

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 09  
ELSE - IF**



**Disusun Oleh :**

**NAMA : IMROATUN SHOLIKHA**

**NIM : 109082500111**

**Asisten Praktikum**

- Adithana Dharma Putra
- Renisa Assyifa Putri

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

# TUGAS

## 1. Tugas 1

### Source code

```
package main

import "fmt"

func main() {
    var beratParsel, totalBiaya, biayaPerGram, biayaTambahan int
    fmt.Print("Berat Parsel: ")
    fmt.Scan(&beratParsel)

    beratKg := beratParsel / 1000
    sisaGram := beratParsel % 1000
    biayaBerat := beratKg * 10000

    gratisSisa := beratParsel >= 10000

    if sisaGram > 0 && !gratisSisa {
        biayaPerGram = 0
        if sisaGram >= 500 {
            biayaPerGram = 5
            biayaTambahan = sisaGram * biayaPerGram
        } else {
            biayaPerGram = 15
            biayaTambahan = sisaGram * biayaPerGram
        }
    }

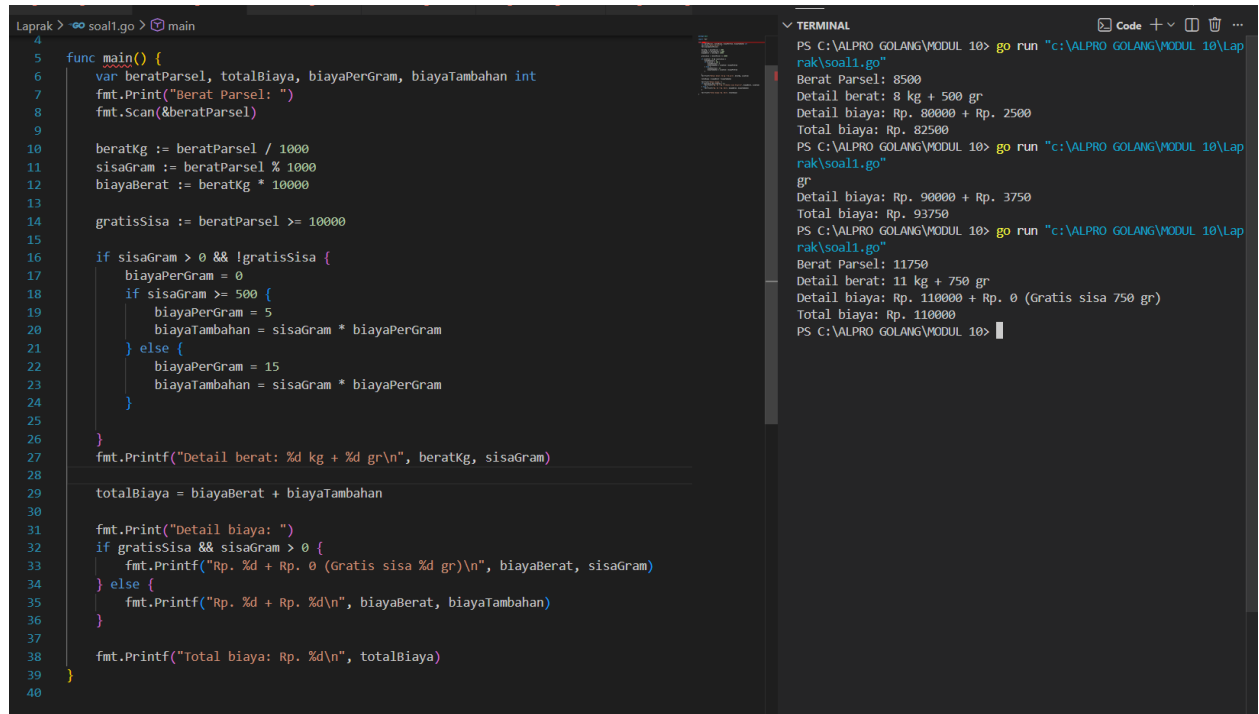
    fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg,
sisaGram)

    totalBiaya = biayaBerat + biayaTambahan

    fmt.Print("Detail biaya: ")
    if gratisSisa && sisaGram > 0 {
        fmt.Printf("Rp. %d + Rp. 0 (Gratis sisa %d gr)\n",
biayaBerat, sisaGram)
    } else {
        fmt.Printf("Rp. %d + Rp. %d\n", biayaBerat,
biayaTambahan)
    }

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

## Screenshoot program



The screenshot shows a Go program in a text editor and its execution in a terminal. The program calculates shipping costs based on weight and a free allowance.

