LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE IF

Disusun oleh:

M MAHDAN ARGYA SYARIF

109082500059

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

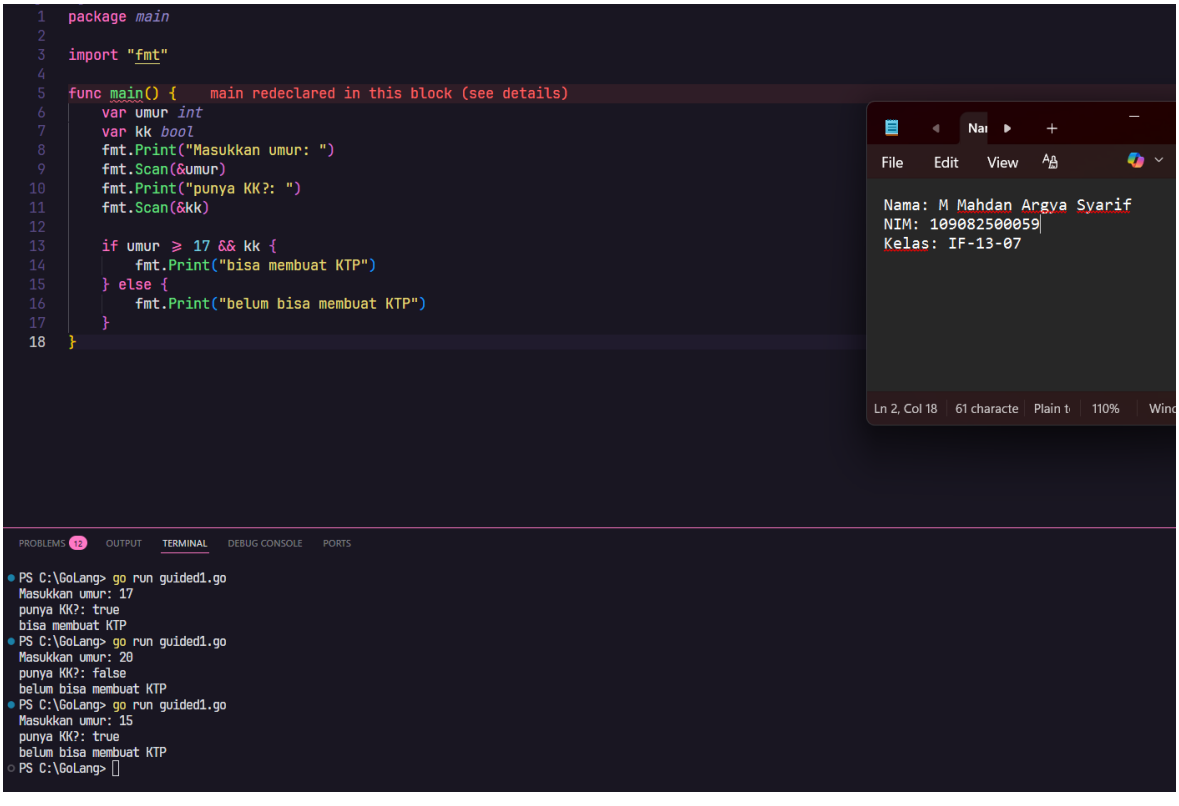
LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main


import "fmt"


func main() {
    var umur int
    var kk bool
    fmt.Print("Masukkan umur: ")
    fmt.Scan(&umur)
    fmt.Print("punya KK?: ")
    fmt.Scan(&kk)


