan Huruf
PS D:\Coding\Go\modul-10>

```
- Notepad:** A small window titled "Untitled - Notepad" is open, displaying the text:


```

109082500101
Raffi Yuliansyah
IF-13-07

```

Deskripsi program

Program tersebut memiliki tujuan menentukan input yang diberikan termasuk ke dalam vokal, konsonan, atau bukan huruf.

Pada awal program akan diinisialisasikan variabel char bertipe rune untuk menyimpan input. Kemudian akan diinputkan menggunakan `fmt.Scan(&x)` suatu karakter dan disimpan di variabel char. Kemudian untuk memeriksa karakter yang diinputkan termasuk vokal, konsonan, atau huruf digunakan `else-if`. Terdapat 3 percabangan,

1. If memiliki kondisi, apabila char berupa salah satu dari a, i, u, e, o, A, I, U, E, ATAU O, maka kondisi ini true dan menampilkan “vokal” menandakan isi dari char termasuk vokal.
2. Else if memiliki kondisi, jika char berada diantara rentang a sampai z ATAU A sampai Z, maka kondisi ini true dan akan menampilkan “konsonan”, menandakan char termasuk konsonan
3. Else, apabila kedua kondisi diatas tidak terpenuhi (false), yang berarti bukan huruf, maka akan menampilkan “bukan huruf”.

Setelah percabangan dilakukan, program selesai.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var bilangan, d1, d2, d3, d4 int

    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 {

        fmt.Print("Terurut Membesar")

    } else if d1 > d2 && d2 > d3 && d3 > d4 {
```

```

        fmt.Print("Terurut Mengecil")

    } else {

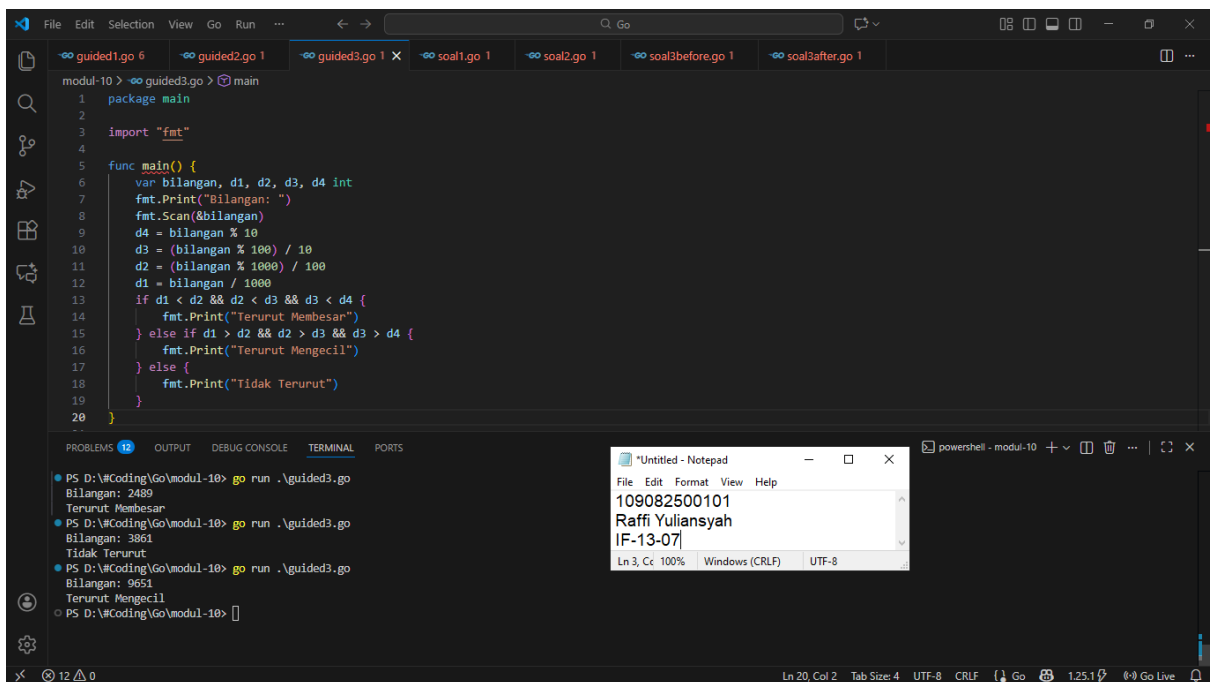
        fmt.Print("Tidak Terurut")

    }

}

