LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE - IF

Disusun oleh:

ALIN KARISA HIZANNAH

109082500010

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

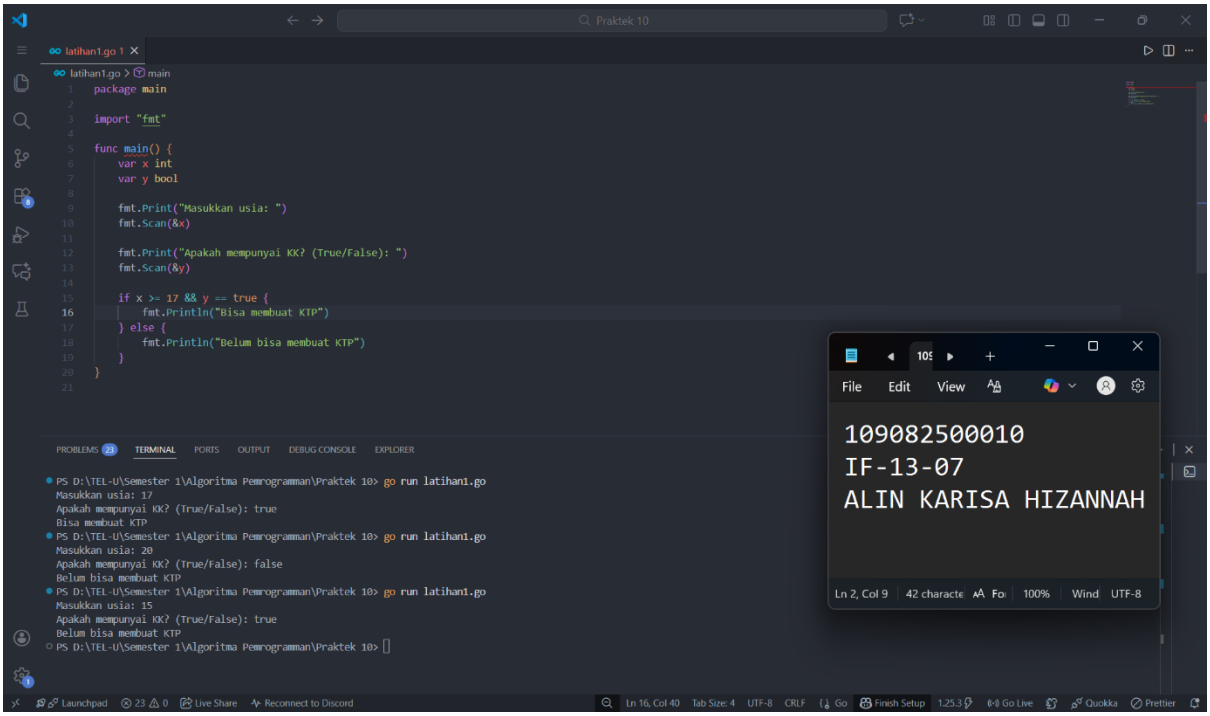
LATIHAN KELAS – GUIDED

1. **Guided 1**

Source Code

```
package main


import "fmt"


func main() {
    var x int
    var y bool


    fmt.Print("Masukkan usia: ")
    fmt.Scan(&x)


    fmt.Print("Apakah mempunyai KK? (True/False): ")
    fmt.Scan(&y)


    if x >= 17 && y == true {
        fmt.Println("Bisa membuat KTP")
    } else {
        fmt.Println("Belum bisa membuat KTP")
    }
}
```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk menentukan apakah seseorang memenuhi syarat untuk membuat KTP berdasarkan usia dan kepemilikan Kartu Keluarga (KK). Pengguna diminta memasukkan usia, kemudian menjawab apakah memiliki KK dengan nilai true atau false. Setelah itu, program memeriksa dua syarat sekaligus yaitu usia minimal 17 tahun dan memiliki KK. Jika kedua syarat terpenuhi maka program menampilkan bahwa pengguna bisa membuat KTP. Jika salah satu atau kedua syarat belum terpenuhi maka program menampilkan bahwa pengguna belum bisa membuat KTP.