    if umur >= 17 && kk {
        fmt.Print("bisa membuat KTP")
    } else {
        fmt.Print("belum bisa membuat KTP")
    }
}
```

Screenshoot program



Deskripsi program

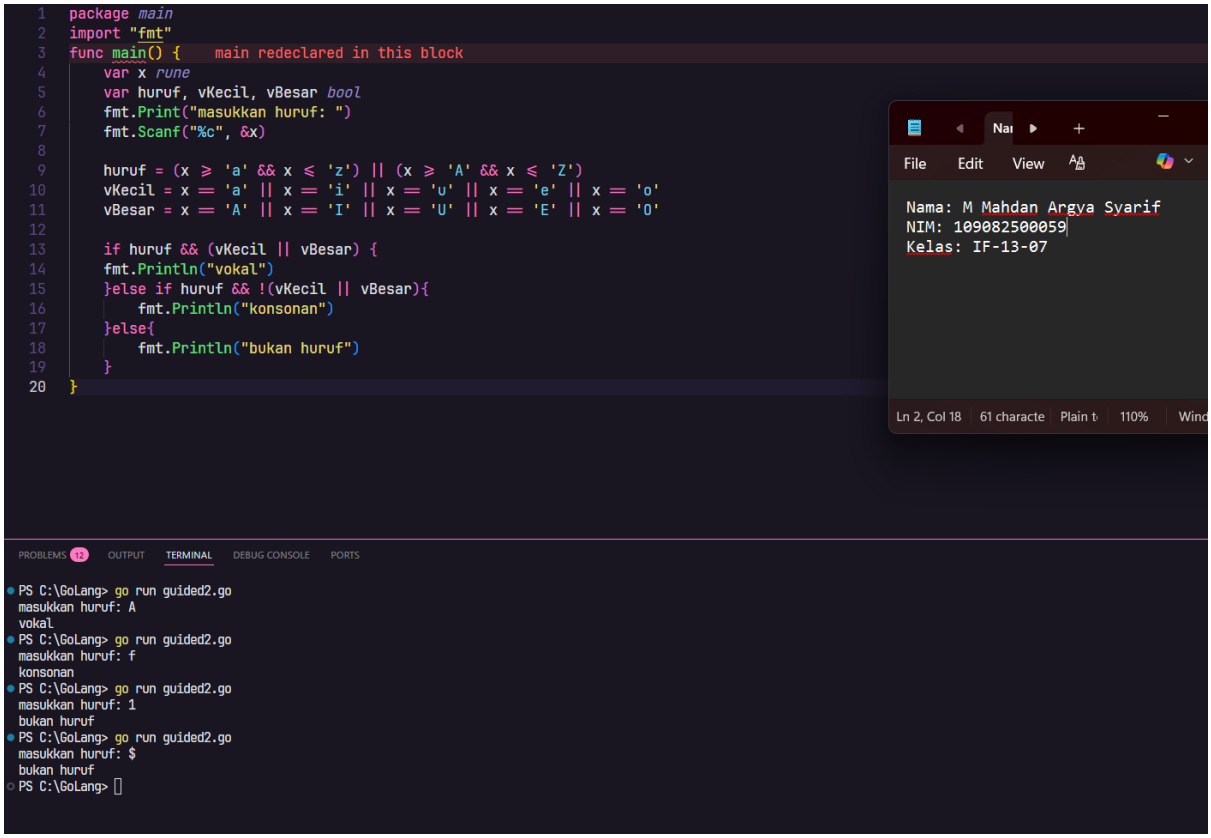
Program ini menggunakan algoritma *if-else* untuk memvalidasi kelayakan pembuatan KTP, di mana syarat mutlak agar bernilai *true* adalah pengguna harus berusia minimal 17 tahun dan memiliki Kartu Keluarga (KK). Sebaliknya, jika kondisi tersebut tidak terpenuhi secara utuh—seperti pengguna yang sudah cukup umur namun tidak memiliki KK, atau pengguna yang belum cukup umur—maka algoritma akan menganggap input tidak memenuhi syarat dan menghasilkan output *false*.

2. Guided 2

Source Code

