

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 10

ELSE-IF



Disusun oleh:

JEREMY CHRISTIAN SIHOMBING

109082500178

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.Print("Masukkan usia : ")

    fmt.Scan(&usia)

    fmt.Print("Apakah sudah punya kk (isi dengan  
true/false) : ")

    fmt.Scan(&kk)

    if usia >= 17 && kk {

        fmt.Println("bisa membuat ktp")

    } else {

        fmt.Println("belum bisa membuat ktp")

    }

}
```

Screenshoot program

```
EXPLORER
...
MODUL 10
  Guide1
  guide1.go
  Guide2
  guide2.go
  Guide3
  guide3.go

Guide1 > guide1.go > main
1 package main
2 import "fmt"
3 func main() {
4
5     var usia int
6     var kk bool
7
8     fmt.Print("Masukkan usia : ")
9     fmt.Scan(&usia)
10    fmt.Print("Apakah sudah punya kk (isi dengan true/false) : ")
11    fmt.Scan(&kk)
12
13    if usia >= 17 && kk {
14        fmt.Println("bisa membuat ktp")
15    } else {
16        fmt.Println("belum bisa membuat ktp")
17    }
18 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide1\guide1.go
Masukkan usia : 17
Apakah sudah punya kk (isi dengan true/false) : true
bisa membuat ktp
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide1\guide1.go
Masukkan usia : 20
Apakah sudah punya kk (isi dengan true/false) : false
belum bisa membuat ktp
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide1\guide1.go
Masukkan usia : 15
Apakah sudah punya kk (isi dengan true/false) : true
belum bisa membuat ktp
PS C:\Users\LENOVO\Downloads\Modul 10> 
```

Deskripsi program

Program ini digunakan untuk menentukan apakah seseorang sudah memenuhi syarat untuk membuat KTP. Ketika program dijalankan, pengguna diminta memasukkan usia dan menjawab apakah ia sudah memiliki Kartu Keluarga (KK) dengan nilai boolean *true* atau *false*. Setelah kedua input diterima, program memeriksa dua kondisi sekaligus: apakah usia pengguna minimal 17 tahun dan apakah ia sudah memiliki KK. Jika kedua syarat tersebut terpenuhi, program menampilkan pesan "bisa membuat ktp". Namun, jika salah satu atau kedua kondisi tidak terpenuhi, program memberikan output "belum bisa membuat ktp". Dengan demikian, program memastikan kelayakan pembuatan KTP berdasarkan usia dan kepemilikan KK.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var huruf rune

    fmt.Print("Masukkan sebuah huruf : ")

    fmt.Scanf("%c", &huruf)

    if huruf == 'A' || huruf == 'I' || huruf == 'U' ||
huruf == 'E' || huruf == 'O' ||

        huruf == 'a' || huruf == 'i' || huruf == 'u' ||
huruf == 'e' || huruf == 'o' {

        fmt.Println("Vokal")

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

        fmt.Println("Konsonan")

    } else {

        fmt.Println("Bukan huruf")

    }

}
```

Screenshoot program

The screenshot shows a Go IDE with three files: guide1.go, guide2.go, and guide3.go. The active file is guide2.go, which contains the following code:

```
1 package main
2 import "fmt"
3 func main() {
4
5     var huruf rune
6
7     fmt.Print("Masukkan sebuah huruf : ")
8     fmt.Scanf("%c", &huruf)
9
10    if huruf == 'A' || huruf == 'I' || huruf == 'U' || huruf == 'E' || huruf == 'O' ||
11        huruf == 'a' || huruf == 'i' || huruf == 'u' || huruf == 'e' || huruf == 'o' {
12        fmt.Println("Vokal")
13    } else if (huruf >= 'A' && huruf <= 'Z') || (huruf >= 'a' && huruf <= 'z') {
14        fmt.Println("Konsonan")
15    } else {
16        fmt.Println("Bukan huruf")
17    }
18 }
```

