

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE-IF



**Telkom
University
PURWOKERTO**

Disusun oleh:

Dharma Chandra Viriya

109082500052

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 - Menentukan Seseorang Bisa Membuat KTP Source Code

```
package main

import "fmt"

func main() {
    var umur int
    var memilikiKtp bool

    fmt.Print("Masukkan umur & apakah sudah memiliki ktp (true/false): ")

    fmt.Scan(&umur, &memilikiKtp)
    if umur >= 17 && memilikiKtp {
        fmt.Println("bisa membuat KTP")
    } else {
        fmt.Println("belum bisa membuat KTP")
    }
}
```

Screenshot program

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder structure under "PERTEMUAN10" containing "Guided1", "Guided2", "Guided3", "Soal1", "Soal2", and "Soal3", each with its own "main.go" file.
- Code Editor:** The "main.go" file in the "Guided1" folder is open. The code is identical to the one shown in the text block above.
- Terminal:** The terminal window shows three runs of the program:
 - Run 1: "Masukkan umur & apakah sudah memiliki ktp (true/false): 17 true" (User input 17, output true)
 - Run 2: "Masukkan umur & apakah sudah memiliki ktp (true/false): 20 false" (User input 20, output false)
 - Run 3: "Masukkan umur & apakah sudah memiliki ktp (true/false): 15 true" (User input 15, output true)
- Status Bar:** Shows the current file path as "iamcelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan10\$".

Deskripsi program

Kode program di atas merupakan kode program untuk menentukan apakah seseorang sudah dapat membuat KTP berdasarkan umur dan status kepemilikan KTP. Pada bagian awal kode, program mendeklarasikan package dengan inisialisasi main, kemudian mengimpor library "fmt" dari Go yang digunakan untuk menangani proses input dan output. Selanjutnya, pada function main, program mendeklarasikan dua buah variabel yaitu umur dengan tipe data int dan memilikiKtp dengan tipe data bool. Program kemudian meminta input dari user menggunakan fmt.Scanln untuk mengisi kedua variabel tersebut. Setelah itu, program melakukan pengecekan dengan menggunakan kondisi if untuk memeriksa apakah umur bernilai 17 atau lebih dan apakah user sudah memiliki KTP. Jika kedua kondisi tersebut terpenuhi, maka program akan mencetak pesan "bisa membuat KTP". Namun, jika salah satu atau kedua kondisi tidak terpenuhi, program akan mencetak pesan "belum bisa membuat KTP" ke console menggunakan fmt.Println.

2. Guided 2 - Menentukan Alfabet Adalah Vokal atau Konsonan

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var input string

    fmt.Print("Masukkan Alfabet: ")
    fmt.Scan(&input)

    alf := strings.ToUpper(input)

    if alf == "A" || alf == "I" || alf == "U" || alf == "E" || alf == "O" {
        fmt.Println("vokal")
    } else if alf >= "A" && alf <= "Z" {
        fmt.Println("konsonan")
    }
}
```