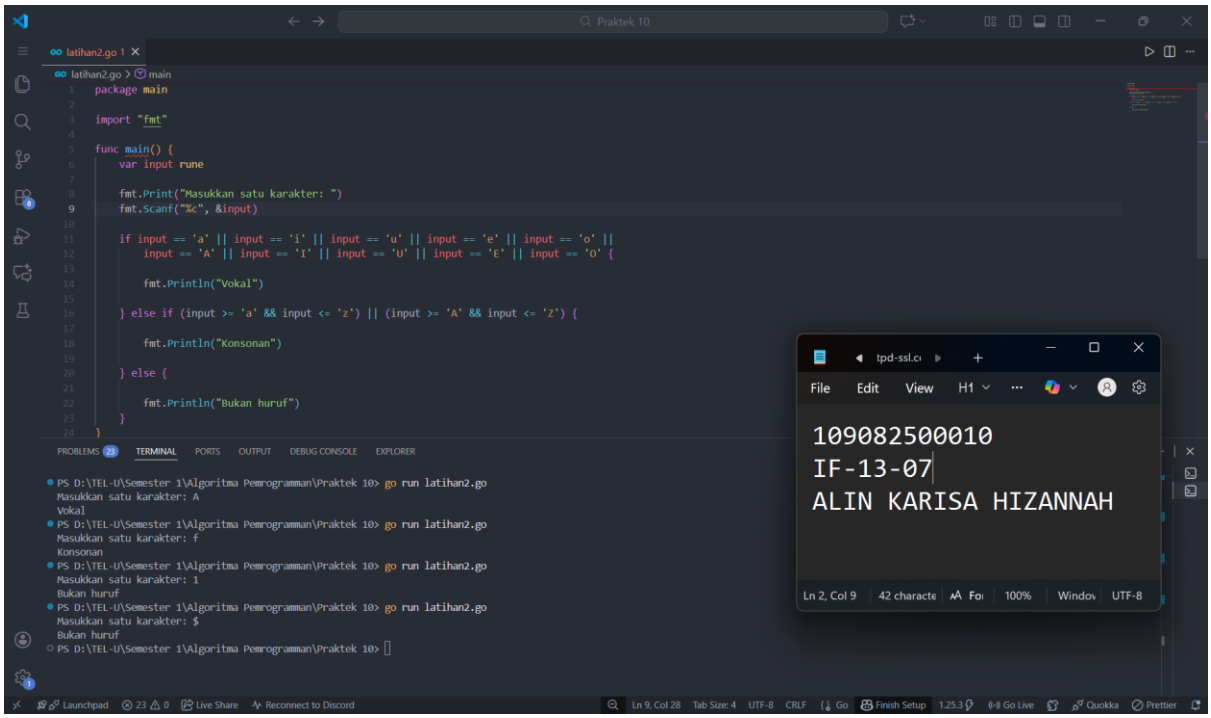
Berdasarkan hasil demo di terminal, program berhasil menguji berbagai kombinasi input seperti usia 17 dengan KK true yang menghasilkan izin membuat KTP, serta contoh lain seperti usia 20 dengan KK false dan usia 15 dengan KK true yang keduanya menghasilkan output belum bisa membuat KTP sehingga menunjukkan bahwa logika percabangannya berjalan dengan benar.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var input rune

    fmt.Print("Masukkan satu karakter: ")
    fmt.Scanf("%c", &input)

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

        fmt.Println("Vokal")

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

        fmt.Println("Konsonan")