```

Screenshoot program



Deskripsi program

Program tersebut memiliki tujuan memeriksa apakah digit yang berurutan dari suatu bilangan ribuan (1000 – 9999) terurut secara menaik, menurun, atau tidak terurut (acak)

Program memiliki 5 variabel integer, diantaranya `bilangan`, `d1`, `d2`, `d3`, `d4`. Variabel `bilangan` berfungsi menampung input angka ribuan, kemudian `d1`, `d2`, `d3`, `d4` untuk menyimpan masing-masing digit dari bilangan. Selanjutnya program akan menerima input lewat `fmt.Scan(&bilangan)`. Untuk menentukan jenis urutan perlu memisahkan tiap digit dengan cara sebagai berikut.

1. Digit pertama (ribuan) akan disimpan di `d1` yang didapat dari membagi `bilangan` dengan 1000 (`bilangan/1000`).
2. Digit kedua (ratusan) akan disimpan di `d2` yang didapat dari me-modulo `bilangan` dengan 1000, kemudian dibagi 100 (`(bilangan%1000)/100`).
3. Digit ketiga (puluhan) akan disimpan di `d3` yang didapat dari me-modulo `bilangan` dengan 1000, kemudian dibagi 10 (`(bilangan%1000)/10`).

4. Digit keempat (satuan) akan disimpan di d4 yang didapat dari me-modulo bilangan dengan 10 (bilangan%10)

Setelah keempat digit didapatkan, bilangan ditentukan terurut menaik atau menurun dengan membandingkan masing masing digit dalam 3 kondisi.

1. Ketika (if) $d1 < d2$ DAN $d2 < d3$ DAN $d3 < d4$, maka akan ditampilkan bahwa bilangan tersebut "Terurut menaik".
2. Ketika (else if) $d1 > d2$ DAN $d2 > d3$ DAN $d3 > d4$, maka akan ditampilkan bilangan tersebut "Terurut menurun".
3. Ketika kedua kondisi tidak terpenuhi (else), maka akan ditampilkan bilangan tersebut "Tidak terurut"

Kemudian program selesai.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var berat int

    fmt.Print("Berat parsel (gram) : ")

    fmt.Scan(&berat)

    fmt.Println("Detai berat : ", berat/1000, " kg", " + ",
berat%1000, " gram")

    if berat >= 10000 {

        if berat%1000 >= 500 {

            fmt.Println("Detail Biaya : ", "Rp. ",
(berat/1000)*10000, " + ", "Rp. ", (berat%1000)*5)
```

```
        fmt.Println("Total Biaya : ", "Rp. ", ((berat /
1000) * 10000))

        } else {

            fmt.Println("Detail Biaya : ", "Rp. ",
(berat/1000)*10000, " + ", "Rp. ", (berat%1000)*15)

            fmt.Println("Total Biaya : ", "Rp. ", ((berat /
1000) * 10000))

        }

    } else if berat%1000 >= 500 {

        fmt.Println("Detail Biaya : ", "Rp. ",
(berat/1000)*10000, " + ", "Rp. ", (berat%1000)*5)

        fmt.Println("Total Biaya : ", "Rp. ",
((berat/1000)*10000)+((berat%1000)*5))

    } else {

        fmt.Println("Detail Biaya : ", "Rp. ",
(berat/1000)*10000, " + ", "Rp. ", (berat%1000)*15)

        fmt.Println("Total Biaya : ", "Rp. ",
((berat/1000)*10000)+((berat%1000)*15))