```

    } else {

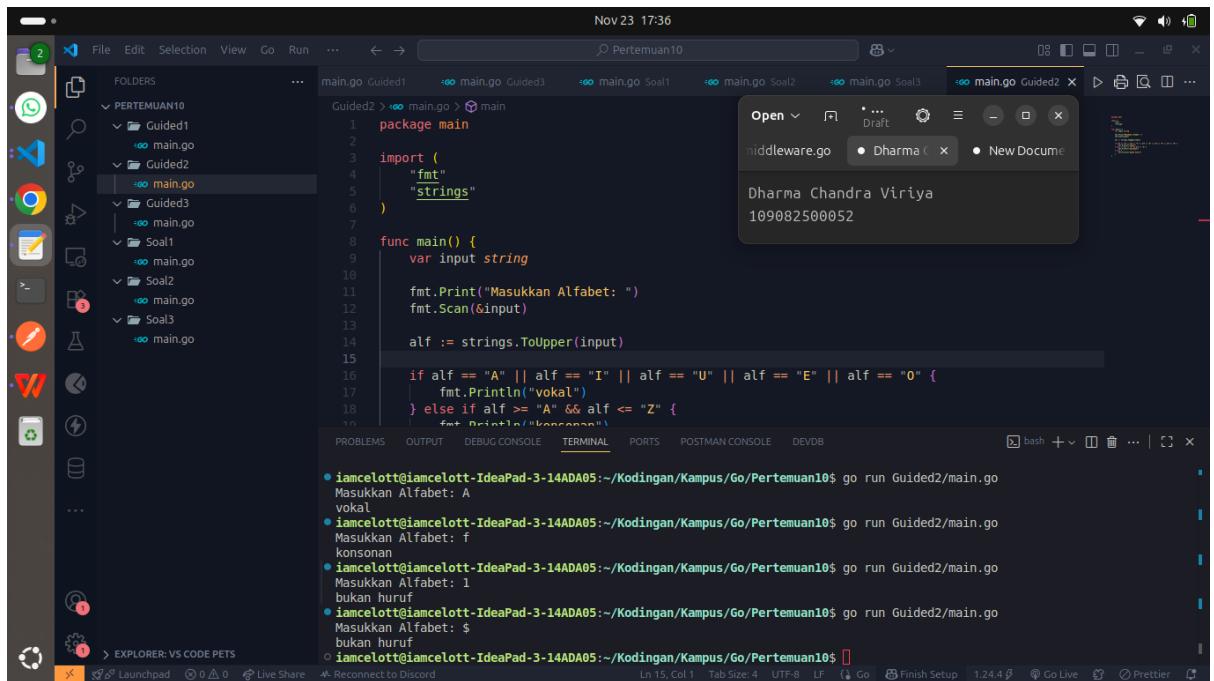
        fmt.Println("bukan huruf")

    }

}

```

Screenshot program



Deskripsi program

Kode program di atas digunakan untuk menentukan apakah input yang diberikan user merupakan huruf vokal, konsonan, atau bukan huruf. Pada awal program, package main diinisialisasi dan library "fmt" serta "strings" diimpor untuk keperluan input/output dan manipulasi teks. Di dalam function main, program mendeklarasikan variabel input bertipe string, kemudian menerima input dari user menggunakan fmt.Scan. Nilai input tersebut diubah menjadi huruf kapital menggunakan strings.ToUpper dan disimpan ke dalam variabel alf. Selanjutnya, program melakukan pengecekan menggunakan kondisi if. Jika alf merupakan salah satu huruf vokal ("A", "I", "U", "E", atau "O"), program mencetak "vokal". Jika alf masih berada dalam rentang huruf "A" sampai "Z", program mencetak "konsonan". Namun, jika tidak memenuhi kedua kondisi tersebut, program mencetak "bukan huruf".

3. Guided 3 - Menentukan Digit Bilangan Terurut Membesar, Mengcil, atau Tidak Berurut
- Source Code**

```
package main

import "fmt"

func main() {
    var n int
    var tampung []int
    var menaik bool = true
    var menurun bool = true

    fmt.Print("Masukkan n: ")
    fmt.Scan(&n)

    tampung = append(tampung, n%10)

    for i := len(fmt.Sprintf("%d", n)) - 1; i >= 1; i-
    {
        n = n / 10
        nTampung := n % 10

        tampung = append(tampung, nTampung)
    }

    for i := 0; i < len(tampung)-1; i++ {
        if tampung[i] < tampung[i+1] {
            menaik = false
        }
        if tampung[i] > tampung[i+1] {
            menurun = false
        }
    }
}
```

```

        if menaik {

            fmt.Println("Digit terurut membesar")

        } else if menurun {

            fmt.Println("Digit terurut mengecil")

