

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE - IF



Disusun oleh:

ELSA DWI RIZQIYANTI

109082500090

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

Source Code

```
package main

import "fmt"

func main() {
    var usia int
    var status bool

    fmt.Scanln(&usia)
    fmt.Scanln(&status)

    if usia >= 17 && status {
        fmt.Println("bisa membuat KTP")
    } else {
        fmt.Println("belum bisa membuat KTP")
    }
}
```

Screenshot program

The screenshot shows a Go development environment with the following details:

- File Explorer:** Shows a project named "MODUL 10" containing a file "guide1.go".
- Code Editor:** Displays the content of "guide1.go":

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var usia int
7     var status bool
8
9     fmt.Scanln(&usia)
10    fmt.Scanln(&status)
11
12    if usia >= 17 && status {
13        fmt.Println("bisa membuat KTP")
14    } else {
15        fmt.Println("belum bisa membuat KTP")
16    }
17 }
18
```
- Terminal:** Shows command-line output:

```
PS C:\praktikum\alpro\modul 10> go run guide1.go
17
true
bisa membuat KTP
PS C:\praktikum\alpro\modul 10> go run guide1.go
20
false
belum bisa membuat KTP
PS C:\praktikum\alpro\modul 10> go run guide1.go
15
true
belum bisa membuat KTP
PS C:\praktikum\alpro\modul 10>
```
- Output Window:** A modal window titled "ELSA DWI" showing the results of the program execution:

ELSA DWI RIZQIYANTI
IF-13-02
109082500090
- Status Bar:** Shows "Ln 3, Col 13 - 41 karakter" and "10.27 17/11/2025".

Deskripsi program

var usia int
var status bool
Mendeklarasikan dua variabel:
usia bertipe int (bilangan bulat)
status bertipe bool (benar/salah, true/false)

fmt.Scanln(&usia)
fmt.Scanln(&status)
Program membaca input dari pengguna:

Input pertama dimasukkan ke variabel usia
Input kedua dimasukkan ke variabel status

```
if usia >= 17 && status {
    fmt.Println("bisa membuat KTP")
} else {
    fmt.Println("belum bisa membuat KTP")
}
```

Ini adalah bagian logika program.

Program mengecek dua syarat secara bersamaan:

Syarat 1: usia >= 17

Artinya: umur harus 17 tahun atau lebih.

Syarat 2: status

Variabel status harus bernilai true.

Operator &&

Berarti dua kondisi harus sama-sama benar.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var alfabet string

    fmt.Scan(&alfabet)

    if (alfabet >= "A" && alfabet <= "Z") || (alfabet >= "a"
&& alfabet <= "z") {
        if alfabet == "A" || alfabet == "a" ||
            alfabet == "I" || alfabet == "i" ||
            alfabet == "U" || alfabet == "u" ||
            alfabet == "E" || alfabet == "e" ||
            alfabet == "O" || alfabet == "o" {
            fmt.Println("vokal")
        } else {
            fmt.Println("konsonan")
        }
    }
}
```

```

    } else {

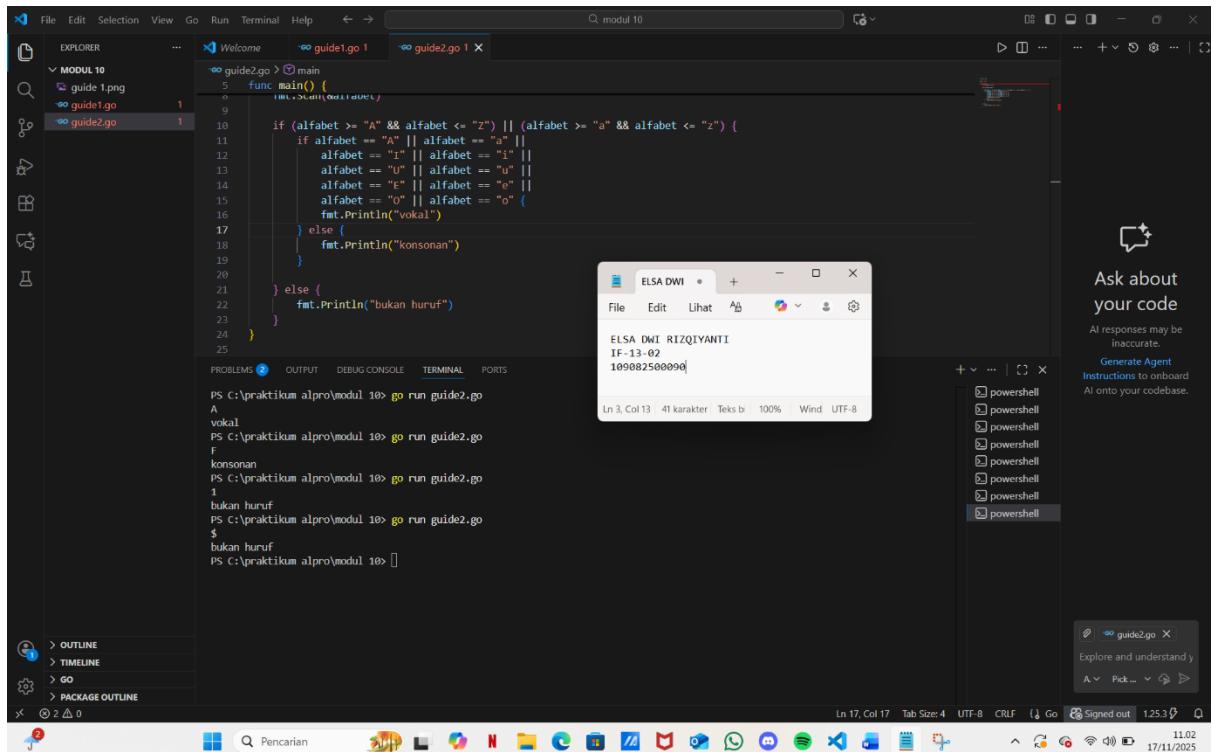
        fmt.Println("bukan huruf")

    }

}

