

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 10

ELSE-IF



Disusun oleh:

BAGUS IRSYAD KAMAL

109082500215

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 kk bool

    fmt.Scan(&usia, &kk)

    if usia >= 17 && kk {

        fmt.Println("bisa membuat KTP")

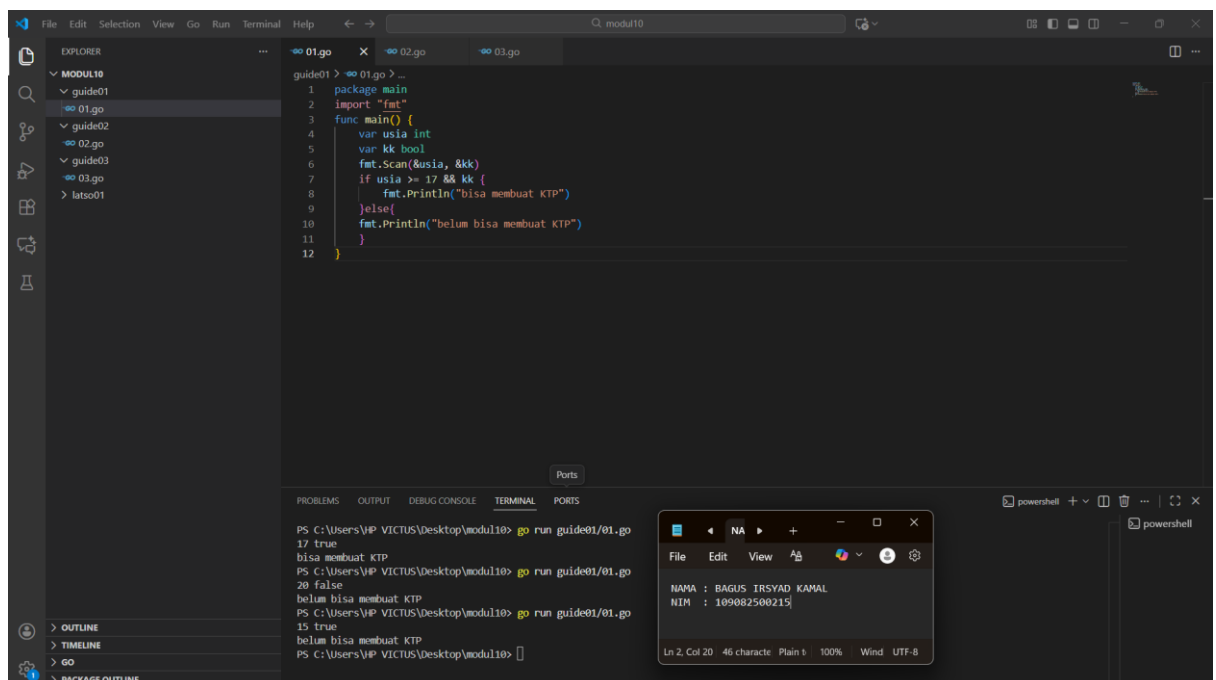
    }else{

        fmt.Println("belum bisa membuat KTP")

    }

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menentukan apakah seorang penduduk **bisa membuat KTP** atau **belum bisa membuat KTP**.

Untuk bisa membuat KTP, seseorang harus memenuhi **dua syarat utama**:

1. **Berusia minimal 17 tahun**
2. **Memiliki kartu keluarga (KK)**

Program membaca **dua input** dari pengguna.

- Input pertama adalah **bilangan bulat** yang menyatakan **usia**.
- Input kedua adalah **nilai boolean** (true atau false) yang menyatakan apakah pengguna **punya KK atau tidak**.

Setelah data dibaca, program melakukan pengecekan menggunakan struktur **if**:

- Jika usia ≥ 17 dan memiliki **KK = true**, maka program menampilkan "bisa membuat KTP".
- Jika salah satu syarat tidak terpenuhi, maka program menampilkan "belum bisa membuat KTP".

