

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 10

ELSE-IF



Disusun oleh:

MUHAMMAD FIRDAUS ARDIANSYAH

109082500126

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

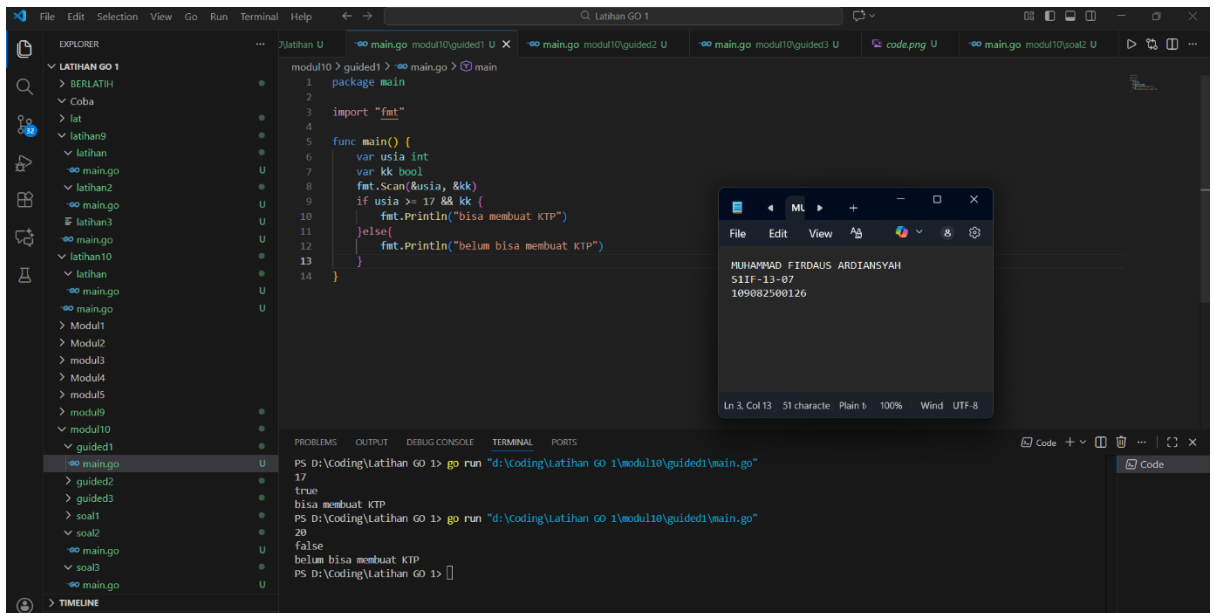
1. Guided 1 Source Code

```
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 berfungsi untuk memvalidasi kelayakan seseorang dalam membuat Kartu Tanda Penduduk (KTP). Di dalam fungsi main, dideklarasikan dua variabel input: usia dengan tipe data *integer* (bilangan bulat) dan kk dengan tipe data *boolean* (benar/salah). Program menggunakan fungsi `fmt.Scan` untuk menerima input langsung dari pengguna melalui terminal, yang kemudian disimpan ke dalam alamat memori kedua variabel tersebut.