```
4
5 func main() {
6     var beratParsel, totalBiaya, biayaPerGram, biayaTambahan int
7     fmt.Print("Berat Parsel: ")
8     fmt.Scan(&beratParsel)
9
10    beratKg := beratParsel / 1000
11    sisaGram := beratParsel % 1000
12    biayaBerat := beratKg * 10000
13
14    gratisSisa := beratParsel >= 10000
15
16    if sisaGram > 0 && !gratisSisa {
17        biayaPerGram = 0
18        if sisaGram >= 500 {
19            biayaPerGram = 5
20            biayaTambahan = sisaGram * biayaPerGram
21        } else {
22            biayaPerGram = 15
23            biayaTambahan = sisaGram * biayaPerGram
24        }
25    }
26
27    fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, sisaGram)
28
29    totalBiaya = biayaBerat + biayaTambahan
30
31    fmt.Print("Detail biaya: ")
32    if gratisSisa && sisaGram > 0 {
33        fmt.Printf("Rp. %d + Rp. 0 (Gratis sisa %d gr)\n", biayaBerat, sisaGram)
34    } else {
35        fmt.Printf("Rp. %d + Rp. %d\n", biayaBerat, biayaTambahan)
36    }
37
38    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
39 }
40
```

The terminal output shows two test cases:

```
PS C:\ALPRO GOLANG\MODUL 10> go run "c:\ALPRO GOLANG\MODUL 10\Lap
rak\soal1.go"
Berat Parsel: 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\ALPRO GOLANG\MODUL 10> go run "c:\ALPRO GOLANG\MODUL 10\Lap
rak\soal1.go"
Berat Parsel: 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0 (Gratis sisa 750 gr)
Total biaya: Rp. 110000
PS C:\ALPRO GOLANG\MODUL 10>
```

## Deskripsi program

Program ini menggunakan bahasa pemrograman golang, dengan yang pertama package main dan func main sebagai titik eksekusi program dan mengimport “fmt” yang menangani proses input dan output.

Selanjutnya, program mendeklarasikan beberapa variabel bilangan bulat (integer), termasuk variabel untuk berat parsel, total biaya, biaya per gram, dan biaya tambahan. Program meminta dan menerima data masukan berupa berat parsel dalam satuan gram dari pengguna menggunakan fmt.Scan.

Berat dalam kilogram (beratKg) dihitung dengan membagi berat total parsel dengan 1000. Sisa berat dalam gram (sisaGram) dihitung menggunakan operator modulus, yaitu sisa dari pembagian berat total dengan 1000. Biaya untuk berat kilogram (biayaBerat) dihitung dengan mengalikan berat dalam kg dengan tarif Rp 10.000 per kg. Sebuah penanda boolean (gratisSisa) diatur menjadi benar (true) jika berat parsel total mencapai 10.000 gram atau lebih.

Program mengecek sebuah kondisi utama, yaitu apakah terdapat sisa berat dalam gram AND parsel tersebut belum mencapai berat minimum 10.000 gram yang memberikannya fasilitas sisa gratis. Jika kondisi ini terpenuhi, program akan melanjutkan untuk menentukan biaya tambahan berdasarkan jumlah sisa gram yang ada: apabila sisa gram sama dengan atau lebih dari 500 gram, biaya per gram ditetapkan Rp 5, dan biaya tambahan dihitung; namun, jika sisa gram kurang dari

500 gram, biaya per gram akan ditetapkan sebesar Rp 15, dan perhitungan biaya tambahan akan dilakukan.

Program menampilkan detail berat parcel (dalam kg dan gram). Kemudian, total biaya dihitung sebagai penjumlahan dari biaya per kilogram dan biaya tambahan.

Pada bagian Keluaran Detail Biaya, program akan menampilkan biaya per kilogram ditambah biaya tambahan. Khusus untuk parcel yang memenuhi syarat gratis sisa, biaya tambahannya akan ditampilkan sebagai Rp 0 disertai keterangan "(Gratis sisa ... gr)". Setelah semua proses selesai dijalankan, program menampilkan total biaya keseluruhan yang harus dibayar.

## 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)
}
```