Tujuan program ini adalah membantu memutuskan apakah seseorang sudah memenuhi syarat administrasi untuk membuat KTP berdasarkan **usia dan kepemilikan KK**.

Contoh

Input:

17

true

Karena usia 17 sudah memenuhi syarat dan memiliki KK (true), maka output:

"bisa membuat KTP"

Input:

20

false

Walaupun usia cukup, tetapi tidak memiliki KK.

Output:

"belum bisa membuat KTP"

Input:

15

true

Memiliki KK, tetapi belum berusia 17.

Output:

"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")

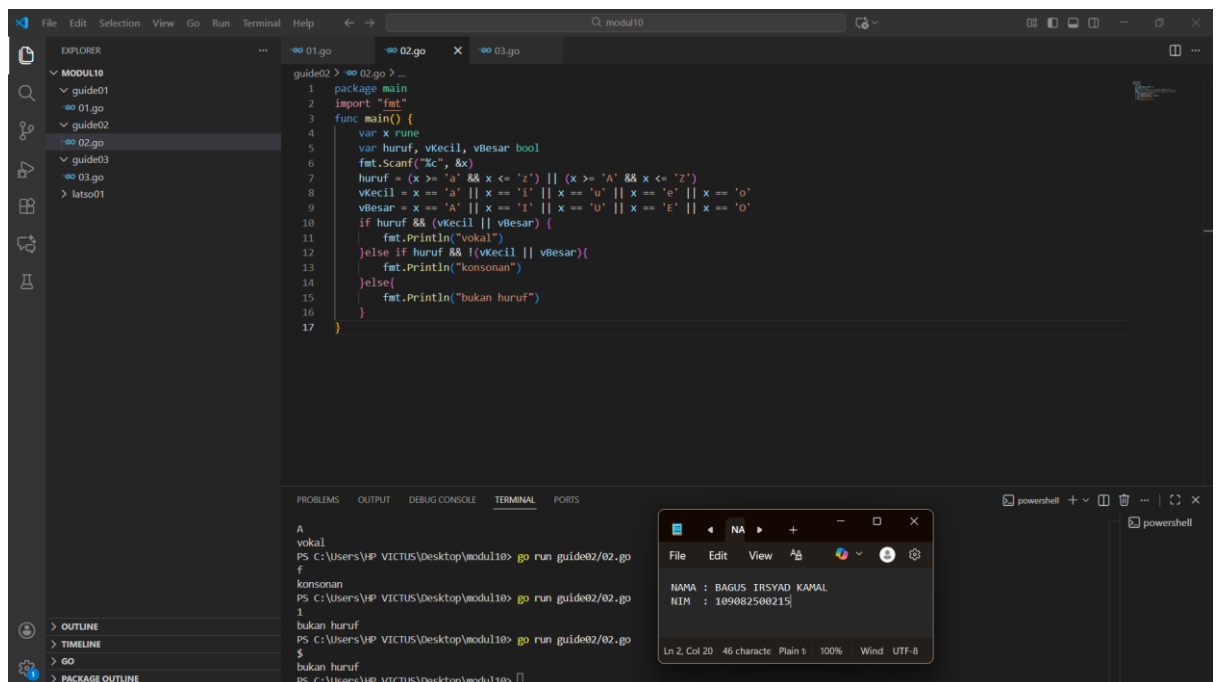
    }else{

        fmt.Println("bukan huruf")

    }

}
```

Screenshoot program



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

Terminal output:

```
A
vokal
PS C:\Users\HP VICTUS\Desktop\modul10> go run guide02/02.go
f
konsonan
PS C:\Users\HP VICTUS\Desktop\modul10> go run guide02/02.go
1
bukan huruf
PS C:\Users\HP VICTUS\Desktop\modul10> go run guide02/02.go
$
bukan huruf
PS C:\Users\HP VICTUS\Desktop\modul10>
```

Deskripsi program

Program ini digunakan untuk menentukan apakah sebuah karakter yang dimasukkan oleh pengguna merupakan **huruf vokal**, **huruf konsonan**, atau **bukan huruf**.

Program menerima **satu input**, yaitu sebuah karakter (misalnya A, f, 3, \$, dan sebagainya). Pertama, program menyiapkan variabel bertipe **rune** untuk menyimpan karakter input.

Program juga menyiapkan beberapa variabel boolean untuk pengecekan, yaitu:

- **huruf** → menandai apakah karakter merupakan huruf alfabet (A–Z atau a–z).
- **vKecil** → menandai apakah karakter termasuk vokal huruf kecil (a, i, u, e, o).
- **vBesar** → menandai apakah karakter termasuk vokal huruf besar (A, I, U, E, O).

Setelah pengguna memasukkan sebuah karakter, program melakukan pengecekan:

1. Menentukan apakah karakter adalah huruf

Program membandingkan karakter dengan rentang alfabet kecil ('a'–'z') dan besar ('A'–'Z').

Jika memenuhi salah satu, maka variabel huruf = true.

2. Menentukan apakah huruf tersebut vokal atau bukan

Program memeriksa apakah karakter termasuk salah satu dari:

- Vokal kecil: a, i, u, e, o
- Vokal besar: A, I, U, E, O

3. Menampilkan hasil

- Jika **huruf** dan **vokal**, maka output: "vokal"
- Jika **huruf** dan **bukan vokal**, maka output: "konsonan"
- Jika karakter **bukan huruf**, maka output: "bukan huruf"

Dengan demikian, program mampu mengklasifikasikan input menjadi tiga kategori: vokal, konsonan, atau bukan huruf.

Contoh :

Input:

A

A adalah huruf alfabet dan termasuk huruf vokal.

Output: "**vokal**"

Input:

f

f adalah huruf alfabet, tetapi bukan vokal.

Output: "**konsonan**"

Input:

1

1 bukan huruf alfabet.

Output: "**bukan huruf**"

Input:

\$

\$ bukan huruf alfabet.

Output: "**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) / 10
    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

The screenshot shows a Go IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code in the editor is as follows:

```
1 package main
2 import "fmt"
3 func main() {
4     var bilangan, d1, d2, d3, d4 int
5     var teks string
6     fmt.Print("Bilangan: ")
7     fmt.Scan(&bilangan)
8     d4 = bilangan % 10
9     d3 = (bilangan % 100) / 10
10    d2 = (bilangan % 1000) / 100
11    d1 = bilangan / 1000
12    if d1 < d2 && d2 < d3 && d3 < d4 {
13        teks = "terurut membesar"
14    } else if d1 > d2 && d2 > d3 && d3 > d4 {
15        teks = "terurut mengecil"
16    } else {
17        teks = "tidak terurut"
18    }
19    fmt.Println("Digit pada bilangan", bilangan, teks)
20 }
```

The terminal output shows the program being run three times with different inputs:

```
$
bukan huruf
PS C:\Users\VP VICTUS\Desktop\modul10> go run guide03/03.go
Bilangan: 2489
Digit pada bilangan 2489 tidak terurut
PS C:\Users\VP VICTUS\Desktop\modul10> go run guide03/03.go
Bilangan: 3861
Digit pada bilangan 3861 tidak terurut
PS C:\Users\VP VICTUS\Desktop\modul10> go run guide03/03.go
Bilangan: 9651
Digit pada bilangan 9651 tidak terurut
PS C:\Users\VP VICTUS\Desktop\modul10>
```

Deskripsi program

Program meminta pengguna memasukkan **satu bilangan empat digit**. Setelah itu, program memecah bilangan tersebut menjadi digit satuan, puluhan, ratusan, dan ribuan.

Digit-digit ini kemudian dibandingkan untuk mengetahui apakah urutannya:

1. **Terurut membesar** → setiap digit berikutnya lebih besar dari digit sebelumnya.
2. **Terurut mengecil** → setiap digit berikutnya lebih kecil dari digit sebelumnya.
3. **Tidak terurut** → jika tidak sesuai dua pola di atas.

Hasilnya ditampilkan dalam bentuk teks seperti:

"Digit pada bilangan ... terurut membesar/mengecil/tidak terurut."

Contoh Input dan Output

Contoh 1

Input:

2489

Penjelasan: digit 2 → 4 → 8 → 9 (urutan naik tidak beraturan)

Output:

“Digit pada bilangan 2489 tidak terurut”

Contoh 2

Input:

1357

Penjelasan: 1 < 3 < 5 < 7

Output:

“Digit pada bilangan 1357 terurut membesar”

Contoh 3**Input:**

9641

Penjelasan: $9 > 6 > 4 > 1$

Output:

“Digit pada bilangan 9641 terurut mengecil”

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var gram int

    var kg, sisa int

    var biayaKg, biayaSisa, total int


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

    fmt.Scan(&gram)


    kg = gram / 1000

    sisa = gram % 1000

    biayaKg = kg * 10000


    if kg > 10 {

        biayaSisa = 0

    } else {

        if sisa <= 500 {

            biayaSisa = sisa * 5

        } else {

            biayaSisa = sisa * 15

        }

    }

}
```

```

    }

    total = biayaKg + biayaSisa

    fmt.Println("Detail berat:", kg, "kg +", sisa, "gr")

    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.",
biayaSisa)

    fmt.Println("Total biaya: Rp.", total)

}

```

Screenshoot program

The screenshot shows a Go IDE with the following components:

- EXPLORER:** A sidebar on the left showing a project structure with files like `guide01.go`, `guide02.go`, `guide03.go`, `latso01.go`, and `latso01.go`.
- EDITOR:** The main area displaying the source code of `latso01.go`. The code defines a `main` function that takes a weight in grams, calculates the cost based on weight and remaining grams, and prints the details and total cost.
- TERMINAL:** A window at the bottom showing the output of the program when executed. It displays the total cost (Rp. 82500), the weight in grams (9250), the breakdown of costs (9 kg at Rp. 90000 and 250 gr at Rp. 1250), and the final total (Rp. 91250).
- DEBUG CONSOLE:** A window on the right showing the execution of the program, including the input of the weight (11750) and the resulting output.

Deskripsi program

Program ini digunakan untuk menghitung **biaya pengiriman parcel** berdasarkan berat dalam gram. Berat parcel dihitung dalam dua komponen, yaitu **kilogram** dan **sisa gram**.

Biaya pengiriman ditentukan berdasarkan aturan:

- Setiap **1 kg** dikenakan biaya **Rp 10.000**.
- Untuk sisa gram:
 - Jika sisa ≤ 500 gr, biaya tambahan **Rp 5 per gram**.
 - Jika sisa > 500 gr, biaya tambahan **Rp 15 per gram**.

- Jika total berat lebih dari **10 kg**, maka **biaya sisa gram digratiskan**.
Setelah pengguna memasukkan berat dalam gram, program menghitung kilogram, sisa gram, biaya per kg, biaya sisa, dan total biaya.
Terakhir, program menampilkan detail perhitungan dan total biaya pengiriman.

Contoh Input & Output

Contoh 1

Input:

8500

Proses:

- $8500 \text{ gr} \rightarrow 8 \text{ kg} + 500 \text{ gr}$
- Biaya kg: $8 \times 10.000 = 80.000$
- Sisa $\leq 500 \rightarrow 500 \times 5 = 2.500$

Output:

Total biaya: **82.500**

Contoh 2

Input:

9250

Proses:

- $9250 \text{ gr} \rightarrow 9 \text{ kg} + 250 \text{ gr}$
- Biaya kg: $9 \times 10.000 = 90.000$
- Sisa $\leq 500 \rightarrow 250 \times 5 = 3.750$

Output:

Total biaya: **93.750**

Contoh 3

Input:

11750

Proses:

- $11750 \text{ gr} \rightarrow 11 \text{ kg} + 750 \text{ gr}$
- Karena $\text{kg} > 10$, biaya sisa **gratis**
- Biaya kg: $11 \times 10.000 = 110.000$

Output:

Total biaya: **110.000**

2. Tugas 2

a. Jika NAM = 80.1, apa output program di gambar?

logikanya:

if nam > 80 → AB

Jadi output program salah adalah:

NMK = AB

Padahal menurut tabel:

$72.5 < \text{NAM} \leq 88 \rightarrow \text{AB}$

Jadi untuk 80.1 memang AB sebenarnya benar.

