

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 10

ELSE-IF



Disusun oleh:

Yedija Johanan Siregar

109082500075

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

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var usia int
8
9     var kk bool
10
11    fmt.Scan(&usia, &kk)
12
13
14    if usia >= 17 && kk {
15
16        fmt.Println("bisa membuat KTP")
17
18    } else{
19
20        fmt.Println("belum bisa membuat KTP")
21
22    }
23
24 }
```

Screenshot program

The screenshot shows a code editor window with a dark theme. On the left, the Go code is displayed with line numbers 1 through 24. Lines 1-16 are part of the main function definition, while lines 17-24 contain the conditional logic and output statements. To the right of the code editor is a terminal window titled 'discord_bac1' showing the execution of the program. The terminal output includes the user's ID (109082500075), date (S1IF-13-07), and name (YEDIJA JOHANAN SIREGAR). Below the terminal, the command 'go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"' is shown along with the resulting output: 'true' and 'false'.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var usia int
7     var kk bool
8     fmt.Scan(&usia, &kk)
9
10    if usia >= 17 && kk {
11        fmt.Println("bisa membuat KTP")
12    } else{
13        fmt.Println("belum bisa membuat KTP")
14    }
15
16 }
17
18
19
20
21
22
23
24
```

```
109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR
Ln 2, Col 4 | 46 characters | AA Formatte | 100% | Windows (CP) | UTF-8
```

```
TERMINAL PROBLEMS (17) OUTPUT DEBUG CONSOLE PORTS
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
17
true
bisa membuat KTP
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
20
false
belum bisa membuat KTP
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>
```

Deskripsi program

Program ini meminta pengguna untuk memasukkan dua data, yaitu usia dalam bentuk angka bulat dan status kepemilikan Kartu Keluarga (KK) dalam bentuk boolean (*true* atau *false*). Program kemudian akan memeriksa apakah kedua kondisi terpenuhi, di mana usia harus 17 tahun atau lebih dan harus memiliki KK. Jika kedua syarat tersebut terpenuhi, program akan menampilkan pesan "bisa membuat KTP". Namun, jika salah satu atau kedua syarat tidak terpenuhi, program akan menampilkan pesan "belum bisa membuat KTP".

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var x rune
    var huruf, vKecil, vBesar bool
    fmt.Scanf("%c", &x)

    huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <=
'Z')

    vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e'
    || x == 'o'

    vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E'
    || x == 'O'

    if huruf && (vKecil || vBesar) {
        fmt.Println("vokal")
    }
}
```

```

} else if huruf && !(vKecil || vBesar) {

    fmt.Println("konsonan")

} else{

    fmt.Println("bukan huruf")

}

}

```

Screenshoot program

```

demo_soal.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x rune
7     var huruf, vKecil, vBesar bool
8     fmt.Scanf("%c", &x)
9
10    huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
11    vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o'
12    vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'
13
14    if huruf && (vKecil || vBesar) {
15        fmt.Println("vokal")
16    } else if huruf && !(vKecil || vBesar){
17        fmt.Println("konsonan")
18    } else{
19        fmt.Println("bukan huruf")
20    }
21 }

```

109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR

Ln 2, Col 4 46 characters AA Formatte 100% Windows (CF) UTF-8

TERMINAL PROBLEMS 17 OUTPUT DEBUG CONSOLE PORTS

```

PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
A
vokal
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
F
konsonan
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>

```

Deskripsi program

Program ini meminta pengguna untuk memasukkan satu karakter tunggal dan kemudian menganalisis jenis karakter tersebut. Program akan membaca input karakter dan melakukan pengecekan bertahap: pertama memastikan apakah karakter tersebut termasuk huruf (baik huruf kecil a-z maupun huruf besar A-Z), kemudian jika termasuk huruf, program akan memeriksa apakah karakter tersebut merupakan huruf vokal (a, i, u, e, o dalam bentuk kecil atau besar) atau justru huruf konsonan. Berdasarkan hasil pemeriksaan ini, program akan menampilkan output yang sesuai berupa "vokal" untuk huruf vokal, "konsonan" untuk huruf non-vokal, atau "bukan

huruf" jika karakter yang dimasukkan tidak termasuk dalam kategori huruf sama sekali.

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) / 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)

}
```

Screenshot program

```

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

TERMINAL PROBLEMS 17 OUTPUT DEBUG CONSOLE PORTS
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
Bilangan: 2489
Digit pada bilangan 2489 terurut membesar
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
Bilangan: 9651
Digit pada bilangan 9651 terurut mengecil
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
Bilangan: 9651
Digit pada bilangan 9651 terurut mengecil
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>

```

The terminal window shows the execution of the Go program. It prompts for a four-digit number. When '2489' is entered, it outputs 'terurut membesar'. When '9651' is entered, it outputs 'terurut mengecil'. Both inputs are correctly analyzed by the program.