```

Screenshot program



Deskripsi program

var alfabet string

Membuat variabel bernama alfabet bertipe string.

Variabel ini akan digunakan untuk menampung 1 karakter yang dimasukkan oleh pengguna.

fmt.Scan(&alfabet)

Program meminta pengguna memasukkan sebuah karakter (huruf), lalu menyimpannya ke variabel alfabet.

if (alfabet >= "A" && alfabet <= "Z") || (alfabet >= "a" && alfabet <= "z") {

Bagian ini mengecek:

apakah alfabet adalah huruf kapital (A–Z), atau

huruf huruf kecil (a–z)

Jika salah satu benar: masuk ke proses selanjutnya

Jika tidak: program langsung menampilkan “bukan huruf”

Cara kerjanya:

String bisa dibandingkan secara alfabetis, jadi "B" lebih besar dari "A", "C" lebih besar dari "B", dst.

Pemeriksaan Huruf Vokal atau Konsonan

```
if alfabet == "A" || alfabet == "a" ||  
    alfabet == "I" || alfabet == "i" ||  
    alfabet == "U" || alfabet == "u" ||  
    alfabet == "E" || alfabet == "e" ||  
    alfabet == "O" || alfabet == "o" {  
        fmt.Println("vokal")  
    } else {  
        fmt.Println("konsonan")  
    }
```

Penjelasan:

Program mengecek apakah huruf tersebut termasuk vokal: A, I, U, E, O (baik besar maupun kecil).

Jika cocok: tampilkan "vokal".

Jika tidak termasuk daftar di atas : berarti huruf tersebut adalah konsonan.

Bagian Jika Bukan Huruf

```
} else {  
    fmt.Println("bukan huruf")  
}
```

Jika input bukan huruf A–Z atau a–z, program menampilkan:

bukan huruf

Contoh input yang menghasilkan bukan huruf:

1, *, #, @, atau angka dan simbol lainnya.

