LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10 – ELSE-IF

ALGORITMA DAN PEMROGRAMAN 1

Disusun oleh:

NAMA : HAFIZD SAMA'I SYAMSI

NIM : 109082500183

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

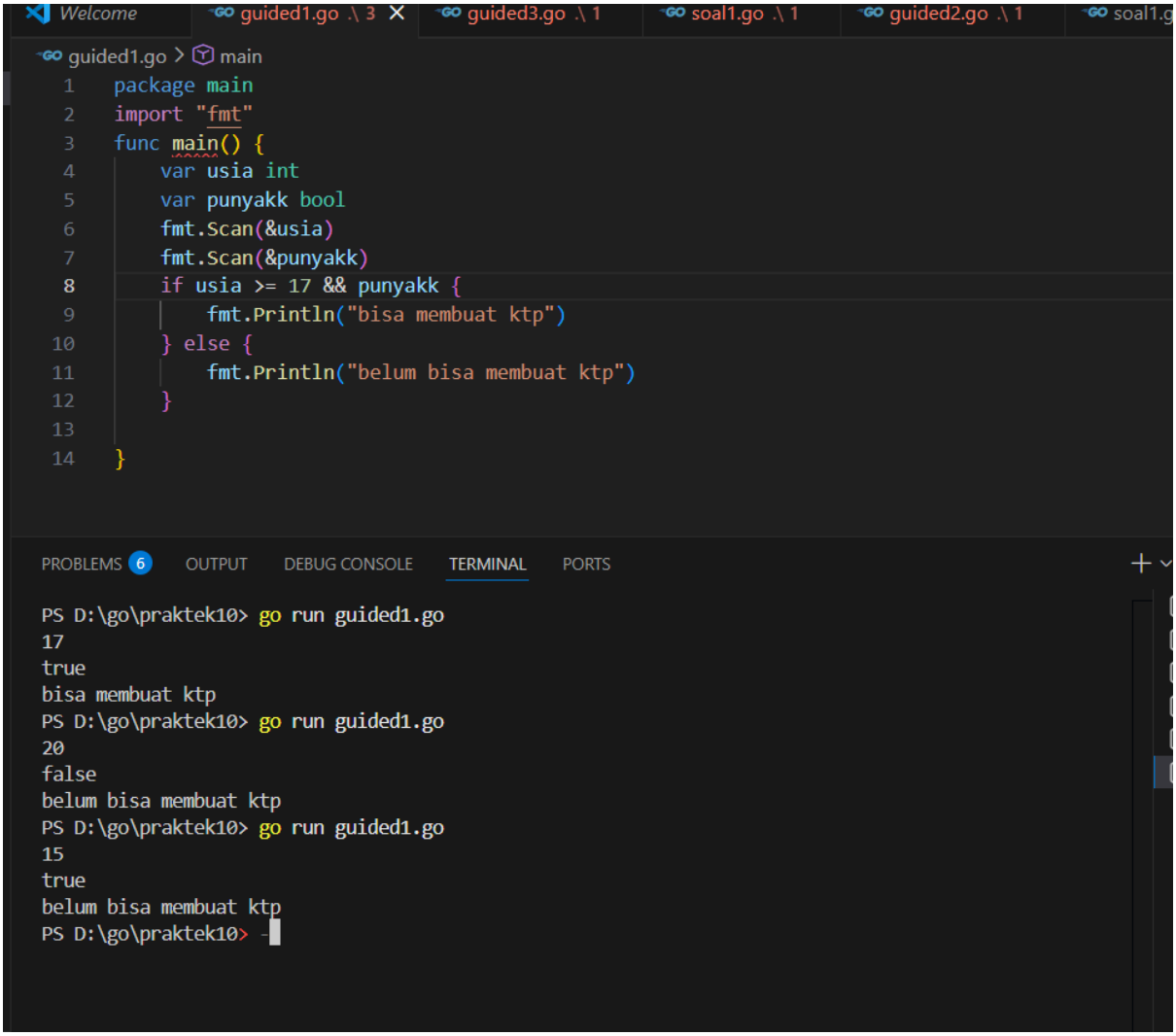
LATIHAN KELAS – GUIDED

1. **Guided 1**

Source Code

```
package main
import "fmt"
func main() {
    var usia int
    var punyakk bool
    fmt.Scan(&usia)
    fmt.Scan(&punyakk)
    if usia >= 17 && punyakk {
        fmt.Println("bisa membuat ktp")
    } else {
        fmt.Println("belum bisa membuat ktp")
    }

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menentukan apakah seseorang sudah memenuhi syarat untuk membuat