

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL

Else If



**Telkom
University
PURWOKERTO**

Disusun oleh:

FAREL TRI JULIAN

109082500163

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

Source Code

```
package main

import "fmt"

func main() {
    var usia int
    var n bool

    fmt.Print("masukan usia: ")
    fmt.Scan(&usia)

    fmt.Print("masukan true/false: ")
    fmt.Scan(&n)

    if usia >= 17 && n == true {
        fmt.Print("bisa buat KTP")
    } else {
        fmt.Print("tidak bisa buat KTP")
    }
}
```

Screenshot program

The screenshot shows a Go development environment. The code editor displays a file named `guided01.go` with the following content:

```
17nowv.go > ⌂ guided01.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var usia int
7     var n bool
8
9     fmt.Print("masukan usia: ")
10    fmt.Scan(&usia)
11    fmt.Scan(&n)
12
13    if usia >= 17 && n == true {
14        fmt.Print("bisa buat KTP")
15    } else {
16        fmt.Print("tidak bisa buat KTP")
17    }
18 }
```

The terminal tab shows the command `go run "e:\pemrograman\17nowv.go\guided01.go"` being run, followed by the output:

```
PS E:\pemrograman> go run "e:\pemrograman\17nowv.go\guided01.go"
masukan usia: 17
true
bisa buat KTP
PS E:\pemrograman> []
```

A modal window titled "NAMA" is open, displaying the text "NAMA : FAREL TRI JULIAN" and "NIM : 109082500163".

Deskripsi program

Program ini meminta pengguna memasukkan dua data, yaitu bilangan bulat untuk usia dan nilai boolean (true/false) untuk variabel `n`. Logika penentuannya menggunakan operator logika AND (`&&`), di mana sistem hanya akan mencetak "bisa buat KTP" jika syarat usia minimal 17 tahun terpenuhi dan input boolean bernilai true secara bersamaan; jika salah satu atau kedua syarat tidak terpenuhi, program akan mencetak "tidak bisa buat KTP".

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var n string

    fmt.Print("masukan apa aja: ")

    fmt.Scan(&n)

    if n == "A" || n == "I" || n == "U" || n == "E" || n ==
    "O" || n == "a" || n == "i" || n == "u" || n == "e" || n ==
    "o" {
        fmt.Print("vokal")
    } else if (n >= "b" && n <= "z") || (n >= "B" && n < "z")
    {
        fmt.Print("konsonan")
    } else {
        fmt.Print("bukan huruf")
    }
}
```

Screenshoot program

The screenshot shows a code editor interface with a dark theme. On the left is the code editor pane displaying a Go program. On the right is a terminal window showing the execution and output of the program. The terminal output is as follows:

```
masukan apa aja: 1
bukan huruf
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\guided02.go"
masukan apa aja: E
vokal
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\guided02.go"
masukan apa aja: e
vokal
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\guided02.go"
masukan apa aja: b
konsonan
PS E:\pemrograman>
```

The terminal window also displays the user's account information and settings at the top.

Deskripsi program

Input karakter tunggal dari pengguna ke dalam kategori huruf vokal, konsonan, atau bukan huruf. Menggunakan struktur percabangan bertingkat if-else if-else, kode pertama-tama mendeteksi apakah input cocok dengan salah satu huruf vokal (baik kapital maupun kecil) secara eksplisit menggunakan operator logika OR (||), kemudian memeriksa apakah karakter tersebut berada dalam rentang urutan alfabet untuk menentukan status konsonan, dan terakhir menangani input lain (seperti angka atau simbol) sebagai kategori "bukan huruf" jika kedua syarat sebelumnya tidak terpenuhi.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("masukan 4 angka: ")

    fmt.Scan(&n)

    if n >= 1000 && n <= 9999 {
        d1 := n / 1000
        d2 := (n / 100) % 10
        d3 := (n / 10) % 10
        d4 := n % 10

        if d1 < d2 && d2 < d3 && d3 < d4 {
            fmt.Println("bilangan terurut membesar")
        } else if d1 > d2 && d2 > d3 && d3 > d4 {
            fmt.Println("bilangan terurut mengecil")
        } else {
            fmt.Println("bilangan tidak terurut")
        }
    } else {
        fmt.Println("masukan antara 1000-9999")
    }
}
```