3. Guided 3

Source Code

```
package main

import (
    "fmt"
)

func main() {
    var n int
    fmt.Print("Bilangan: ")
    fmt.Scanln(&n)

    d1 := n / 1000
    d2 := (n / 100) % 10
    d3 := (n / 10) % 10
    d4 := n % 10

    if d1 < d2 && d2 < d3 && d3 < d4 {
        fmt.Printf("Digit pada bilangan %d terurut
membesar\n", n)
    } else if d1 > d2 && d2 > d3 && d3 > d4 {
        fmt.Printf("Digit pada bilangan %d terurut
mengecil\n", n)
    }
}
```

```

        } else {

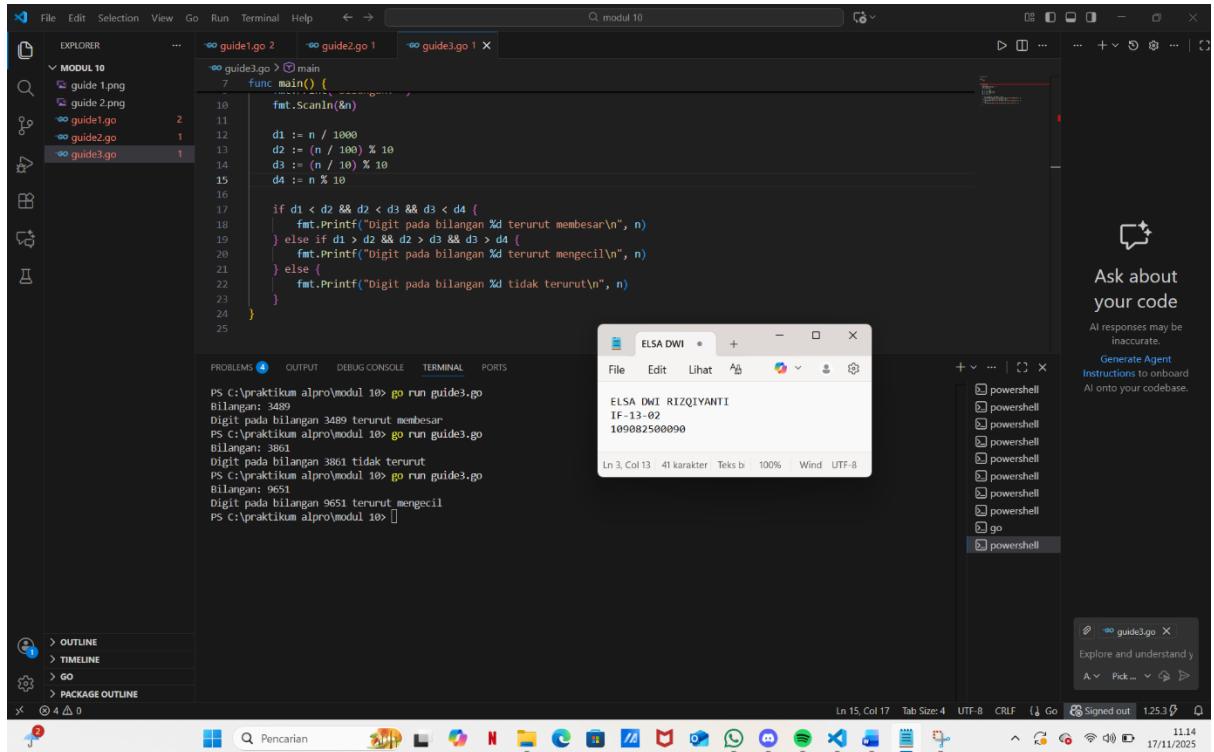
            fmt.Printf("Digit pada bilangan %d tidak
terurut\n", n)

        }

    }

```

Screenshot program



Deskripsi program

Bagian Input Bilangan

var n int

```

fmt.Print("Bilangan: ")
fmt.Scanln(&n)

```

Variabel `n` bertipe `int`, digunakan untuk menyimpan bilangan 4 digit dari pengguna.

Program mencetak teks: Bilangan:

Program menunggu input angka (misalnya 1234).

Bagian Memisahkan Digit Satu per Satu

Misal **n = 3579**, maka:

`d1 := n / 1000`

Mengambil digit ribuan

Contoh: $3579 / 1000 = 3$

```
d2 := (n / 100) % 10  
Mengambil digit ratusan  
3579 / 100 = 35  
35 % 10 = 5
```

```
d3 := (n / 10) % 10  
Mengambil digit puluhan  
3579 / 10 = 357  
357 % 10 = 7
```

```
d4 := n % 10  
Mengambil digit satuan  
3579 % 10 = 9  
Jadi jika input 3579  
d1 = 3, d2 = 5, d3 = 7, d4 = 9.
```