Tetapi program tetap salah karena urutan pengecekan tidak sesuai untuk nilai lain.

b. Apa saja kesalahan program?

Ringkas:

1. Urutan pengecekan rentang nilai salah.
2. Beberapa batas ($<$, \leq) tidak sesuai tabel.
3. Nilai 88–100 tidak ditangani dengan benar untuk grade A.
4. Struktur logika tidak mengikuti spesifikasi soal.

Source code

```
package main

import "fmt"

func main() {

    var nam float64

    var nmk string

    fmt.Print("Masukkan nilai akhir mata kuliah (NAM):")
    "\n"

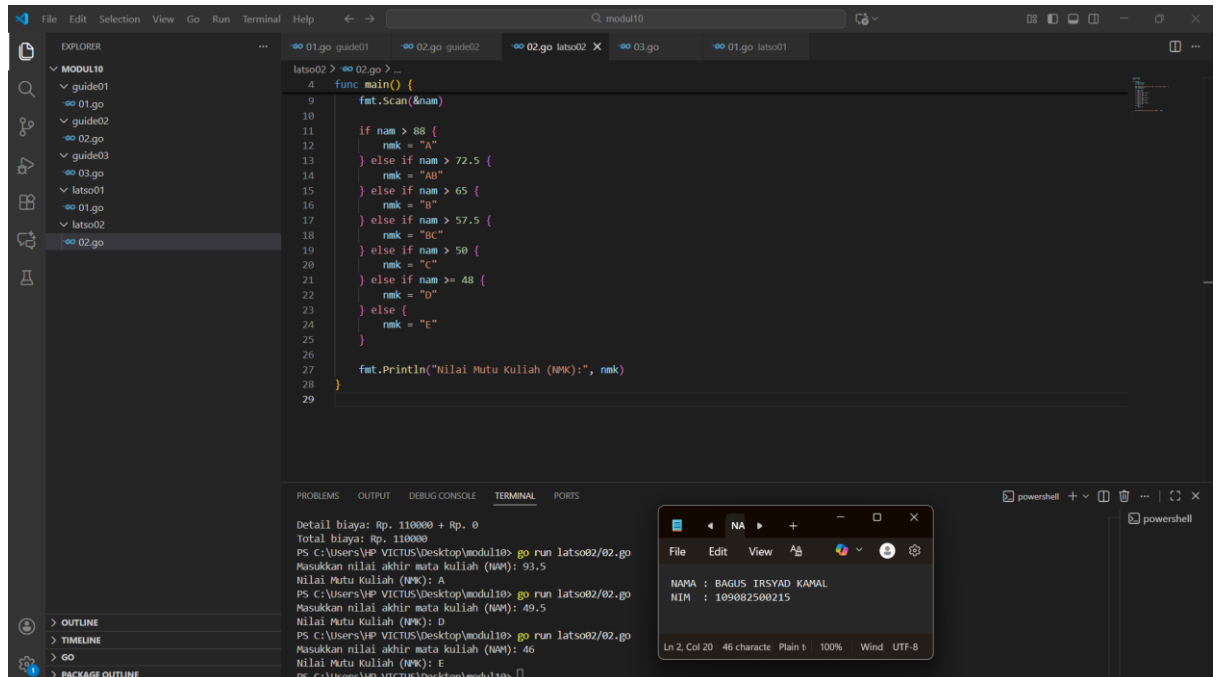
    fmt.Scan(&nam)

    if nam > 88 {

        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 >= 48 {  
            nmk = "D"  
        } else {  
            nmk = "E"  
        }  
  
        fmt.Println("Nilai Mutu Kuliah (NMK):", nmk)  
    }
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menentukan **Nilai Mutu Kuliah (NMK)** berdasarkan **Nilai Akhir Mata Kuliah (NAM)** yang diinput oleh pengguna. Nilai angka (0–100) akan dikonversi menjadi nilai huruf berdasarkan standar penilaian yang telah ditentukan.

2. Cara Kerja Program

1. Program meminta pengguna untuk memasukkan **nilai akhir mata kuliah (NAM)**.
2. Setelah nilai dimasukkan, program membandingkan nilai tersebut dengan interval nilai yang tersedia.
3. Berdasarkan interval yang cocok, program menentukan **nilai huruf (NMK)**.
4. Program menampilkan hasil akhir berupa nilai mutu (NMK).

3. Aturan Penentuan Nilai Mutu (NMK)

Program menggunakan aturan berikut untuk menentukan nilai huruf:

Rentang NAM	NMK
$NAM > 88$	A
$72.5 \leq NAM \leq 88$	AB
$65 \leq NAM < 72.5$	B
$57.5 \leq NAM < 65$	BC

Rentang NAM	NMK
$50 \leq \text{NAM} < 57.5$	C
$48 \leq \text{NAM} < 50$	D
$\text{NAM} < 48$	E

4. Alur Logika Program

- Jika nilai lebih dari 88 \rightarrow hasil A
- Jika nilai tidak lebih dari 88 tetapi $\geq 72.5 \rightarrow$ hasil AB
- Jika nilai tidak memenuhi syarat AB tetapi $\geq 65 \rightarrow$ hasil B
- Jika nilai tidak memenuhi syarat B tetapi $\geq 57.5 \rightarrow$ hasil BC
- Jika nilai tidak memenuhi syarat BC tetapi $\geq 50 \rightarrow$ hasil C
- Jika nilai tidak memenuhi syarat C tetapi $\geq 48 \rightarrow$ hasil D
- Jika nilai lebih kecil dari 48 \rightarrow hasil E

Program mengevaluasi kondisi secara berurutan dari nilai tertinggi ke terendah.

Tugas 3

Source code