```
package main
import "fmt"
func main() {
    var x rune
    var huruf, vKecil, vBesar bool
    fmt.Print("masukkan huruf: ")
    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 bertujuan mengklasifikasikan input karakter menjadi huruf vokal, konsonan, atau bukan huruf dengan memanfaatkan tipe data rune untuk representasi Unicode. Alur program dimulai dengan mendeklarasikan variabel boolean bantu (huruf, vKecil, vBesar) untuk memetakan rentang alfabet dan karakter vokal baik kapital maupun kecil, yang kemudian dievaluasi menggunakan struktur kendali *if-else*. Berdasarkan logika tersebut, program akan mencetak output "vokal" jika input sesuai dengan himpunan huruf A, I, U, E, O, menghasilkan "konsonan" untuk huruf alfabet lainnya, atau menampilkan "bukan huruf" jika input berupa simbol atau angka di luar definisi alfabet.

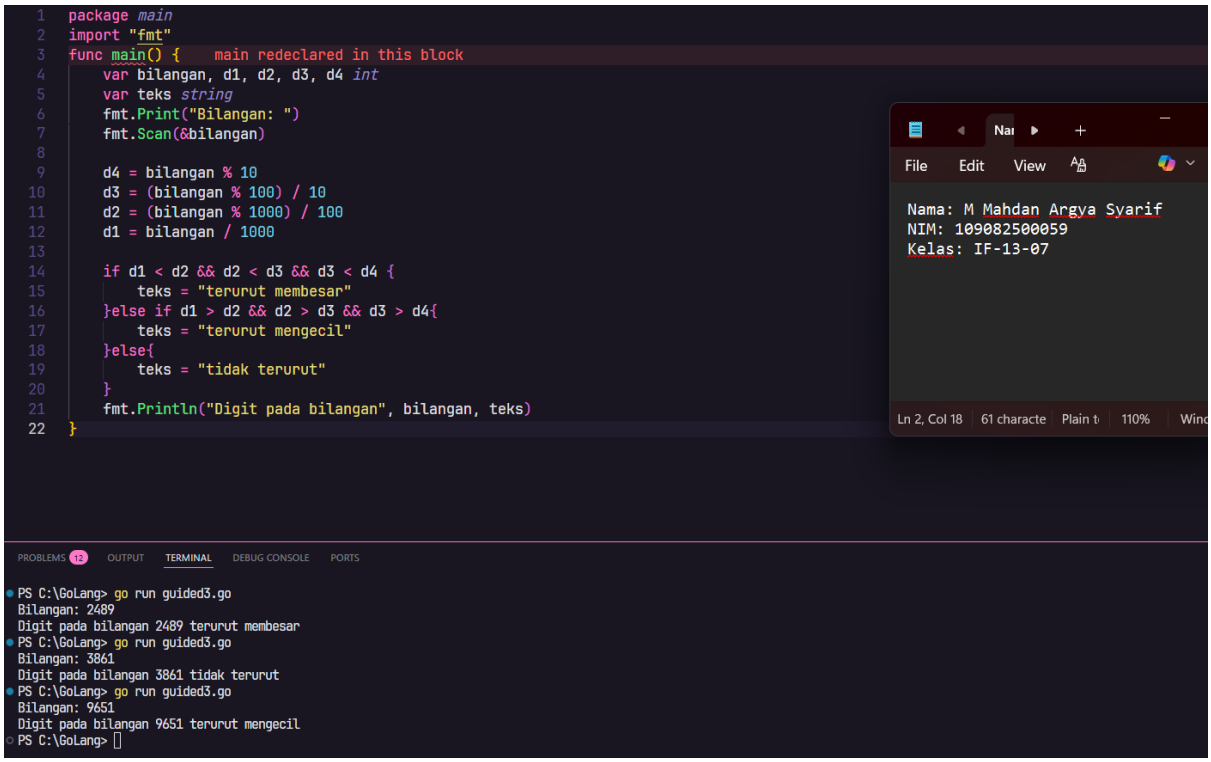
3. Guided 3

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



Deskripsi program

Program ini berfungsi untuk mengidentifikasi pola urutan pada input bilangan empat digit, menentukan apakah digit-digitnya terurut membesar, mengecil, atau tidak beraturan. Proses teknisnya dimulai dengan memecah bilangan menjadi empat variabel digit terpisah (d1, d2, d3, d4) menggunakan operasi matematika pembagian dan modulus untuk mengekstrak nilai ribuan hingga satuan. Selanjutnya, algoritma menerapkan struktur kendali kondisional untuk membandingkan relasi antar digit; jika memenuhi kondisi d1 < d2 < d3 < d4 maka statusnya terurut membesar, jika d1 > d2 > d3 > d4 maka terurut mengecil, dan jika tidak memenuhi kedua pola tersebut, maka dikategorikan sebagai tidak terurut.