Mengecek Pola Digit

```
if d1 < d2 && d2 < d3 && d3 < d4 {  
    Kondisi ini mengecek apakah:  
    digit 1 lebih kecil dari digit 2  
    digit 2 lebih kecil dari digit 3  
    digit 3 lebih kecil dari digit 4  
  
} else if d1 > d2 && d2 > d3 && d3 > d4 {  
    Kondisi ini mengecek apakah digit-digit mengecil dari depan ke belakang, seperti  
    9876, 4210, 7642, dll.  
    Jika benar, tampilkan: "Digit ... terurut mengecil"
```

```
} else {  
    fmt.Printf("Digit pada bilangan %d tidak terurut\n", n)  
}
```

Jika kedua kondisi sebelumnya tidak terpenuhi, berarti digit-digit bilangan tersebut acak atau tidak berurutan, contoh:

```
1328  
5542  
4831  
2003
```

Maka program menampilkan: "Digit ... tidak terurut"

TUGAS

1. Tugas 1

Source code

```
package main

import (
    "fmt"
)

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

    kg := berat / 1000
    sisa := berat % 1000
    biayaKg := kg * 10000
    biayaSisa := 0

    if kg > 10 {
        biayaSisa = 0
    } else {
        if sisa >= 500 {
            biayaSisa = sisa * 5
        } else if sisa > 0 {
            biayaSisa = sisa * 15
        }
    }
}
```

```
    }

    totalBiaya := biayaKg + biayaSisa

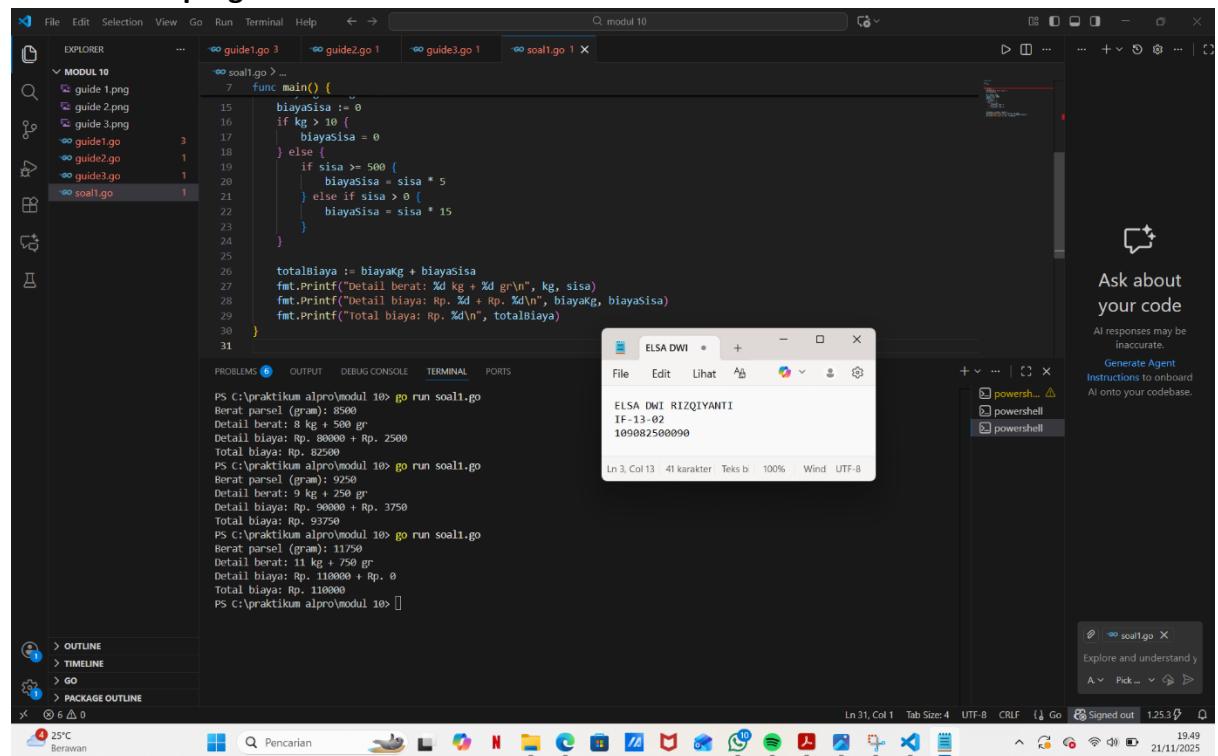
    fmt.Printf("Detail berat: %d kg + %d gr\n", kg, sisa)

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

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

}
```

Screenshot program



Deskripsi program

Bagian Input Berat Parsel

var berat int

```
fmt.Println("Berat parsel (gram): ")  
fmt.Scan(&berat)
```

Program menyiapkan variabel berat bertipe int.