Logika penentu keputusan menggunakan struktur percabangan `if-else` dengan operator logika **AND** (`&&`). Agar program menampilkan pesan "bisa membuat KTP", dua kondisi harus terpenuhi secara bersamaan: nilai variabel usia harus minimal 17 (`>= 17`) **dan** nilai variabel kk harus bernilai `true`. Jika salah satu atau kedua kondisi tersebut tidak terpenuhi (seperti pada contoh kedua di terminal di mana kk bernilai `false`), program akan mengeksekusi blok `else` dan mencetak output "belum bisa membuat KTP".

2. Guided 2

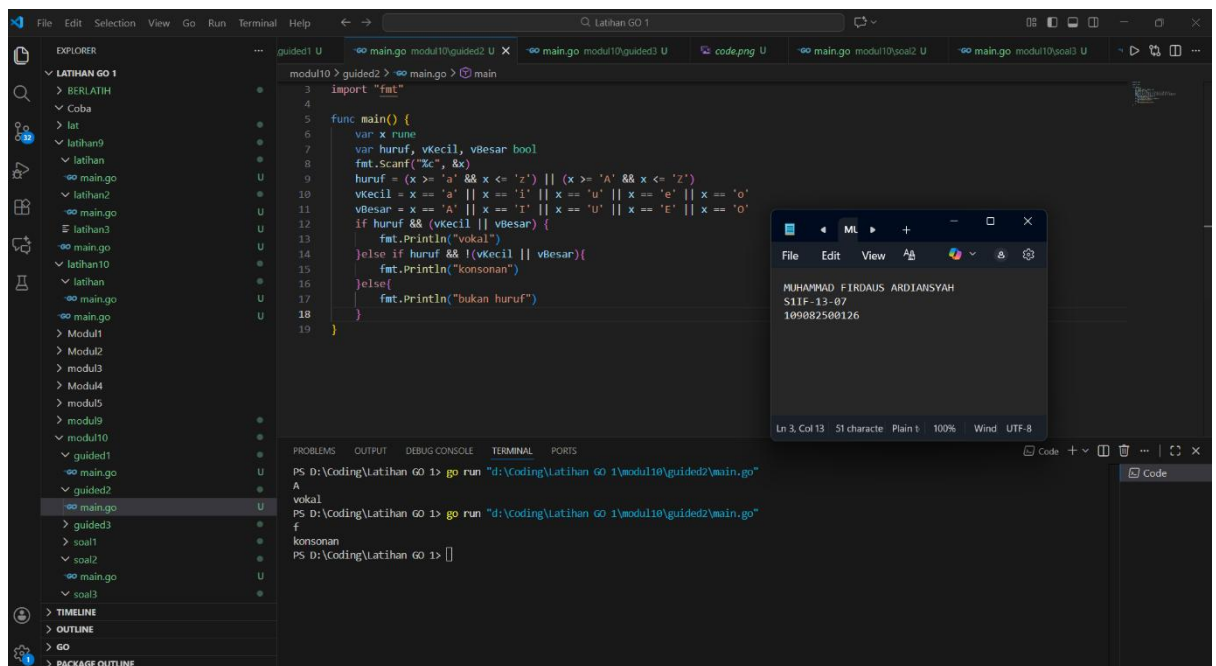
Source Code