    } else {

        fmt.Println("Bukan huruf")
    }
}
```

Screenshoot program



Deskripsi program

Program ini meminta pengguna memasukkan satu karakter, kemudian karakter tersebut dianalisis untuk menentukan apakah termasuk huruf vokal, huruf konsonan, atau bukan huruf. Karakter disimpan dalam variabel bertipe rune sehingga dapat dikenali sebagai satu simbol.

Program memeriksa apakah karakter tersebut adalah salah satu huruf vokal baik dalam bentuk huruf kecil maupun huruf besar. Jika cocok, maka program menampilkan bahwa karakter tersebut adalah vokal. Jika bukan vokal tetapi masih berada dalam rentang huruf alfabet, program menampilkannya sebagai konsonan. Namun apabila karakter yang dimasukkan berada di luar alfabet, seperti angka atau simbol, program akan menyatakan bahwa input tersebut bukan huruf. Berdasarkan hasil demo di terminal, input seperti "A" menghasilkan output vokal, input "f"

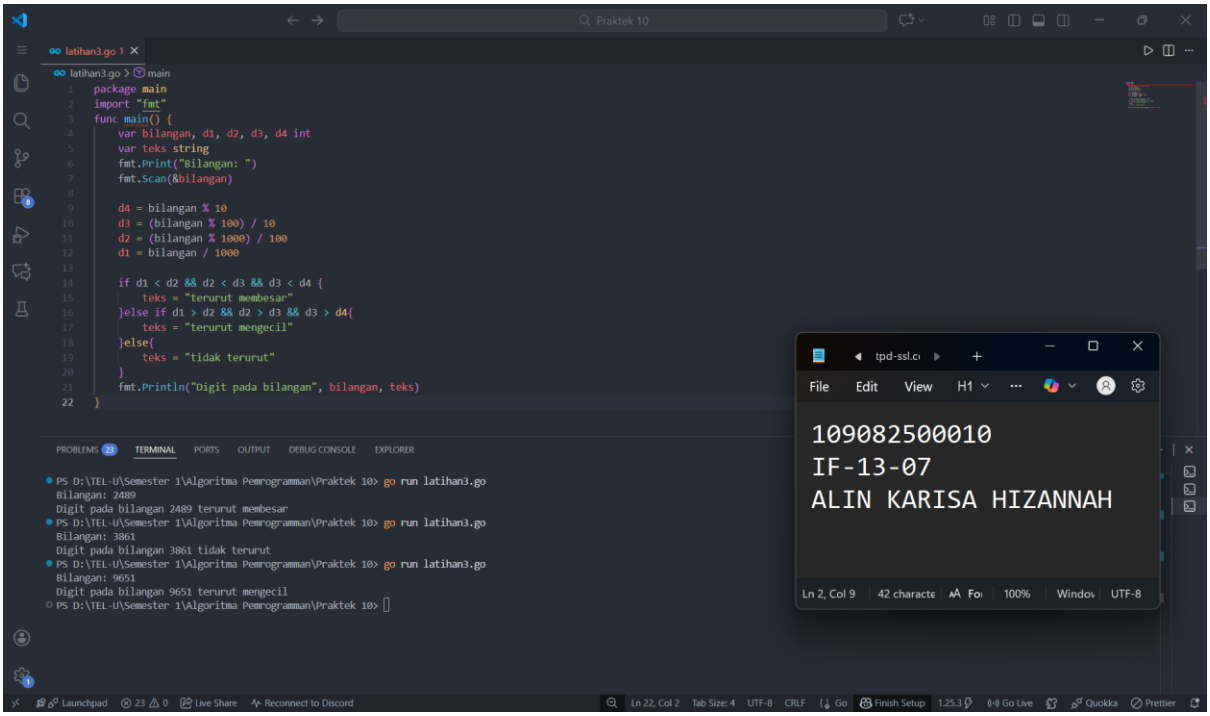
menghasilkan konsonan, dan input "1" maupun "$" menghasilkan output bukan huruf yang menunjukkan bahwa pengecekan berjalan sesuai kondisi yang telah dibuat.

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