Screenshot program

The screenshot shows a code editor interface with a dark theme. On the left is the code editor pane displaying a Go file named 17novv.go. The code prompts the user to enter a four-digit number and then checks if the digits are in ascending or descending order. The code editor has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the command PS E:\pemrograman> go run "e:\pemrograman\17novv.go\guided03.go" followed by three examples of input and output. The first example shows the digits 1234 as 'bilangan terurut membesar'. The second example shows 1215 as 'bilangan tidak terurut'. The third example shows 6543 as 'bilangan terurut mengecil'. To the right of the terminal is a floating window titled 'NAMA : FAREL TRI JULIAN' with the text 'NIM : 109082500163'. At the bottom of the screen is a status bar with various icons and text.

```
17novv.go > guided03.go > main
1    package main
2
3    import "fmt"
4
5    func main() {
6        var n int
7
8        fmt.Print("masukan 4 angka: ")
9        fmt.Scan(&n)
10
11       if n >= 1000 && n <= 9999 {
12           d1 := n / 1000
13           d2 := (n / 100) % 10
14           d3 := (n / 10) % 10
15           d4 := n % 10
16
17           if d1 < d2 && d2 < d3 && d3 < d4 {
18               fmt.Print("bilangan terurut membesar")
19           } else if d1 > d2 && d2 > d3 && d3 > d4 {
20               fmt.Print("bilangan terurut mengecil")
21           } else {
22               fmt.Print("bilangan tidak terurut")
```

PROBLEMS 36 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\guided03.go"
masukan 4 angka: 1234
bilangan terurut membesar
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\guided03.go"
masukan 4 angka: 1215
bilangan tidak terurut
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\guided03.go"
masukan 4 angka: 6543
bilangan terurut mengecil
PS E:\pemrograman> 
```

NAMA : FAREL TRI JULIAN
NIM : 109082500163

Ln 2, Col 19 | 42 karakter | Terusan | 100% | Windows | UTF-8

Ln 15, Col 21 | Tab Size: 4 | UTF-8 | CRLF | Go | Signed out | 1.25.1 |

Deskripsi program

Program ini untuk menganalisis pola urutan angka pada sebuah bilangan bulat empat digit yang diinputkan oleh pengguna. Setelah memvalidasi bahwa input berada dalam rentang 1000 hingga 9999, program memecah bilangan tersebut menjadi digit-digit terpisah (ribuan, ratusan, puluhan, dan satuan) menggunakan kombinasi operasi pembagian bilangan bulat dan modulo (%). Program kemudian membandingkan digit-digit tersebut untuk menentukan apakah susunannya membentuk urutan yang membesar secara ketat ($d1 < d2 < d3 < d4$) atau mengecil secara ketat, lalu mencetak status urutan tersebut atau memberikan pesan peringatan jika input tidak memenuhi syarat jumlah digit.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var kg, gram, hargaperkg, jasa int

    fmt.Print("Masukan berat total : ")

    fmt.Scan(&gram)

    hargaperkg = 10000

    kg = gram / 1000

    gram = gram % 1000

    hargakg := kg * hargaperkg

    if kg >= 10 {

        jasa = 0

    } else {

        if gram >= 500 {

            jasa = gram * 5

        } else {

            jasa = gram * 15

        }

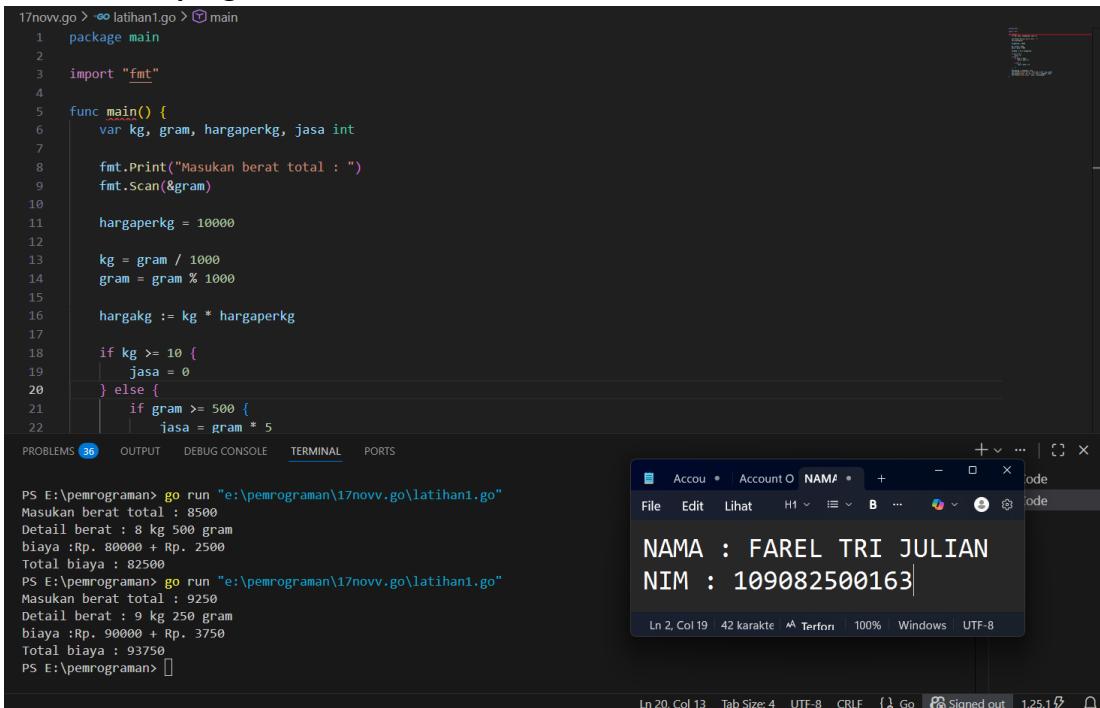
    }

    totalharga := hargakg + jasa

    fmt.Printf("Detail berat : %d kg %d gram\n", kg, gram)
    fmt.Printf("biaya :Rp. %d + Rp. %d\n", hargakg, jasa)
    fmt.Printf("Total biaya : %d\n", totalharga)

}
```