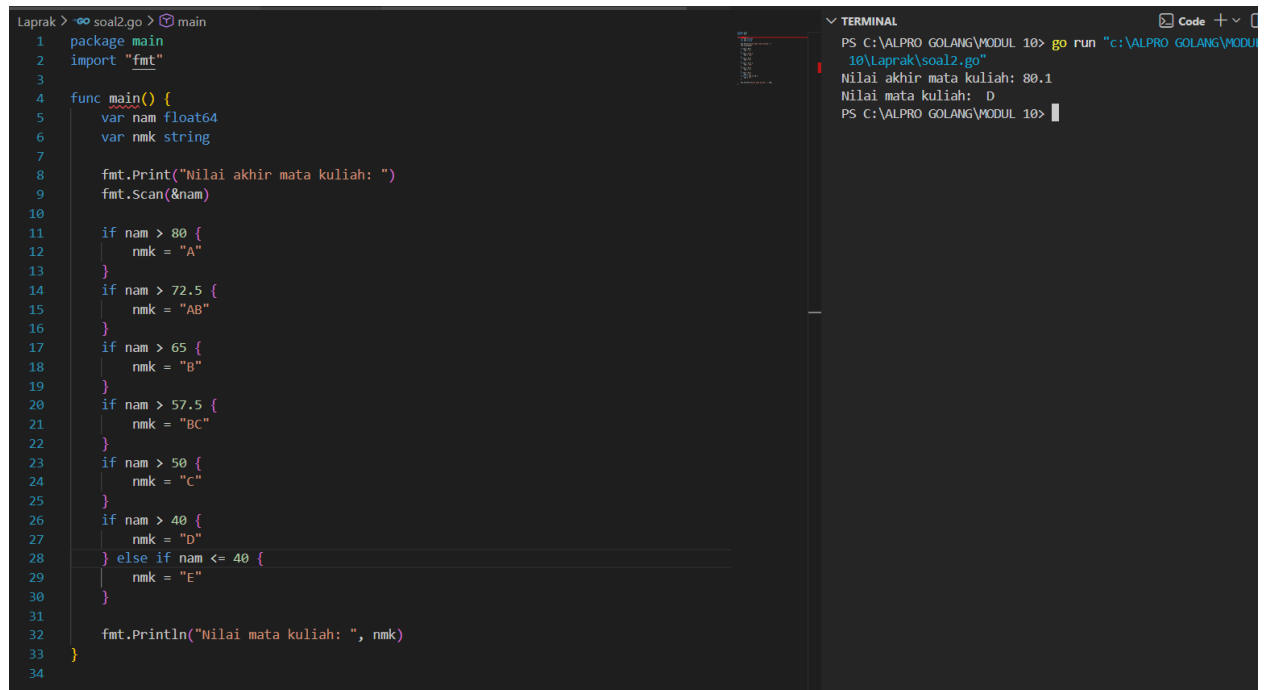
Jawablah pertanyaan-pertanyaan berikut:

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

Jawab : Tidak sesuai dengan spesifikasi soal. Pertama program tersebut tidak akan bisa dijalankan karena keluarannya adalah stringnya kan nmk bukan nam dan harus diganti terlebih dahulu , selanjutnya menurut program diatas, jika  $80 < \text{NAM} \leq 100$ , seharusnya NMK adalah "A".

Karena input 80.1 seharusnya menghasilkan “A” tetapi program mengeluarkan “D” eksekusi program ini salah.

## Screenshoot program



The screenshot shows a Go program in a text editor and its execution in a terminal. The program is a simple grading system that takes a score as input and outputs a letter grade. The logic is as follows:

```
1 package main
2 import "fmt"
3
4 func main() {
5     var nam float64
6     var nmk string
7
8     fmt.Print("Nilai akhir mata kuliah: ")
9     fmt.Scan(&nam)
10
11     if nam > 80 {
12         nmk = "A"
13     }
14     if nam > 72.5 {
15         nmk = "AB"
16     }
17     if nam > 65 {
18         nmk = "B"
19     }
20     if nam > 57.5 {
21         nmk = "BC"
22     }
23     if nam > 50 {
24         nmk = "C"
25     }
26     if nam > 40 {
27         nmk = "D"
28     } else if nam <= 40 {
29         nmk = "E"
30     }
31
32     fmt.Println("Nilai mata kuliah: ", nmk)
33 }
34
```