Kode Program tersebut dibuat untuk menganalisis susunan empat digit pada sebuah bilangan empat angka. Ketika pengguna memasukkan bilangan, program memisahkan angka tersebut menjadi digit pertama, kedua, ketiga, dan keempat. Digit paling kanan diambil menggunakan operasi sisa pembagian, sedangkan digit lain diperoleh melalui kombinasi pembagian dan sisa pembagian.

Setelah empat digit itu didapatkan, program membandingkannya untuk mengetahui pola urutannya. Jika setiap digit dari kiri ke kanan selalu lebih besar dari digit sebelumnya maka bilangan tersebut dianggap memiliki pola urut membesar. Jika setiap digit justru selalu lebih kecil maka hasilnya urut mengecil. Namun bila ada digit yang tidak mengikuti pola tersebut maka dinyatakan tidak terurut.

Hasil pengujian yang terlihat menunjukkan bahwa bilangan seperti 2489 termasuk urut membesar karena setiap angkanya meningkat dari kiri. Bilangan 9651 termasuk

urut mengecil karena setiap angkanya menurun. Sementara bilangan seperti 3861 tidak memenuhi kedua pola sehingga dinyatakan tidak terurut. Jendela teks di sisi kanan hanya berisi data identitas dan tidak memiliki hubungan dengan proses analisis bilangan pada program tersebut.**TUGAS**

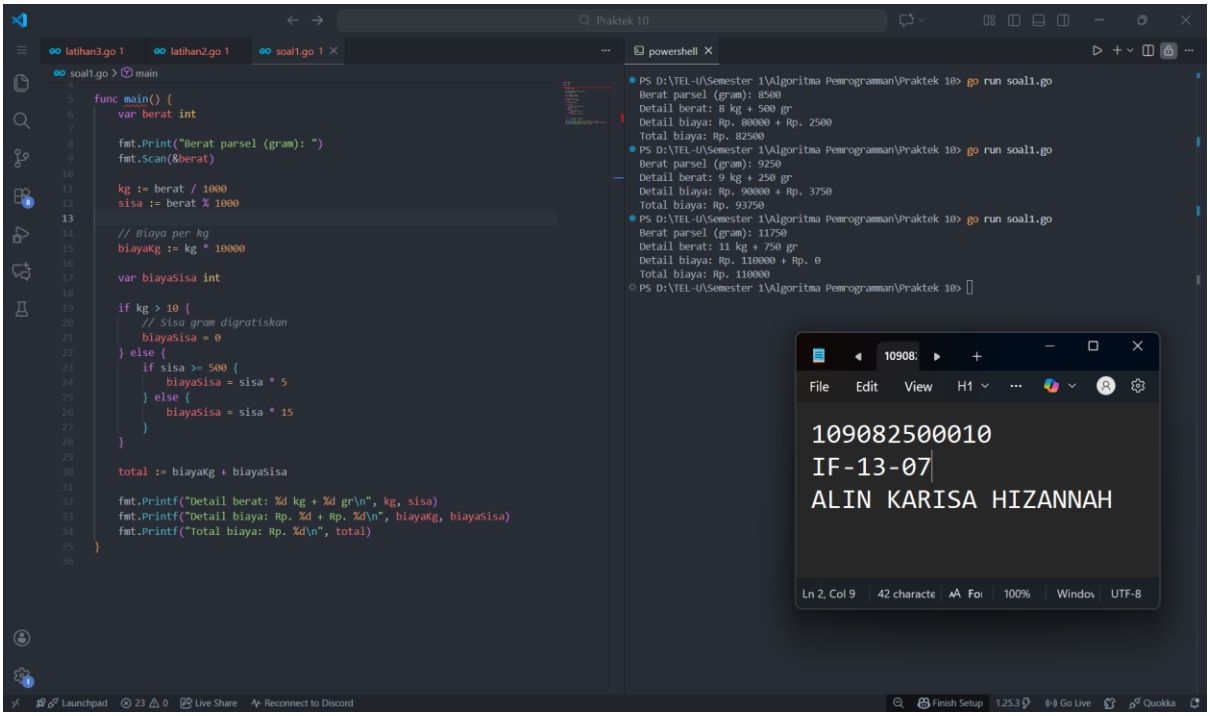
1. **Tugas 1**

   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


    // Biaya per kg
    biayaKg := kg * 10000


    var biayaSisa int


    if kg > 10 {
        // Sisa gram digratiskan
```