KTP atau belum. Program meminta dua input dari pengguna: **usia** dan status kepemilikan KK. Usia dimasukkan sebagai angka, sedangkan status KK dimasukkan sebagai nilai boolean (true atau false).

Setelah mendapatkan kedua input tersebut, program melakukan pengecekan. Jika usia pengguna 17 tahun atau lebih *dan* pengguna sudah memiliki KK, program menampilkan pesan "bisa membuat ktp". Namun jika salah satu syarat tidak terpenuhi, program menampilkan "belum bisa membuat ktp".

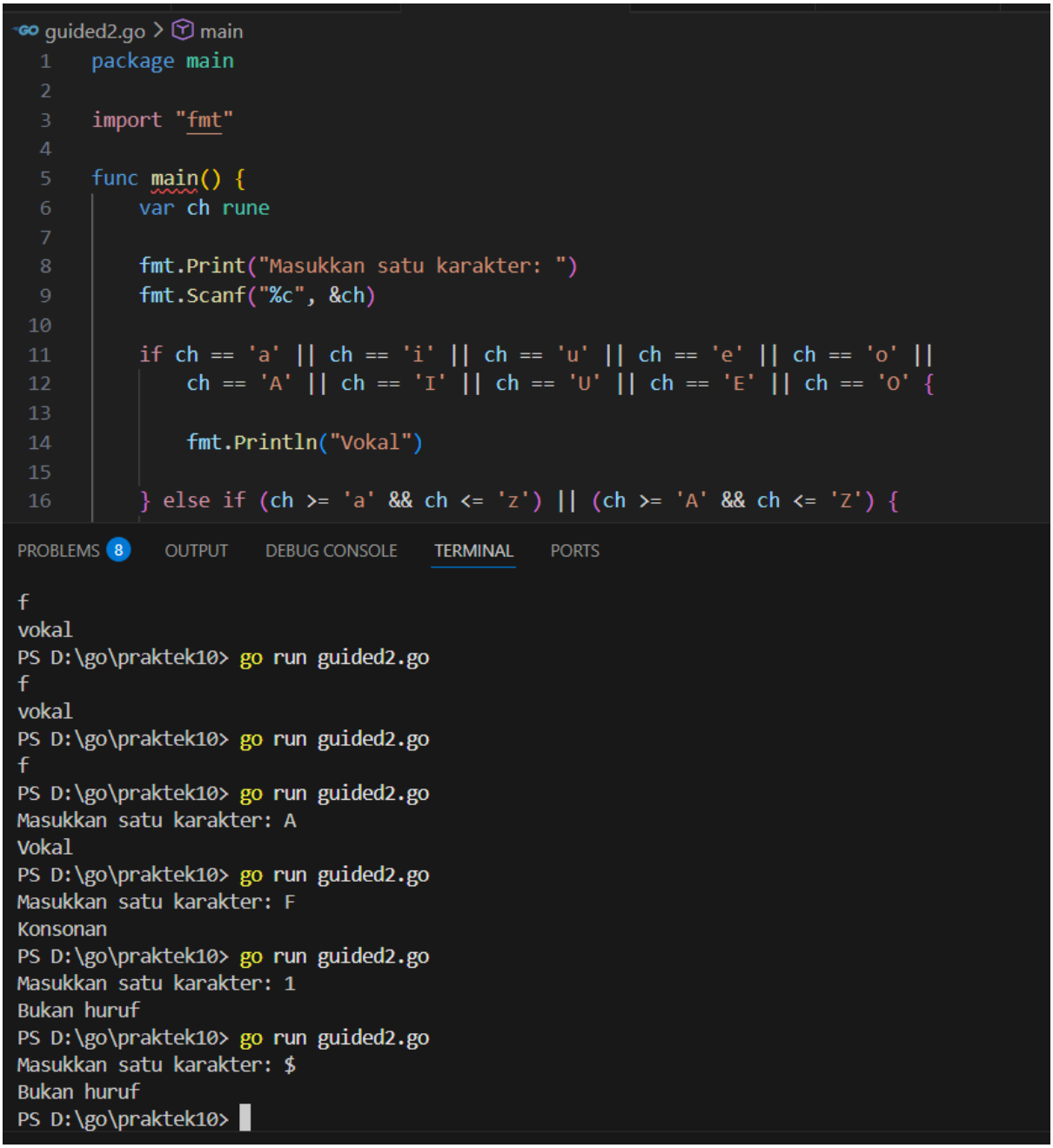
Intinya, program ini mengecek dua syarat utama pembuatan KTP: cukup umur dan punya KK. Jika dua-duanya terpenuhi, baru bisa buat KTP.

2. **Guided 2**

Source Code