```
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



Deskripsi program

Program ini berfungsi untuk mengklasifikasikan karakter input menjadi huruf vokal, konsonan, atau bukan huruf. Program mendeklarasikan variabel *x* dengan tipe data *rune* untuk menampung karakter tunggal, serta beberapa variabel *boolean* (*huruf*, *vKecil*, *vBesar*) sebagai penanda (*flag*). Setelah menerima input melalui *fmt.Scanf*, program mengevaluasi apakah karakter tersebut termasuk dalam rentang abjad (a-z atau A-Z) dan apakah karakter tersebut merupakan huruf vokal spesifik, baik huruf kecil maupun kapital.

Logika penentu output menggunakan struktur percabangan *if-else if-else* yang menggabungkan hasil evaluasi boolean tersebut. Jika karakter teridentifikasi sebagai huruf dan merupakan vokal (salah satu dari *vKecil* atau *vBesar*), program mencetak "vokal". Sebaliknya, jika karakter adalah huruf tetapi bukan vokal (menggunakan operator negasi *!*), program mencetak "konsonan". Jika karakter input tidak memenuhi kriteria sebagai huruf (misalnya angka atau simbol), program akan menjalankan blok terakhir dan mencetak "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{

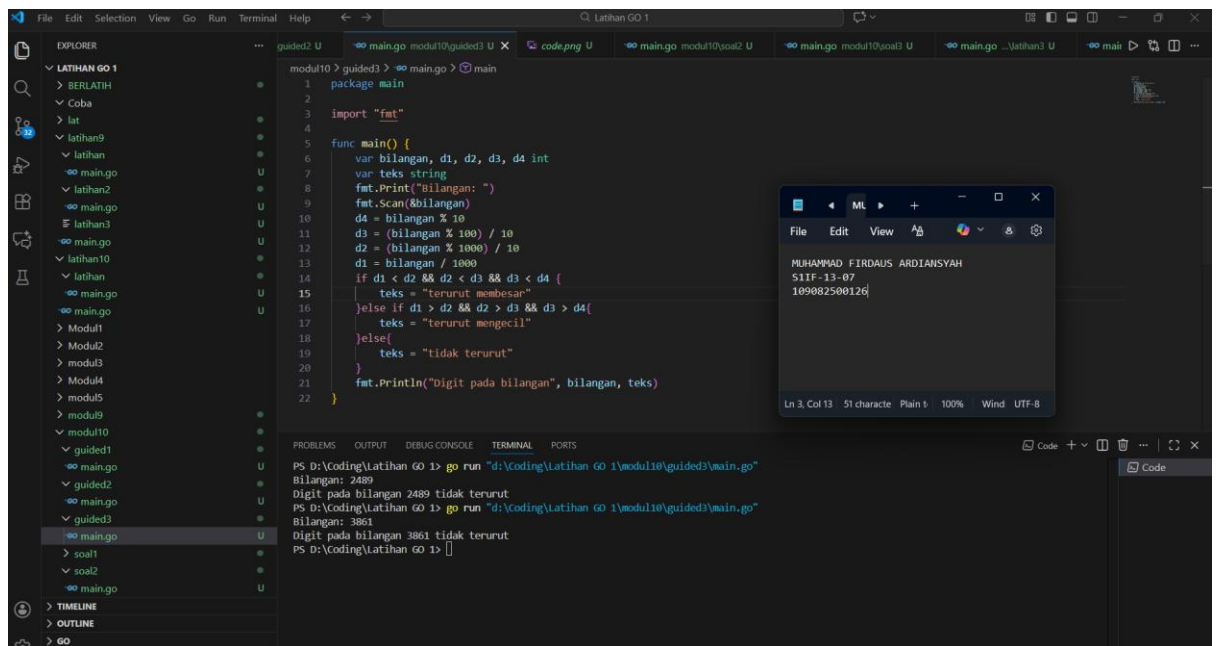
    teks = "tidak terurut"

}

fmt.Println("Digit pada bilangan", bilangan, teks)

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menganalisis pola urutan digit pada sebuah bilangan bulat empat digit (maksimal 9999). Program menerima satu input *integer*, lalu memecah bilangan tersebut menjadi digit-digit penyusunnya (ribuan, ratusan, puluhan, dan satuan) menggunakan operasi aritmatika. Operasi modulus (%) digunakan untuk mengambil sisa bagi (mendapatkan digit terakhir), sementara pembagian integer (/) digunakan untuk membuang digit belakang, sehingga nilai d1 hingga d4 dapat merepresentasikan setiap posisi digit dari bilangan tersebut.

Setelah digit-digit berhasil dipisahkan, program melakukan evaluasi logika menggunakan struktur percabangan if-else. Program memeriksa apakah digit-digit tersebut memiliki urutan nilai yang naik secara konsisten ($d1 < d2 < d3 < d4$) untuk menetapkan status "terurut membesar", atau turun secara konsisten ($d1 > d2 > d3 > d4$) untuk status "terurut mengecil". Jika digit-digit tersebut acak atau memiliki angka yang sama sehingga tidak memenuhi kedua pola tersebut, program menetapkan status "tidak terurut" dan menampilkan hasilnya ke layar.