TUGAS

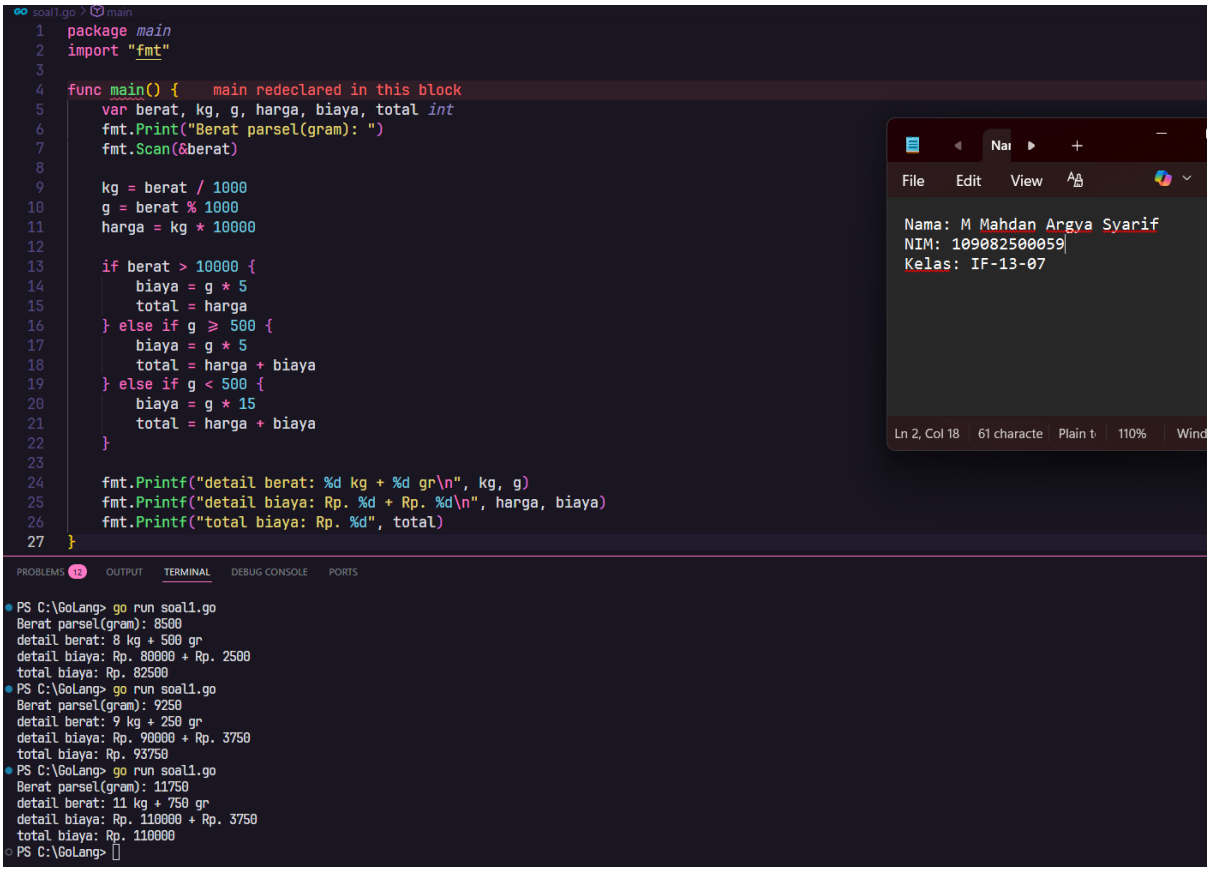
1. **Tugas 1**

   Source code

```
package main

import "fmt"


func main() {

    var berat, kg, g, harga, biaya, total int

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

    fmt.Scan(&berat)


    kg = berat / 1000

    g = berat % 1000

    harga = kg * 10000


    if berat > 10000 {

        biaya = g * 5

        total = harga

    } else if g >= 500 {

        biaya = g * 5

        total = harga + biaya

    } else if g < 500 {

        biaya = g * 15

        total = harga + biaya

    }
```

```
        fmt.Printf("detail berat: %d kg + %d gr\n", kg, g)

        fmt.Printf("detail biaya: Rp. %d + Rp. %d\n", harga,
    biaya)

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

    }
```

Screenshoot program



Deskripsi program

Program ini dirancang untuk menghitung biaya pengiriman parsel menggunakan tipe data *integer* untuk seluruh variabel operasional dan memanfaatkan pustaka fmt untuk penanganan input-output. Logika bisnisnya menetapkan tarif dasar Rp10.000 per kilogram, dengan penerapan aturan kondisional khusus untuk sisa gram: biaya tambahan digratiskan jika total berat melebihi 10 kg, namun dikenakan tarif bertingkat (Rp5 per gram jika $\ge$ 500g, atau Rp15 per gram jika < 500g) untuk berat standar. Seluruh hasil kalkulasi, mulai dari rincian berat hingga total harga akhir, kemudian ditampilkan secara terstruktur dan terperinci kepada pengguna.

