

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE-IF



Disusun oleh:

OFI ANDRE KHOIRUNIZA

109082500061

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 kk bool

    fmt.Scan(&usia, &kk)

    if usia >= 17 && kk {
        fmt.Println("bisa membuat KTP")
    } else {
        fmt.Println("belum bisa membuat KTP")
    }
}
```

Screenshot program

The screenshot shows a code editor with a dark theme. A Go program named 'guide1.go' is open. The code defines a main function that reads 'usia' (age) and 'kk' (family card status) from standard input. It then checks if both conditions are true using an if statement. If true, it prints "bisa membuat KTP"; otherwise, it prints "belum bisa membuat KTP". The code editor has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and SPELL CHECKER. The TERMINAL tab shows command-line history for running the program with different inputs. A floating terminal window titled 'Nama Ofi' displays the output of the program for a specific input.

```
guide1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var usia int
7     var kk bool
8
9     fmt.Scan(&usia, &kk)
10
11    if usia >= 17 && kk {
12        fmt.Println("bisa membuat KTP")
13    } else {
14        fmt.Println("belum bisa membuat KTP")
15    }
16 }
```

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 5

PS C:\Kuliah\Semester 1\Algoritma Pemrograman\latihan\latihan minggu 10> go run guide1.go
17 true
bisa membuat KTP
PS C:\Kuliah\Semester 1\Algoritma Pemrograman\latihan\latihan minggu 10> go run guide1.go
17 false
belum bisa membuat KTP
PS C:\Kuliah\Semester 1\Algoritma Pemrograman\latihan\latihan minggu 10> go run guide1.go
16 true
belum bisa membuat KTP
PS C:\Kuliah\Semester 1\Algoritma Pemrograman\latihan\latihan minggu 10>

Nama Ofi
File Edit Lihat Aa 8 N
Nama: Ofi Andre Khoiruniza
Kelas: S1IF-13-07
NIM: 109082500061

Deskripsi program

Program ini membaca dua input, yaitu usia dan status kepemilikan kartu keluarga. Setelah data diterima, program memeriksa apakah kedua syarat pembuatan KTP terpenuhi: usia minimal 17 tahun dan memiliki kartu keluarga. Pemeriksaan dilakukan menggunakan operator logika AND (`&&`) sehingga keduanya harus benar. Jika syarat terpenuhi, program mencetak “bisa membuat KTP”, dan jika tidak maka mencetak “belum bisa membuat KTP”.

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

```

:go guide2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x rune
7     var huruf, vKecil, vBesar bool
8
9     fmt.Scanf("%c", &x)
10
11    huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
12    vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o'
13    vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'
14
15    if huruf && (vKecil || vBesar) {
16        fmt.Println("vokal")
17    } else if huruf && !(vKecil || vBesar) {
18        fmt.Println("konsonan")
19    }

```

PROBLEMS 37 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 16

powerShell + - X

- PS C:\Kuliah\Smester 1\Algoritma Pemrograman\latihan\latihan minggu 10> go run guide2.go
 - a
vokal
 - B
konsonan
 - 1
bukan huruf
 - PS C:\Kuliah\Smester 1\Algoritma Pemrograman\latihan\latihan minggu 10>

Nama Ofi · +

File Edit Lihat AA ⚡⚙️

Nama: Ofi Andre Khoiruniza
Kelas: S1IF-13-07
NIM: 109082500061

Deskripsi program

Program ini membaca satu karakter dan memeriksa apakah karakter tersebut merupakan huruf vokal, huruf konsonan, atau bukan huruf. Pengecekan dilakukan dengan membandingkan karakter terhadap rentang alfabet, lalu menentukan apakah karakter tersebut termasuk vokal kecil atau besar. Setelah itu, struktur if-else digunakan untuk menentukan kategori berdasarkan kombinasi kondisi huruf dan vokal.

3. Guided 3

Source Code

```

package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3, d4 int
    var teks string
}

```

```
fmt.Println("Bilangan: ")  
bilangan := scanner.ScanInt()  
  