TUGAS

1. Tugas 1

Source code

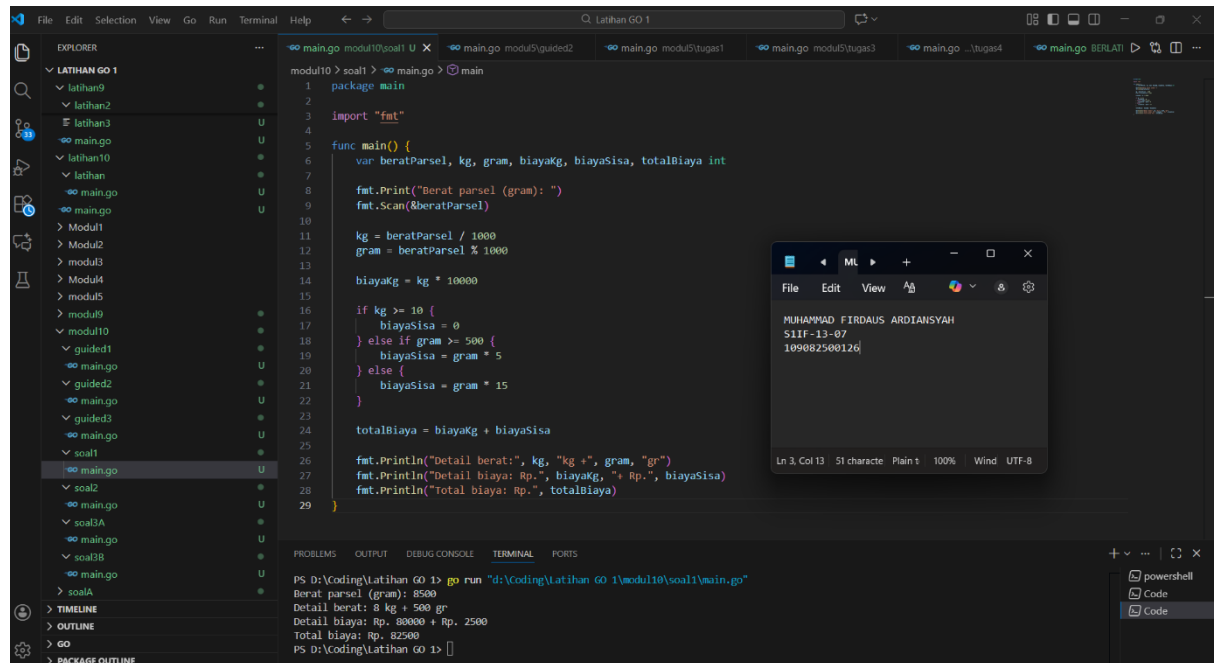
```
package main

import "fmt"
```

```
func main() {  
  
    var beratParsel, kg, gram, biayaKg, biayaSisa,  
    totalBiaya int  
  
    fmt.Print("Berat parsel (gram): ")  
    fmt.Scan(&beratParsel)  
  
    kg = beratParsel / 1000  
    gram = beratParsel % 1000  
  
    biayaKg = kg * 10000  
  
    if kg >= 10 {  
        biayaSisa = 0  
    } else if gram >= 500 {  
        biayaSisa = gram * 5  
    } else {  
        biayaSisa = gram * 15  
    }  
  
    totalBiaya = biayaKg + biayaSisa  
  
    fmt.Println("Detail berat:", kg, "kg +", gram,  
    "gr")  
  
    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.",  
    biayaSisa)  
  
    fmt.Println("Total biaya: Rp.", totalBiaya)
```




Screenshoot program



Deskripsi program

Program ini menerima input berat total parsel dalam gram, kemudian memecahnya menjadi komponen kilogram dan sisa gram menggunakan operasi pembagian integer dan modulus (%). Biaya dasar pengiriman tetap dihitung berdasarkan berat kilogram mutlak dikalikan dengan tarif Rp. 10.000.

Perubahan signifikan terdapat pada logika penentuan biaya tambahan (biayaSisa) menggunakan struktur if-else. Program kini memiliki aturan prioritas: jika berat paket mencapai 10 kg atau lebih, biaya sisa gram diabaikan (di-set menjadi 0). Jika beratnya kurang dari 10 kg, barulah aturan berjenjang diterapkan: sisa gram ≥ 500 dikenakan tarif Rp. 5 per gram, sedangkan sisa di bawah 500 gram dikenakan tarif lebih mahal, yaitu Rp. 15 per gram.