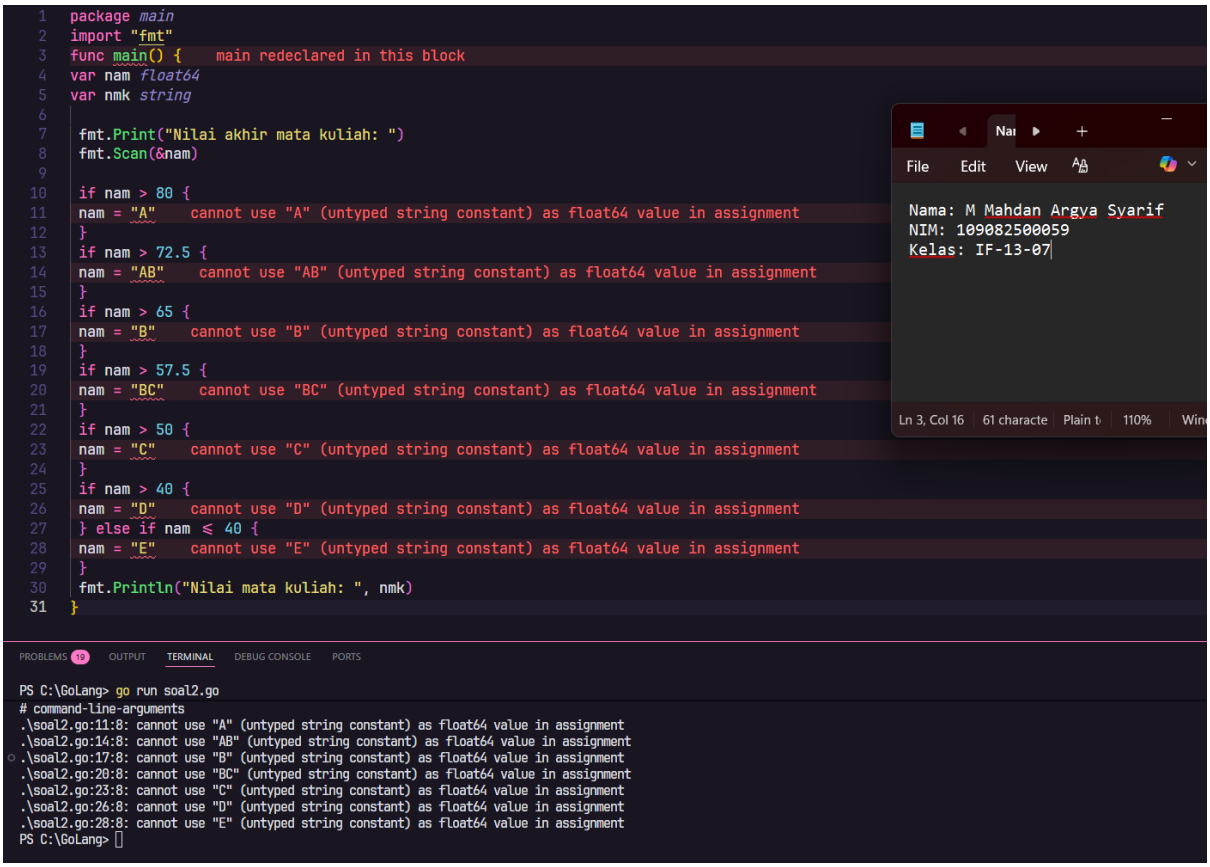
2. **Tugas 2**

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 {
 nam = "A"
 }
 if nam > 72.5 {
 nam = "AB"
 }
 if nam > 65 {
 nam = "B"
 }
 if nam > 57.5 {
 nam = "BC"
 }
if nam > 50 {
 nam = "C"
```

```
        }

        if nam > 40 {

        nam = "D"

        } else if nam <= 40 {

        nam = "E"