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

        total := 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", total)
    }
```

Screenshoot program



Deskripsi program

Program tersebut digunakan untuk menghitung biaya pengiriman berdasarkan berat paket dalam satuan gram. Setelah pengguna memasukkan berat, program memecah nilai tersebut menjadi dua bagian yaitu jumlah kilogram penuh serta sisa gram yang tidak mencapai satu kilogram. Bagian kilogram dihitung dengan membagi berat dengan seribu sedangkan sisa gram dihitung menggunakan operasi sisa pembagian.

Biaya utama berasal dari jumlah kilogram yang telah lengkap. Setiap kilogram dikenakan tarif tetap sehingga total biaya dasar berasal dari jumlah kilogram dikalikan harga per kilogram. Setelah itu program menghitung biaya tambahan dari sisa gram. Jika berat paket mencapai sepuluh kilogram atau lebih maka sisa gram tidak dikenai biaya sehingga biaya tambahan bernilai nol. Namun jika beratnya kurang dari sepuluh kilogram maka sisa gram dihargai berbeda bergantung jumlahnya. Bila sisa lebih dari setengah kilogram maka setiap gram dihitung dengan tarif lebih ringan. Bila kurang dari atau sama dengan setengah kilogram maka tarif per gramnya lebih tinggi.

Setelah semua komponen biaya dihitung program menampilkan hasil secara rinci. Tampilan yang diperlihatkan pada layar memperlihatkan beberapa contoh

perhitungan. Contoh pertama adalah paket seberat lima ribu delapan ratus gram yang setara dengan delapan kilogram dan lima ratus gram sehingga dikenakan biaya kilogram serta biaya gram tambahan. Contoh lain menunjukkan paket yang beratnya melewati sepuluh kilogram sehingga sisa gramnya tidak lagi dihitung. Jendela teks di sebelah kanan masih berisi informasi identitas dan tidak berkaitan dengan proses perhitungan biaya pengiriman.

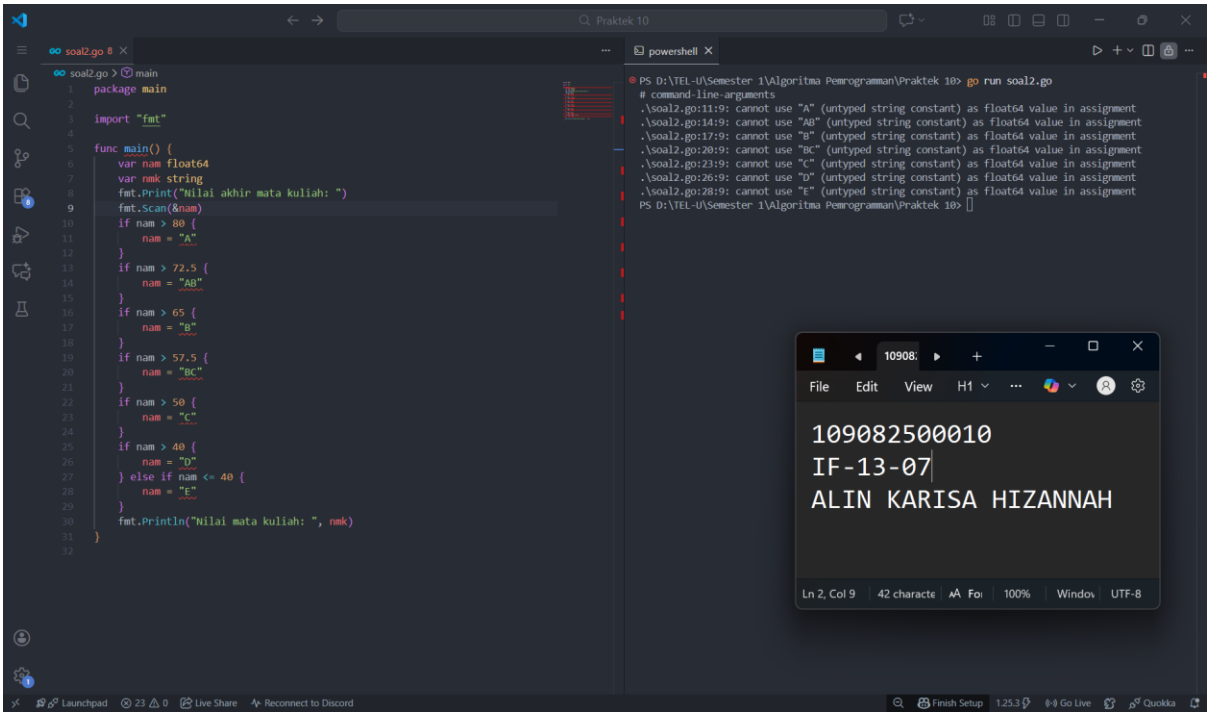
2. **Tugas 2**

Source code

```
//Program yang salah sesuai soal
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



a. Jika NAM diberikan 80.1 apa keluaran dari program tersebut? Apakah eksekusi program sesuai spesifikasi soal?

Jika nilai yang diberikan adalah 80.1 maka hasil yang dihasilkan oleh program adalah huruf A. Hasil ini memang sesuai dengan kategori nilai pada tabel penilaian karena nilai di atas atau sama dengan delapan puluh memang dikategorikan sebagai A. Namun kecocokan ini hanya terjadi karena nilainya berada tepat pada batas atas sehingga langsung masuk pada kondisi pertama yang dicek oleh program.

Meskipun hasilnya benar, eksekusi program sebenarnya tidak mengikuti spesifikasi yang diharapkan. Program melakukan pengecekan kondisi secara terpisah dan tidak tersusun secara berurutan sesuai rentang nilai yang telah ditentukan pada tabel. Akibatnya kondisi-kondisi tersebut tidak saling mengunci dan dapat menyebabkan penilaian yang salah pada nilai-nilai lain meskipun untuk nilai 80.1 hasilnya kebetulan benar.

