

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 10**

**ELSE-IF**



**Disusun oleh:**

**Nama: Nabyla Zahirah Ramadhani**

**NIM: 109082500104**

**Kelas: 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 umur int

    var KK bool

    fmt.Print("Masukkan umur: ")

    fmt.Scan(&umur)

    fmt.Print("Memiliki KK: ")

    fmt.Scan(&KK)

    if umur >= 17 && KK == true {

        fmt.Println("bisa membuat KTP")

    } else {

        fmt.Println("belum bisa membuat KTP")

    }

}
```

**Screenshoot program**

The screenshot shows a Go IDE with a file named `contoh1.go`. The code defines a `main` function that declares `umur` as an integer and `KK` as a boolean. It prompts the user to enter their age and whether they have a KTP. It then checks if the age is greater than or equal to 17 and if `KK` is true. If both conditions are met, it prints "Bisa membuat KTP", otherwise it prints "belum bisa membuat KTP".

```
1 package main
2 import "fmt"
3 func main(){
4
5     var umur int
6     var KK bool
7
8     fmt.Print("Masukkan umur: ")
9     fmt.Scan(&umur)
10    fmt.Print("Memiliki KK: ")
11    fmt.Scan(&KK)
12
13    if umur >= 17 && KK == true {
14        fmt.Println("Bisa membuat KTP")
15    } else {
16        fmt.Println("belum bisa membuat KTP")
17    }
18 }
```

Below the code editor, the terminal shows the execution of the program with three test cases:

```
go run "/Users/nabylaaa/Documents/go/week 10/contoh1.go"
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents/go/week 10/contoh1.go"
Masukkan umur: 17
Memiliki KK: true
Bisa membuat KTP
nabylaaa@Nabylas-MacBook-Air week 10 %
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents/go/week 10/contoh1.go"
Masukkan umur: 20
Memiliki KK: false
belum bisa membuat KTP
nabylaaa@Nabylas-MacBook-Air week 10 %
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents/go/week 10/contoh1.go"
Masukkan umur: 15
Memiliki KK: true
belum bisa membuat KTP
nabylaaa@Nabylas-MacBook-Air week 10 %
nabylaaa@Nabylas-MacBook-Air week 10 %
```

A student ID card is visible in the background of the terminal area:

Nama: Nabylia Zaherah Ramadhani  
NIM: 109082500104  
Kelas: S1IF-13-07

## Deskripsi program

Program ini untuk menentukan seseorang apakah bisa membuat KTP atau tidak, variabelnya umur dengan tipe datanya int dan KK yang bertipe bool untuk menyimpan false atau true. Pertama program ini meminta untuk input umur dan KK menggunakan `fmt.Print("Masukkan umur: ")` dan `fmt.Print("Memiliki KK")` dan inputan dari user akan dibaca menggunakan `fmt.Scan(&umur)` dan untuk `fmt.Scan(&KK)` akan membaca nilai boolean. Untuk `if umur >= 17 && KK == true` program akan memeriksa dua syarat, yang pertama `umur >= 17` ini itu memeriksa apakah umur lebih dari sama dengan 17 dan yang kedua `KK == true` itu artinya apakah KK itu sama dengan true?, untuk operator `&&` itu artinya keduanya harus benar agar kondisi terpenuhi. Jika kedua syarat tadi terpenuhi maka program akan menjalankan `fmt.Println("Bisa membuat KTP")` tetapi jika salah satu syarat tidak memenuhi maka program beralih menjalankan `fmt.Println("Belum bisa membuat KTP ")`.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var huruf rune

    fmt.Print("Masukkan 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")

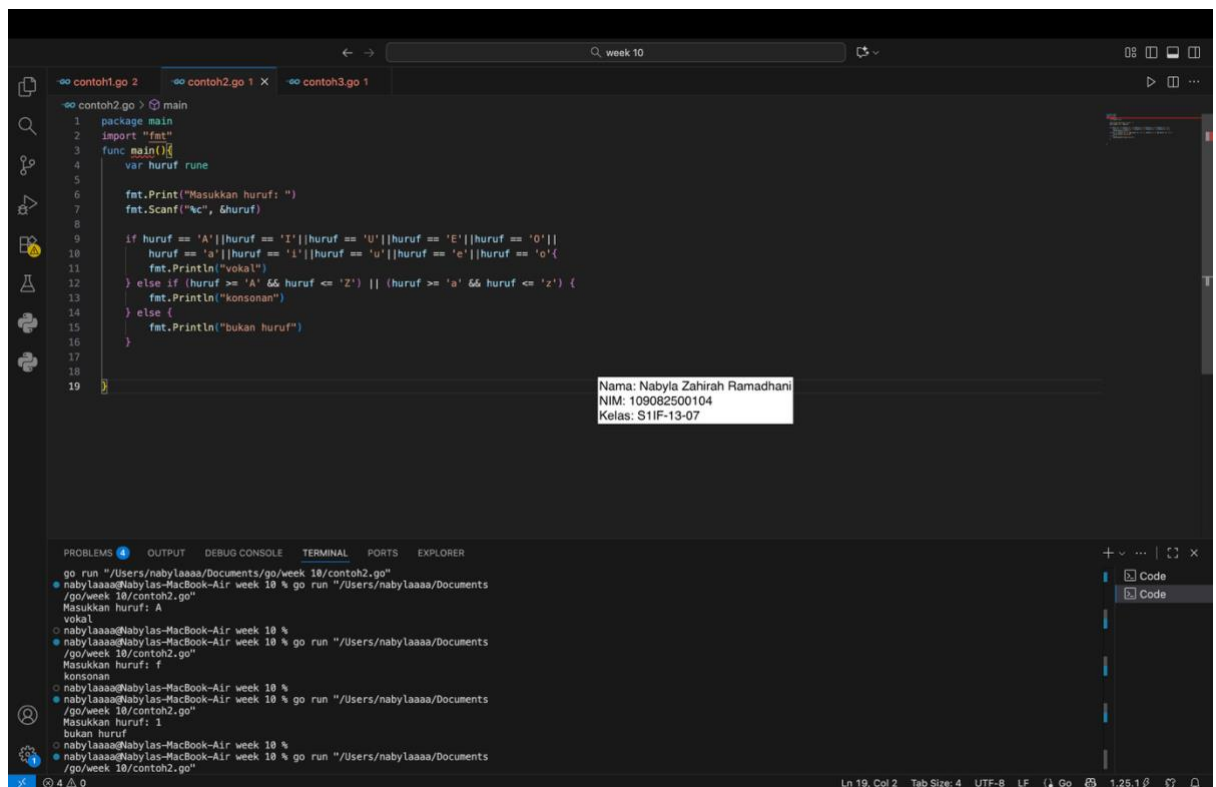
    } else {

        fmt.Println("bukan huruf")

    }

}
```

**Screenshoot program**



```
1 package main
2 import "fmt"
3 func main() {
4     var huruf rune
5
6     fmt.Print("Masukkan huruf: ")
7     fmt.Scanf("%c", &huruf)
8
9     if huruf == 'A' || huruf == 'I' || huruf == 'U' || huruf == 'E' || huruf == 'O' ||
10        huruf == 'a' || huruf == 'i' || huruf == 'u' || huruf == 'e' || huruf == 'o' {
11         fmt.Println("vokal")
12     } else if (huruf >= 'A' && huruf <= 'Z') || (huruf >= 'a' && huruf <= 'z') {
13         fmt.Println("konsonan")
14     } else {
15         fmt.Println("bukan huruf")
16     }
17 }
18
19
```