Screenshot program



```
17novv.go > ⌂ lathan1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kg, gram, hargaperkg, jasa int
7
8     fmt.Println("Masukan berat total : ")
9     fmt.Scan(&gram)
10
11     hargaperkg = 10000
12
13     kg = gram / 1000
14     gram = gram % 1000
15
16     hargakg := kg * hargaperkg
17
18     if kg >= 10 {
19         jasa = 0
20     } else {
21         if gram >= 500 {
22             jasa = gram * 5
23         }
24     }
25
26     biaya := hargakg + jasa
27
28     fmt.Println("Detail berat : ", kg, "kg", gram, "gram")
29     fmt.Println("biaya :Rp.", hargakg, "+ Rp.", jasa)
30     fmt.Println("Total biaya : ", biaya)
31
32 }
```

PROBLEMS 36 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\latihan1.go"
Masukan berat total : 8500
Detail berat : 8 kg 500 gram
biaya :Rp. 80000 + Rp. 2500
Total biaya : 82500
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\latihan1.go"
Masukan berat total : 9250
Detail berat : 9 kg 250 gram
biaya :Rp. 90000 + Rp. 3750
Total biaya : 93750
PS E:\pemrograman>
```

File Edit Lihat H1 ... B ... Accou Account NAMA + - × Code

NAMA : FAREL TRI JULIAN
NIM : 109082500163

Ln 2, Col 13 42 karakter ⌂ Terfor 100% Windows UTF-8

Ln 20, Col 13 Tab Size: 4 CRLF ⌂ Go ⌂ Signed out 1.25.1 ⌂

Deskripsi program

Program ini menghitung total biaya pengiriman logistik dengan cara mengonversi input berat total (dalam gram) menjadi satuan kilogram dan sisa gram menggunakan operasi pembagian dan modulo (%). Program menerapkan logika percabangan bersarang untuk menentukan biaya jasa tambahan: biaya jasa menjadi gratis jika berat mencapai 10 kg atau lebih, namun jika kurang, biaya jasa dikenakan berdasarkan sisa gram (tarif lebih rendah Rp5/gram jika sisa > 500 gram, dan tarif lebih tinggi Rp15/gram jika sisa < 500 gram). Hasil akhirnya menampilkan rincian konversi berat, komponen biaya terpisah, dan total harga yang harus dibayar menggunakan format string `fmt.Printf`.

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 = "B"
    } else if nam > 40 {
        nmk = "C"
    } else {
        nmk = "D"
    }
}
```

```
fmt.Println("Nilai mata kuliah: ", nmk)  
}
```

