

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL PRAKTIKUM 10

ELSE-IF



Disusun oleh:

Hanifan Bintang Wiraaji

109082500007

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 n int
    var kk bool
    fmt.Scan(&n)
    fmt.Scan(&kk)
    if n >= 17 && kk {
        fmt.Println("Bisa membuat ktp")
    } else {
        fmt.Println("Belum bisa membuat")
    }
}
```

Screenshot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     var kk bool
8     fmt.Scan(&n)
9     fmt.Scan(&kk)
10    if n >= 17 && kk {
11        fmt.Println("Bisa membuat ktp")
12    } else {
13        fmt.Println("Belum bisa membuat")
14    }
15 }
```

```
ed 1\main.go
17
true
Bisa membuat ktp
PS D:\Project\Coding\Go\ngampus\ALPRO\Modul 10> go run "d:\Project\Coding\Go\ngampus\ALPRO\Modul 10\1\main.go"
ed 1\main.go
false
false
Belum bisa membuat
PS D:\Project\Coding\Go\ngampus\ALPRO\Modul 10> go run "d:\Project\Coding\Go\ngampus\ALPRO\Modul 10\1\main.go"
ed 1\main.go
15
true
Belum bisa membuat
```

Deskripsi program

Program ini untuk memeriksa apakah seseorang bisa membuat ktp atau tidak. User akan diminta untuk input umur dan Boolean (true / false) mengenai apakah dia punya kk (kartu keluarga) atau tidak. Kemudian dalam program akan masuk ke pengkodisian if dimana jika user berumur lebih dari atau sama dengan 17 dan memiliki kk yang berarti booleannya True maka user bisa membuat ktp. Selain itu, jika salah satu atau kedua kondisi tidak terpenuhi maka user belum bisa membuat ktp.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var c rune

    fmt.Scanf("%c", &c)
```

```

if c == 'a' || c == 'i' || c == 'u' || c == 'e' || c == 'o' ||
c == 'A' || c == 'I' || c == 'U' || c == 'E' || c == 'O' {

    fmt.Println("vokal")

} else if (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') {

    fmt.Println("konsonan")

} else {

    fmt.Println("bukan huruf")

}

}

```

Screenshot program

The screenshot shows the Visual Studio Code interface. In the center, there is a code editor with the following Go code:

```

package main
import "fmt"
func main() {
    var c rune
    fmt.Scan("%c", &c)
    if c == 'a' || c == 'i' || c == 'u' || c == 'e' || c == 'o' ||
        c == 'A' || c == 'I' || c == 'U' || c == 'E' || c == 'O' {
        fmt.Println("vokal")
    } else if (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') {
        fmt.Println("konsonan")
    } else {
        fmt.Println("bukan huruf")
    }
}

```

Below the code editor is a terminal window showing the execution of the program:

```

$ go run "d:\Project\Coding\Go\ngampus\ALPRO\Modul 10\Guided 2\main.go"
109082500007
Kelas 7
Hanifan Bintang Wiraaji

```

Deskripsi program

Program ini untuk memeriksa apakah suatu huruf adalah vocal, konsonan atau malah bukan huruf. User akan diminta input 1 huruf bebas. Input ini akan dikategorikan sebagai variable rune karena hanya sesimbol dan untuk memudahkan. Kemudian akan masuk ke pengkondisian if yang pertama; akan diperiksa apakah huruf itu masuk ke jenis vocal yang termasuk a, i, u, e, o. Jika hurufnya adalah vocal maka outputnya adalah vocal. Pengkondisian kedua jika bukan huruf vocal tapi masih huruf abjad maka dikategorikan sebagai konsonan, dan outputnya adalah konsonan. Yang terakhir jika

bukan huruf sama sekali maka akan dikategorikan sebagai bukan huruf dan outputnya adalah bukan huruf.

3. Guided 3

Source Code

```
package main

import (
    "fmt"
)

func main() {
    var n int
    fmt.Scan(&n)

    angka1 := n / 1000
    angka2 := n % 1000 / 100
    angka3 := n % 1000 % 100 / 10
    angka4 := n % 10

    if angka1 <= angka2 && angka2 <= angka3 && angka3 <= angka4 {
        fmt.Printf("Digit pada bilangan %v terurut membesar", n)
    } else if angka1 >= angka2 && angka2 >= angka3 && angka3 >= angka4 {
        fmt.Printf("Digit pada bilangan %v terurut mengecil", n)
    }else {
        fmt.Printf("Digit pada bilangan %v tidak terurut", n)
    }
}
```

```
}
```

Screenshot program

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The main area is a code editor with a dark theme, displaying a Go program named 'main.go'. The code checks if a four-digit number is in ascending order (large), descending order (small), or not in order. It uses modulus and division to separate digits. The terminal below shows the program's output for three test cases: 2489, 3861, and 9651, with outputs indicating the digit order and the name 'Hanifan Bintang Wiraaji'.