```
package main


import "fmt"


func main() {
    var ch rune


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


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


        fmt.Println("Vokal")


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


        fmt.Println("Konsonan")


    } else {


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

Screenshoot program



```
guided2.go > main
1   package main
2
3   import "fmt"
4
5   func main() {
6       var ch rune
7
8       fmt.Print("Masukkan satu karakter: ")
9       fmt.Scanf("%c", &ch)
10
11      if ch == 'a' || ch == 'i' || ch == 'u' || ch == 'e' || ch == 'o' ||
12          ch == 'A' || ch == 'I' || ch == 'U' || ch == 'E' || ch == 'O' {
13
14          fmt.Println("Vokal")
15
16      } else if (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') {
```

```
PROBLEMS 8   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

f
vokal
PS D:\go\praktek10> go run guided2.go
f
vokal
PS D:\go\praktek10> go run guided2.go
f
PS D:\go\praktek10> go run guided2.go
Masukkan satu karakter: A
Vokal
PS D:\go\praktek10> go run guided2.go
Masukkan satu karakter: F
Konsonan
PS D:\go\praktek10> go run guided2.go
Masukkan satu karakter: 1
Bukan huruf
PS D:\go\praktek10> go run guided2.go
Masukkan satu karakter: $
Bukan huruf
PS D:\go\praktek10>
```

Deskripsi program

Program ini digunakan untuk menentukan jenis suatu karakter yang dimasukkan oleh pengguna. Jenis karakter yang ingin diketahui adalah apakah karakter tersebut huruf vokal, huruf konsonan, atau bukan huruf.

Pertama, program meminta pengguna untuk memasukkan satu karakter, lalu menyimpannya dalam variabel ch.

Setelah itu, program melakukan pengecekan:

- Jika karakter Adalah huruf vokal
  Prorgam akan mengecek apakah ch termasuk salah satu huruf vocal (a, i, u, e, o) baik huruf kecil maupun huruf besar.
- Jika karakter Adalah huruf alfabet tapi bukan vocal

Program mengecek apakah ch berada dalam rentang huruf a-z atau A-Z. Kalau iya, berarti karakter tersebut Adalah konsonan , dan program menampilkan “kosonan”