        } else {

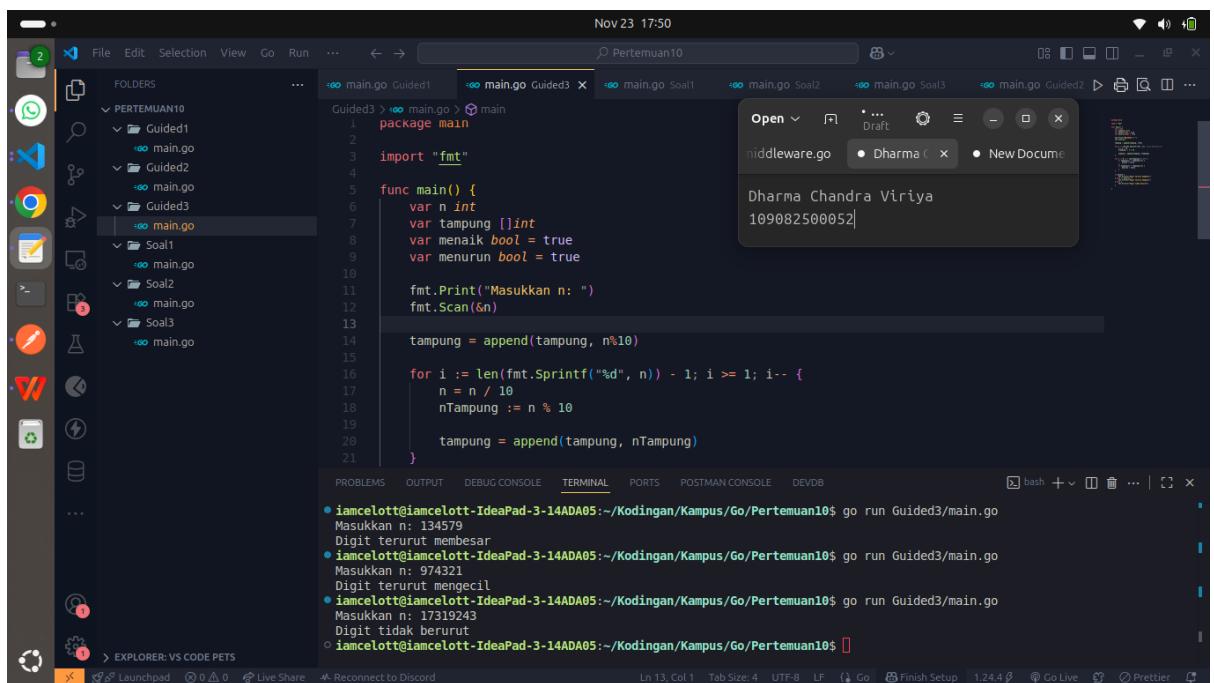
            fmt.Println("Digit tidak berurut")

        }

    }

```

Screenshot program



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a file tree under the 'PERTEMUAN10' folder. It contains several subfolders like 'Guided1', 'Guided2', 'Guided3', 'Soal1', 'Soal2', and 'Soal3', each containing a 'main.go' file.
- Code Editor (Center):** Displays a Go code snippet. The code reads an integer from the user, then iterates through its digits from right to left, appending them to a slice if they are in increasing order. It then prints the result.
- Terminal (Bottom):** Shows the output of running the program with different inputs. For example, with input 134579, it prints "Digit terurut membesar".
- Status Bar:** Shows the date and time (Nov 23 17:50), the current file path (~/.Kodingan/Kampus/Go/Pertemuan10/main.go), and various status icons.

Deskripsi program

Kode program di atas merupakan kode program untuk memeriksa apakah digit-digit pada sebuah angka tersusun secara menaik, menurun, atau tidak berurut. Pada awal

program, package main diinisialisasi dan library "fmt" diimpor untuk menangani input dan output. Di dalam function main, program mendeklarasikan variabel n sebagai bilangan input, tampung sebagai slice untuk menyimpan setiap digit, serta dua variabel boolean menaik dan menurun untuk menandai jenis urutan digit.

Program kemudian meminta user memasukkan nilai n menggunakan fmt.Scan, dan digit terakhir angka disimpan ke dalam slice tampung. Selanjutnya, program melakukan perulangan untuk memisahkan setiap digit angka n dari belakang ke depan, lalu menyimpannya ke dalam slice tersebut. Setelah semua digit tersimpan, program melakukan pengecekan berurutan dengan membandingkan setiap digit dengan digit setelahnya. Jika ditemukan digit yang lebih besar dari digit berikutnya, maka urutan menaik dianggap tidak valid. Sebaliknya, jika ditemukan digit yang lebih kecil dari digit berikutnya, maka urutan menurun dianggap tidak valid.

Pada bagian akhir, program mencetak hasil berdasarkan nilai variabel menaik dan menurun. Jika menaik bernilai true, program menampilkan "Digit terurut membesar". Jika menurun bernilai true, program menampilkan "Digit terurut mengecil". Namun, jika keduanya tidak terpenuhi, program mencetak "Digit tidak berurut".

TUGAS

1. Tugas 1 - Program BiayaPOS

Source code