```
package main

import "fmt"

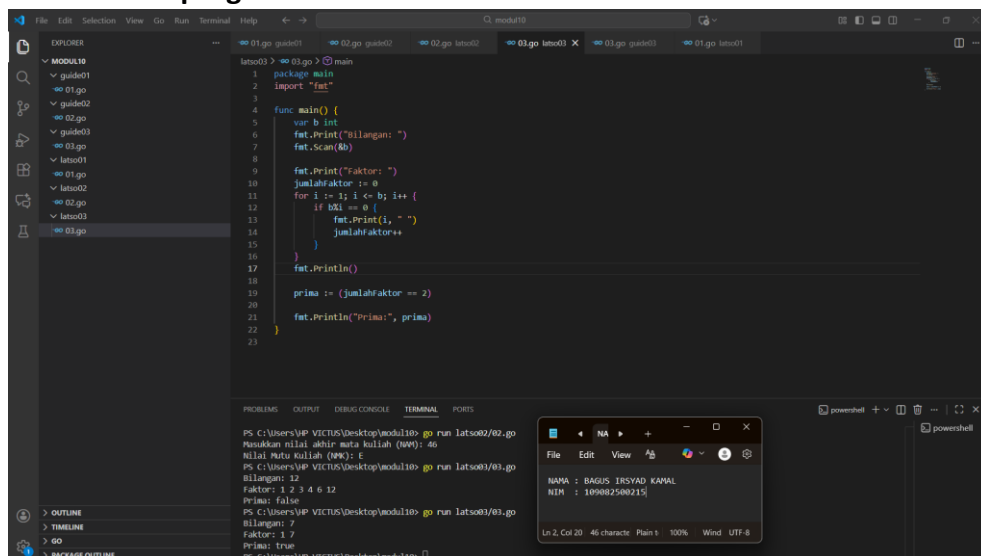
func main() {
    var b int
    fmt.Print("Bilangan: ")
    fmt.Scan(&b)