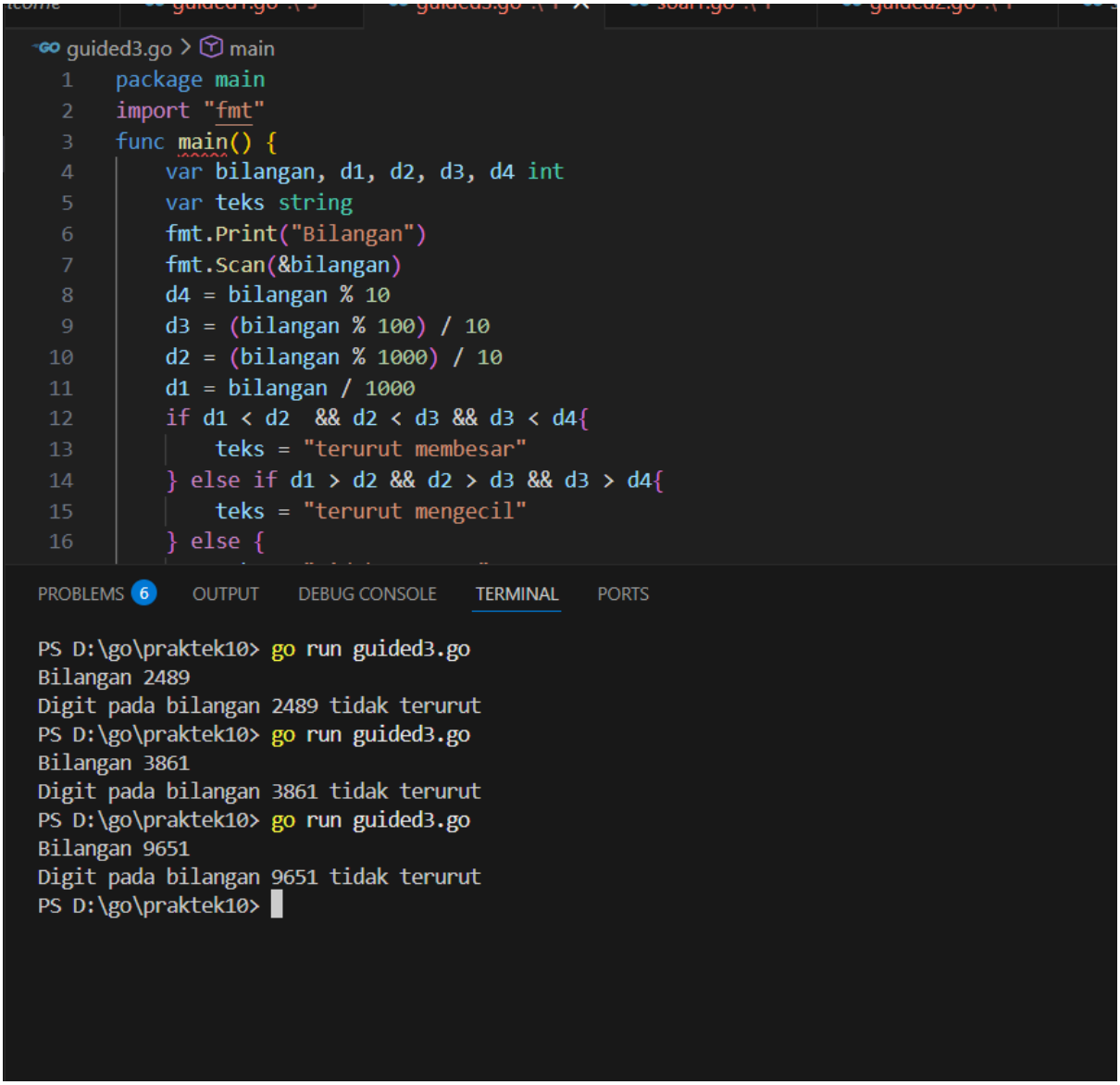
- Jika buakn huruf sama sekali
  Misalnya angka, symbol, atau karakter lain
  Program akan menampilkan “Bukan Huruf”

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



guided3.go > main

```go
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 {
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\go\praktek10> go run guided3.go
Bilangan 2489
Digit pada bilangan 2489 tidak terurut
PS D:\go\praktek10> go run guided3.go
Bilangan 3861
Digit pada bilangan 3861 tidak terurut
PS D:\go\praktek10> go run guided3.go
Bilangan 9651
Digit pada bilangan 9651 tidak terurut
PS D:\go\praktek10>
```

Deskripsi program

Program ini digunakan untuk menentukan apakah susunan digit pada sebuah bilangan empat angka terurut membesar, terurut mengecil, atau tidak terurut.
Pertama, program meminta pengguna memasukkan sebuah bilangan empat digit, lalu menyimpannya dalam variabel bilangan.
Setelah itu, program memisahkan bilangan tersebut menjadi empat digit:

- d4 = digit ribuan paling kanan (bilangan % 10)
- d3 = digit ratusan alias digit ketiga
- d2 = digit kedua
- d1 = digit pertama (ribuan)

Dengan empat digit ini, program melakukan pengecekan:

➢ Terurut membesar

Jika digit pertama lebih kecil dari digit kedua, digit kedua lebih kecil dari digit ketiga, dan digit ketiga leboh kecil dari digit keempat (d1 < d2 < d3 < d4), makan program menyatakan bahwa digit bilangan tersebut “terurut membesar”