The terminal output shows the program being run with the input 80.1, resulting in the output "D".

```
PS C:\ALPRO GOLANG\MODUL 10> go run "c:\ALPRO GOLANG\MODUL 10\Laprak\soal2.go"
Nilai akhir mata kuliah: 80.1
Nilai mata kuliah: D
PS C:\ALPRO GOLANG\MODUL 10>
```

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

Jawab : Masalah logika, Karena setiap blok if dievaluasi terpisah, program tidak menghentikan pengecekan setelah menemukan kondisi yang benar. Sebagai contoh, untuk 80.1, ia memenuhi  $nam > 80$ , tetapi program tetap memeriksa dan memenuhi  $nam > 72.5$ ,  $nam > 65$ , dan seterusnya, yang mengakibatkan variabel nmk ditimpa berulang kali hingga mendapatkan nilai yang salah ("D") dari kondisi terakhir yang terpenuhi.

Alur program yang seharusnya adalah :

Untuk mengkonversi nilai berdasarkan rentang, program seharusnya menggunakan struktur if-else if yang memungkinkan pemeriksaan kondisi secara eksklusif. Alur yang benar adalah memeriksa dari batas nilai tertinggi ke terendah, seperti:

1. If  $nam > 80$  (Mendapatkan 'A').
2. Else if  $nam > 72.5$  (Jika kondisi sebelumnya salah, cek apakah  $72.5 < NAM \leq 80$  untuk 'AB').
3. Else if  $nam > 65$  (Jika kondisi sebelumnya salah, cek apakah  $65 < NAM \leq 72.5$  untuk 'B').

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

Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

Jawab :

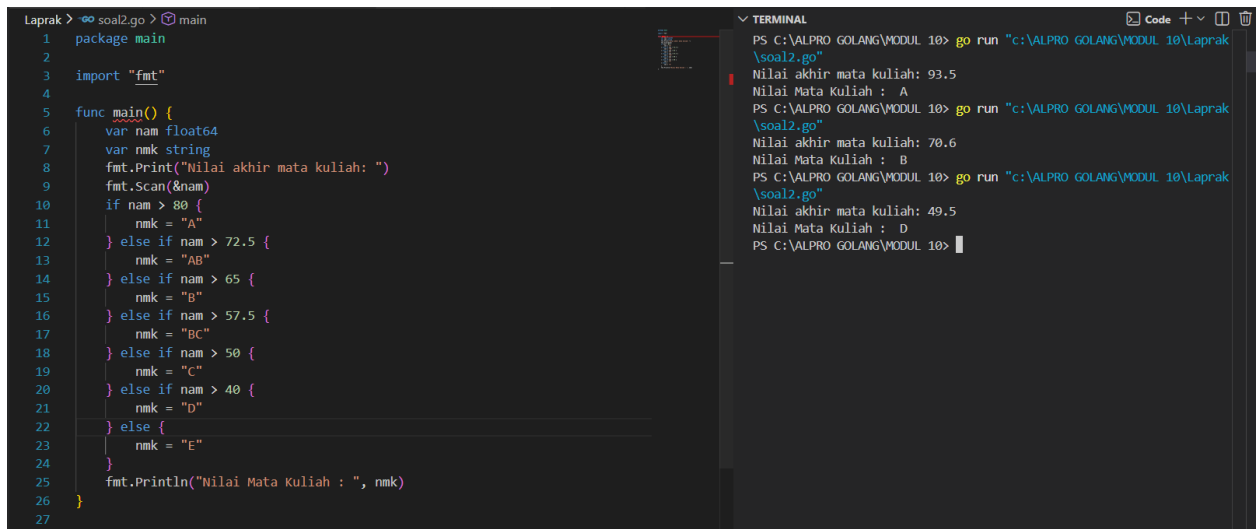
## Source code

```
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 code editor with a Go program and a terminal window showing its execution. The program is a simple conditional logic that assigns a letter grade (A, AB, B, BC, C, D, E) based on a numerical score. The terminal shows three separate runs of the program with different input scores and the corresponding output grades.