Nama: Nabyla Zahirah Ramadhani  
NIM: 109082500104  
Kelas: S1IF-13-07

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS EXPLORER

```
go run "/Users/nabylaaa/Documents/go/week 10/contoh2.go"
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents
/go/week 10/contoh2.go"
Masukkan huruf: A
vokal
nabylaaa@Nabylas-MacBook-Air week 10 %
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents
/go/week 10/contoh2.go"
Masukkan huruf: f
konsonan
nabylaaa@Nabylas-MacBook-Air week 10 %
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents
/go/week 10/contoh2.go"
Masukkan huruf: 1
bukan huruf
nabylaaa@Nabylas-MacBook-Air week 10 %
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents
/go/week 10/contoh2.go"
```

## Deskripsi program

Program ini untuk menentukan inputan dari user itu vocal, konsonan atau bukan huruf, variabelnya huruf yang bertipe rune, tipe rune digunakan untuk menyimpan karakter. Pertama program ini meminta input huruf dari user menggunakan *fmt.Print("Masukkan huruf: ")* dan akan dibaca menggunakan *fmt.Scanf("%c", &huruf)*, *%c* digunakan untuk membaca suatu karakter. Program dimulai dengan kondisi *if huruf == 'A' || huruf == 'I' || huruf == 'U' || huruf == 'E' || huruf == 'O' || huruf == 'a' || huruf == 'i' || huruf == 'u' || huruf == 'e' || huruf == 'o'* untuk mengecek apakah huruf yang dimasukkan adalah vocal, mau itu huruf besar atau kecil. Jika memenuhi program akan menampilkan *fmt.Println("vokal")*. Jika kondisi tadi tidak memenuhi maka program akan melanjutkan ke kondisi yang kedua, *else if (huruf >= 'A' && huruf <= 'Z') || (huruf >= 'a' && huruf <= 'z')* kondisi ini akan mengecek apakah karakter yang di input huruf alfabet a-z, jika benar program akan menampilkan *fmt.Println("konsonan")*. Jika tidak terpenuhi lagi maka program akan masuk ke *else {fmt.Println("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("Masukkan 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

The screenshot shows a Go IDE with a file named `contoh3.go`. The code defines a `main` function that prompts the user to enter a number, reads it, and then extracts its four digits (`d1`, `d2`, `d3`, `d4`) using modulus and division. It then checks if the digits are in ascending order (`d1 < d2 && d2 < d3 && d3 < d4`), descending order (`d1 > d2 && d2 > d3 && d3 > d4`), or neither. The results are printed to the console.

```
1 package main
2 import "fmt"
3 func main(){
4     var bilangan, d1, d2, d3, d4 int
5     var teks string
6
7     fmt.Print("Masukkan bilangan: ")
8     fmt.Scan(&bilangan)
9     d4 = bilangan % 10
10    d3 = (bilangan % 100) / 10
11    d2 = (bilangan % 1000) / 100
12    d1 = bilangan / 1000
13    if d1 < d2 && d2 < d3 && d3 < d4{
14        teks = "terurut membesar"
15    }else if d1 > d2 && d2 > d3 && d3 > d4{
16        teks = "terurut mengecil"
17    }else{
18        teks = "tidak terurut"
19    }
20    fmt.Println("Digit pada bilangan:", bilangan, teks)
21 }
```

The terminal output shows three test cases: 2489 (ascending), 3861 (descending), and 9651 (no order).

```
go run "/Users/nabylaaa/Documents/go/week 10/contoh3.go"
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents/go/week 10/contoh3.go"
Masukkan bilangan: 2489
Digit pada bilangan 2489 terurut membesar
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents/go/week 10/contoh3.go"
Masukkan bilangan: 3861
Digit pada bilangan 3861 tidak terurut
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents/go/week 10/contoh3.go"
Masukkan bilangan: 9651
Digit pada bilangan 9651 terurut mengecil
nabylaaa@Nabylas-MacBook-Air week 10 %
```