    }

}
```

Screenshoot program


```
modul-10 > go soal1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var berat int
7     fmt.Print("Berat parcel (gram) : ")
8     fmt.Scan(&berat)
9
10    fmt.Println("Detail berat : ", berat/1000, " kg", " + ", berat%1000, " gram")
11    if berat >= 10000 {
12        if berat%1000 >= 500 {
13            fmt.Println("Detail Biaya : ", "Rp. ", (berat/1000)*10000, " + ", "Rp. ", (berat%1000)*5)
14            fmt.Println("Total Biaya : ", "Rp. ", ((berat / 1000) * 10000))
15        } else {
16            fmt.Println("Detail Biaya : ", "Rp. ", (berat/1000)*10000, " + ", "Rp. ", (berat%1000)*15)
17            fmt.Println("Total Biaya : ", "Rp. ", ((berat / 1000) * 10000))
18        }
19    } else if berat%1000 >= 500 {
20        fmt.Println("Detail Biaya : ", "Rp. ", (berat/1000)*10000, " + ", "Rp. ", (berat%1000)*5)
21        fmt.Println("Total Biaya : ", "Rp. ", ((berat/1000)*10000)+((berat%1000)*5))
22    } else {
23        fmt.Println("Detail Biaya : ", "Rp. ", (berat/1000)*10000, " + ", "Rp. ", (berat%1000)*15)
24        fmt.Println("Total Biaya : ", "Rp. ", ((berat/1000)*10000)+((berat%1000)*15))
25    }
26 }
```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Total Biaya : Rp. 110000
PS D:\VCoding\Go\modul-10> []

Untitled - Notepad
File Edit Format View Help
109082500101
Raffi Yuliansyah
IF-13-07
Ln 3, Cc 100% Windows (CRLF) UTF-8

```
modul-10 > go soal1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var berat int
7     fmt.Print("Berat parcel (gram) : ")
8     fmt.Scan(&berat)
9
10    fmt.Println("Detail berat : ", berat/1000, " kg", " + ", berat%1000, " gram")
11    if berat >= 10000 {
12        if berat%1000 >= 500 {
13            fmt.Println("Detail Biaya : ", "Rp. ", (berat/1000)*10000, " + ", "Rp. ", (berat%1000)*5)
14            fmt.Println("Total Biaya : ", "Rp. ", ((berat / 1000) * 10000))
15        } else {
16            fmt.Println("Detail Biaya : ", "Rp. ", (berat/1000)*10000, " + ", "Rp. ", (berat%1000)*15)
17            fmt.Println("Total Biaya : ", "Rp. ", ((berat / 1000) * 10000))
18        }
19    } else if berat%1000 >= 500 {
20        fmt.Println("Detail Biaya : ", "Rp. ", (berat/1000)*10000, " + ", "Rp. ", (berat%1000)*5)
21        fmt.Println("Total Biaya : ", "Rp. ", ((berat/1000)*10000)+((berat%1000)*5))
22    } else {
23        fmt.Println("Detail Biaya : ", "Rp. ", (berat/1000)*10000, " + ", "Rp. ", (berat%1000)*15)
24        fmt.Println("Total Biaya : ", "Rp. ", ((berat/1000)*10000)+((berat%1000)*15))
25    }
26 }
```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Terurut Menecill
● PS D:\VCoding\Go\modul-10> go run .\soal1.go
Berat parcel (gram) : 8500
Detail berat : 8 kg + 500 gram
Detail Biaya : Rp. 80000 + Rp. 2500
Berat parcel (gram) : 8500
● Detail berat : 8 kg + 500 gram
Detail Biaya : Rp. 80000 + Rp. 2500
Total Biaya : Rp. 82500
PS D:\VCoding\Go\modul-10> go run .\soal1.go
Berat parcel (gram) : 9250
● Detail berat : 9 kg + 250 gram
Detail Biaya : Rp. 90000 + Rp. 3750
Detail berat : 9 kg + 250 gram
Detail Biaya : Rp. 90000 + Rp. 3750
Total Biaya : Rp. 93750
PS D:\VCoding\Go\modul-10> go run .\soal1.go
PS D:\VCoding\Go\modul-10> go run .\soal1.go
Berat parcel (gram) : 11750
Detail berat : 11 kg + 750 gram
Detail Biaya : Rp. 110000 + Rp. 3750
Total Biaya : Rp. 110000
PS D:\VCoding\Go\modul-10> []

Untitled - Notepad
File Edit Format View Help
109082500101
Raffi Yuliansyah
IF-13-07
Ln 3, Cc 100% Windows (CRLF) UTF-8

Deskripsi program

Program tersebut memiliki tujuan menghitung berat dan biaya pengiriman parcel.

Pada awal program akan diinisialisasikan variabel berat bertipe data integer untuk menyimpan berat parcel yang diinput. Kemudian akan diinputkan berat parcel dalam satuan gram dengan `fmt.Scan(&berat)` dan disimpan di variabel berat. Untuk menghitung berat parcel secara spesifik maka akan ditampilkan berat dalam kg didapatkan dari `berat/1000` dan ditampilkan sisa berat dalam gram dari `berat%1000`.

Untuk menghitung biaya pengiriman dan rinciannya menggunakan `else if` dengan 3 kondisi sesuai dengan ketentuan.

- If memiliki kondisi apabila berat lebih dari sama dengan 10kg, maka biaya total untuk pengiriman hanya dikenakan biaya per 1000 gram saja $(\text{berat}/1000) \times 10000$.

- b. Else if memiliki kondisi apabila sisa dari berat persatuan kg lebih dari sama dengan 500 pada kode program $(\text{berat} \% 1000) \geq 500$, maka biaya pengiriman adalah $((\text{berat}/1000)*10000) + (\text{berat}\%1000)*5$
 - c. Else, ketika semua kondisi diatas bernilai salah, yang memiliki arti sisa dari berat persatuan kg kurang dari 500 $(\text{berat} \% 1000) < 500$, maka biaya pengiriman adalah $((\text{berat}/1000)*10000) + (\text{berat}\%1000)*15$
- Program Selesai.

2. Tugas 2

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 if nam <= 40 {

        nmk = "E"

    }

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

}