```
package main
```

```
import "fmt"

func main() {
    var beratParsel int

    fmt.Println("Berat Parsel (gram): ")
    fmt.Scan(&beratParsel)

    if beratParsel <= 0 {
        fmt.Println("Berat parsel tidak boleh kurang atau sama dengan 0 (gram)")

        return
    }

    detailBeratParselKg, detailBeratParselGr := beratParsel/1000, beratParsel%1000
    detailBiayaKg, detailBiayaGr := detailBeratParselKg*10000, 0

    if detailBeratParselGr >= 500 {
        detailBiayaGr = detailBeratParselGr * 5
    } else {
        detailBiayaGr = detailBeratParselGr * 15
    }

    if beratParsel%1000 != 0 {
        fmt.Printf("Detail berat: %d kg + %d gr\n",
            detailBeratParselKg, detailBeratParselGr)
        fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n",
            detailBiayaKg, detailBiayaGr)
    } else {
        fmt.Printf("Detail berat: %d kg\n",
            detailBeratParselKg)
```

```

        fmt.Printf("Detail biaya: Rp. %d\n",
detailBiayaKg)

    }

totalBiaya := 0

if detailBeratParselKg < 10 {
    totalBiaya += detailBiayaKg + detailBiayaGr
} else {
    totalBiaya += detailBiayaKg
}

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

```

Screenshoot program

The screenshot shows a VS Code interface with multiple tabs open. The main editor tab contains the following Go code:

```

package main
import "fmt"
func main() {
    var beratParsel int
    fmt.Println("Berat Parsel (gram): ")
    fmt.Scan(&beratParsel)
    if beratParsel <= 0 {
        fmt.Println("Berat parcel tidak boleh kurang atau sama dengan 0 (gram)")
        return
    }
    detailBeratParselKg, detailBeratParselGr := beratParsel/1000, beratParsel%1000
    detailBiayaKg, detailBiayaGr := detailBeratParselKg*1000, 0
    if detailBeratParselGr >= 500 {
        detailBiayaGr = detailBeratParselGr * 5
    } else {

```

The terminal tab shows the output of running the program with two different inputs:

```

Berat Parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
Berat Parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 3750
Total biaya: Rp. 110000

```

Deskripsi program

Kode program di atas merupakan kode program untuk menghitung biaya pengiriman parsel berdasarkan berat yang dimasukkan oleh user. Pada awal program, package main diinisialisasi dan library "fmt" diimpor untuk keperluan input dan output. Di dalam function main, program mendeklarasikan variabel beratParsel bertipe int,

kemudian meminta user memasukkan berat dalam satuan gram menggunakan `fmt.Scan`. Program terlebih dahulu melakukan validasi untuk memastikan bahwa berat yang diinput tidak kurang atau sama dengan nol.

Selanjutnya, program memisahkan berat parsel menjadi kilogram dan gram menggunakan operasi pembagian dan modulo. Biaya dasar dihitung berdasarkan kilogram (`detailBiayaKg`), sedangkan biaya tambahan dihitung dari sisa gram (`detailBiayaGr`). Jika sisa gram lebih besar atau sama dengan 500, biaya per gram dihitung menggunakan tarif 5 rupiah; jika kurang dari 500, tarif yang digunakan adalah 15 rupiah per gram.

Program kemudian menampilkan detail berat dan biaya, baik untuk parsel yang memiliki sisa gram maupun yang beratnya bulat dalam kilogram. Setelah itu, program menghitung total biaya. Jika berat parsel kurang dari 10 kg, total biaya merupakan penjumlahan biaya kilogram dan biaya gram. Namun, jika beratnya mencapai atau melebihi 10 kg, total biaya hanya dihitung berdasarkan biaya kilogram. Terakhir, program mencetak total biaya yang harus dibayarkan.

2. Tugas 2 - Program NAM dan NMK

Source code module