- Terurut mencecil
  Jika digit petama lebih besar dari digit kedua, digit kedua lebih besar dari digit ketiga, dan ketiga lebih besar dari idgit keempat (d1 > d2 > d3 > d4), program akan menampilkan “terurut megecil”

- Tidak terurut
  Jika tidak memenuhi salah satu kondisi di atas, berarti digit bilangan tidak tersusun secara naik atau turun, sehingga, program akan men

TUGAS

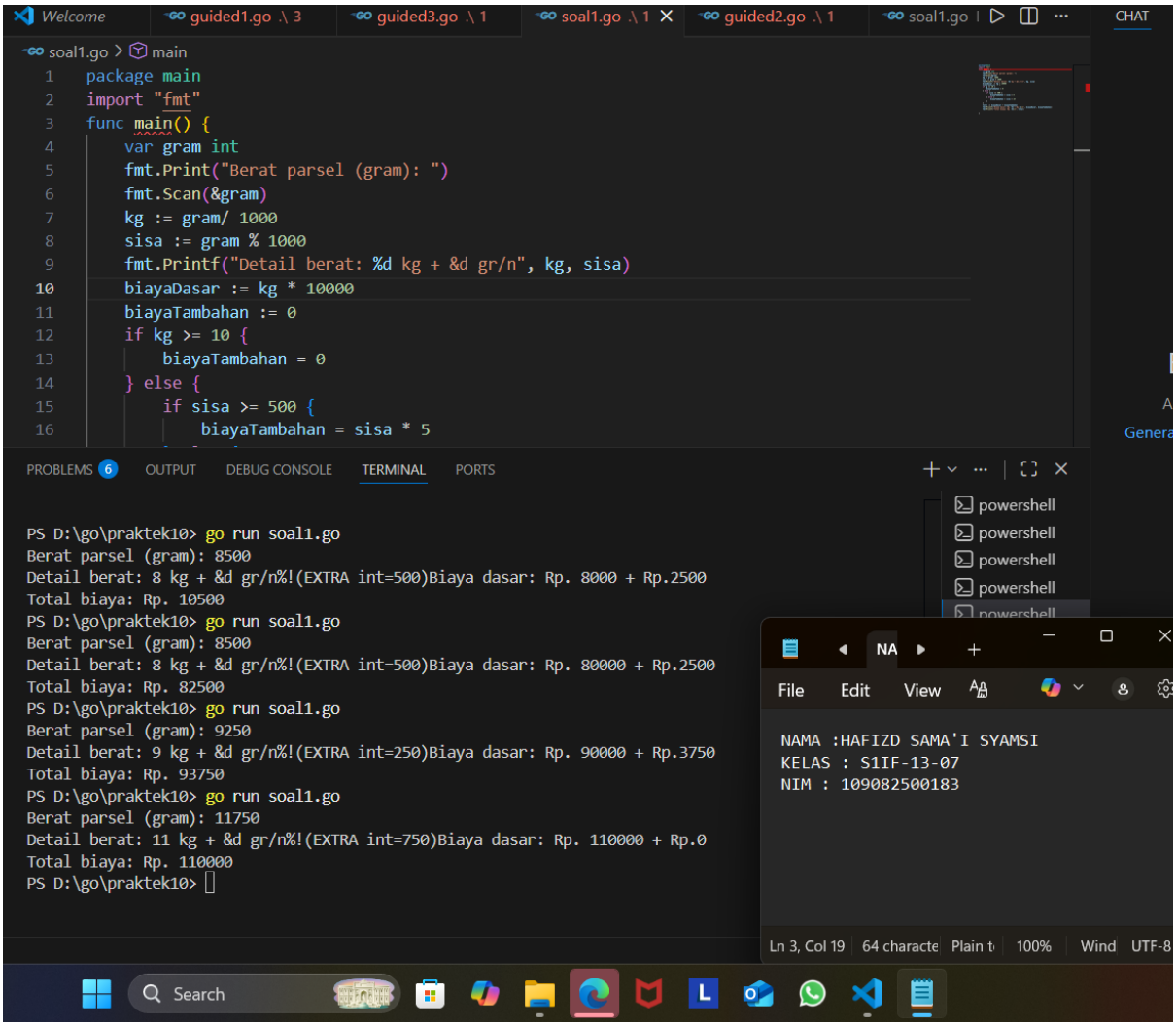
1. **Tugas 1**

   Source code

```
package main
import "fmt"
func main() {
    var gram int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&gram)
    kg := gram/ 1000
    sisa := gram % 1000
    fmt.Printf("Detail berat: %d kg + &d gr/n", kg, sisa)
    biayaDasar := kg * 10000
    biayaTambahan := 0
    if kg >= 10 {
        biayaTambahan = 0
    } else {
        if sisa >= 500 {
            biayaTambahan = sisa * 5
        } else {
```

```
            biayaTambahan = sisa * 15

        }

    }

    total := biayaDasar + biayaTambahan

    fmt.Printf("Biaya dasar: Rp. %d + Rp.%d\n", biayaDasar,
biayaTambahan)

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


}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menghitung biaya pengiriman parsel berdasarkan berat yang dimasukkan dalam satuan gram. Perhitungan biaya dilakukan dengan memisahkan berat menjadi kilogram dan sisa gram, lalu menghitung biaya dasar serta biaya tambahan sesuai aturan berat.