Program meminta pengguna memasukkan berat parsel dalam gram.

Contoh input: 2500 7300 11500 , dll

Menghitung Berat dalam Kilogram dan Gram

kg := berat / 1000

sisa := berat % 1000

Program memisahkan berat menjadi:

Kg: berat kilogram (pembagian bulat)

sisa: sisa gram (modulus)

Menghitung Biaya Berdasarkan Kilogram

biayaKg := kg * 10000

Setiap 1 kg dikenakan biaya Rp10.000.

Menghitung Biaya Sisa Gram

Jika berat lebih dari 10 kg, sisa gram tidak dihitung

if kg > 10 {

 biayaSisa = 0

Artinya, jika paket beratnya besar sekali (lebih dari 10 kg), maka sisa gram gratis.

Jika sisa gram \geq 500 gr, dihitung Rp5 per gram

} else {

 if sisa \geq 500 {

 biayaSisa = sisa * 5

Contoh:

sisa = 650 \rightarrow biayaSisa = $650 \times 5 = \text{Rp } 3.250$

Jika sisa gram $<$ 500 dan $> 0 \rightarrow$ dihitung Rp15 per gram

} else if sisa > 0 {

 biayaSisa = sisa * 15

}

}

Contoh:

sisa = 350: biayaSisa = $350 \times 15 = \text{Rp } 5.250$

Jika sisa = 0: biayaSisa tetap 0.

Menghitung Total Biaya

totalBiaya := biayaKg + biayaSisa

Menjumlahkan biaya kilogram + biaya sisa gram.

Output Program

```
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", totalBiaya)
```

Program menampilkan:

Detail berat (kg dan sisa gram)

Detail biaya (biaya dari kg + biaya dari sisa gram)

Total biaya keseluruhan

Contoh Perhitungan

Berat parsel (gram): 2650

Maka:

kg = 2

sisa = 650

biayaKg = $2 \times 10.000 = 20.000$

sisa $\geq 500 \rightarrow$ biaya = $650 \times 5 = 3.250$

total = $20.000 + 3.250 = 23.250$

2. Tugas 2

a. Jawabannya yaitu tidak sesuai spesifikasi.

Menurut tabel penilaian di soal, nilai 80.1 seharusnya mendapatkan "AB", bukan "D".

Kesalahan terjadi karena program tidak menggunakan else if, sehingga semua kondisi dieksekusi dan nilai akhir terus berubah sampai kondisi terakhir.

1. nam > 80
80.1 lebih besar dari 80 kondisi benar, maka nmk = "A"
2. nam > 72.5
80.1 juga lebih besar dari 72.5 kondisi benar, maka nilai sebelumnya ("A") ditimpak menjadi "AB"
3. nam > 65
80.1 juga lebih besar dari 65 kondisi benar, nmk berubah lagi menjadi "B"
4. nam > 57.5
Benar nmk menjadi "BC"
5. nam > 50
Benar nmk menjadi "C"
6. nam > 40
Benar nmk menjadi "D"
7. else if nam <= 40
Ini dilewati karena kondisi sebelumnya sudah benar.

Jadi nilai akhir yang tersimpan di variabel nmk adalah: D

b. Kesalahan program

1. Semua kondisi menggunakan if terpisah, bukan if – else if

Semua if ini dicek satu per satu, tidak berhenti setelah satu kondisi terpenuhi.

Jika nilai memenuhi beberapa syarat sekaligus, hasil NMK akan ditimpak terus oleh kondisi berikutnya.

2. Rentang nilai tidak sesuai dengan tabel

Dalam tabel soal, setiap nilai punya rentang yang jelas, contoh:

$$72.5 \leq \text{NAM} < 80 \text{ AB}$$

$$65 \leq \text{NAM} < 72.5 \text{ B}$$

3. Kondisi D dan E ditulis tidak tepat

Kondisi >40 harusnya berlaku 40 sampai <50 , bukan semua yang >40 .

Tidak ada batas atas untuk D sehingga semua nilai ≥ 50 pun masuk ke "D" (sebelum ditimpak kondisi lain).