```
package main

import "fmt"

func main() {
    var nam float64
```

```

var nmk string

fmt.Println("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)
}

```

Deskripsi program

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

Jawab: Jika nam diberikan adalah 80.1 maka keluaran dari program tersebut adalah "D", maka hasil tersebut tidaklah sesuai dengan spesifikasi soal.

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

Jawab: Terdapat beberapa kesalahan dari kode tersebut, berikut:

1. Penggunaan tanda kutip yang salah/tidak sesuai
2. Variable yang menampung nilai akhir seharusnya bukanlah nam, melainkan nmk
3. Flow pengkondisian menggunakan if yang salah, dimana tidak ada limit setiap kondisi, contoh nam > 72.5 && nam <= 80, sehingga ketika nam yang diinputkan > 40, maka hasil dari nmk akan selalu "D"

Alur program yang benar seharusnya:

1. Input nilai
2. Tampung nilai ke dalam variable nam
3. Cek dengan pengkondisian:

```

jika nam > 80 && nam <= 100 {
    nmk = "A"
}
jika nam > 72.5 && nam <= 80 {
    nmk = "AB"
}
jika nam > 65 && nam <= 72.5 {
    nmk = "B"
}
jika nam > 57.5 && nam <= 65 {
    nmk = "BC"
}
jika nam > 50 && nam <= 57.5 {
    nmk = "C"
}
jika nam > 40 && nam <= 50 {
    nmk = "D"
}
lain jika nam <= 40 {
    nmk = "E"
}

```

4. Print hasil nmk

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

Jawab:

Source code perbaikan kode

```

package main

import "fmt"

func main() {
    var nam float64

```

```
var nmk string  
fmt.Println("Nilai akhir mata kuliah: ")  
fmt.Scan(&nam)  
if nam > 80 && nam <= 100 {  
    nmk = "A"  
}  
if nam > 72.5 && nam <= 80 {  
    nmk = "AB"  
}  
if nam > 65 && nam <= 72.5 {  
    nmk = "B"  
}  
if nam > 57.5 && nam <= 65 {  
    nmk = "BC"  
}  
if nam > 50 && nam <= 57.5 {  
    nmk = "C"  
}  
if nam > 40 && nam <= 50 {  
    nmk = "D"  
} else if nam <= 40 {  
    nmk = "E"  
}  
fmt.Println("Nilai mata kuliah: ", nmk) }
```

Screenshot program

A screenshot of the Visual Studio Code (VS Code) interface. The title bar shows "Nov 23 18:31". The left sidebar has a "FOLDERS" tree with several projects like "PERTEMUAN10", "Guided1", "Guided2", "Guided3", "Soal1", "Soal2", and "Soal3". The main editor area displays a Go file named "main.go" with the following code:

```
32 import "fmt"
33
34 func main() {
35     var nam float64
36     var nmk string
37     fmt.Println("Nilai akhir mata kuliah: ")
38     fmt.Scan(&nam)
39     if nam > 80 && nam <= 100 {
40         nmk = "A"
41     }
42     if nam > 72.5 && nam <= 80 {
43         nmk = "AB"
44     }
45     if nam > 65 && nam <= 72.5 {
46         nmk = "B"
47     }
48     if nam > 57.5 && nam <= 65 {
49         nmk = "BC"
50     }
51     if nam > 50 && nam <= 57.5 {
52         nmk = "D"
53     }
54 }
```

The terminal below shows the execution of the code with different input values:

```
iamcelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan10$ go run Soal2/main.go
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
● iamcelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan10$ go run Soal2/main.go
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
● iamcelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan10$ go run Soal2/main.go
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
○ iamcelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan10$
```

3. Tugas 3 - Menentukan Faktor dan Prima Dari Suatu Bilangan

Source code

```
package main

import "fmt"