1. Input berat parsel
Program meminta pengguna memasukkan berat parsel dalam gram, lalu menyimpannya ke variabel gram.

2. Mengubah gram menjadi kilogram dan sisa gram
Berat total dipisahkan menjadi dua bagian:

- kg = jumlah kilogram hasil pembagian (gram / 1000)
- sisa = sisa gram setelah dikonversi (gram % 1000)

Program kemudian menampilkan detail berat dalam format:
x kg + y gr

3. Menghitung biaya dasar
Biaya dasar dihitung dari total kilogram:

biayaDasar = kg * 10000
Setiap 1 kg dikenakan biaya Rp10.000.

4. Menghitung biaya tambahan
Aturan biaya tambahan:

- Jika berat parsel ≥ 10 kg, tidak ada biaya tambahan (biayaTambahan = 0).
- Jika berat kurang dari 10 kg, maka biaya tambahan dihitung dari sisa gram:
    - Jika sisa gram ≥ 500, biaya tambahan = sisa * 5
    - Jika sisa gram < 500, biaya tambahan = sisa * 3

Dengan kata lain, semakin berat sisa gramnya, semakin besar tarif per gramnya.

5. Menghitung total biaya
Total biaya dihitung dengan menjumlahkan:

total = biayaDasar + biayaTambahan

6. Menampilkan hasil akhir
Program menampilkan:

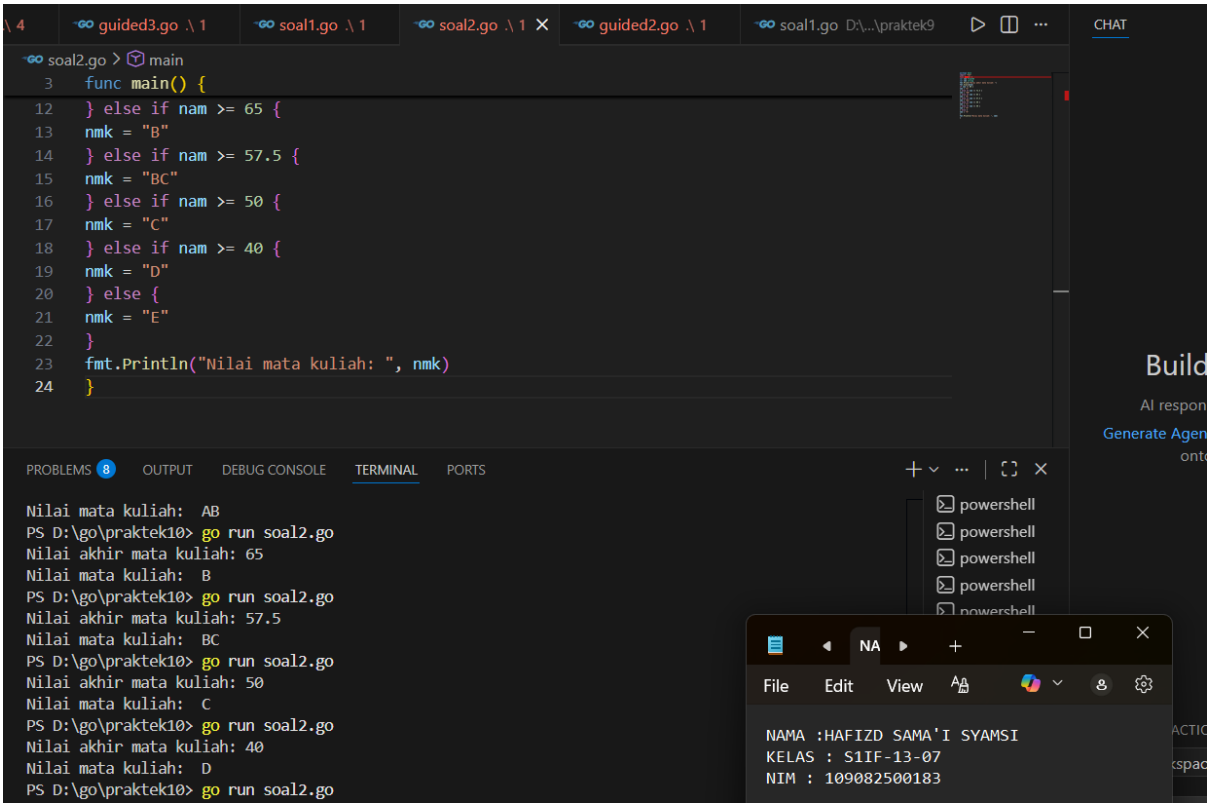
- Biaya dasar
- Biaya tambahan
- Total biaya pengiriman

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 {
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