The terminal output shows the program being run three times with different inputs:

```
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide2\guide2.go
Masukkan sebuah huruf : A
Vokal
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide2\guide2.go
Masukkan sebuah huruf : f
Konsonan
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide2\guide2.go
Masukkan sebuah huruf : 1
Bukan huruf
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide2\guide2.go
Masukkan sebuah huruf : $
Bukan huruf
PS C:\Users\LENOVO\Downloads\Modul 10>
```

A floating window on the right displays the following information:

```
NIM 10908
KELAS S1IF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING
```

The status bar at the bottom of the floating window shows: Ln 3, Col 32 | 65 character | Plain t | 100% | Wind UTF-8

Deskripsi program

Program ini digunakan untuk menentukan apakah karakter yang dimasukkan pengguna merupakan huruf vokal, huruf konsonan, atau bukan huruf. Ketika program dijalankan, pengguna diminta memasukkan satu karakter. Nilai yang diterima disimpan dalam variabel bertipe *rune* agar dapat diproses sebagai karakter tunggal.

Setelah input diterima, program melakukan pengecekan pertama dengan membandingkan karakter tersebut dengan seluruh huruf vokal, baik huruf besar maupun huruf kecil (A, I, U, E, O, dan versi kecilnya). Jika karakter cocok dengan salah satu di antaranya, program menampilkan output "Vokal".

Jika bukan vokal, program melanjutkan pengecekan kedua untuk melihat apakah karakter tersebut masih termasuk dalam rentang huruf alfabet, baik huruf besar maupun kecil. Apabila karakter berada dalam rentang tersebut tetapi bukan vokal, maka program menyatakan karakter tersebut sebagai "Konsonan".

Jika kedua kondisi sebelumnya tidak terpenuhi, berarti karakter yang dimasukkan bukan huruf, sehingga program menampilkan pesan "Bukan huruf".