Screenshot program

```
17novv.go > 100 latihan2.go > main  
1 package main  
2  
3 import "fmt"  
4  
5 func main() {  
6     var nam float64  
7     var nmk string  
8     fmt.Println("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 = "B"  
20    } else if nam > 40 {  
21        nmk = "C"  
22    } else {  
PROBLEMS 36 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
Nilai akhir mata kuliah: 80.1  
Nilai mata kuliah: A  
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\latihan2.go"  
Nilai akhir mata kuliah: 93.5  
Nilai mata kuliah: A  
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\latihan2.go"  
Nilai akhir mata kuliah: 70.6  
Nilai mata kuliah: B  
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\latihan2.go"  
Nilai akhir mata kuliah: 49.5  
Nilai mata kuliah: C  
PS E:\pemrograman>  
Ln 10, Col 18 Tab Size: 4 UTF-8 CRLF { } Go ⚙️ Signed out 1.25.1 ⚙️
```

Deskripsi program

- pada kode yang diberikan sebelumnya (saat belum di perbaiki), Ketika kita input nilai 80.1 maka output nya akan error. Karena ada kesalahan di dalam tipe data variable.
- Program tersebut memiliki kesalahan tipe data yang fatal karena variabel nam dideklarasikan sebagai float64 (angka desimal) namun dipaksa menerima nilai string (huruf "A", "B", dll) yang seharusnya disimpan ke dalam variabel nmk, serta penggunaan tanda kutip yang akan menyebabkan syntax error. Selain itu, terdapat kesalahan logika karena menggunakan serangkaian if terpisah tanpa else, yang menyebabkan nilai input tinggi (misalnya 90) akan terus memenuhi syarat kondisi di bawahnya ($90 > 72.5$, $90 > 65$, dst.) sehingga predikat akhirnya tertimpa menjadi nilai yang salah (misalnya "D"), alur yang benar seharusnya menggunakan struktur if-else if agar pengecekan berhenti segera setelah satu kondisi terpenuhi, dan hasil huruf mutu disimpan ke variabel nmk untuk dicetak di akhir.
- program yang sudah diperbaiki seperti pada program yang telah saya berikan / pada screenshot.

3. Tugas 3

Source code

```
package main
import "fmt"
func main() {
    var n int
    fmt.Print("masukan bilangan bulat :")
    fmt.Scan(&n)
    fmt.Print("Faktor :")
    prima := true

    for i := 1; i <= n; i++ {
        if n%i == 0 {
            fmt.Print(i, " ")
            if i == 1 || i == n {
            } else if i > 1 && i < n {
                prima = false
            }
        }
    }
    if n == 1 {
        prima = false
    }
    fmt.Println()
    fmt.Println("Prima :", prima)
}
```

Screenshoot program

```
17novv.go > ⚡ latihan3.go > main
4
5 func main() {
6     var n int
7     fmt.Print("masukan bilangan bulat :")
8     fmt.Scan(&n)
9     fmt.Print("Faktor :")
10    prima := true
11
12    for i := 1; i <= n; i++ {
13        if n%i == 0 {
14            fmt.Print(i, " ")
15            if i == 1 || i == n {
16                } else if i > 1 && i < n {
17                    prima = false
18                }
19            }
20        }
21    if n == 1 {
22        prima = false
23    }
24    fmt.Println()
25
PROBLEMS 37 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\latihan3.go"
masukan bilangan bulat :12
1 2 3 4 6 12
Prima : false
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\tempCodeRunnerFile.go"
17novv.go\tempCodeRunnerFile.go:1:1: expected 'package', found fmt
PS E:\pemrograman> go run "e:\pemrograman\17novv.go\latihan3.go"
masukan bilangan bulat :12
Faktor :1 2 3 4 6 12
Prima : false
PS E:\pemrograman> []

```

NAMA : FAREL TRI JULIAN
NIM : 109082500163

Deskripsi program

Program untuk mencari seluruh faktor dari sebuah bilangan bulat input sekaligus mengidentifikasi apakah bilangan tersebut termasuk bilangan prima atau bukan. Melalui struktur perulangan for yang berjalan dari 1 hingga n, program memeriksa keterbagian angka menggunakan operator modulo (%) untuk mencetak setiap faktor yang ditemukan. Secara bersamaan, logika validasi di dalam loop akan mengubah status variabel boolean prima (yang awalnya diset true) menjadi false apabila ditemukan faktor pembagi selain angka 1 dan bilangan itu sendiri, serta program juga secara eksplisit menangani pengecualian bahwa angka 1 bukan bilangan prima sebelum menampilkan status akhirnya.