

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE IF



Disusun oleh:

NABIL NAILUR RIDHO

109082530008

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 a bool

    fmt.Scan(&usia)

    fmt.Scan(&a)

    if usia >= 17 && a == true{

        fmt.Println("bisa membuat ktp")

    } else {

        fmt.Println("belum bisa memmbuat ktp")

    }

}
```

Screenshot program

```
coba2.go
10
Positif
PS C:\TestMinggu9> go run coba2.go
-3
Bukan positif
PS C:\TestMinggu9> go run coba2.go
5
Positif
PS C:\TestMinggu9> go run coba2.go
0
Bukan positif
PS C:\TestMinggu9> ]
```

Deskripsi program

Program Go ini untuk memilih kondisi untuk memvalidasi persyaratan pembuatan Kartu Tanda Penduduk (KTP). Sistem menerima dua parameter masukan, yakni usia pengguna dan status kelengkapan berkas (dalam bentuk *boolean*). Menggunakan operator logika AND (`&&`), program menetapkan aturan ketat bahwa izin pembuatan KTP hanya diberikan jika usia minimal 17 tahun dan status boolean bernilai true secara bersamaan. Jika salah satu syarat saja tidak terpenuhi, maka output akan menyatakan bahwa pengguna belum layak 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")

    }

}
```

Screenshot program

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows a file tree under the 'TEST' folder. Files listed include: assesment, Minggu5, Minggu6, Minggu9, Minggu10 (which contains lat10.1.go, lat10.2.go, lat10.3.go, nyoba.go, nyoba1.go, nyoba2.go), contoh2.go, contoh3.go, forloop.go, guided1.go, guided2.go, guided3.go, lat1.go, lat2.go, lat3.go, and test.go.
- EDITOR:** Displays the content of the 'nyoba2.go' file. The code defines a package 'main' with a function 'main()' that reads a character from standard input and prints whether it is a vowel ('vokal'), consonant ('konsonan'), or neither ('bukan huruf').
- TERMINAL:** Shows the command 'go run nyoba2.go' being run in the directory 'C:\Test\Minggu10'. The output asks for a character input, which is then categorized as a vowel ('vokal'), consonant ('konsonan'), or neither ('bukan huruf').
- OUTPUT:** A floating terminal window titled 'Nama Nal' displays the input and output from the terminal command, along with the student's name, NIM, and class information.

Deskripsi program

Program Go ini buat mengidentifikasi dan mengelompokkan karakter input ke dalam kategori vokal, konsonan, atau karakter non-alfabet. Pengecekan dimulai dengan membandingkan input terhadap daftar huruf vokal (baik huruf besar maupun kecil) menggunakan operator OR. Apabila input bukan vokal, program melanjutkannya dengan memeriksa apakah karakter tersebut berada dalam rentang huruf abjad standar (a-z atau A-Z) untuk diklasifikasikan sebagai konsonan. Karakter di luar kedua definisi tersebut, seperti angka atau tanda baca, akan ditandai sebagai "bukan huruf".

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var bilangan, d1, d2, d3, d4 int
```

```
var urutan string

fmt.Println("Bilangan: ")

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.Println("Digit pada bilangan", bilangan,
urutan)

}
```

Screenshot program

```
1 package main
2 import "fmt"
3 func main(){
4     var bilangan, d1, d2, d3, d4 int
5     var urutan string
6
7     fmt.Println("Bilangan: ")
8     fmt.Scan(&bilangan)
9     d1 = bilangan / 1000
10    d2 = (bilangan % 1000) / 100
11    d3 = (bilangan % 100) / 10
12    d4 = bilangan % 10
13
14    if d1 < d2 && d2 < d3 && d3 < d4 {
15        urutan = "terurut membesar"
16    } else if d1 > d2 && d2 > d3 && d3 > d4{
17        urutan = "terurut mengecil"
18    } else {
19        urutan = "tidak terurut"
20    }
21    fmt.Println("Digit pada bilangan", bilangan, urutan)
22 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Test\Minggu10> go run nyoba3.go
Bilangan: 2489
Digit pada bilangan 2489 terurut membesar
PS C:\Test\Minggu10> go run nyoba3.go
Bilangan: 3861
Digit pada bilangan 3861 tidak terurut
PS C:\Test\Minggu10> go run nyoba3.go
Bilangan: 9651
Digit pada bilangan 9651 terurut mengecil
PS C:\Test\Minggu10>
```

Nama Nal • + - X
File Edit View A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Nama : Nabil Nailur Ridho
NIM : 109082530008
Kelas: SIIF-13-07

Deskripsi program

Program Go ini buat membedah struktur sebuah bilangan bulat empat digit untuk menentukan pola urutan angkanya. Proses dimulai dengan teknik ekstraksi digit, di mana bilangan utama dipecah menjadi satuan-satuan terpisah (d_1 hingga d_4) melalui kombinasi operasi pembagian dan modulus. Setelah setiap digit didapatkan, program mengevaluasi hubungan antar digit tersebut secara sekuensial. Hasil evaluasi akan menyimpulkan apakah digit-digit tersebut tersusun dalam format yang konsisten membesar, konsisten mengecil, atau tidak memiliki keteraturan pola sama sekali.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var parsel, kilogram, gram, harga, biaya, total int
    fmt.Println("Berat parsel(gram): ")
    fmt.Scan(&parsel)

    kilogram = parsel / 1000
    gram = parsel % 1000
    harga = kilogram * 10000

    if parsel > 10000 {

        biaya = 0
        total = harga
    } else if gram >= 500 {

        biaya = gram * 5
        total = harga + biaya
    } else if gram < 500 {

        biaya = gram * 15
        total = harga + biaya
    }
}
```

```

        fmt.Printf("detail berat: %d kg + %d gr\n", kilogram,
gram)

        fmt.Printf("detail biaya: Rp. %d + Rp. %d\n", harga,
biaya)

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

}

