

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 10

ELSE-IF



Disusun oleh:

Akhmad Noval Annur

109082500100

S1IF-13-04

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 bilangan int
    fmt.Scan(&bilangan)

    if bilangan < 0 {
        bilangan = -bilangan
    }

    fmt.Println(bilangan)
}
```

Screenshot program

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)

CONTOH TANGKAPAN LAYAR:

The screenshot shows a Windows desktop environment. On the left is a code editor window titled 'Untitled (Workspace)' containing Go code. The code defines a package 'main' with a function 'main()' that reads an integer from standard input and prints it if it's non-negative, or its negative value if it's negative. In the bottom right corner of the code editor, there's a status bar with file information like 'modul10 > guided1.go > main' and line numbers 1 through 14. To the right of the code editor is a terminal window titled 'TERMINAL'. It shows command-line interactions in a dark-themed terminal. The user runs 'go run' on the file 'guided1.go' located at 'C:\SEMESTER 1\modul10'. The terminal output shows the program's behavior: it prints 'true' and 'bisa membuat ktp' when given a positive number, and 'false' and 'belum bisa membuat ktp' when given a negative number. The terminal also displays the current date and time: 'PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided1.go" 17 true bisa membuat ktp PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided1.go" 20 false belum bisa membuat ktp PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided1.go" 15 true belum bisa membuat ktp PS C:\SEMESTER 1\modul10>'. At the very bottom of the screen is a taskbar with various icons for system functions like search, file explorer, and browser.

Deskripsi program

Program ini digunakan untuk menentukan apakah seseorang memenuhi syarat untuk membuat KTP. Program meminta dua input, yaitu usia dan status kepemilikan Kartu Keluarga (KK). Setelah menerima masukan, program memeriksa apakah usia yang diberikan minimal 17 tahun dan nilai boolean KK bernilai true. Jika kedua syarat tersebut terpenuhi, program menampilkan pesan “bisa membuat KTP”. Namun jika salah satu atau kedua syarat tidak terpenuhi misalnya usianya belum cukup atau tidak memiliki KK program menghasilkan keluaran ”belum bisa membuat KTP”. Contoh output menunjukkan tiga kasus berbeda yang menggambarkan bagaimana kondisi program bekerja berdasarkan variasi input usia dan status KK.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var bilangan int
    var teks string
    fmt.Scan(&bilangan)
    teks = "bukan positif"
    if bilangan > 0 {
        teks = "positif"
    }
    fmt.Println(teks)
}
```

Screenshot program

The screenshot shows a Visual Studio Code (VS Code) interface. The Explorer sidebar on the left lists files in the workspace, including 'latsol4.go', 'latsol5.go', 'guided2.go' (the active file), 'guided1.go', 'guided3.go', 'soal1.go', 'soal2.go', and 'soal3.go'. The code editor displays the following Go code:

```
modul10 > guided2.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     huruf = (x >= 'a' && x <= 'z') || (x >= 'A'
10    vKecil = x == 'a' || x == 'i' || x == 'u'
11    vBesar = x == 'A' || x == 'I' || x == 'U'
12    if huruf && (vKecil || vBesar) {
13        fmt.Println("vokal")
14    } else if huruf && !(vKecil || vBesar) {
15        fmt.Println("konsonan")
16    } else {
17        fmt.Println("bukan huruf")
18    }
19 }
```

The Terminal tab on the right shows the output of running the program:

```
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided2.go"
A
vokal
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided2.go"
F
konsonan
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided2.go"
1
bukan huruf
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided2.go"
$
bukan huruf
PS C:\SEMESTER 1\modul10>
```

Deskripsi program

Program ini digunakan untuk menentukan apakah sebuah karakter yang dimasukkan merupakan huruf vokal, huruf konsonan, atau bukan huruf sama sekali. Input berupa satu karakter, kemudian program memeriksa apakah karakter tersebut termasuk huruf kecil atau huruf besar. Jika termasuk huruf, program kembali mengecek apakah karakter tersebut ada dalam kumpulan huruf vokal baik versi besar maupun kecil. Apabila kondisi vokal terpenuhi, program mencetak “vokal”. Jika karakter adalah huruf tetapi bukan vokal, maka hasilnya “konsonan”. Namun bila input tidak termasuk alfabet, program langsung menampilkan “bukan huruf”. Contoh output pada PDF menunjukkan berbagai kemungkinan masukan seperti huruf kapital, huruf kecil, angka, dan simbol.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var bilangan int
    var hasil bool
    fmt.Scan(&bilangan)
    hasil = bilangan%2 == 0 && bilangan < 0
    fmt.Println(hasil)
}
```

Screenshot program

```
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 mengcil"
    } else {
        teks = "tidak terurut"
    }
    fmt.Println("Digit pada bilangan", bilangan)
}
```

Deskripsi program

Program ini digunakan untuk mengevaluasi apakah digit-digit dari suatu bilangan empat angka tersusun secara membesar, mengcil, atau tidak teratur. Bilangan yang dimasukkan harus

berada pada rentang 1000–9999. Program memisahkan setiap digit dari ribuan, ratusan, puluhan, hingga satuan menggunakan operasi modulo dan pembagian. Setelah digit didapatkan, program membandingkan urutan nilainya. Jika digit ribuan < ratusan < puluhan < satuan, maka bilangan dikatakan “terurut membesar”. Jika kebalikannya, yaitu digit ribuan > ratusan > puluhan > satuan, maka dinamakan “terurut mengecil”. Selain dua pola itu, kondisi lain dianggap “tidak terurut”. Contoh output menunjukkan beberapa variasi bilangan dan hasil keterurutannya.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var o, m int
    fmt.Scan(&o)

    m = o / 2
    if o%2 != 0 {
        m = m + 1
    }

    fmt.Println(m)
}
```