## Deskripsi program

Program ini untuk menentukan apakah digit dalam suatu bilangan terurut membesar, mengecil, atau tidak terurut. Variabelnya `bilangan`, `d1`, `d2`, `d3`, `d4` dan bertipe data `int`, dan `teks` yang bertipe `string`. Pertama program meminta input angka empat digit melalui `fmt.Print("Masukkan bilangan: ")`, kemudian akan dibaca nilainya menggunakan `fmt.Scan(&bilangan)`. Setelah mendapat input dari user program akan mengambil digit satu per satu:

`d4 = bilangan % 10` mengambil digit satuan, untuk `% 10` itu modulus sisa pembagian bilangan dengan 10.

`d3 = (bilangan % 100) / 10` mengambil digit puluhan, , untuk `% 100` itu modulus sisa pembagian bilangan dengan 100 dan hasilnya akan dibagi 10

`d2 = (bilangan % 1000) / 100` mengambil digit ratusan. untuk `% 1000` itu modulus sisa pembagian bilangan dengan 1000 dan hasilnya akan dibagi 100

`d1 = bilangan / 1000` mengambil digit ribuan, yaitu bilangan dibagi 1000.

Setelah semua sudah diperoleh, program akan memeriksa apakah digit-digit tadi terurut naik atau turun. Pada kondisi pertama `if d1 < d2 && d2 < d3 && d3 < d4 {teks = "terurut membesar"}` bagian ini mengecek apakah digit ribuan `d1` lebih kecil dari digit ratusan `d2`, `d2` lebih kecil dari `d3`, dan `d3` lebih kecil dari `d4`. Jika semua benar, berarti seluruh digit urut dari kecil ke besar, sehingga program menyimpan teks "terurut membesar". Jika tidak, program mengecek kondisi kedua `} else if d1 > d2 && d2 > d3 && d3 > d4 {teks = "terurut mengecil"}` bagian ini mengecek apakah digit-digitnya urut dari besar ke kecil. Jika `d1` lebih besar dari `d2`, lalu `d2` lebih besar dari `d3`, dan `d3` lebih besar dari `d4`, maka digit-digit tersebut terurut mengecil, sehingga program menyimpan teks itu ke variabel `teks`.

Jika kedua kondisi tersebut tidak terpenuhi, berarti digit-digitnya tidak membentuk pola urutan tertentu tidak semuanya naik dan tidak semuanya turun. Maka program masuk ke bagian `else` dan mengisi teks = "tidak terurut". Terakhir program menampilkan hasil `fmt.Println("Digit pada bilangan", bilangan, teks)` akan memberi tau apakah bilangan itu terurut membesar, terurut mengecil, atau tidak terurut.



## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main(){

    var gram int

    fmt.Print("Masukkan berat parsel (gram): ")

    fmt.Scan(&gram)


    kg := gram / 1000

    sisa := gram % 1000

    biayaKg := kg * 10000


    var tambahan int

    if gram > 10000 {

        tambahan = 0

    } else if sisa >= 500 {

        tambahan = sisa * 5

    } else if sisa > 0 {

        tambahan = sisa * 15

    } else {

        tambahan = 0

    }


    total := biayaKg + tambahan
```

```

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

    fmt.Printf("Detail biaya: Rp %d", biayaKg)

    if tambahan > 0 {

        fmt.Printf(" + Rp %d\n", tambahan)

    } else {

        fmt.Printf(" + Rp %d (sisa gratis)\n", tambahan)

    }

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

}

```

## Screenshoot program

The screenshot shows a Go program in VS Code. The editor displays the source code for `soal1.go`, which calculates shipping costs based on weight in grams. The code includes variables for `kg`, `sisa` (remaining grams), `biayaKg` (cost in kg), and `tambahan` (additional cost). It uses `fmt` for input/output and includes conditional logic for additional charges based on the remaining grams.

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