```

Screenshot program

Deskripsi program

Program ini berfungsi untuk menghitung biaya pengiriman parsel berdasarkan berat total dalam gram. Pertama, program menerima input berat parsel lalu mengubahnya menjadi satuan kilogram dan sisa gram. Selanjutnya, harga dasar dihitung dari kilogram dikali 10.000. Setelah itu, program menentukan biaya tambahan berdasarkan jumlah gram: jika berat lebih dari 10.000 gram maka tidak ada biaya tambahan, jika $gram \geq 500$ maka biaya adalah $gram \times 5$, dan jika $gram < 500$ biaya adalah $gram \times 15$. Program kemudian menampilkan rincian berat, perhitungan biaya tambahan, dan total biaya pengiriman.

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)
```

```
}
```

Screenshoot program

The screenshot shows a Visual Studio Code interface with the following details:

- EXPLORER:** Shows files in the 'TEST' folder, including 'lat10.2.go' which is currently selected.
- CODE EDITOR:** Displays the content of 'lat10.2.go'. The code defines a package 'main' with a function 'main()' that prints the grade based on a numerical input. The output window shows the execution of the program with various inputs and their corresponding letter grades.
- OUTPUT:** Shows the terminal output where the program is run multiple times with different inputs (80.1, 93.5, 70.6, 49.5) and outputs the letter grades (A, A, B, D).
- DEBUG CONSOLE:** Shows the input and output of the program, including the user's name, NIM, and class information.

Deskripsi program

Program ini buat mengonversi nilai akhir mata kuliah menjadi nilai huruf berdasarkan rentang tertentu. Pengguna memasukkan nilai numerik, kemudian program membandingkannya melalui serangkaian kondisi untuk menentukan nilai huruf: nilai di atas 80 menjadi A, di atas 72.5 menjadi AB, di atas 65 menjadi B, di atas 57.5 menjadi BC, di atas 50 menjadi C, di atas 40 menjadi D, dan sisanya E. Program ini memanfaatkan struktur *if–else* bertingkat, sehingga setiap nilai numerik dapat diklasifikasikan secara tepat sesuai standar penilaian.

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

Jawaban : maka keluarannya nmk : A, dan sesuai, tetapi jika belum di perbaiki maka keluarannya Adalah nam : D, dan tidak sesuai.

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

Jawaban: Program memiliki kesalahan sintaks karena variabel nam (tipe *float*) keliru digunakan untuk menyimpan data huruf (*string*), padahal seharusnya menggunakan variabel nmk, dan terdapat kesalahan logika fatal akibat penggunaan if terpisah tanpa else yang menyebabkan nilai huruf terus tertimpa oleh kondisi di bawahnya (misalnya nilai 80.1 yang seharusnya "A" akan berubah menjadi "D"). Perbaikannya adalah mengganti target variabel menjadi nmk dan mengubah struktur kendali menjadi if–else if agar program berhenti mengecek setelah satu kondisi terpenuhi.

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

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

Jawaban: Sudah di perbaiki seperti pada gambar yang terdapat di atas soal dan deskripsi program berikut.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var n, i, faktor int
    var prima bool
    fmt.Print("masukkan bilangan: ")
    fmt.Scan(&n)
    faktor = 0

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

    fmt.Println()

    fmt.Print("Prima: ")
    if faktor == 2 {
        prima = true
    }
    fmt.Println(prima)
}
```

Screenshot program

The screenshot shows a Visual Studio Code (VS Code) interface with a dark theme. The left sidebar has an 'EXPLORER' view showing a file tree under 'TEST'. The 'lat10.3.go' file is selected. The main editor area contains the following Go code:

```
1 package main
2 import "fmt"
3
4 func main() {
5     var n, i, faktor int
6     var prima bool
7     fmt.Println("masukkan bilangan: ")
8     fmt.Scan(&n)
9     faktor = 0
10
11    fmt.Print("faktor: ")
12    for i = 1; i <= n; i++ {
13        if n%i == 0 {
14            fmt.Print(" ", i)
15            faktor++
16        }
17    }
18
19    fmt.Println()
20
21    fmt.Print("Prima: ")
22    if faktor == 2 {
23        prima = true
24    } else {
25        prima = false
26    }
27
28    fmt.Println(prima)
29}
```

The terminal at the bottom shows the execution of the program:

```
PS C:\Test\Minggu10> go run lat10.3.go
masukkan bilangan: 12
faktor: 1 2 3 4 6 12
Prima: false
PS C:\Test\Minggu10> go run lat10.3.go
masukkan bilangan: 7
faktor: 1 7
Prima: true
PS C:\Test\Minggu10>
```

Deskripsi program

Program ini buat menentukan apakah sebuah bilangan merupakan bilangan prima serta menampilkan semua faktor dari bilangan tersebut. Pengguna memasukkan bilangan bulat, lalu program menggunakan perulangan dari 1 hingga n untuk mencari semua nilai yang dapat membagi n (faktor). Setiap faktor ditampilkan dan dihitung jumlahnya. Setelah perulangan selesai, jika jumlah faktor tepat dua (yaitu 1 dan dirinya sendiri), maka bilangan tersebut dinyatakan prima; jika tidak, maka bukan bilangan prima. Program ini menggabungkan konsep perulangan, pengecekan modulus, dan logika sederhana untuk mengidentifikasi bilangan prima.