        }

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

      }
```

Screenshoot program



- **SOAL A**

  Jika **nam** diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?

  Jawab:

  Program tersebut tidak dapat dieksekusi karena terdapat kesalahan fatal pada prosedur pengkondisian, di mana terjadi ketidaksesuaian tipe data

antara variabel nam (bertipe *float64*) dan nilai yang diproses. Kesalahan terletak pada logika yang mencoba membandingkan atau menetapkan nilai *string* "A" langsung pada variabel numerik nam, padahal seharusnya kondisi mengevaluasi nilai nam (misalnya $\ge$ 80) untuk kemudian menyimpan predikat huruf mutunya ke dalam variabel nmk (bertipe *string*), sehingga konflik tipe data ini menyebabkan program mengalami *error*.

- **SOAL B**

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

  Jawab:

  Program ini memiliki dua kesalahan fundamental, yaitu ketidaksesuaian tipe data saat penetapan variabel dan cacat logika pada struktur kendali. Masalah sintaksis terjadi karena pencampuran tipe data float64 dan string yang seharusnya dipisahkan antara kondisi dan penugasan nilai, namun kesalahan logika yang lebih fatal terletak pada penggunaan rangkaian if independen (bukan if-else if). Akibatnya, input nilai 80.1 yang seharusnya menghasilkan gred "A" justru tertimpa menjadi "D", karena program terus mengevaluasi kondisi hingga baris terakhir tanpa mengunci hasil yang benar sebelumnya, membuat alur penyeleksian kondisi menjadi tidak akurat.

- **SOAL C**

  Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5.

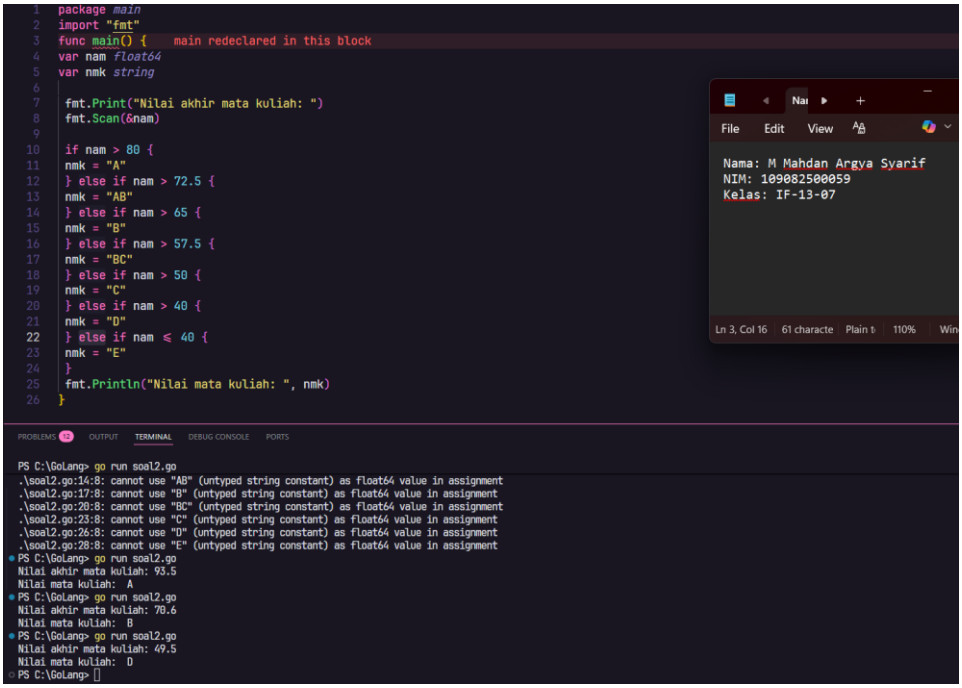
  Seharusnya keluaran yang diperoleh adalah ‘A’, ‘B’, dan ‘D’.

  Jawab:

```
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 & bukti jawaban soal c



Deskripsi program

Program ini merupakan sistem konversi nilai akademik berbasis bahasa Go yang mengubah input angka presisi ganda (float64) menjadi indeks huruf mutu (string) melalui struktur percabangan if-else if berjenjang. Menggunakan paket fmt untuk manajemen input-output, algoritma mengevaluasi rentang nilai secara menurun dari batas atas hingga terendah untuk menentukan klasifikasi yang tepat. Fungsionalitas program ini telah divalidasi melalui uji coba input angka 93.5, 70.6, dan 49.5, yang secara akurat menghasilkan output klasifikasi huruf mutu berturut-turut A, B, dan D.

3. Tugas 3A

Source code

```
package main
import "fmt"

func main() {
    var n, i int
    fmt.Print("masukkan bilangan: ")
    fmt.Scan(&n)

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