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 mengecil"  
} else {  
    teks = "tidak terurut"  
}  
fmt.Println("Digit pada bilangan", bilangan, teks)  
}
```

Screenshot program

```

:oo guide3.go > main
4
5 func main() {
6     var bilangan, d1, d2, d3, d4 int
7     var teks string
8     fmt.Println("Bilangan: ")
9     fmt.Scan(&bilangan)
10
11     d4 = bilangan % 10
12     d3 = (bilangan % 100) / 10
13     d2 = (bilangan % 1000) / 100
14     d1 = bilangan / 1000
15
16     if d1 < d2 && d2 < d3 && d3 < d4 {
17         teks = "terurut membesar"
18     } else if d1 > d2 && d2 > d3 && d3 > d4 {
19         teks = "terurut mengecil"
20     } else {
21         teks = "tidak terurut"
22     }
23
24     fmt.Println(teks)
25 }

```

PROBLEMS 42 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 21

powerShell + - X

- PS C:\Kuliah\Smester 1\Algoritma Pemrograman\latihan\latihan minggu 10> go run guide3.go

Bilangan: 1234

Digit pada bilangan 1234 terurut membesar
- PS C:\Kuliah\Smester 1\Algoritma Pemrograman\latihan\latihan minggu 10>

Bilangan: 4321

Digit pada bilangan 4321 terurut mengecil
- PS C:\Kuliah\Smester 1\Algoritma Pemrograman\latihan\latihan minggu 10>

Bilangan: 2143

Digit pada bilangan 2143 tidak terurut

Nama Ofi: + - X

File Edit Lihat AA ⚡ ⚡ ⚡

Nama: Ofi Andre Khoiruniza
 Kelas: S1IF-13-07
 NIM: 109082500061

Deskripsi program

Program ini membaca sebuah bilangan empat digit, memisahkan masing-masing digit menggunakan operasi modulo dan pembagian, lalu memeriksa apakah digit-digit tersebut membentuk urutan membesar, mengecil, atau tidak terurut. Dengan membandingkan urutan digit dari ribuan ke satuan, program menentukan kategori urutan dan mencetak hasilnya bersama bilangan asal.

TUGAS

1. Tugas 1

Source code

```

package main

import "fmt"

func main() {
    var gram, kg, sisa int
    var biayaKg, biayaSisa, total int
}

```

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

fmt.Scan(&gram)

kg = gram / 1000

sisa = gram % 1000

biayaKg = kg * 10000

if kg > 10 {

    biayaSisa = 0

} else if sisa >= 500 {

    biayaSisa = sisa * 5

} else {

    biayaSisa = sisa * 15

}

total = biayaKg + biayaSisa

fmt.Println("Detail berat:", kg, "kg +", sisa, "gr")

fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.",
biayaSisa)

fmt.Println("Total biaya: Rp.", total)

}
```

Screenshoot program

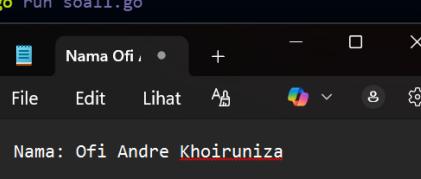


```
soal1.go > main
 5 func main() {
 6     gram := 1234
 7
 8     kg := gram / 1000
 9     sisa := gram % 1000
10
11     biayaKg := kg * 10000
12
13     if kg > 10 {
14         biayaSisa := 0
15     } else if sisa >= 500 {
16         biayaSisa := sisa * 5
17     } else {
18         biayaSisa := sisa * 15
19     }
20
21     total := biayaKg + biayaSisa
22
23
24
25
26 }
```

PROBLEMS 48 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 27

powershell + - ×

- PS C:\Kuliah\Semester 1\Algoritma Pemrograman\latihan\latihan minggu 10> go run soal1.go
Berat parsel (gram): 1234
Detail berat: 1 kg + 234 gr
Detail biaya: Rp. 10000 + Rp. 3510
Total biaya: Rp. 13510
- PS C:\Kuliah\Semester 1\Algoritma Pemrograman\latihan\latihan minggu 10>



Nama Ofi Andre Khoiruniza
Kelas: S1IF-13-07
NIM: 109082500061

Deskripsi program

Program ini menghitung biaya pengiriman parsel berdasarkan total berat dalam gram. Berat dikonversi menjadi kilogram dan sisa gram, lalu biaya dasar dihitung dari jumlah kilogram. Sisa gram dihitung sesuai aturan: gratis jika lebih dari 10 kg, tarif 5 per gram jika sisanya ≥ 500 , dan tarif 15 per gram jika sisanya < 500 . Setelah itu semua biaya dijumlahkan, dan program menampilkan detail berat serta total biaya.