```

go run "/Users/nabylaaaa/Documents/go/week 10/soal1.go"
nabylaaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaaa/Documents/go/week 10/soal1
.go"
Masukkan berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp 80000 + Rp 2500
Total biaya: Rp 82500
nabylaaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaaa/Documents/go/week 10/soal1
.go"
Masukkan berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp 90000 + Rp 3750
Total biaya: Rp 93750
nabylaaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaaa/Documents/go/week 10/soal1
.go"
Masukkan berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp 110000 + Rp 0 (sisa gratis)
Total biaya: Rp 110000
nabylaaaa@Nabylas-MacBook-Air week 10 %

```

A student ID card is visible in the background of the terminal window:

```

Nama: Nabyla Zahrah Ramadhani
NIM: 109082500104
Kelas: S1IF-13-07

```

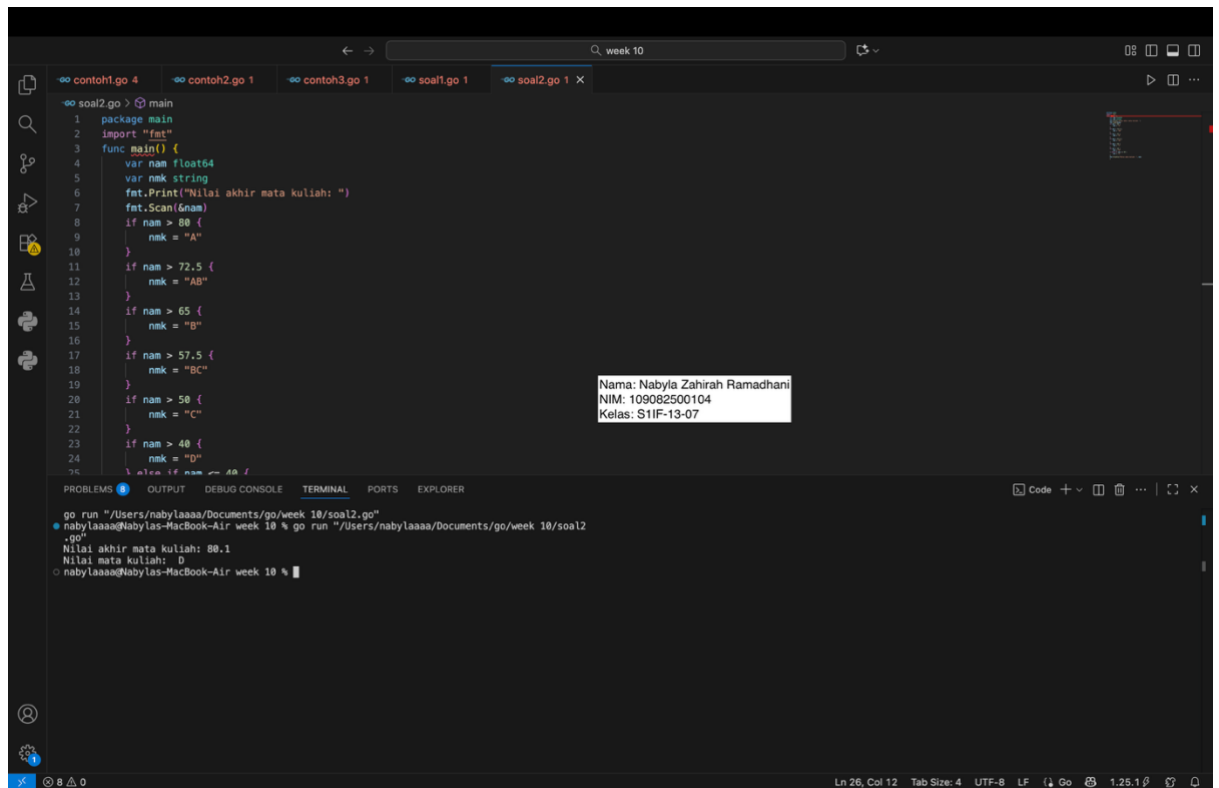
## Deskripsi program

Program ini untuk menghitung biaya pengiriman variabelnya gram yang bertipe int. Program meminta input berat dalam satuan gram menggunakan `fmt.Print("Masukkan berat parcel (gram): ")` untuk meminta pengguna memasukkan berat. Nilai yang dimasukkan kemudian dibaca menggunakan `fmt.Scan(&gram)` dan disimpan ke variabel `gram`. Setelah berat diterima, program mulai memecah berat tersebut menjadi dua bagian. Baris `kg := gram / 1000` (gram dibagi 1000) menghitung berapa kilogram dari total berat dari gram dibagi dengan 1000, sedangkan `sisa := gram % 1000`

(gram modulus 1000, sisa pembagian gram dengan 1000) menyimpan gram yang tersisa setelah dikeluarkan bagian kilogramnya. Lalu biaya dasar dihitung menggunakan  $biayaKg := kg * 10000$  (kilogram dikali 10000), yang berarti setiap 1 kg dikenai biaya Rp 10.000. Program membuat variabel lagi *var