Deskripsi program

Program ini meminta pengguna untuk memasukkan sebuah bilangan bulat 4 digit, kemudian menganalisis apakah keempat digit pada bilangan tersebut tersusun secara terurut. Program akan membaca input bilangan, memisahkan setiap digitnya (ribuan, ratusan, puluhan, satuan), lalu memeriksa pola urutan digit-digit tersebut. Hasil analisis akan menampilkan apakah digit-digit pada bilangan tersebut terurut membesar (contoh: 1234), terurut mengecil (contoh: 4321), atau tidak terurut (contoh: 1324). Program ini berguna untuk mengenali pola urutan dalam bilangan 4 digit yang dimasukkan oleh pengguna.

TUGAS

1. Tugas 1

Source code

```

package main

import "fmt"

func main() {

    var beratGram int

    var kg, sisaGram int

```

```
var biayaPerKg, biayaSisa, totalBiaya int

fmt.Println("Berat parsel (gram): ")

fmt.Scan(&beratGram)

kg = beratGram / 1000

sisaGram = beratGram % 1000

biayaPerKg = kg * 10000

if sisaGram >= 500 {

    biayaSisa = sisaGram * 5

} else {

    biayaSisa = sisaGram * 15

}

if kg > 10 {

    biayaSisa = 0

}

totalBiaya = biayaPerKg + biayaSisa

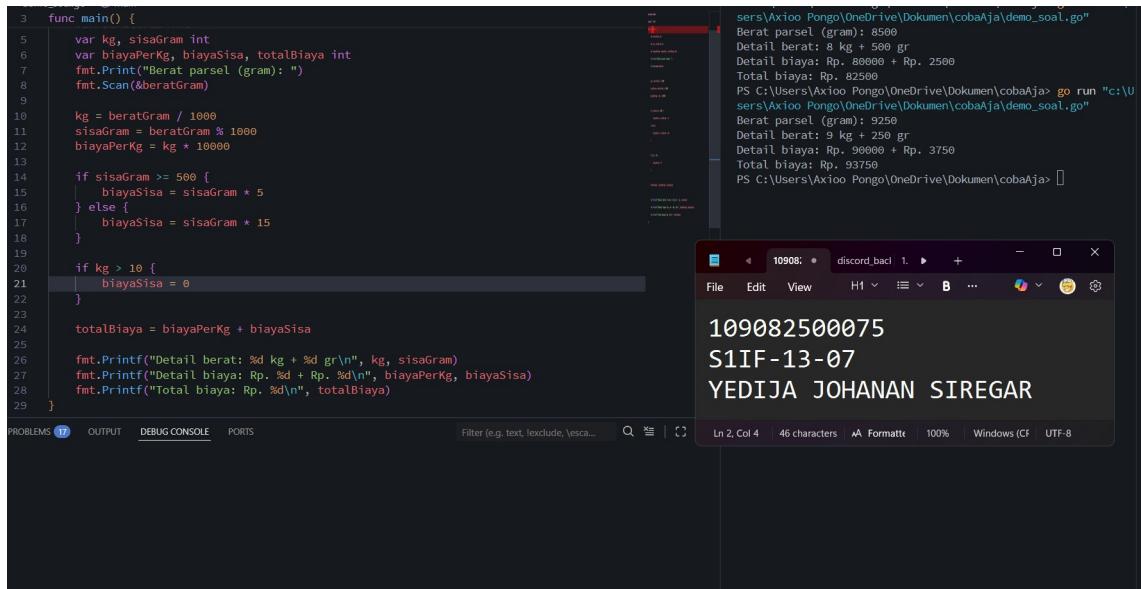
fmt.Printf("Detail berat: %d kg + %d gr\n", kg, sisaGram)

fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaPerKg,
biayaSisa)

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

}
```

Screenshot program



```
3 func main() {
4     var kg, sisaGram int
5     var biayaPerKg, biayaSisa, totalBiaya int
6     fmt.Println("Berat parsel (gram): ")
7     fmt.Scan(&beratGram)
8
9     kg = beratGram / 1000
10    sisaGram = beratGram % 1000
11    biayaPerKg = kg * 10000
12
13    if sisaGram >= 500 {
14        biayaSisa = sisaGram * 5
15    } else {
16        biayaSisa = sisaGram * 15
17    }
18
19    if kg > 10 {
20        biayaSisa = 0
21    }
22
23    totalBiaya = biayaPerKg + biayaSisa
24
25    fmt.Printf("Detail berat: %d kg + %d gr\n", kg, sisaGram)
26    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaPerKg, biayaSisa)
27    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
28
29 }
```

The screenshot shows a Go code editor with the code above. To the right is a terminal window titled "discord_bacl 1." displaying the output of running the program. The output shows the breakdown of the weight (8 kg + 500 gr), the cost per kilogram (Rp. 80000 + Rp. 2500), and the total cost (Rp. 93750).