2. Tugas 2

Jawaban Program

a. Jika nam = 80.1, apa keluarannya? Apakah sesuai spesifikasi soal?

Tidak Sesuai. Karena program seharusnya menghasilkan nilai huruf, tetapi:

- Program ini salah tipe data (mengisi string ke variabel float).
- Program ini tidak memakai else if.
- Nilai huruf ditulis ke variabel nam, bukan nmk.
- Variabel nmk selalu kosong → output tidak benar.
- Logika kondisi tumpang tindih → hasil tidak pernah benar

b. Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!

1. Kesalahan tipe data

nam bertipe float64, tetapi diisi string seperti "A", "AB" → **error** dan program tidak bisa dikompilasi.

2. Variabel yang salah digunakan

- Variabel yang benar untuk nilai huruf adalah nm, tetapi program menulis huruf ke nam.
- Karena itu nm tetap kosong → output tidak ada nilai huruf.

3. Semua kondisi menggunakan if terpisah

4. Rentang kondisi tumpang tindih

nam > 80, nam > 72.5, nam > 65 semua overlap.

5. Struktur if–else if–else yang salah

6. Output tidak sesuai format soal

Soal ingin huruf nilai tampil, tetapi variabel yang dicetak (nm) kosong.

c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

Source code

```
package main

import "fmt"

func main() {

    var nam float64

    var nm string

    fmt.Print("Nilai akhir mata kuliah: ")

    fmt.Scan(&nam)

    if nam > 80 {

        nm = "A"

    } else if nam > 72.5 {

        nm = "AB"

    } else if nam > 65 {

        nm = "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"

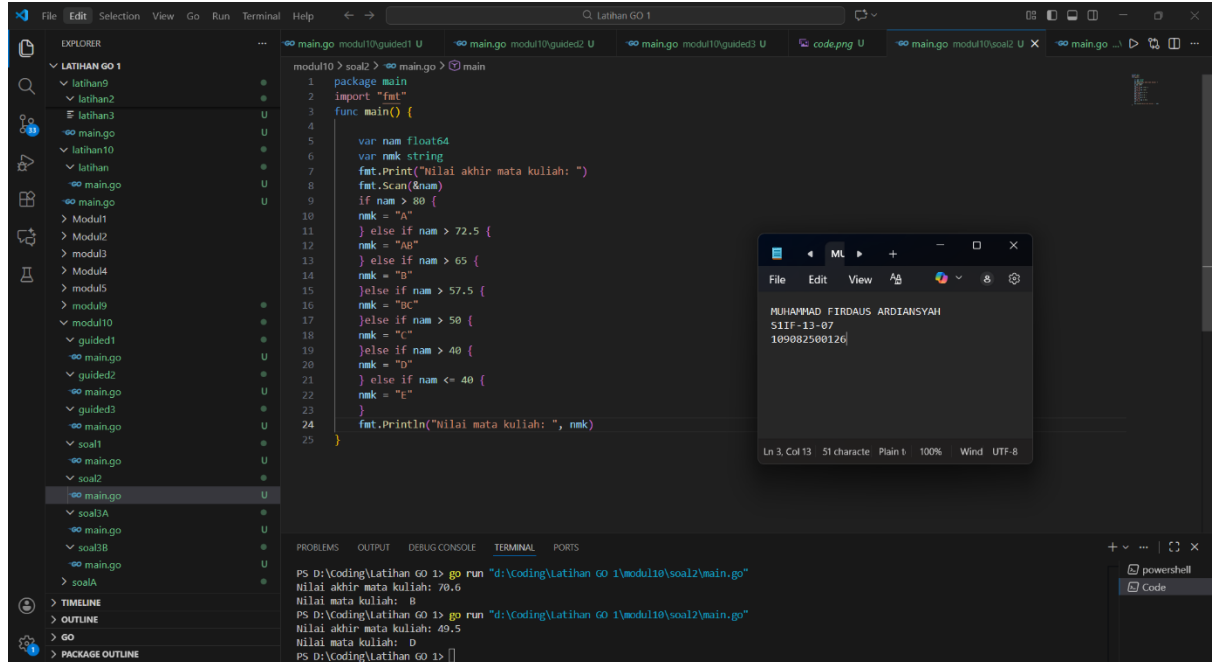
    }

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

}