Alur Program Seharusnya

1. Mengecek nilai dari rentang tertinggi ke terendah.
2. Menggunakan if – else if – else supaya ketika satu kondisi benar, kondisi lainnya tidak diperiksa lagi.
3. Menuliskan rentang nilai yang tidak tumpang tindih, seperti:
Jika nilai ≥ 80 A
Jika tidak, tetapi nilai ≥ 72.5 AB
Jika tidak, tetapi nilai ≥ 65 B
Dan seterusnya sampai E.

Dengan begitu program akan berhenti pada kategori yang tepat dan tidak menimpa hasil sebelumnya.

c. Tabel nilai (yang benar):

NAM	NMK
$\text{NAM} \geq 80$	A
$72.5 \leq \text{NAM} < 80$	AB
$65 \leq \text{NAM} < 72.5$	B
$57.5 \leq \text{NAM} < 65$	BC
$50 \leq \text{NAM} < 57.5$	C
$40 \leq \text{NAM} < 50$	D
$\text{NAM} < 40$	E

Alur program seharusnya

```
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

The screenshot shows a Go development environment with the following components:

- Code Editor:** Displays a Go file named `soal2.go` containing a program that calculates a grade based on a score. The code includes imports for `fmt` and `float64`, defines a `main` function, and uses an `if-else` ladder to determine the grade (`A`, `AB`, `B`, `BC`) based on the score.
- Terminal:** Shows the command `go run soal2.go` being run three times, producing output: "Nilai akhir mata kuliah: 93.5", "Nilai akhir mata kuliah: A", and "Nilai akhir mata kuliah: 49.5".
- Browser:** An instance of Microsoft Edge is open, titled "ELSA DWI", displaying the text "ELSA DWI RIZQIYANTI" and "IF-13-02 109082500090".
- CodeLens:** A sidebar on the right shows "Ask about your code" and "Instructions to onboard AI onto your codebase".
- System Taskbar:** At the bottom, the Windows taskbar shows various pinned icons and the date/time: "21/11/2025 20:22".

3. Tugas 3

Source code

```
package main

import "fmt"

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

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

    fmt.Print("Faktor: ")
    count := 0
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Print(i, " ")
            count++
        }
    }
    fmt.Println()
    if count == 2 {
        fmt.Println("Prima: true")
    } else {
        fmt.Println("Prima: false")
    }
}
```

Screenshot program

The screenshot shows a Visual Studio Code interface for a Go project named 'modul 10'. The 'soal3.go' file is open in the editor, containing the following code:

```
func main() {
    fmt.Println("Faktor: ")
    count := 0
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Println(i, " ")
            count++
        }
    }
    fmt.Println()
}

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

The terminal window shows the output of running the program with input '12':

```
Bilangan: 12
Faktor: 1 2 3 4 6 12
```

A powershell window titled 'ELSA DWI' shows the user's name and student ID:

```
ELSA DWI RIZQIYANTI
IF-13-02
109082500090
```

The status bar at the bottom right indicates the file is signed out.

Deskripsi program

Menampilkan Faktor Bilangan

```
fmt.Println("Bilangan: ")
```

```
fmt.Scan(&b)
```

Program meminta pengguna memasukkan satu bilangan.

Nilainya disimpan ke variabel b.

```
fmt.Println("Faktor: ")
```

```
for i := 1; i <= b; i++ {
```

```
    if b%i == 0 {
```

```
        fmt.Println(i, " ")
```

```
}
```

```
}
```

Cara kerja:

Program mencari faktor dari bilangan b dengan cara:

Melakukan loop dari angka 1 sampai b.

Jika $b \% i == 0$ berarti i adalah faktor.

Contoh: b = 12

Faktor = 1, 2, 3, 4, 6, 12

Menghitung Banyaknya Faktor (untuk cek prima)

Mengecek jumlah faktor

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

count digunakan untuk menghitung jumlah faktor.

Jika sebuah bilangan hanya punya 2 faktor, yaitu 1 dan dirinya sendiri bilangan prima.

Cek Bilangan Prima

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

Jika faktor = 2 bilangan prima

Jika faktor lebih dari 2 bukan prima

Contoh: b = 12, Jumlah faktor = 6 Prima: false

b = 7, Jumlah faktor = 2 Prima: true