```
package main
import (
    "fmt"
)
func main() {
    var n int
    fmt.Scan(&n)
    angka1 := n / 1000
    angka2 := n % 1000 / 100
    angka3 := n % 1000 % 100 / 10
    angka4 := n % 10
    if angka1 <= angka2 && angka2 <= angka3 && angka3 <= angka4 {
        fmt.Printf("Digit pada bilangan %v terurut membesar", n)
    } else if angka1 >= angka2 && angka2 >= angka3 && angka3 >= angka4 {
        fmt.Printf("Digit pada bilangan %v terurut mengecil", n)
    }else {
        fmt.Printf("Digit pada bilangan %v tidak terurut", n)
    }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Project\Coding\Go\ngampus\ALPRO\Modul 10> go run "d:\Project\Coding\Go\ngampus\ALPRO\Modul 10\Gu...
2489
Digit pada bilangan 2489 terurut membesar
PS D:\Project\Coding\Go\ngampus\ALPRO\Modul 10> go run "d:\Project\Coding\Go\ngampus\ALPRO\Modul 10\Gu...
3861
Digit pada bilangan 3861 tidak terurut
PS D:\Project\Coding\Go\ngampus\ALPRO\Modul 10> go run "d:\Project\Coding\Go\ngampus\ALPRO\Modul 10\Gu...
9651
Digit pada bilangan 9651 terurut mengecil
%PS D:\Project\Coding\Go\ngampus\ALPRO\Modul 10>

File Edit View Aa 105 ◀ ▶ - + ×

109882500007
Kelas 7
Hanifan Bintang Wiraaji

Ln 5, Col 24 46 character Plain t 100% Wind UTF-8

Deskripsi program

Program ini untuk memeriksa apakah suatu bilangan ribuan (4 angka) adalah bilangan berurut membesar, mengecil atau tidak berurut. User akan diminta memasukan bilangan ribuan (4 angka), kemudian program akan memisahkannya menjadi angka 1 – angka 4, rinciannya dengan modulo dan pembagian yang dilakukan di tiap angka seperti di kodingan yang tertera.

Kemudian program akan masuk ke pengkondisian if pertama yang dimana jika, angka 1 kurang dari sama dengan angka 2, dan angka 2 kurang dari sama dengan angka 3, dan angka 3 kurang dari sama dengan angka 4, maka akan masuk ke pengkondisian pertama dan memberi output bilangan ini berurut membesar. Kemudian pengkondisian kedua, hampir mirip seperti yang pengkondisian pertama hanya saja operator perbandingannya adalah lebih dari sama dengan, jika pengkondisianya true maka akan masuk ke if kedua yang memberi output bilangan ini sebagai berurut mengecil. Jika bilangan tidak termasuk ke kedua pengkondisian diatas maka akan masuk else yang memberi output bilangan ini tidak berurut.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var b int

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

    fmt.Scan(&b)

    kgberat := b / 1000

    gberat := b % 1000

    fmt.Printf("Detail berat: %v kg + %v gr\n", kgberat, gberat)

    kgbayaya := kgberat * 10000

    gbayaya := gberat

    if gbayaya >= 500 {
        gbayaya = gbayaya * 5
    } else {
        gbayaya = gbayaya * 15
    }

    fmt.Printf("Detail biaya: Rp. %v + Rp. %v\n", kgbayaya, gbayaya)
```

```

var totalbiaya int

if kgberat > 10 {

    totalbiaya = kgbaya

} else {

    totalbiaya = kgbaya + gbiaya

}

fmt.Printf("Total biaya: Rp. %v", totalbiaya)

}

```

Screenshoot program

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a project structure under "MODUL 10" with files like "main.go" for various sub-modules (1, 2, 3) and a "Soal" folder.
- Code Editor:** The main file "main.go" contains the following Go code:

```

package main

import "fmt"

func main() {
    var b int

    fmt.Println("Berat parcel(kg): ")
    fmt.Scan(&b)

    kgberat := b / 1000
    gberat := b % 1000
    fmt.Printf("Detail berat: %v kg + %v gr\n", kgberat, gberat)

    kgbaya := kgberat * 10000
    gbiaya := gberat
    if gbiaya >= 500 {
        gbiaya = gbiaya * 5
    } else {
        gbiaya = gbiaya * 15
    }
    fmt.Printf("Detail biaya: Rp. %v + Rp. %v\n", kgbaya, gbiaya)

    var totalbiaya int
    if b > 1000 {
        totalbiaya = kgbaya
    } else {
        totalbiaya = kgbaya + gbiaya
    }
    fmt.Printf("Total biaya: Rp. %v", totalbiaya)
}

```
- Terminal:** Shows the command "go run" being run, and the output:

```

109082500007
Kelas 7
Hanifan Bintang Wiraaji

```
- Status Bar:** Shows file statistics: Line 5, Column 24, 46 characters, Plain text, 100% zoom, Wind, and UTF-8 encoding.

The image shows two side-by-side instances of the Microsoft Visual Studio Code (VS Code) code editor. Both instances have the same file structure and code content, demonstrating a comparison or a split-screen view.

File Structure:

- Left window: Explorer sidebar shows a folder named "MODUL 10" containing several sub-folders like "Guided 1", "Guided 2", "Guided 3", "Soal 1", "Soal 2", "Soal 3", and "main.go".
- Right window: Explorer sidebar shows a similar file structure.

Code Content:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var b int
7
8     fmt.Print("Berat parsel(gram): ")
9     fmt.Scan(&b)
10
11     kberat := b / 1000
12     gberat := b % 1000
13     fmt.Printf("Detail berat: %v kg + %v gr\n", kberat, gberat)
14
15     kgbiaya := kberat * 10000
16     gbiaya := gberat
17     if gbiaya >= 500 {
18         gbiaya = gbiaya * 5
19     } else {
20         gbiaya = gbiaya * 15
21     }
22     fmt.Printf("Detail biaya: Rp. %v + Rp. %v\n", kgbiaya, gbiaya)
23
24     var totalbiaya int
25     if b > 10000 {
26         totalbiaya = kgbiaya
27     } else {
28         totalbiaya = kgbiaya + gbiaya
29     }
30     fmt.Printf("Total biaya: Rp. %v", totalbiaya)
31 }
```

Terminal Output:

Both terminals show the same command-line output:

```

PS D:\Project\Coding\Go\ngampus\ALPRO\Modul 10> go run "d:\Project\Coding\Go\ngampus\ALPRO\Modul 10\Soal 1"
Berat parsel(gram): 9258
Detail berat: 9 kg + 258 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\Project\Coding\Go\ngampus\ALPRO\Modul 10>
```

Below the terminals, status bars indicate the current line (Ln 5), column (Col 24), and character count (46 characters).

Deskripsi program

Program ini untuk menghitung rincian berat dan biaya ongkos sebuah parsel. Perhitungan biayanya adalah sebagai berikut;

1. Tiap 1 kg seharga Rp. 10.000
2. Tiap lebihan yang lebih dari sama dengan 500 gram, akan di hargai Rp. 5 per gram nya
3. Tiap lebihan yang kurang dari 500 gram, akan di hargai Rp. 15 per gram nya
4. Tiap parsel yang beratnya lebih dari 10 kg, user hanya perlu membayar tiap kg nya tidak perlu membayar lebihannya yg kurang dari 1 kg

User akan diminta input berat parsel dalam gram. Kemudian program akan mengkonversi berat parsel menjadi rincian berupa berat dalam kg dan gram sisanya. Kg didapat dari membagi berat parsel dengan 1000, dan gram sisanya didapat dari

modulo berat parsel dengan 1000. Kemudian rincian beratnya ini akan di tampilkan sebagai output dengan keterangan berat dalam (angka) kg + (angka) gram.

Lalu akan diperiksa biaya tiap beratnya. Harga Kg akan dihitung dengan mengkalikan berat kg dengan 10.000. Harga dalam gram akan di hitung dengan pengkondisian. Pertama jika berat lebih dari sama dengan 500, harganya akan dihitung dengan mengkalikan berat gram dengan 5. Kedua jika beratnya kurang dari 500, maka harganya akan dihitung dengan mengkalikan berat gram dengan 15. Kemudian hasil rincian biaya tiap berat akan di tampilkan sebagai output dengan keterangan detail biaya dalam harga kg + harga gram.