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

    prima := (jumlahFaktor == 2)

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

Screenshoot program



The screenshot shows the Go program being executed in a terminal window within VS Code. The code is as follows:

```
1 package main
2 import "fmt"
3
4 func main() {
5     var b int
6     fmt.Print("Bilangan: ")
7     fmt.Scan(&b)
8
9     fmt.Print("Faktor: ")
10    jumlahFaktor := 0
11    for i := 1; i <= b; i++ {
12        if b%i == 0 {
13            fmt.Print(i, " ")
14            jumlahFaktor++
15        }
16    }
17    fmt.Println()
18    prima := (jumlahFaktor == 2)
19    fmt.Println("Prima:", prima)
20 }
```

The terminal output shows the program running successfully with the input 46:

```
PS C:\Users\VP VICTUS\Desktop\modul10> go run latso02/02.go
Masukkan nilai akhir mata kuliah (NKK): 46
Bilangan: 46
Faktor: 1 2 23 46 12
Prima: false
PS C:\Users\VP VICTUS\Desktop\modul10> go run latso03/03.go
Bilangan: 7
Faktor: 1 7
Prima: true
PS C:\Users\VP VICTUS\Desktop\modul10>
```

Deskripsi program

Program ini digunakan untuk:

1. **Menerima input sebuah bilangan bulat b (>1)**
2. **Menampilkan semua faktor dari bilangan tersebut**
3. **Menentukan apakah bilangan tersebut bilangan prima atau bukan**

Penjelasan di bawah mengikuti alur contoh pada gambar modul.

1. Penginputan

Program meminta pengguna memasukkan sebuah bilangan bulat.

Contoh:

Bilangan: 12

2. Mencari Faktor

Sebuah bilangan bulat **f adalah faktor dari b** apabila:

b habis dibagi f (sisa = 0)

Program mengecek semua angka dari **1 sampai b** , dan setiap angka yang dapat membagi b tanpa sisa akan ditampilkan sebagai faktor.

Contoh 1

Jika $b = 12$, maka pengecekan:

- $12 \% 1 = 0 \rightarrow$ faktor
- $12 \% 2 = 0 \rightarrow$ faktor
- $12 \% 3 = 0 \rightarrow$ faktor
- $12 \% 4 = 0 \rightarrow$ faktor
- $12 \% 5 \neq 0 \rightarrow$ bukan faktor
- $12 \% 6 = 0 \rightarrow$ faktor
- $12 \% 12 = 0 \rightarrow$ faktor

Maka faktor 12 adalah:

1 2 3 4 6 12

Contoh 2

Jika $b = 7$, pengecekan:

- $7 \% 1 = 0 \rightarrow$ faktor
- $7 \% 7 = 0 \rightarrow$ faktor

Maka faktor 7 adalah:

1 7

3. Menentukan Apakah Bilangan Prima

Sebuah bilangan dikatakan **prima** jika memiliki **tepat dua faktor**, yaitu:

1. angka 1
2. dirinya sendiri

Program menghitung **jumlah faktor** yang ditemukan.

- Jika **jumlah faktor = 2**, maka **prima = true**

- Jika **lebih dari 2**, maka **prima = false**

Contoh 1

b = 12

Jumlah faktor: 6 → **Prima: false**

Contoh 2

b = 7

Jumlah faktor: 2 → **Prima: true**

4. Contoh Interaksi Program (seperti modul)

Contoh 1

Input:

Bilangan: **12**

Output:

Faktor: 1 2 3 4 6 12

Prima: **false**

Contoh 2

Input:

Bilangan: **7**

Output:

Faktor: 1 7

Prima: **true**