```

Screenshoot program



Deskripsi program

Fungsi program ini adalah menerjemahkan nilai angka (Nilai Akhir Mata Kuliah) menjadi nilai huruf atau indeks prestasi. Program menggunakan variabel `nam` dengan

tipe data float64 untuk mengakomodasi input angka desimal yang presisi, serta variabel nmk bertipe string untuk menyimpan hasil konversi hurufnya. Input diambil langsung dari pengguna melalui fungsi fmt.Scan.

Logika konversi diimplementasikan menggunakan struktur percabangan bertingkat if-else if. Program memeriksa nilai input secara berurutan mulai dari batas tertinggi (di atas 80) hingga batas terendah. Jika sebuah kondisi terpenuhi (misalnya nilai > 72.5), program langsung menetapkan huruf mutu yang sesuai (dalam hal ini "AB") dan mengabaikan kondisi di bawahnya. Proses ini berlanjut hingga kondisi terakhir, di mana nilai yang kurang dari atau sama dengan 40 (≤ 40) akan dikategorikan sebagai grade "E".

d. Tugas 3A

Source code

```
package main

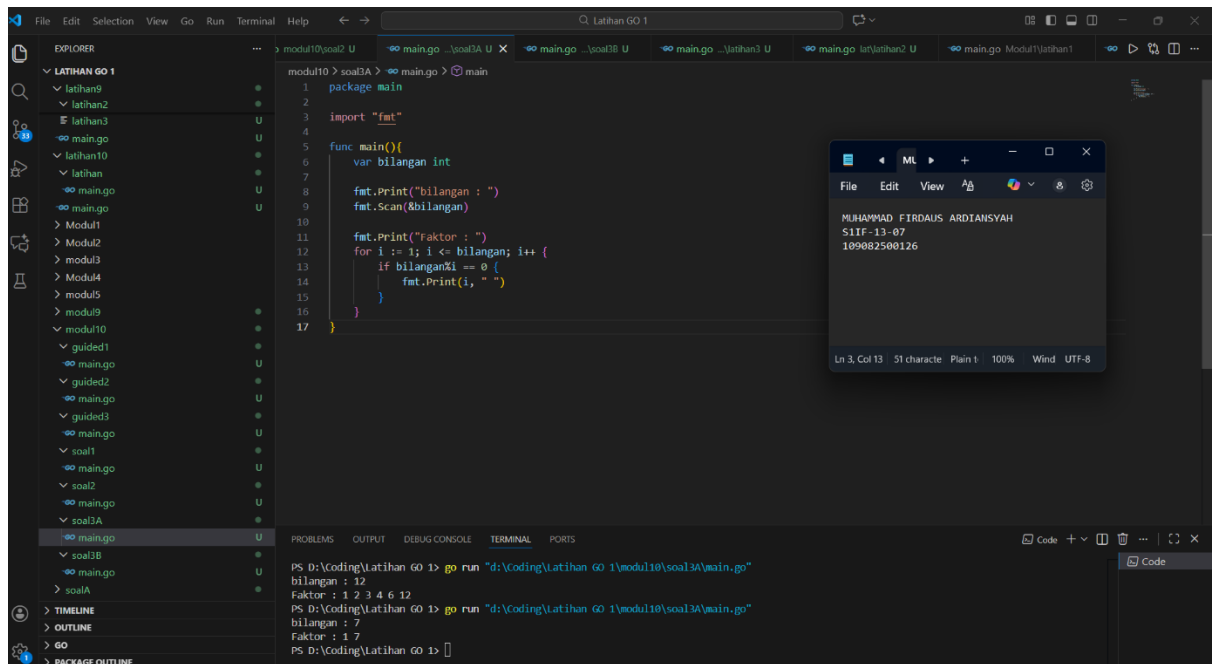
import "fmt"

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, " ")
        }
    }
}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk mencari dan menampilkan semua faktor dari sebuah bilangan bulat positif. Program mendeklarasikan variabel bilangan dengan tipe data *integer*, lalu meminta kita memasukkan sebuah angka melalui fungsi `fmt.Scan`. Angka yang diinputkan ini akan menjadi batas atas untuk proses pencarian faktor.

Logika inti program menggunakan struktur perulangan `for` yang melakukan iterasi mulai dari angka 1 hingga mencapai nilai bilangan itu sendiri. Di dalam setiap putaran loop, terdapat kondisi `if` yang menggunakan operator modulus (`%`) untuk memeriksa apakah bilangan habis dibagi oleh angka iterator saat ini (`i`). Jika hasil bagi tersebut tidak memiliki sisa (sama dengan 0), maka `i` dianggap sebagai faktor dari bilangan tersebut dan langsung dicetak ke layar.

Tugas 3B

Source code