Screenshot program

```
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\soal1.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\soal1.go"
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\soal1.go"
Berat parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS C:\SEMESTER 1\modul10>
```

Deskripsi program

Program ini digunakan untuk menghitung biaya pengiriman paket berdasarkan berat totalnya dalam gram. Mula-mula, pengguna memberikan input berupa berat parsel. Program kemudian mengonversi berat tersebut menjadi kilogram dan sisa gram yang tidak mencapai satu kilogram penuh. Biaya pengiriman dasar dihitung dari jumlah kilogram dikali Rp10.000. Sisa gram dikenai biaya tambahan sesuai aturan: jika sisa berat kurang dari 500 gram, biaya tambahan adalah Rp15 per gram; jika lebih atau sama dengan 500 gram, biaya tambahan hanya Rp5 per gram. Namun, apabila total berat paket lebih dari 10 kilogram, maka sisa gram diigratiskan sepenuhnya sehingga hanya biaya kilogram yang dihitung. Contoh output pada PDF menggambarkan tiga kasus berbeda yang menunjukkan bagaimana aturan tersebut diterapkan untuk berbagai kombinasi berat paket.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var bilangan int
    var teks string

    fmt.Scan(&bilangan)

    teks = "bukan"
    if bilangan%2 == 0 && bilangan < 0 {
        teks = "genap negatif"
    }

    fmt.Println(teks)
}
```

Screenshot program

The screenshot shows a Windows desktop environment with the Visual Studio Code (VS Code) application open. The code editor displays a Go file named `soal2.go` with the following content:

```
modul10 > soal2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n float64
7     var m string
8
9     fmt.Scan(&n)
10
11    if n > 80 {
12        m = "A"
13    } else if n > 72.5 {
14        m = "AB"
15    } else if n > 65 {
16        m = "B"
17    } else if n > 57.5 {
18        m = "BC"
19    } else if n > 50 {
20        m = "C"
21    } else if n > 40 {
22        m = "D"
23    } else {
24        m = "E"
}
Ln 17, Col 25 Tab Size: 4 UTF-8 CRLF
```

The terminal window on the right shows the command `go run "c:\SEMESTER 1\modul10\soal2.go"` being run three times, resulting in outputs A, B, and D respectively, corresponding to input values 93.5, 70.6, and 49.5.

Deskripsi program

Latihan soal 2 berisi program yang seharusnya mengonversi nilai akhir mata kuliah (NAM) menjadi nilai huruf (NMK) menggunakan beberapa kondisi berurutan. Program membaca nilai desimal, kemudian menentukan kategori A, AB, B, BC, C, D, atau E berdasarkan rentang nilai yang telah ditentukan. Meski tujuannya jelas, program yang ada pada modul masih memiliki beberapa kesalahan yang membuat hasilnya tidak sesuai. Kesalahan utamanya adalah variabel nam yang bertipe angka tetapi malah diisi teks sehingga logikanya menjadi tidak valid. Selain itu, variabel nmk yang seharusnya digunakan untuk menyimpan nilai huruf justru tidak pernah diberi nilai sehingga output menjadi kosong. Struktur percabangan juga salah karena semua kondisi menggunakan `if` secara terpisah, bukan `else-if`, sehingga nilai kategori bisa saling menimpa. Urutan kondisi juga tidak berjenjang sehingga hasil akhir tidak bisa ditentukan dengan benar untuk semua nilai. Dengan kesalahan-kesalahan tersebut, program gagal menampilkan nilai huruf yang sesuai dan perlu diperbaiki agar dapat bekerja sebagaimana mestinya.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Scan(&x, &y)

    fmt.Println(y%x == 0)
    fmt.Println(x%y == 0)
}
```

Screenshot program

```
File Edit Selection View ... ← → Q Untitled (Workspace)
EXPLORER UNTITLED (WORKSPACE)
    > gotraining
    > tulis
    > laprak2
    > laprak3
    > laprak modul 5...
    > evaluasi
    > tulisbahasalain
    > modul9
    > modul10
        > guided1.go
        > guided2.go
        > guided3.go
        > soal1.go
        > soal2.go
        > soal3.go
modul10 > soal3.go > main
5 func main() {
15 }
16
17
18
19
20
21
22
23
24
25
26
TERMINAL ...
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\soal3.go"
12
1 2 3 4 6 12
false
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\soal3.go"
7
1 7
true
PS C:\SEMESTER 1\modul10>
Activate Windows
Go to Settings to activate Windows.
1.25.1 Go Live Prettier
Ln 23, Col 22 Tab Size: 4 UTF-8 CRLF Go Finish Setup 1:04 24°C 22:39 23/11/2025
```

Deskripsi program

Program ini digunakan untuk mencari semua faktor dari sebuah bilangan bulat b yang lebih dari 1. Setelah menerima input bilangan, program melakukan iterasi dari 1 hingga b untuk memeriksa setiap angka apakah merupakan faktor dengan cara mengecek apakah b habis dibagi angka tersebut. Semua faktor yang memenuhi kondisi ditampilkan secara berurutan. Setelah itu, program melanjutkan menentukan apakah bilangan tersebut merupakan bilangan prima dengan cara menghitung jumlah faktornya. Jika hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri, maka bilangan tersebut dikategorikan sebagai prima; sebaliknya, jika jumlah faktornya lebih

dari dua, maka bukan prima. Contoh output menunjukkan dua kasus yang memperlihatkan perbedaan antara bilangan komposit dan bilangan prima.