```

Screenshoot program

The screenshot shows a Go IDE with a file named `soal2.go`. The code defines a `main` function that takes a float `nam` and a string `nmk`. It uses a series of `if-then` statements to assign values to `nmk` based on the value of `nam`. The logic is as follows:

- If `nam > 80`, set `nmk = "A"`.
- Else if `nam > 72.5`, set `nmk = "AB"`.
- Else if `nam > 65`, set `nmk = "B"`.
- Else if `nam > 57.5`, set `nmk = "BC"`.
- Else if `nam > 50`, set `nmk = "C"`.
- Else if `nam > 40`, set `nmk = "D"`.
- Else if `nam <= 40`, set `nmk = "E"`.

The terminal output shows the program being run with `nam = 93.5`. The output is:

```

PS D:\Coding\Go\modul-10> go run .\soal2.go
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: D

```

The output is incorrect because the program should have assigned `nmk = "A"` for `nam = 93.5`. The error is due to the `if-then` logic, where the final `else if` condition (`nam <= 40`) is always false, but the previous conditions are not mutually exclusive, leading to the final condition being the only one that executes.

Jawaban :

- Apabila dijalankan sesuai dengan soal maka akan menampilkan Nilai D.
- Program tersebut memiliki kesalahan, yaitu percabangan menggunakan If-then yang mengakibatkan isi dari variabel `nmk` pada if then pertama (ketika `nam > 80`) terganti dengan if yang terakhir (ketika `nam > 40`), pada contoh kasusnya apabila `nam = 93,5` maka yang seharusnya isi variabel `nmk` adalah A, tetapi yang tertampil `nmk` adalah D, sehingga terjadi kesalahan dari program. Hal tersebut terjadi karena percabangan menggunakan if-then yang menyebabkan semua percabangan tereksekusi, sehingga aksi mengganti isi variabel `nmk = "A"` pada if then pertama tertimpa (terganti) dengan aksi if-then terakhir yang mengganti isi variabel `nmk = "D"`.