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

Kesalahan utama pada program terletak pada cara pengecekan nilai yang tidak menggunakan struktur percabangan berurutan. Setiap kondisi ditulis berdiri sendiri

sehingga nilai yang seharusnya termasuk dalam satu kategori masih berpotensi dievaluasi oleh kondisi lainnya. Selain itu batas nilai yang dibandingkan tidak disusun secara sistematis dari rentang tertinggi ke terendah sehingga hasilnya bisa tidak sesuai tabel.

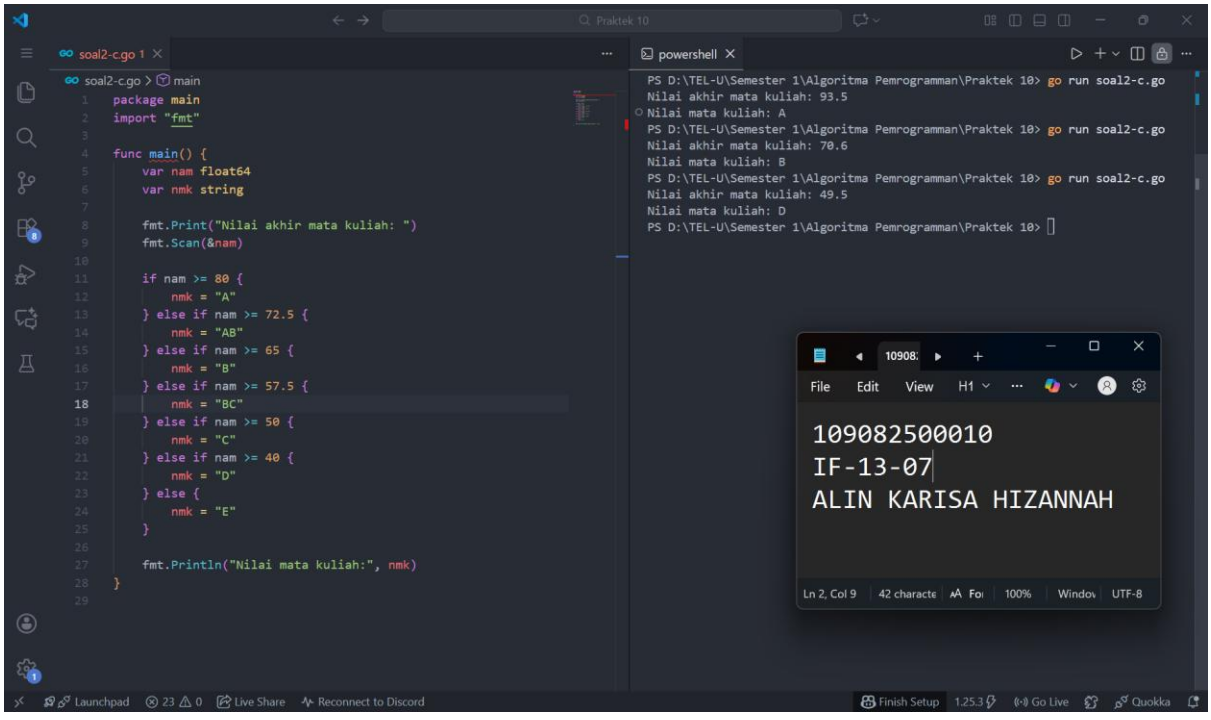
Alur yang benar seharusnya memeriksa nilai berdasarkan urutan dari kategori tertinggi menuju kategori terendah. Setiap nilai harus hanya mungkin masuk ke satu rentang saja sehingga struktur percabangan harus berurutan dan eksklusif. Dengan mengikuti aturan rentang yang tepat sesuai tabel penilaian maka nilai huruf yang dihasilkan akan konsisten dan tidak menimbulkan ambiguitas.