3. Guided 3

Source Code

```
package main

import "fmt"

func main(){

    var bilangan,d1, d2, d3, d4 int

    var urutan string

    fmt.Print("masukkan bilangan 4 digit :")

    fmt.Scan(&bilangan)

    d1 = bilangan / 1000

    d2 = (bilangan % 1000) / 100

    d3 = (bilangan % 100) / 10

    d4 = bilangan % 10

    if d1 < d2 && d2 < d3 && d3 < d4 {

        urutan = "terurut membesar"

    } else if d1 > d2 && d2 > d3 && d3 > d4 {

        urutan = "terurut mengecil"

    } else {

        urutan = "tidak terurut"

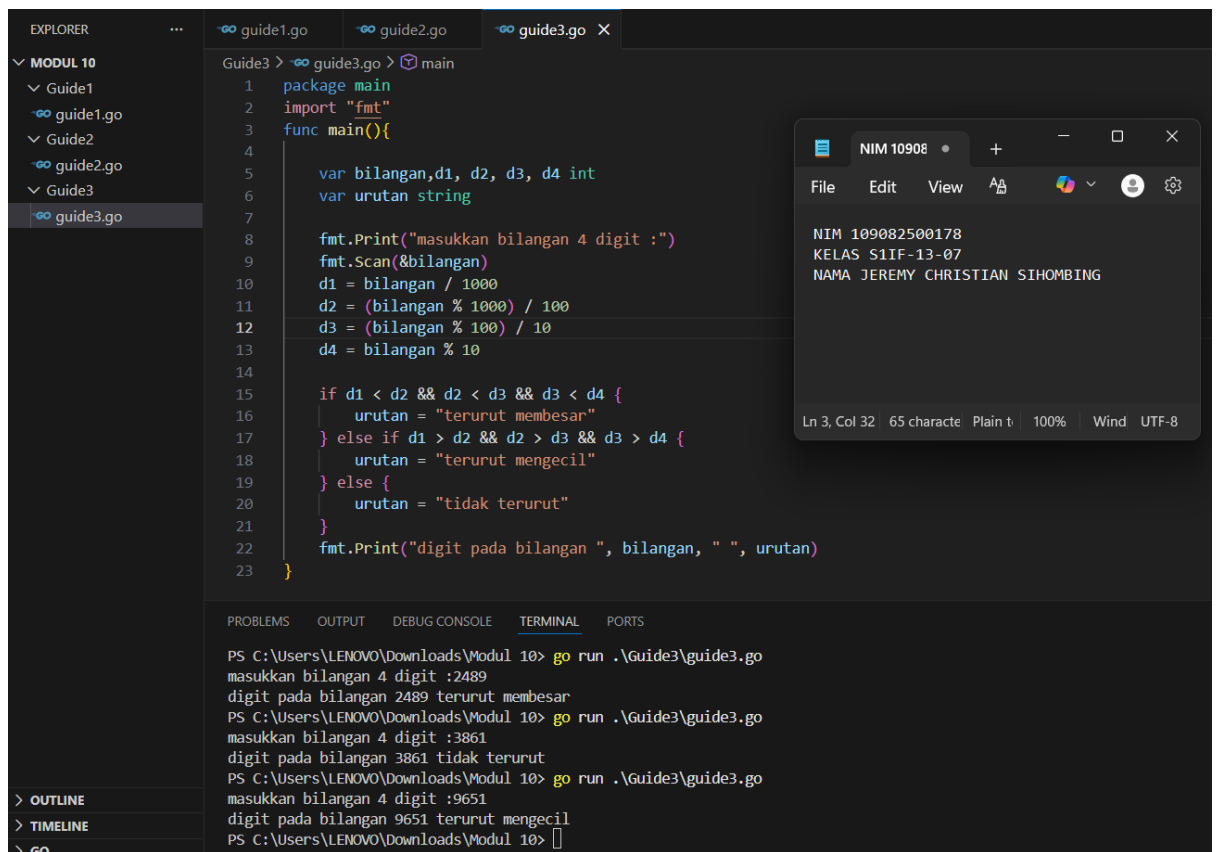
    }

    fmt.Print("digit pada bilangan ", bilangan, " ",

urutan)

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk mengevaluasi apakah susunan empat digit pada sebuah bilangan membentuk urutan yang membesar, mengecil, atau tidak teratur. Ketika program dijalankan, pengguna diminta memasukkan sebuah bilangan yang terdiri dari empat digit. Setelah input diperoleh, program memisahkan setiap digit dengan membagi dan mengambil sisa pembagian dari bilangan tersebut. Digit pertama diambil dengan membagi bilangan dengan 1000, digit kedua diperoleh dari sisa 1000 dibagi 100, digit ketiga diambil dari sisa 100 dibagi 10, dan digit keempat diperoleh dari sisa pembagian 10.

Setelah seluruh digit berhasil diambil, program membandingkan digit-digit tersebut. Jika setiap digit lebih kecil dari digit berikutnya ($d1 < d2 < d3 < d4$), program menentukan bahwa bilangan tersebut memiliki pola "terurut membesar". Jika setiap digit justru lebih besar dari digit berikutnya ($d1 > d2 > d3 > d4$), maka pola yang ditampilkan adalah "terurut mengecil". Namun, apabila tidak memenuhi kedua pola tersebut, program menyimpulkan bahwa digit-digit pada bilangan tersebut "tidak teratur". Program kemudian menampilkan hasil akhir berupa pola urutan digit dari bilangan yang dimasukkan.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var beratParsel, kg, gram, biayaKg, biayaSisa, total 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

    }

    total = biayaKg + biayaSisa

    fmt.Println("Detail berat:", kg, "kg +", gram, "gr")

    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.",
biayaSisa)

    fmt.Println("Total biaya: Rp.", total)

}
```

Screenshoot program

The screenshot displays an IDE with the following components:

- EXPLORER:** A file tree on the left showing a project structure under 'MODUL 10' with files 'Guide1', 'Guide2', 'Guide3', 'Soal1', 'soal1.go', 'Soal2', and 'soal2.go'.
- Editor:** The main workspace showing the code for 'soal1.go'. The code defines a function 'main()' that takes a weight in grams and calculates the total cost based on whether the weight is in kilograms or grams.
- Terminal:** A terminal window at the bottom showing the execution of the program. It displays the input weight and the calculated total cost for three different inputs: 8500g, 9250g, and 11750g.
- Output:** A small window on the right side of the editor showing the output of the program, which includes the NIM number, class name, and student name.

```
1 package main
2 import "fmt"
3 func main() {
4     var beratParsel, kg, gram, biayaKg, biayaSisa, total int
5     fmt.Print("Berat parsel (gram): ")
6     fmt.Scan(&beratParsel)
7     kg = beratParsel / 1000
8     gram = beratParsel % 1000
9     biayaKg = kg * 10000
10    if kg >= 10 {
11        biayaSisa = 0
12    } else if gram >= 500 {
13        biayaSisa = gram * 5
14    } else {
15        biayaSisa = gram * 15
16    }
17    total = biayaKg + biayaSisa
18    fmt.Println("Detail berat:", kg, "kg +", gram, "gr")
19    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.", biayaSisa)
20    fmt.Println("Total biaya: Rp.", total)
21 }
```

PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal1\soal1.go
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal1\soal1.go
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal1\soal1.go
Berat parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000

NIM 109082500178
KELAS SIIF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING

Deskripsi program

Program ini digunakan untuk menghitung biaya pengiriman parcel berdasarkan berat total dalam satuan gram. Ketika program dijalankan, pengguna diminta memasukkan berat parcel. Program kemudian memisahkan berat tersebut menjadi dua bagian, yaitu berat dalam kilogram dan sisa berat dalam gram. Nilai kilogram diperoleh dengan membagi total gram dengan 1000, sedangkan sisanya dihitung menggunakan operasi modulo.

Setelah berat terbagi, program menghitung biaya berdasarkan ketentuan tertentu. Biaya dasar dihitung dari jumlah kilogram dikalikan dengan tarif Rp10.000 per kilogram. Untuk sisa berat, program menerapkan aturan tambahan: jika berat kilogram mencapai 10 kg atau lebih, tidak ada biaya tambahan untuk sisa gram. Jika sisa gram kurang dari 500, biaya dihitung dengan tarif Rp15 per gram. Namun, jika sisa gram mencapai 500 gram atau lebih, tarifnya menjadi Rp5 per gram. Seluruh biaya ini kemudian dijumlahkan untuk memperoleh total biaya pengiriman.

Terakhir, program menampilkan rincian berat dalam bentuk kilogram dan gram, rincian biaya berdasarkan perhitungan tarif, serta total biaya keseluruhan yang harus dibayar oleh pengguna.

2. Tugas 2

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

Jika inputnya adalah 80.1, program tidak akan menghasilkan output apa pun karena akan muncul Error (Kegagalan Kompilasi). Alasannya, variabel nam dideklarasikan sebagai float64 (angka desimal), tetapi dalam blok if, kode berusaha memasukkan string ("A", "AB", dan seterusnya) ke dalam variabel tersebut. Di bahasa Go, ini tidak diperbolehkan (Ketidakcocokan Tipe). Kesesuaian Spesifikasinya, program tidak memenuhi spesifikasi karena adanya kesalahan dalam sintaks (aturan bahasa pemrograman) dan logika.

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

1. Kesalahan Tipe Data (Ketidakcocokan Tipe):

- Salah: `nam = "A"`. Penjelasannya, `nam` merupakan tempat penyimpanan angka (float), sehingga tidak dapat diisi dengan huruf. Seharusnya, huruf grade disimpan di variabel `nmk` (yang bertipe string).

2. Kesalahan Logika Percabangan (If Mandiri):

- Salah: Memakai beberapa if yang terpisah tanpa else.

Penjelasannya, dengan if terpisah, program akan mengevaluasi semua kondisi.

- Contoh skenario: Jika nilai 90.

- `if nam > 80`: Terpenuhi -> `nmk` diubah menjadi "A".

- `if nam > 72.5`: Terpenuhi juga (karena $90 > 72.5$) -> `nmk` diganti lagi menjadi "AB".