Deskripsi program

Program BiayaPos menghitung biaya pengiriman parsel berdasarkan berat dalam gram. Program memisahkan berat menjadi kilogram dan sisa gram, dengan biaya dasar Rp. 10.000 per kg. Untuk sisa gram: jika ≥ 500 gram dikenakan Rp. 5/gram, jika < 500 gram dikenakan Rp. 15/gram. Jika total berat lebih dari 10kg, biaya sisa gram diigratiskan. Program menampilkan detail berat, rincian biaya, dan total biaya yang harus dibayar.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var nam float64

    var nmk string

    fmt.Println("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)  
}
```

Screenshot program

The screenshot shows a Go development environment. On the left, the code editor displays a Go file named demo_soal.go with the following content:

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

The terminal window on the right shows the command PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go" followed by several lines of output showing different test cases and their results.

The text editor window in the foreground contains student information:

109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR

Deskripsi program

- Jika nam diberikan adalah 80.1, maka keluaran nya adalah E. Program tersebut tidak tereksekusi dengan baik dan benar, yang seharusnya keluaran nya adalah A
- Kesalahan utama nya: urutan kondisi dari terbesar ke terkecil, yang seharusnya dari nilai tertinggi ke terendah, lalu semua kondisi dieksekusi karena menggunakan if bertingkat tanpa else if, dan nilai akhir selalu ditimpak oleh kondisi terakhir yang terpenuhi.

Alur pemrogramannya:

- Untuk nilai 80.1, if nam > 80 → true → nmk = "A"
- Untuk nilai 93.5, if nam > 80 → true → nmk = "A"
- Untuk nilai 70.6, if nam > 80 → false
 - else if nam > 72.5 → true → nmk = "B"
- Untuk nilai 49.5, if nam > 80 → false
 - else if nam > 72.5 → false
 - else if nam > 65 → false
 - else if nam > 50 → false
 - else if nam > 40 → true → nmk = "D"

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var b int
    fmt.Print("Masukkan sebuah bilangan bulat: ")
    fmt.Scan(&b)

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

    count := 0
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            count++
        }
    }

    prima := count == 2
    fmt.Println("Apakah bilangan prima?", prima)
}
```

Screenshot program

The screenshot shows a code editor with a Go file named `demo_soal.go`. The code defines a function `main()` that prompts the user for a number, finds all factors up to that number, and then checks if the number is prime. The output window shows the execution of the program, including the factors of 10908 and the result of the primality check for 7.

```
1 demo_soal.go > main
2
3
4
5 func main() {
6     var b int
7     fmt.Print("Masukkan sebuah bilangan bulat: ")
8     fmt.Scan(&b)
9
10    fmt.Println("Faktor: ")
11    for i := 1; i <= b; i++ {
12        if b%i == 0 {
13            fmt.Println(i, " ")
14        }
15    }
16    fmt.Println()
17
18    count := 0
19    for i := 1; i <= b; i++ {
20        if b%i == 0 {
21            count++
22        }
23    }
24
25    prima := count == 2
26    fmt.Println("Apakah bilangan prima?", prima)
27 }
```

```
PS C:\Users\Axioo Pongo\OneDrive\Kuliah\Golang\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Kuliah\Golang\cobaAja\demo_soal.go"
Faktor: 1 2 3 4 6 12
Apakah bilangan prima? false
PS C:\Users\Axioo Pongo\OneDrive\Kuliah\Golang\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Kuliah\Golang\cobaAja\demo_soal.go"
Masukkan sebuah bilangan bulat: 7
Faktor: 1 7
Apakah bilangan prima? true
PS C:\Users\Axioo Pongo\OneDrive\Kuliah\Golang\cobaAja>
```

Deskripsi program

Program ini dirancang untuk menganalisis sifat matematis dari sebuah bilangan bulat yang dimasukkan oleh pengguna. Pertama-tama, program akan meminta pengguna untuk memasukkan sebuah bilangan bulat positif, kemudian secara otomatis melakukan proses pencarian semua faktor dari bilangan tersebut. Faktor-faktor tersebut dicari dengan metode iterasi dari angka 1 hingga bilangan itu sendiri, dimana setiap angka yang dapat membagi habis bilangan input akan diidentifikasi sebagai faktor dan ditampilkan secara berurutan.

Setelah menampilkan semua faktor yang ditemukan, program melanjutkan dengan menganalisis apakah bilangan tersebut merupakan bilangan prima atau bukan. Analisis ini dilakukan dengan menghitung jumlah total faktor yang telah ditemukan sebelumnya. Bilangan dinyatakan sebagai bilangan prima jika dan hanya jika jumlah faktornya tepat dua, yaitu angka 1 dan bilangan itu sendiri. Hasil dari penentuan status bilangan prima ini kemudian ditampilkan kepada pengguna, memberikan informasi lengkap tentang karakteristik matematis dari bilangan yang diinputkan.