c. Perbaiki program tersebut. Ujilah dengan masukan 93.5, 70.6, dan 49.5. Seharusnya keluaran: A, B, dan D.

```
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



Pada bagian logika yang diperbaiki, perubahan utamanya adalah bahwa seluruh proses pengecekan nilai tidak lagi menggunakan variabel nam seperti pada versi program sebelumnya, tetapi telah diganti menjadi nmk. Perubahan ini menyebabkan

sumber nilai yang dievaluasi dalam setiap kondisi menjadi berbeda dari sebelumnya. Jika nilai yang dibaca ke dalam variabel nmk benar-benar sesuai dengan nilai akhir mata kuliah yang ingin dinilai, maka seluruh struktur if–else kini bekerja berdasarkan variabel yang tepat. Hal ini membuat proses penentuan kategori nilai huruf menjadi konsisten dan tidak lagi salah membaca variabel, sehingga setiap pengecekan kondisi sesuai dengan data yang benar.

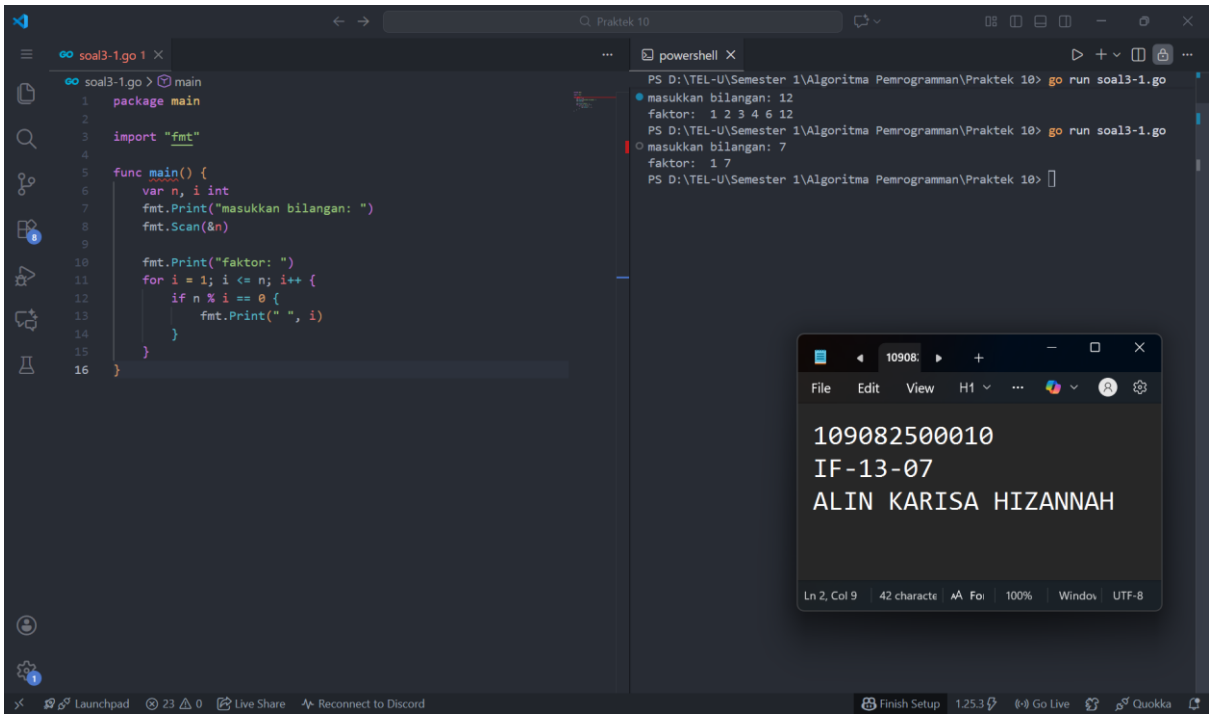
Selain itu, perubahan variabel tersebut juga memastikan bahwa alur percabangan berjalan sebagaimana yang telah ditentukan dalam tabel penilaian. Ketika nilai disimpan dengan benar ke dalam variabel nmk, setiap rentang nilai dapat dipetakan ke kategori yang tepat tanpa terjadi konflik seperti sebelumnya. Setelah perbaikan ini dilakukan, hasil uji coba menggunakan nilai 93.5 menghasilkan A, nilai 70.6 menghasilkan B, dan nilai 49.5 menghasilkan D. Ini menunjukkan bahwa mengganti penggunaan variabel nam menjadi nmk pada logika if–else telah membuat program bekerja sesuai standar yang benar.

3. Tugas 3 - 1

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 tersebut digunakan untuk menampilkan faktor dari sebuah bilangan sekaligus menentukan apakah bilangan tersebut merupakan bilangan prima atau tidak. Setelah pengguna memasukkan sebuah nilai, program memeriksa setiap angka mulai