

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 ktp int
    var kk bool

    fmt.Scan(&ktp, &kk)
    if ktp >= 17 && kk {
        fmt.Println("bisa membuat ktp")
    } else{
        fmt.Println("belum bisa membuat ktp")
    }
}
```

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, there is a file explorer window titled 'UNTITLED (WORKSPACE)' containing several Go source files: 'guided1.go' (selected), 'guided2.go', 'guided3.go', 'soal1.go', 'soal2.go', and 'soal3.go'. In the center, a code editor window titled 'Untitled (Workspace)' displays the content of 'guided1.go'. The code is identical to the one shown in the previous text block. On the right, a terminal window is open with the command 'PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided1.go"' followed by two runs of the program. The first run, with input '17', outputs 'bisa membuat ktp'. The second run, with input '20', outputs 'belum bisa membuat ktp'. At the bottom, the taskbar shows various icons for system functions like search, task view, and system status.

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 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")
    }
}
```

Screenshot program

The screenshot shows a Microsoft 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 Terminal tab on the right shows the command-line output of running the 'guided2.go' program. The code in 'guided2.go' is a Go script that reads a character from standard input, checks if it's a vowel ('a', 'i', 'u', 'A', 'I', 'U'), and prints 'vokal' if true, or 'konsonan' if false. The terminal shows several runs of the program with different inputs like 'A', 'F', 'k', and '\$'.

```
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 }
```

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, d1, d2, d3, d4 int
    var teks string
    fmt.Print("Bilangan: ")
    fmt.Scan(&bilangan)
    d4 = bilangan % 10
    d3 = (bilangan % 100) / 10
    d2 = (bilangan % 1000) / 10
    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

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. The Explorer sidebar on the left lists several Go files: latsol5.go, guided2.go, guided3.go (which is currently selected), soal1.go, soal2.go, and soal3.go. The Terminal window on the right shows command-line output from running the program. The terminal output includes several lines of text starting with 'PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided3.go"' followed by digit sequences like 2489, 3861, and 9651, each preceded by the message 'Digit pada bilangan' and followed by either 'tidak terurut' or 'terurut'. The status bar at the bottom of the VS Code window shows the file path 'C:\SEMESTER 1\modul10\guided3.go', line 23, column 1, tab size 4, and UTF-8 encoding.

```
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided3.go"
Bilangan: 2489
Digit pada bilangan 2489 tidak terurut
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided3.go"
Bilangan: 3861
Digit pada bilangan 3861 tidak terurut
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\guided3.go"
Bilangan: 9651
Digit pada bilangan 9651 tidak terurut
PS C:\SEMESTER 1\modul10>
```

Deskripsi program

Program ini digunakan untuk mengevaluasi apakah digit-digit dari suatu bilangan empat angka tersusun secara membesar, mengecil, 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 g, k, s int
    var bkg, bs, t int

    fmt.Println("Berat parsel (gram): ")
    fmt.Scan(&g)

    k = g / 1000
    s = g % 1000

    bkg = k * 10000

    if k > 10 {
        bs = 0
    } else {
        if s >= 500 {
            bs = s * 5
        } else {
            bs = s * 15
        }
    }

    t = bkg + bs

    fmt.Printf("Detail berat: %d kg + %d gr\n", k, s)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", bkg, bs)
    fmt.Printf("Total biaya: Rp. %d\n", t)
```

}

Screenshot program

The screenshot shows a Windows desktop environment with the Visual Studio Code application open. The title bar reads "Untitled (Workspace)". The left sidebar displays the "EXPLORER" view with a tree structure of files under "UNTITLED (WORKSPACE)", including "guided2.go", "guided3.go", "soal1.go", "soal1.go", "soal2.go", and "soal3.go". The main editor area shows the content of "soal1.go". The terminal tab is active, showing command-line output for running Go programs. The bottom status bar shows file paths like "C:\SEMESTER 1\modul10\soal1.go", execution times, and system information.

```
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 n float64
    var m string

    fmt.Scan(&n)

    if n > 80 {
        m = "A"
    } else if n > 72.5 {
        m = "AB"
    } else if n > 65 {
        m = "B"
    } else if n > 57.5 {
        m = "BC"
    } else if n > 50 {
        m = "C"
    } else if n > 40 {
        m = "D"
    } else {
        m = "E"
    }

    fmt.Println(m)
}
```

```
fmt.Println(teks)  
}
```

Screenshot program

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. The left sidebar (EXPLORER) shows a file tree with several Go files: guided3.go, soal1.go, soal2.go, and soal3.go. The right pane (TERMINAL) shows a command-line interface (PS) running a Go program. The terminal output is as follows:

```
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\soal2.go"
93.5
A
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\soal2.go"
70.6
B
PS C:\SEMESTER 1\modul10> go run "c:\SEMESTER 1\modul10\soal2.go"
49.5
D
PS C:\SEMESTER 1\modul10>
```

At the bottom of the screen, the taskbar displays various icons for common Windows applications like File Explorer, Edge, and File History. The system tray shows the date (23/11/2025), time (1:05), battery level (24°C), and network status.

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 b int
    fmt.Scan(&b)

    c := 0

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

    fmt.Println()

    if c == 2 {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }
}
```

Screenshot program

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window titled 'Untitled (Workspace)'. The workspace contains three files: 'soal1.go', 'soal2.go', and 'soal3.go'. The 'soal3.go' file is open in the editor, displaying the Go code provided above. The code editor has syntax highlighting for Go, with keywords like 'func', 'main', 'if', and 'fmt' in blue, and numbers in green. The terminal tab to the right of the editor shows command-line output. The terminal window title is 'Terminal' and it displays the following session:

```
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>
```

At the bottom of the screen, the taskbar is visible with various icons for applications like File Explorer, Task View, and the Start button. A system tray icon for a battery level of 24% is also present.

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