```
package main

import "fmt"

func main() {
    var bilangan int
    var prima bool
```

```
var jumlahFaktor int = 0

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

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

fmt.Println()

if jumlahFaktor == 2 {
    prima = true
} else {
    prima = false
}

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

Screenshoot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7     var prima bool
8     var jumlahFaktor int = 0
9
10    fmt.Print("bilangan : ")
11    fmt.Scan(&bilangan)
12
13    fmt.Print("Faktor : ")
14    for i := 1; i <= bilangan; i++ {
15        if bilangan%i == 0 {
16            fmt.Print(i, " ")
17            jumlahFaktor++
18        }
19    }
20
21    fmt.Println()
22
23    if jumlahFaktor == 2 {
24        prima = true
25    } else {
26        prima = false
27    }
28
29    fmt.Println("Prima:", prima)
30 }
```

More options

MUHAMMAD FIRDAUS ARDIANSYAH
SIIP-13-07
109082500126

Ln 3, Col 13 51 character Plain text 100% Wind UTF-8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Bilangan : 12
Faktor : 1 2 3 4 6 12
Prima: false
PS D:\Coding\Latihan GO 1> go run "d:\coding\Latihan GO 1\modul10\soal3B\main.go"

Bilangan : 7
Faktor : 1 7
Prima: true

Deskripsi program

Program ini merupakan pengembangan lebih lanjut dari soal 3A yang bertujuan untuk menentukan apakah sebuah angka adalah bilangan prima atau bukan. Logika intinya masih sama, yaitu menggunakan perulangan for untuk mencari faktor-faktor bilangan. Namun, program ini menambahkan variabel jumlahFaktor yang bertugas menghitung ada berapa banyak faktor yang ditemukan. Setelah perulangan selesai, program melakukan evaluasi akhir: jika jumlahFaktor tepat berjumlah 2 (karena bilangan prima hanya bisa dibagi oleh angka 1 dan dirinya sendiri), maka variabel boolean prima akan bernilai true, dan sebaliknya false.

Hubungannya dengan kode sebelumnya adalah kode ini memanfaatkan data yang dihasilkan oleh kode sebelumnya untuk menarik kesimpulan baru. Jika program sebelumnya ("Soal 3A") hanya berfungsi untuk menampilkan (output) deretan angka faktor, program ini ("Soal 3B") memproses data tersebut lebih lanjut dengan cara menghitungnya. Kode sebelumnya adalah dasar algoritmanya (mekanisme pencarian), sedangkan kode ini adalah implementasi logika matematika untuk mengklasifikasikan bilangan tersebut berdasarkan hasil pencarian faktor tadi.