Seharusnya program tersebut memiliki alur sebagai berikut, yaitu program akan membuat variabel `nam` bertipe `float64` dan `nmk` yang bertipe `string`. Setelah itu program akan menerima input `nam`. Kemudian `nmk` akan ditentukan menggunakan `nam` dengan percabangan `else-if` (dipilih karena hanya satu kondisi saja yang akan dieksekusi) sesuai nilai yang ditentukan. If `nam > 80` maka `nmk` berisi "A", else if `nam > 72,5` maka `nmk` berisi "AB", else if `nam > 65` maka `nmk` berisi "B", else if `nam > 57.5` maka `nmk` berisi "BC", else if `nam > 50` maka `nmk` berisi "C", else if `nam > 40` maka `nmk` berisi "D", else if `nam <= 40` maka `nmk` berisi "E". Kemudian isi variabel `nmk` ditampilkan.

- c. Setelah diperbaiki, apabila diberi masukan 93,5 akan menampilkan `nmk` yaitu A, dengan masukan 70,6 akan menampilkan `nmk` yaitu B, dengan masukan 49,5 akan menampilkan D

3. Tugas 3

Source code

Before

```
package main

import "fmt"

func main(){
    var bilangan int
    fmt.Print("Bilangan : ")
    fmt.Scan(&bilangan)
    fmt.Print("Faktor : ")
    for i:=1; i<=bilangan; i++){
        if bilangan%i == 0 {
            fmt.Print(i, " ")
        }
    }
}
```

After

```
package main
```

```
import "fmt"

func main() {

    var bilangan int

    var prima bool

    prima = true

    fmt.Print("Bilangan : ")

    fmt.Scan(&bilangan)

    fmt.Print("Faktor : ")

    for i := 1; i <= bilangan; i++ {

        if bilangan%i == 0 {

            fmt.Print(i, " ")

        }

    }

    if bilangan == 0 || bilangan == 1 {

        prima = false

    }

    for i := 2; i <= bilangan-1; i++ {

        if bilangan%i == 0 {

            prima = false

            break

        }

    }

    if prima {

        fmt.Print("Prima : ", prima)

    } else {
```