func main() {
    var b int
    var isPrima bool = true

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

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

```

        }

    }

    fmt.Println()

    if counter != 2 {

        isPrima = false

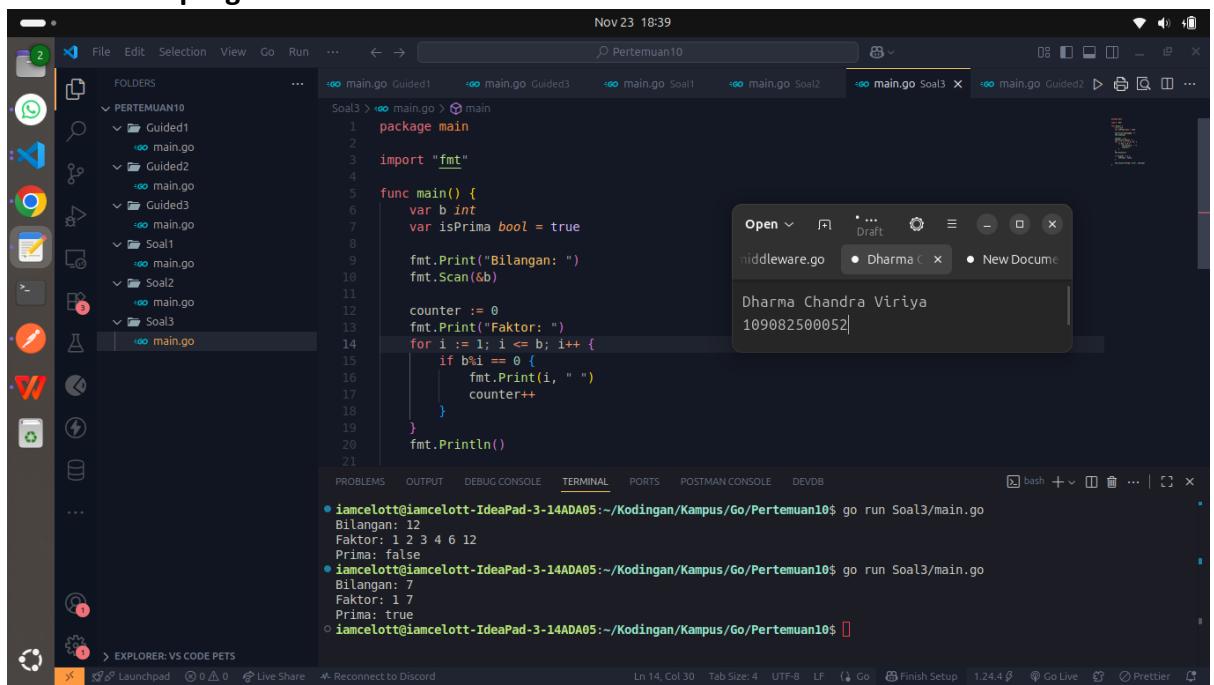
    }

    fmt.Printf("Prima: %t\n", isPrima)

}

```

Screenshot program



The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows a tree view of files under "PERTEMUAN10" and "Soal3".
- Code Editor:** Displays a Go file named "main.go" with the following code:


```

package main

import "fmt"

func main() {
    var b int
    var isPrima bool = true

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

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

```
- Terminal:** Shows command-line output for two runs of the program:


```

iamcelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan10$ go run Soal3/main.go
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
• iamcelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan10$ go run Soal3/main.go
Bilangan: 7
Faktor: 1 7
Prima: true
○ iamcelott@iamcelott-IdeaPad-3-14ADA05:~/Kodingan/Kampus/Go/Pertemuan10$ 

```
- Status Bar:** Shows the current line (Ln 14), column (Col 30), tab size (Tab Size: 4), encoding (UTF-8), line feed (LF), and other developer tools like Minify Setup, 1.24.4 beta, Go Live, and Prettier.

Deskripsi program

Kode program di atas merupakan kode program untuk menentukan apakah sebuah bilangan termasuk bilangan prima sekaligus menampilkan semua faktor dari bilangan tersebut. Pada awal program, package main diinisialisasi dan library "fmt" diimpor untuk menangani proses input dan output. Di dalam function main, program mendeklarasikan variabel b untuk menerima input bilangan dari user serta variabel boolean isPrima yang secara default bernilai true.

Program kemudian meminta user memasukkan sebuah bilangan menggunakan fmt.Scan. Setelah itu, program menginisialisasi variabel counter untuk menghitung jumlah faktor. Melalui perulangan for, program memeriksa setiap bilangan dari 1 hingga b dan menampilkan bilangan yang menjadi faktor dari b. Setiap kali ditemukan faktor, nilai counter akan bertambah.

Setelah seluruh faktor ditampilkan, program melakukan pengecekan apakah jumlah faktor tepat dua. Jika jumlahnya tidak sama dengan dua, maka variabel isPrima diubah menjadi false karena bilangan tersebut bukan bilangan prima. Terakhir, program menampilkan hasil pengecekan bilangan prima menggunakan fmt.Printf.