- ... dan begitu seterusnya hingga kondisi terakhir.

- Dampaknya, nilai tinggi malah akan mendapat grade terendah yang masih memenuhi syarat (seperti "D").

3. Kesalahan Variabel Output:

- Salah: `fmt.Println(..., nmk)`. Penjelasannya, Variabel `nmk` yang dicetak di akhir, padahal dalam logika if yang dimodifikasi adalah variabel `nam`. Akibatnya, `nmk` akan tetap kosong.

Alur yang Benar: Program seharusnya menerapkan struktur `if - else if - else`. Setelah satu kondisi terpenuhi (misalnya > 80), program harus menghentikan pemeriksaan kondisi

selanjutnya agar grade "A" tidak digantikan. Selain itu, hasil grade harus disimpan ke variabel nmk.

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 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

The screenshot shows a Go IDE with the following components:

- EXPLORER:** A sidebar on the left showing a project structure with folders 'MODUL 10', 'Guide1', 'Guide2', 'Guide3', 'Soal1', 'Soal2', and 'Soal3a'. The file 'soal2.go' is selected.
- Source Code:** The main editor displays the code for 'soal2.go':


```

1 package main
2 import "fmt"
3 func main() {
4     var nam float64
5     var nmk string
6     fmt.Print("Nilai akhir mata kuliah: ")
7     fmt.Scan(&nam)
8     if nam > 80 {
9         nmk = "A"
10    } else if nam > 72.5 {
11        nmk = "AB"
12    } else if nam > 65 {
13        nmk = "B"
14    } else if nam > 57.5 {
15        nmk = "BC"
16    } else if nam > 50 {
17        nmk = "C"
18    } else if nam > 40 {
19        nmk = "D"
20    } else {
21        nmk = "E"
22    }
23    fmt.Println("Nilai mata kuliah:", nmk)
24 }

```
- Terminal:** The bottom panel shows the output of running the program three times:


```

PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal2\soal2.go
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal2\soal2.go
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal2\soal2.go
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\Users\LENOVO\Downloads\Modul 10>

```
- Output Window:** A small window titled 'NIM 10908' is open, displaying the text:


```

NIM 109082500178
KELAS S1IF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING

```

Deskripsi program

Program ini digunakan untuk menentukan Nilai Mutu Kuliah (NMK) berdasarkan Nilai Akhir Mata Kuliah (NAM) yang dimasukkan oleh pengguna. Program menerima masukan berupa satu bilangan real (float64) yang mewakili nilai akhir mahasiswa.

Setelah nilai NAM dibaca, program membandingkan nilai tersebut dengan rentang standar penilaian. Pemeriksaan dilakukan menggunakan struktur if - else if - else, sehingga hanya satu kondisi yang akan dipilih sebagai hasil akhir. Berdasarkan rentang nilai tersebut, program akan menentukan huruf mutu yang sesuai, mulai dari A untuk nilai yang lebih dari 80, sampai E untuk nilai yang kurang atau sama dengan 40.

Pada akhir program, NMK yang telah ditentukan akan ditampilkan ke layar.

3. 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

The screenshot displays a Go IDE interface. On the left, the Explorer pane shows a project structure with 'MODUL 10' and several files including 'soal3a.go'. The main editor shows the source code of 'soal3a.go', which is identical to the code provided in the 'Source code' section. On the right, a window titled 'NIM 10908' displays personal information: 'NIM 109082500178', 'KELAS S1IF-13-07', and 'NAMA JEREMY CHRISTIAN SIHOMBING'. At the bottom, the Terminal pane shows the execution of the program. It displays the prompt 'PS C:\Users\LENOVO\Downloads\Modul 10>' followed by the command 'go run .\Soal3a\soal3a.go'. The output shows the program running twice: first with 'bilangan : 12' and factors '1 2 3 4 6 12', and then with 'bilangan : 7' and factor '1 7'.

```
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, " ")
        }
    }
}
```

PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal3a\soal3a.go
bilangan : 12
Faktor : 1 2 3 4 6 12
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal3a\soal3a.go
bilangan : 7
Faktor : 1 7
PS C:\Users\LENOVO\Downloads\Modul 10>

Deskripsi program

Program ini digunakan untuk menampilkan seluruh faktor dari sebuah bilangan bulat positif. Program diawali dengan meminta pengguna memasukkan satu bilangan bulat. Setelah bilangan tersebut dibaca, program menampilkan teks "Faktor : " sebagai penanda bahwa faktor-faktor bilangan akan ditampilkan.