tambahan int* untuk menyimpan biaya tambahan berdasarkan sisa gram atau kondisi berat tertentu. Pada bagian *if gram > 10000 { tambahan = 0 }*, artinya jika berat total lebih dari 10.000 gram (lebih dari 10 kg), maka tidak ada biaya tambahan sehingga *tambahan = 0*. Jika kondisi tersebut tidak terpenuhi, program masuk ke *else if sisa >= 500*, jika sisa gram minimal 500 gram, maka program menghitung biaya tambahan dengan tarif Rp 5 per gram melalui  $tambahan = sisa * 5$ . Jika sisa gram kurang dari 500 tetapi masih lebih dari 0, program masuk ke *else if sisa > 0* dan menghitung biaya tambahan menggunakan  $tambahan = sisa * 15$ , yaitu tarif Rp 15 per gram. Jika tidak ada sisa gram sama sekali, program akan masuk ke *else { tambahan = 0 }* dan biaya tambahan tetap 0. Program menghitung total biaya pengiriman dengan  $total := biayaKg + tambahan$ . Untuk menampilkan hasil perhitungan, program menggunakan *fmt.Printf("Detail berat: %d kg + %d gr\n", kg, sisa)* untuk menampilkan berat dalam kilogram dan gram sisa. Kemudian program menampilkan biaya dasar kilogram dengan *fmt.Printf("Detail biaya: Rp %d", biayaKg)*. Setelah itu, program mengecek apakah terdapat biaya tambahan menggunakan kondisi *if tambahan > 0*. Jika tambahan lebih besar dari 0, maka program menampilkan *fmt.Printf(" + Rp %d\n", tambahan)*, sedangkan jika tidak ada biaya tambahan, program menampilkan *fmt.Printf(" + Rp %d (sisa gratis)\n", tambahan)* untuk memberi tahu bahwa sisa gram tidak dikenai biaya. Di akhir, program menampilkan total biaya pengiriman menggunakan *fmt.Printf("Total biaya: Rp %d\n", total)*.

## 2. Tugas 2

- a. Keluaran program= D, tidak sesuai spesifikasi soal karena NAM 80.1 harusnya menghasilkan A



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

Terminal Output:

```
go run "/Users/nabylaaa/Documents/go/week 10/soal2.go"
nabylaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaa/Documents/go/week 10/soal2
.go"
Nilai akhir mata kuliah: 80.1
Nilai mata kuliah: 0
nabylaaa@Nabylas-MacBook-Air week 10 %
```

- b. Kesalahan program ini itu:

- Program di modul juga variabel nam menjadi string nam="A", padahal nam itu float64 jadi tipennya harus angka, yang benar harusnya nmk="A".
- Semua kondisi menggunakan if terpisah bukan else if, karena ini setelah satu kondisi benar maka program tidak berhenti, program akan tetap cek kondisi berikutnya.
- Tidak mencakup rentang yang benar, program tidak menuliskan batas bawah dan batas atas dengan benar
- Bagian nam > 40, nam <= 40 tidak mengikuti table nmk.

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

### Source code bagian c

```
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**

```
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 = "Aa"
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 Output:

```
go run "/Users/nabylaaaa/Documents/go/week 10/soal2.go"
nabylaaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaaa/Documents/go/week 10/soal2
.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
nabylaaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaaa/Documents/go/week 10/soal2
.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
nabylaaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaaa/Documents/go/week 10/soal2
.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
nabylaaaa@Nabylas-MacBook-Air week 10 %
```

Metadata:

Nama: Nabyla Zahirah Ramadhani  
NIM: 109082500104  
Kelas: S1IF-13-07

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {

    var b int

    fmt.Print("Bilangan: ")

    fmt.Scan(&b)


    fmt.Print("Faktor: ")

    count := 0

    for i := 1; i <= b; i++ {

        if b%i == 0 {

            fmt.Print(i, " ")

            count++

        }

    }


    var prima bool

    if count == 2 {
```

```

        prima = true
    } else {
        prima = false
    }

    fmt.Println()
    fmt.Println("Prima:", prima)
}

```

## Screenshoot program

```

soal3.go > main
1 package main
2 import "fmt"
3 func main() {
4     var b int
5     fmt.Print("Bilangan: ")
6     fmt.Scan(&b)
7
8     fmt.Print("Faktor: ")
9     count := 0
10    for i := 1; i <= b; i++ {
11        if b%i == 0 {
12            fmt.Print(i, " ")
13            count++
14        }
15    }
16
17    var prima bool
18    if count == 2 {
19        prima = true
20    } else {
21        prima = false
22    }
23    fmt.Println()
24    fmt.Println("Prima:", prima)
25 }

```

go run "/Users/nabylaaaa/Documents/go/week 10/soal3.go"  
 nabylaaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaaa/Documents/go/week 10/soal3.go"  
 Bilangan: 12  
 Faktor: 1 2 3 4 6 12  
 Prima: false  
 nabylaaaa@Nabylas-MacBook-Air week 10 % go run "/Users/nabylaaaa/Documents/go/week 10/soal3.go"  
 Bilangan: 7  
 Faktor: 1 7  
 Prima: true  
 nabylaaaa@Nabylas-MacBook-Air week 10 %

## Deskripsi program

Program ini untuk mencari dan menampilkan factor dari suatu bilangan yang di input oleh user dan menentukan apakah bilangan tersebut prima atau tidak, variabelnya *b* yang bertipe integer dan *prima* yang bertipe Boolean. Pertama program meminta inputan bilangan dari user menggunakan `fmt.Print("Bilangan: ")` dan akan dibaca menggunakan `fmt.Scan(&b)`. Setelah mendapatkan angkanya, program menampilkan teks `fmt.Print("Faktor: ")`. Lalu dibuat variabel `count := 0` untuk menghitung berapa banyak factor yang dimiliki bilangan tersebut. Selanjutnya ada sebuah perulangan `for i := 1; i <= b; i++ { ... }`. Perulangan ini memeriksa setiap angka dari 1 sampai *b*. Di dalamnya, terdapat pengecekan `if b%i == 0`, itu artinya jika *b* habis dibagi oleh *i*, maka *i* adalah factor dari *b*. Jika benar, program mencetak angka tersebut dan menambah nilai `count` sebanyak 1. Setelah perulangan selesai, program menentukan apakah bilangan tersebut prima atau tidak. Karena itu dibuat variabel `prima bool`. `If count == 2`, maka `prima = true`, artinya bilangan tersebut adalah bilangan prima. Jika tidak sama

dengan 2, maka `prima = false`. Terakhir, program menampilkan hasilnya dengan `fmt.Println("Prima:", prima)` yang memberi tahu apakah bilangan tersebut prima atau bukan.