```
            }
        }
```

Screenshoot program

Deskripsi program

Program ini dirancang untuk mengidentifikasi dan menampilkan faktor-faktor pembentuk suatu bilangan bulat. Dengan menggunakan tipe data integer pada variabel input dan iterator, program mengelola interaksi pengguna melalui paket fmt untuk menangkap nilai dan menampilkan hasil. Mekanisme intinya bertumpu pada struktur perulangan for yang menelusuri angka dari 1 hingga batas nilai input, dikombinasikan dengan logika percabangan if yang memvalidasi kelipatan menggunakan operasi modulus; jika sisa bagi bernilai nol, maka angka tersebut dikonfirmasi dan dicetak sebagai faktor yang valid.

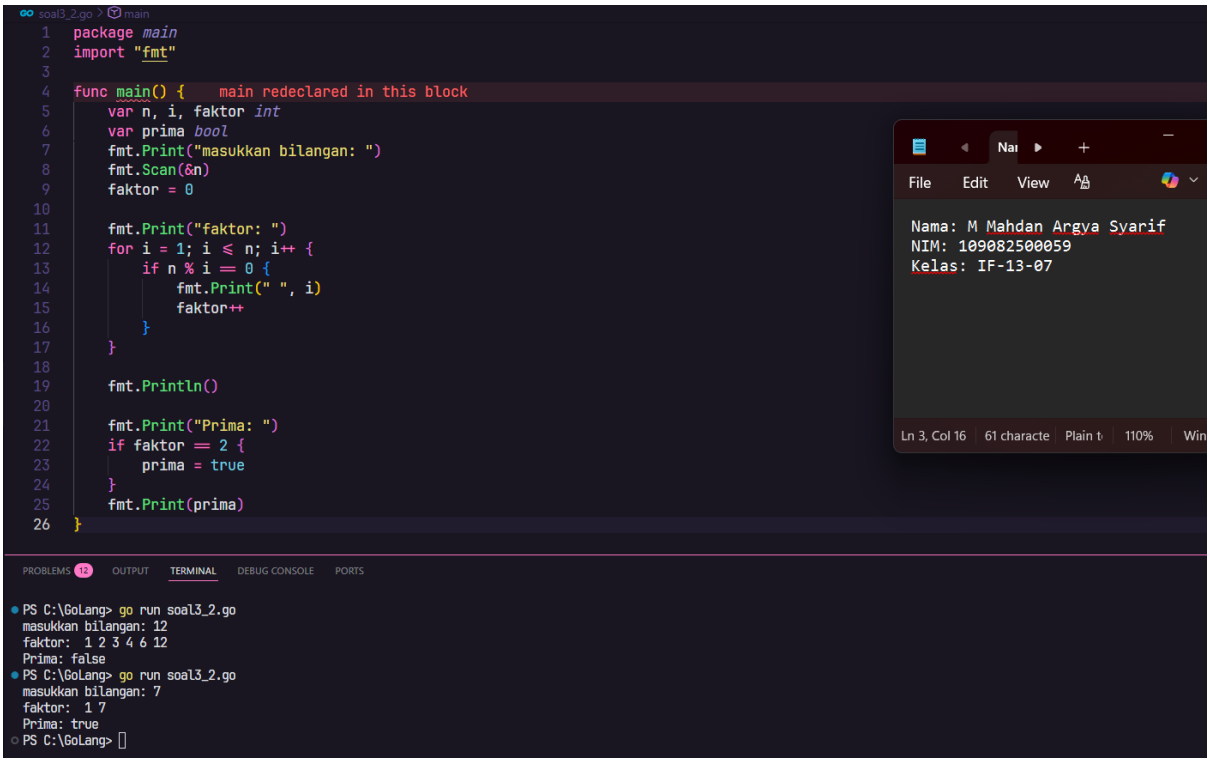
4. Tugas 3B

Source code

```
    package main
    import "fmt"

    func main() {
        var n, i, faktor int
        var prima bool
        fmt.Print("masukkan bilangan: ")
```

```
        fmt.Scan(&n)
        faktor = 0

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

        fmt.Println()

        fmt.Print("Prima: ")
        if faktor == 2 {
            prima = true
        }
        fmt.Print(prima)
    }
```

Screenshoot program



Deskripsi program

Program ini menerapkan logika matematika untuk mengidentifikasi bilangan prima dengan cara memvalidasi jumlah faktor pembaginya. Menggunakan tipe data *integer* untuk pemrosesan angka dan pencacahan, serta *boolean* untuk menyimpan status validitas, algoritma bekerja dengan menelusuri seluruh pembagi potensial melalui perulangan for. Penentuan status prima dilakukan di akhir proses menggunakan struktur kondisional, di mana jika total faktor yang ditemukan tepat berjumlah dua, variabel indikator akan diset menjadi *true* untuk mengonfirmasi bilangan tersebut sebagai prima.