```
Laparak > go soal2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var nam float64
7     var nmk string
8     fmt.Print("Nilai akhir mata kuliah: ")
9     fmt.Scan(&nam)
10    if nam > 80 {
11        nmk = "A"
12    } else if nam > 72.5 {
13        nmk = "AB"
14    } else if nam > 65 {
15        nmk = "B"
16    } else if nam > 57.5 {
17        nmk = "BC"
18    } else if nam > 50 {
19        nmk = "C"
20    } else if nam > 40 {
21        nmk = "D"
22    } else {
23        nmk = "E"
24    }
25    fmt.Println("Nilai Mata Kuliah : ", nmk)
26 }
27
```

TERMINAL

```
PS C:\ALPRO GOLANG\MODUL 10> go run "c:\ALPRO GOLANG\MODUL 10\Laparak\soal2.go"
Nilai akhir mata kuliah: 93.5
Nilai Mata Kuliah : A
PS C:\ALPRO GOLANG\MODUL 10> go run "c:\ALPRO GOLANG\MODUL 10\Laparak\soal2.go"
Nilai akhir mata kuliah: 70.6
Nilai Mata Kuliah : B
PS C:\ALPRO GOLANG\MODUL 10> go run "c:\ALPRO GOLANG\MODUL 10\Laparak\soal2.go"
Nilai akhir mata kuliah: 49.5
Nilai Mata Kuliah : D
PS C:\ALPRO GOLANG\MODUL 10>
```

## Deskripsi program

Program ini ditulis menggunakan Bahasa pemrograman golang dengan yang pertama package main dan func main sebagai titik eksekusi program dan mengimpor “fmt” yang menangani prose input dan output.

Selanjutnya memasukan input variable nam dengan tipe data float64 dan variable nmk menggunakan tipe data string setelah itu menginputkan satu bilangan dengan fmt.Scan(&nam).

Program menggunakan struktur kontrol bersyarat if – else untuk menentukan nilai huruf (nmk) berdasarkan nilai angka yang dimasukkan (nam). Pengecekan dilakukan secara berurutan, dari nilai tertinggi ke terendah.

Karena menggunakan else if, ketika salah satu kondisi terpenuhi, pengecekan akan berhenti dan nilai huruf akan ditetapkan.

Setelah proses selesai dijalankan, program akan menampilkan hasil akhir menggunakan fmt.Println(“Nilai Mata Kuliah : , nmk).

## 3. Tugas 3

### Source code

```
package main

import "fmt"

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

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

    fmt.Print("Faktor: ")

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

    fmt.Println()
    Prima := false
```

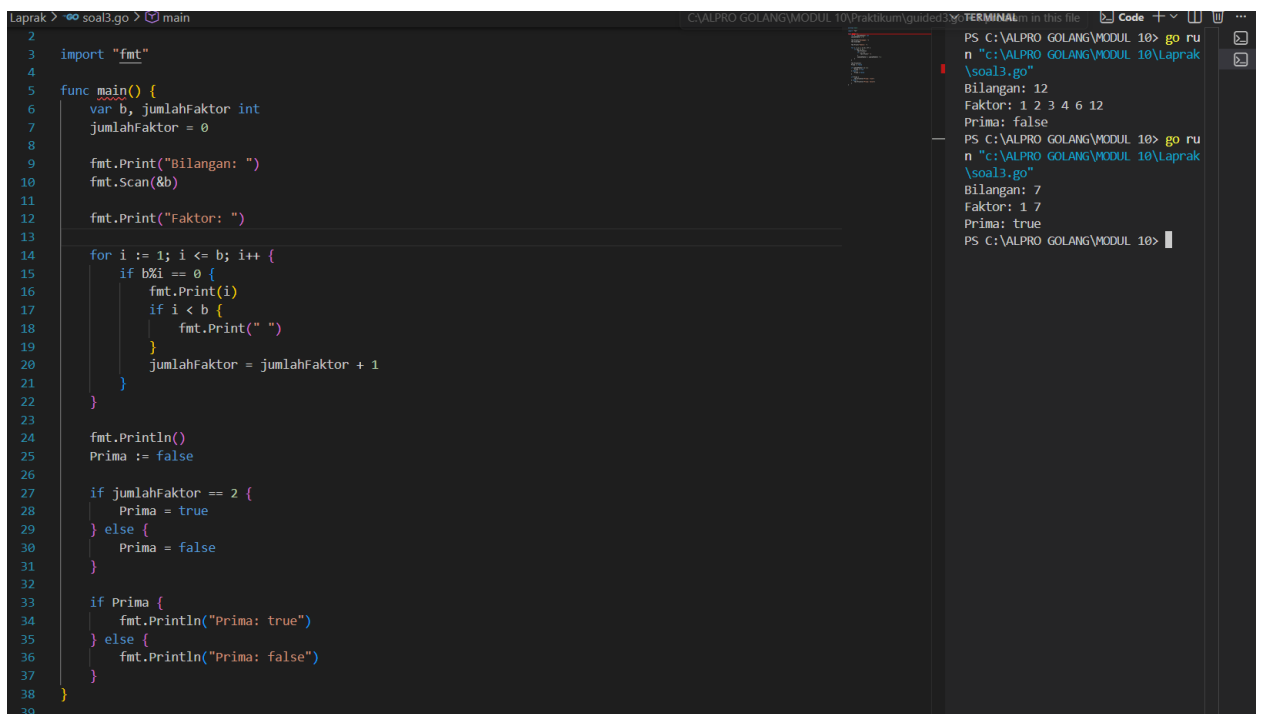
```