Deskripsi program

A. Jika nam diberikan angka 80.1, apa keluaran program?

➢ Program akan masuk ke kondisi:

if nam >= 80 { nmk = "A"
}

Karena 80.1 ≥ 80, maka nilai huruf yang diberikan adalah:

Keluaran program: A

➢ Apakah eksekusi program sesuai dengan spesifikasi soal?

Jika spesifikasi soal menyatakan bahwa:
Nilai ≥ 80 harus mendapat huruf A
maka, program berjalan sesuai spesifikasi karena nilai 80.1 jelas berada di atas batas minimal 80.

B. Kesalahan Program

➢ Rentang nilai tidak jelas

Program hanya mengecek batas bawah (≥), tapi tidak memberi batas atas.
Akibatnya, interval nilai tidak tertutup dengan rapi.

➢ Logika penilaian kurang presisi

Misalnya nilai 100 atau 300 tetap dianggap A karena tidak ada pengecekan batas maksimal.
Penyebabnya

- Karena *if else* berjalan dari atas ke bawah, dan setiap kondisi hanya memeriksa "nilai ≥ batas", tanpa memeriksa "nilai < batas berikutnya".
- Alur yang Seharusnya
- Program harus memeriksa rentang lengkap, misalnya:
  80 100 → A
  72.5 < 80 → AB
  65 < 72.5 → B

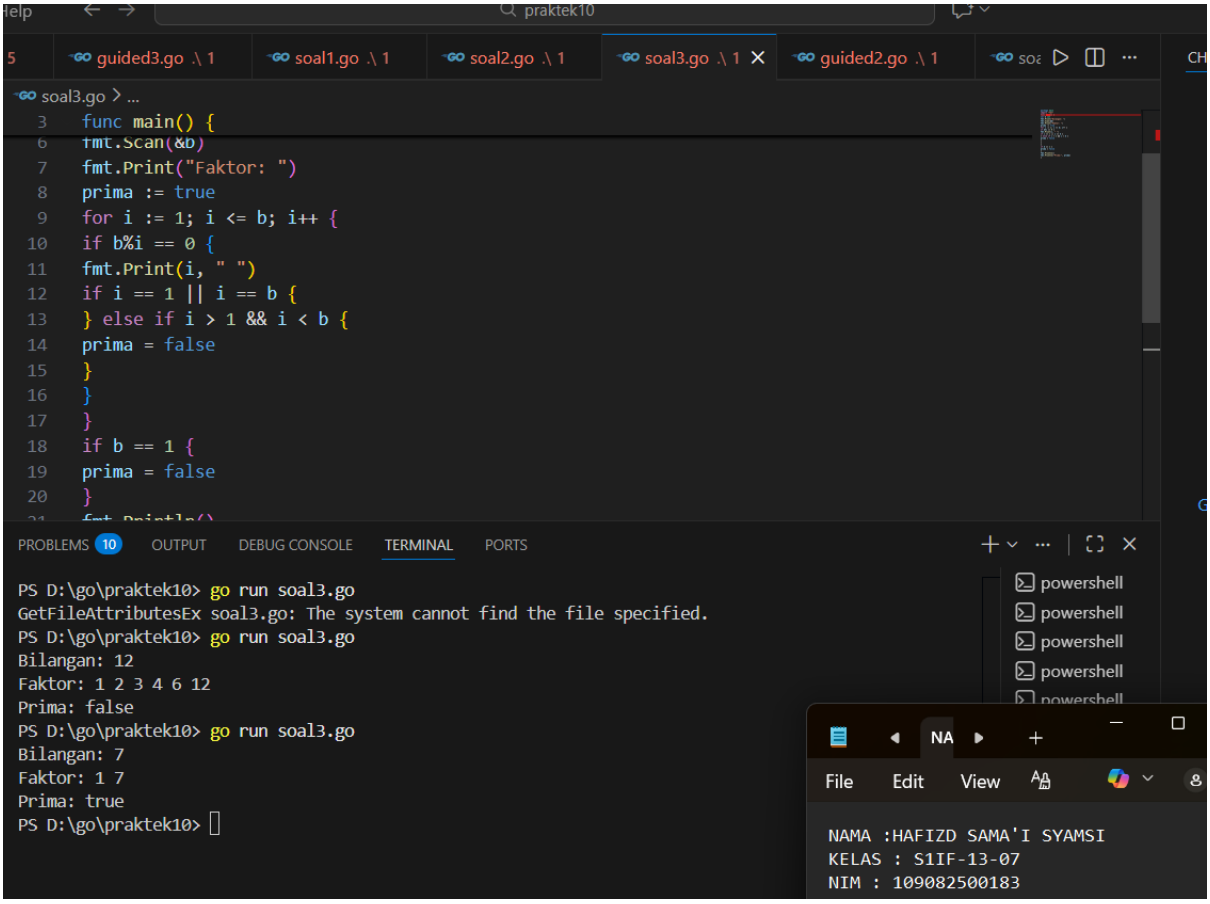
C. Setelah rentang nilainya diperbaiki, program diuji dengan tiga angka: 93.5**,** 70.6**,** dan 49.5**.** Hasil yang benar adalah A**,** B**,** dan D**,** karena masing-masing jatuh pada rentang nilai yang sesuai.

- **Tugas 3**

Source code