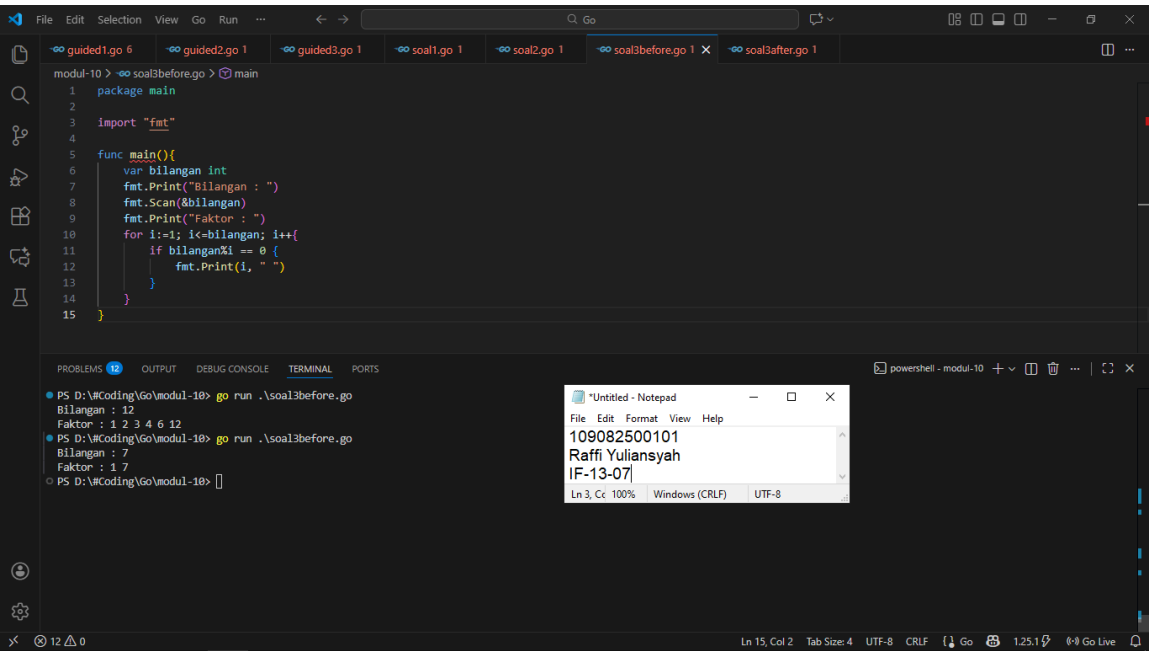
```
        fmt.Print("Prima : ", prima)

    }

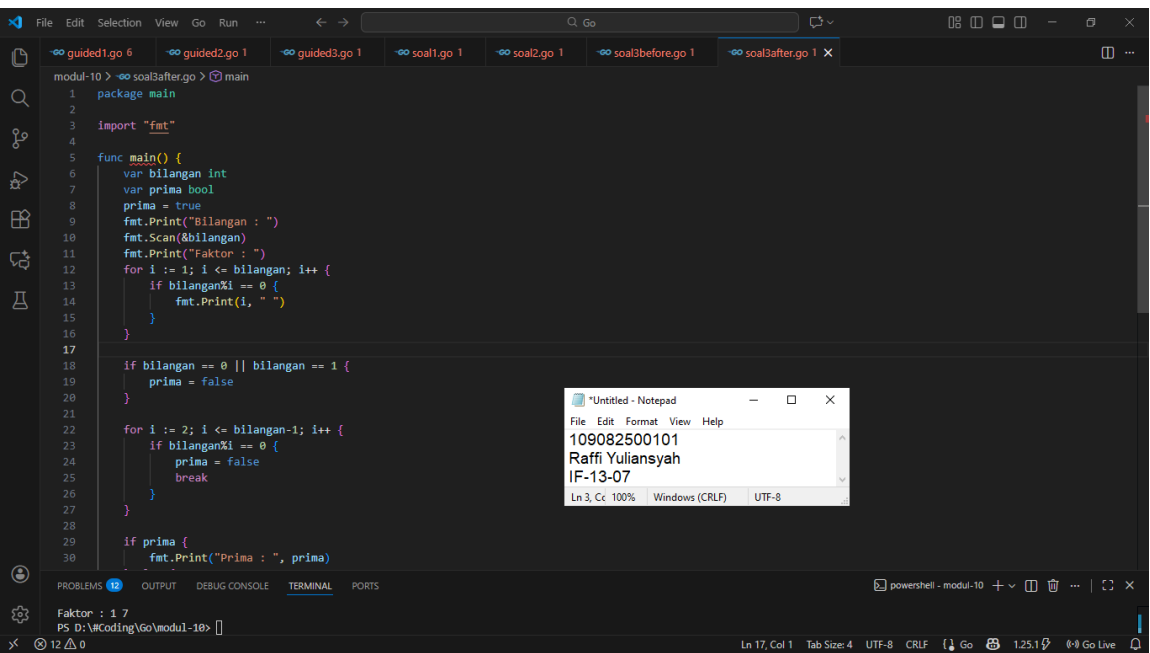
}
```