        if jumlahFaktor == 2 {
            Prima = true
        } else {
            Prima = false
        }

        if Prima {
            fmt.Println("Prima: true")
        } else {
            fmt.Println("Prima: false")
        }
    }
}

```

## Screenshoot program



```

Laparak > soal3.go > main
2
3 import "fmt"
4
5 func main() {
6     var b, jumlahFaktor int
7     jumlahFaktor = 0
8
9     fmt.Print("Bilangan: ")
10    fmt.Scan(&b)
11
12    fmt.Print("Faktor: ")
13
14    for i := 1; i <= b; i++ {
15        if b%i == 0 {
16            fmt.Print(i)
17            if i < b {
18                fmt.Print(" ")
19            }
20            jumlahFaktor = jumlahFaktor + 1
21        }
22    }
23
24    fmt.Println()
25    Prima := false
26
27    if jumlahFaktor == 2 {
28        Prima = true
29    } else {
30        Prima = false
31    }
32
33    if Prima {
34        fmt.Println("Prima: true")
35    } else {
36        fmt.Println("Prima: false")
37    }
38 }
39
PS C:\ALPRO GOLANG\MODUL 10> go run "c:\ALPRO GOLANG\MODUL 10\Laparak\soal3.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS C:\ALPRO GOLANG\MODUL 10> go run "c:\ALPRO GOLANG\MODUL 10\Laparak\soal3.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS C:\ALPRO GOLANG\MODUL 10>

```

## Deskripsi program

Program ini ditulis menggunakan Bahasa pemrograman golang dengan yang pertama package main dan func main sebagai titik eksekusi program dan mengimpor “fmt” yang menangani proses input dan output.

Program mendeklarasikan dua variabel integer: b untuk menyimpan bilangan yang diinput, dan jumlahFaktor yang diinisialisasi ke 0 untuk menghitung total faktor yang ditemukan. Program kemudian meminta pengguna untuk memasukkan bilangan bulat menggunakan `fmt.Print("Bilangan: ")` dan menerima input tersebut ke dalam variabel b menggunakan `fmt.Scan`.



Proses pencarian faktor dilakukan menggunakan perulangan for yang dimulai dari  $i = 1$  hingga  $i$  mencapai nilai bilangan input ( $b$ ). Di dalam perulangan tersebut, program menggunakan kondisi `if b%i == 0` untuk mengecek apakah  $i$  adalah faktor dari  $b$ . Operator `%` (modulus) menentukan sisa pembagian; jika sisa pembagian adalah 0, maka  $i$  adalah faktor.

Setelah perulangan selesai, `fmt.Println()` dipanggil untuk pindah baris, memisahkan daftar faktor dari hasil pengecekan prima. Program kemudian mendeklarasikan variabel boolean `Prima`, yang diinisialisasi ke `false`, apakah bilangan tersebut prima dilakukan berdasarkan hasil perhitungan faktor:

Bilangan disebut prima jika ia hanya memiliki dua faktor (yaitu 1 dan dirinya sendiri). Program mengecek kondisi `if jumlahFaktor == 2`. Jika benar, variabel `Prima` diatur menjadi `true`. Jika tidak (jumlah faktor kurang dari 2 atau lebih dari 2), `Prima` diatur menjadi `false`.

Hasil akhirnya jika `Prima` adalah `true` maka program mencetak "`Prima : true`". Dan jika hasil prima nya adalah `false`, maka program mencetak "`prima : false`".