Terakhir total biaya, disini akan ada 2 pengkondisian. Pertama jika beratnya lebih dari 10 kg atau 10000 gram maka lebihan yang kurang dari 1 kg akan di gratiskan, yang berarti total biaya nya adalah harga kg saja. Kemudian pengkondisian keduanya adalah else atau jika beratnya kurang dari 10 kg atau 10000 gram harga akan di total antara harga kg dan gram. Kemudian total biaya tadi akan di tampilkan sebagai output terakhir.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var nam float64

    var nmk string

    fmt.Print("Nilai akhir mata kuliah: ")

    fmt.Scan(&nam)

    if nam > 80 {

        nam = "A"

    }

    if nam > 72.5 {

        nam = "AB"

    }
}
```

```
if nam > 65 {  
  
    nam = "B"  
  
}  
  
if nam > 57.5 {  
  
    nam = "BC"  
  
}  
  
if nam > 50 {  
  
    nam = "C"  
  
}  
  
if nam > 40 {  
  
    nam = "D"  
  
} else if nam <= 40 {  
  
    nam = "E"  
  
}  
  
fmt.Println("Nilai mata kuliah: ", nmk)  
  
}
```

The fixed code

```
package main  
  
import "fmt"  
  
  
func main() {  
  
    var nam float64  
  
    var nmk string  
  
    fmt.Print("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 if nam <= 40 {

    nmk = "E"

}

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

}
```

Screenshoot program

```

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 = "C"
    } else if nam > 50 {
        nmk = "D"
    } else if nam > 40 {
        nmk = "E"
    } else if nam <= 40 {
        nmk = "T"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}

```

Deskripsi program

Program ini untuk melihat nilai nmk dari nam yang di berikan. User akan diminta menginput nilai nam nya, kemudian program akan masuk pengkondisian untuk melihat rentang skor nmknya dari nilai nam, jika lebih 80 akan A, lebih dari 72.5 akan AB, dan seterusnya. Kemudian hasil nmk nya ditampilkan.

Pertanyaan mengenai program

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

Jawab: Isi kodingannya eror semua, tapi kalo di benerin dikit tanpa merubah if nya, Hasilnya D, tidak sesuai dengan spesifikasi soal

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

Jawab: Kesalahannya cukup banyak;

- Tanda kutipnya tidak sesuai dengan standar vs code
- 5 if berjejer menurun yang tidak berguna karena yg terpakai ujung2nya yang paling bawah, seharusnya if ditulis dengan format if, kemudian else if, dan diakhiri else
- Variable yang diubah oleh if salah, seharusnya nmk karena dia adalah variable string, dan juga karena nmk adalah rentang nilai A, B dll, jadi penggunaan nam pada source code salah

Penjelasan alur program, User diminta input nam yang adalah float64, setelah kode dibenarkan, nam akan diperiksa menurun secara terstruktur di if else, sehingga menghasilkan nmk yang sesuai dengan nam.

- Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

Jawab: kode sudah dibenarkan, hasilnya ada di screenshot program

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {

    var b, prima int

    fmt.Println("Bilangan: ")

    fmt.Scan(&b)

    if b > 1 {

        fmt.Println("Faktor: ")

        for i := 1; i <= b; i++ {

            faktor := b % i

            if faktor == 0 {

                fmt.Print(i, " ")

                prima += i

            }

        }

        fmt.Println("")

    }

    if b > 0 {

        bilPrima := false

        prima -= 1

    }

}
```

```
if prima == b {  
    bilPrima = true  
}  
  
fmt.Printf("Prima: %v", bilPrima)  
}  
}
```

Screenshot program

Deskripsi program

Program ini untuk memeriksa apakah suatu bilangan adalah bilangan prima dan menyajikan faktornya. User diminta input angka, program akan memeriksa faktor bilangan tersebut. Akan ada perulangan dengan i yang bernilai 1 dan terus bertambah sampai sama dengan angka yg diinput, di dalamnya akan menghitung modulo angka yang diinput dengan i , jika hasilnya sama dengan 0 maka di tampilkan sebagai faktor. Ini akan terus berulang sampai semua faktor tertulis, atau sampai i mencapai sejumlah bilangan yg diinput.

Kemudian untuk bilangan primanya, jadi dalam perulangan tadi tiap bilangan factor muncul akan di akumulasi dalam variable prima. Kemudian jika angka yang diinput adalah bilangan prima, contoh 7 adalah angka yang diinput user, maka variable prima akan berisi angka 8, jumlah akumulasi dari factor 7 yang hanya 1 dan 7. Kemudian variable prima akan dikurangi dengan 1, kemudian masuk pengkondisian if, jika

variable prima sama dengan angka yang diinput user maka akan ditampilkan keterangan true pada prima.