```
package main
import "fmt"
func main() {
var b int
fmt.Print("Bilangan: ")
fmt.Scan(&b)
fmt.Print("Faktor: ")
prima := true
for i := 1; i <= b; i++ {
if b%i == 0 {
fmt.Print(i, " ")
if i == 1 || i == b {
} else if i > 1 && i < b {
prima = false
}
}
}
if b == 1 {
prima = false
}
fmt.Println()
fmt.Println("Prima:", prima)
}
```

Screenshoot program



Deskripsi program

Program ini mulai dengan minta satu bilangan dari pengguna. Setelah itu, program langsung nyiapin anggapan awal bahwa bilangan tersebut prima. Lalu program ngecek satu per satu angka dari 1 sampai bilangan itu sendiri.

Setiap kali ada angka yang bisa membagi habis bilangan tersebut, angka itu langsung ditampilin sebagai faktor. Nah, begitu program nemu faktor yang letaknya di Tengah artinya bukan 1 dan bukan bilangan itu sendiri program langsung tandai kalau bilangan itu bukan prima, karena bilangan prima cuma boleh punya dua faktor.

Setelah loop selesai, ada pengecekan khusus: kalau angkanya 1, otomatis dianggap bukan prima karena 1 cuma punya satu faktor.

Terakhir, program cuma tampilin semua faktor yang udah ketemu tadi, dan ngasih tahu apakah bilangan itu prima atau nggak berdasarkan pengecekan tadi.