Untuk menentukan faktor, program menggunakan sebuah perulangan for yang berjalan dari 1 hingga nilai bilangan tersebut. Pada setiap iterasi, program memeriksa apakah nilai pembagi i dapat membagi habis bilangan (dengan memeriksa apakah $\text{bilangan} \% i == 0$). Jika kondisi tersebut benar, maka angka i adalah salah satu faktor dari bilangan, dan angka tersebut langsung dicetak ke layar.

Dengan cara ini, seluruh faktor dari bilangan akan ditampilkan secara berurutan mulai dari 1 sampai bilangan itu sendiri.

4. Tugas 3b

Source code

```
package main
import "fmt"
func main() {
    var bilangan int
    var prima bool
    var jmlFaktor int = 0
    fmt.Print("Bilangan : ")
    fmt.Scan(&bilangan)
    fmt.Print("Faktor : ")
    for i := 1; i <= bilangan; i++ {
        if bilangan%i == 0 {
            fmt.Print(i, " ")
            jmlFaktor++
        }
    }
    fmt.Println()
    if jmlFaktor == 2 {
        prima = true
    } else {
        prima = false
    }
    fmt.Println("Prima:", prima)
}
```

Screenshoot program

```
Soal3b > soal3b.go > main
1 package main
2 import "fmt"
3 func main() {
4     var bilangan int
5     var prima bool
6     var jmlFaktor int = 0
7     fmt.Print("Bilangan : ")
8     fmt.Scan(&bilangan)
9     fmt.Print("Faktor : ")
10    for i := 1; i <= bilangan; i++ {
11        if bilangan%i == 0 {
12            fmt.Print(i, " ")
13            jmlFaktor++
14        }
15    }
16    fmt.Println()
17    if jmlFaktor == 2 {
18        prima = true
19    } else {
20        prima = false
21    }
22    fmt.Println("Prima:", prima)
23 }
```

PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal3b\soal3b.go
Bilangan : 12
Faktor : 1 2 3 4 6 12
Prima: false
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal3b\soal3b.go
Bilangan : 7
Faktor : 1 7
Prima: true
PS C:\Users\LENOVO\Downloads\Modul 10>

NIM 10908
NIM 109082500178
KELAS S1IF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING

Deskripsi program

Program ini digunakan untuk menentukan faktor-faktor dari sebuah bilangan serta mengecek apakah bilangan tersebut merupakan bilangan prima atau bukan. Pertama, program meminta pengguna memasukkan sebuah bilangan bulat, kemudian menampilkan teks "Faktor : " sebagai penanda bahwa faktor-faktor bilangan akan ditampilkan.

Program menggunakan sebuah perulangan for yang berjalan dari 1 hingga nilai bilangan. Pada setiap nilai i, program memeriksa apakah $\text{bilangan} \% i == 0$. Jika kondisi tersebut benar, berarti i adalah salah satu faktor dari bilangan tersebut. Angka ini langsung dicetak, dan variabel penghitung faktor (jmlFaktor) ditambah satu.

Setelah seluruh faktor ditampilkan, program memeriksa apakah jumlah faktor tepat berjumlah 2. Bilangan yang memiliki tepat dua faktor (yaitu 1 dan dirinya sendiri) dikategorikan sebagai bilangan prima, sehingga variabel prima diisi dengan nilai true. Jika tidak, maka prima bernilai false.

Terakhir, program menampilkan hasil pengecekan tersebut melalui tulisan "Prima: true" atau "Prima: false"

.