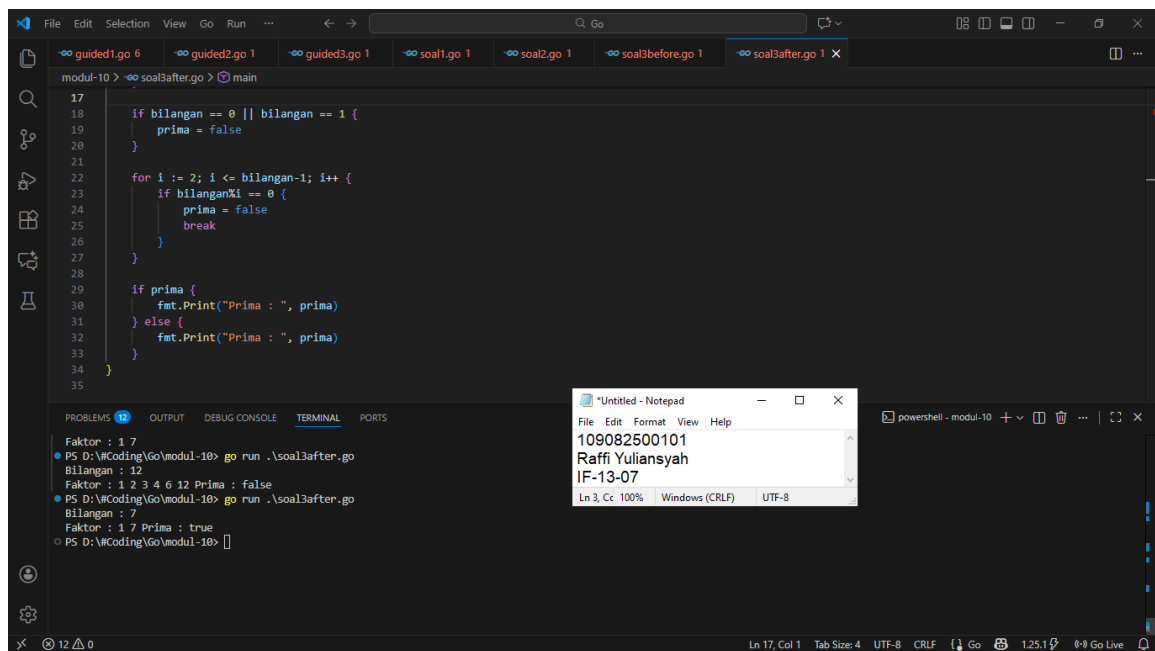
Screenshoot program

Before



After





Deskripsi program

Before

Program tersebut memiliki tujuan menampilkan beberapa angka yang menjadi faktor dari suatu bilangan bulat.

Pada awal program akan diinisialisasikan variabel "bilangan" bertipe data integer untuk menyimpan angka yang diinput. Kemudian akan diinputkan suatu bilangan bulat dengan `fmt.Scan(&bilangan)` dan disimpan di variabel bilangan. Untuk menentukan angka apa saja yang menjadi faktor dari variabel bilangan menggunakan for-loop. Dimana pada forloop tersebut akan dicek dari angka 1 sampai dengan variabel bilangan apakah angka angka tersebut dapat membagi variabel bilangan.

```
for i:=1; i<=bilangan; i++){  
    if bilangan%i == 0 {  
        fmt.Print(i, " ")
```

Apabila var bilangan dimodulo dengan i (angka yang akan dicek berdasarakan loop) memiliki hasil 0 maka angka tersebut ditampilkan sebagai faktor dari variabel bilangan. Program selesai.

After

Program tersebut memiliki tujuan menampilkan beberapa angka yang menjadi faktor dari suatu bilangan bulat dan menentukan apakah bilangan tersebut prima atau bukan.

Pada awal program akan diinisialisasikan variabel "bilangan" bertipe data integer untuk menyimpan angka yang diinput dan variabel "prima" bertipe boolean. Kemudian akan diinputkan suatu bilangan bulat dengan `fmt.Scan(&bilangan)` dan disimpan di variabel bilangan. Untuk menentukan angka apa saja yang menjadi

faktor dari variabel bilangan menggunakan for-loop. Dimana pada forloop tersebut akan memeriksa dari angka 1 sampai dengan variabel bilangan, apakah angka tersebut dapat membagi variabel bilangan.

```
for i:=1; i<=bilangan; i++){  
    if bilangan%i == 0 {  
        fmt.Print(i, " ")
```

Apabila var bilangan dimodulo dengan i (angka yang akan dicek berdasarkan loop) memiliki hasil 0 maka angka tersebut ditampilkan sebagai faktor dari variabel bilangan. Kemudian untuk menentukan apakah bilangan tersebut prima atau bukan menggunakan cara for-loop dimana akan diperiksa dari 2 sampai dengan satu angka sebelum variabel bilangan, apakah angka tersebut bisa membagi variabel bilangan.

```
if bilangan == 0 || bilangan == 1 {  
    prima = false  
}  
  
for i := 2; i <= bilangan-1; i++ {  
    if bilangan%i == 0 {  
        prima = false  
    }  
}  
  
if prima {  
    fmt.Print("Prima : ", prima)
```

Apabila isi dari bilangan adalah 0 dan 1 maka prima bernilai false yang menandakan bilangan bukanlah bilangan prima. Dengan for-loop akan dicek dari 2 sampai (bilangan-1) apabila terdapat i yang habis membagi bilangan maka prima bernilai true yang menandakan bahwa bilangan termasuk prima. Program selesai.