dari satu hingga bilangan itu sendiri. Setiap angka yang dapat membagi bilangan tanpa menyisakan sisa akan ditampilkan sebagai faktor, dan setiap kali kondisi tersebut terpenuhi penghitung faktor akan bertambah satu. Dengan cara ini seluruh faktor akan tampil berurutan dari kecil ke besar.

Setelah seluruh faktor ditemukan, program melanjutkan dengan menentukan apakah bilangan tersebut prima. Suatu bilangan dikategorikan sebagai bilangan prima apabila hanya memiliki dua faktor, yaitu satu dan bilangan itu sendiri. Program kemudian memeriksa jumlah faktor yang telah dihitung sebelumnya. Jika jumlah faktor tepat dua,

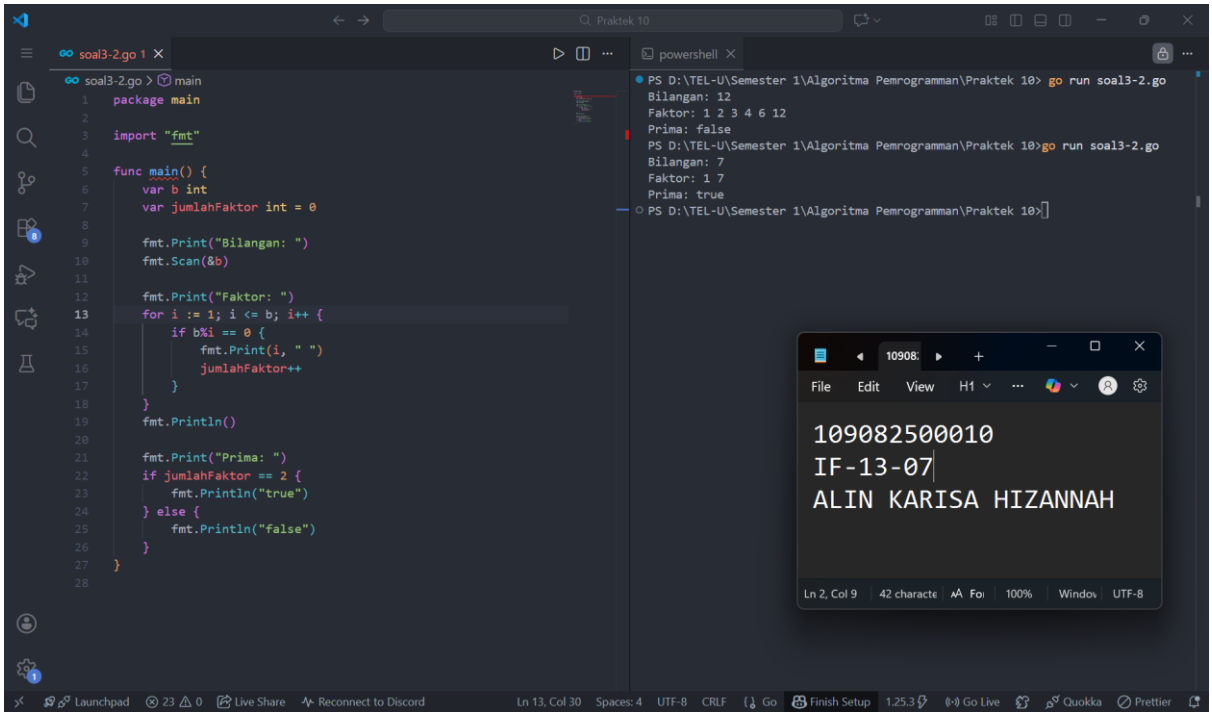
program menampilkan nilai true sebagai tanda bahwa bilangan tersebut prima. Sebaliknya, jika lebih dari dua atau hanya satu faktor, program menampilkan false karena bilangan tersebut bukan termasuk bilangan prima.

4. **Tugas 3 - 2**

Source code

```
package main

import "fmt"

func main() {
    var b int
    var jumlahFaktor int = 0

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

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

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

Screenshoot program



Deskripsi program

Program ini berfungsi untuk menghitung nilai faktorial dari suatu bilangan yang dimasukkan oleh pengguna. Ketika dijalankan, pengguna diminta untuk memasukkan sebuah bilangan bulat positif, lalu program akan melakukan perhitungan faktorial dengan mengalikan setiap bilangan dari 1 hingga bilangan tersebut secara berurutan. Nilai hasil perkalian disimpan dalam variabel faktorial yang awalnya bernilai 1 agar proses perhitungan berjalan dengan benar.

Setelah semua perulangan selesai, hasil akhir perhitungan faktorial akan ditampilkan ke layar disertai dengan bilangan asalnya.