2. Tugas 2

Source code

```
// PROGRAM YANG SALAH

package main

import "fmt"

func main() {
```

```
var nam float64

var nmk string

fmt.Println("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)

}
```

Screenshot program

The screenshot shows a code editor with a Go file named `soal2.go`. The code defines a `main()` function that reads a float value from the user and prints it as a grade letter based on the following scale:

```
5 func main() {
6     var nam float64
7     var nmk string
8     fmt.Print("Nilai akhir mata kuliah: ")
9     fmt.Scan(&nam)
10    if nam > 80 {
11        nmk = "A"
12    }
13    if nam > 72.5 {
14        nmk = "AB"
15    }
16    if nam > 65 {
17        nmk = "B"
18    }
19    if nam > 57.5 {
20        nmk = "BC"
21    }
22    if nam > 50 {
23 }
```

The terminal window below shows the command `go run soal2.go` being run, followed by several error messages indicating that string constants like "A", "AB", etc., cannot be assigned to a float64 variable. The final output shows the program running correctly for the input "Ofi Andre Khoiruniza".

```
PS C:\Kuliah\Semester 1\Algoritma Pemrograman\latihan minggu 10> go run soal2.go
# command-line-arguments
.\soal2.go:11:9: cannot use "A" (untyped string constant) as float64 value in assignment
.\soal2.go:14:9: cannot use "AB" (untyped string constant) as float64 va
.\soal2.go:17:9: cannot use "B" (untyped string constant) as float64 val
.\soal2.go:20:9: cannot use "BC" (untyped string constant) as float64 va
.\soal2.go:23:9: cannot use "C" (untyped string constant) as float64 val
.\soal2.go:26:9: cannot use "D" (untyped string constant) as float64 val
.\soal2.go:28:9: cannot use "E" (untyped string constant) as float64 val
o PS C:\Kuliah\Semester 1\Algoritma Pemrograman\latihan minggu 10>
```

Nama Ofi
Nama: Ofi Andre Khoiruniza
Kelas: S1IF-13-07
NIM: 109082500061

A. Output untuk input 80.1 & apakah sesuai spesifikasi?

Saat nilai 80.1 dimasukkan, baris `nam = "A"` akan dieksekusi sehingga `nam` berubah dari float menjadi string. Setelah itu kondisi-kondisi `if` berikutnya tetap berjalan dan akan terus menimpa nilai sebelumnya. Karena variabel `nmk` tidak pernah diisi, program akhirnya mencetak: "Nilai mata kuliah: (tanpa nilai apapun)". Jadi program tidak sesuai spesifikasi, karena tidak menghasilkan nilai huruf yang benar dan salah secara logika.

B. Kesalahan program & alur yang seharusnya

Kesalahan program:

- Variabel `nam` (float) diubah menjadi string, menyebabkan tipe data bercampur dan program tidak sesuai tujuan.
- Semua kondisi ditulis sebagai `if` terpisah, bukan `else-if`, sehingga kondisi yang lebih rendah tetap berjalan dan menimpa hasil sebelumnya.
- Variabel `nmk` tidak pernah diisi, tetapi justru yang dicetak.
- Struktur logika rentang nilai tidak dibuat berurutan—rule yang benar harus dievaluasi dari nilai tertinggi ke terendah.
- Tidak ada batas bawah/atas yang jelas antar kondisi.

Alur program seharusnya:

Nilai angka (`nam`) dibandingkan terhadap batas nilai secara berurutan menggunakan rangkaian `if ... else-if`, sehingga hanya satu kondisi yang dijalankan. Nilai huruf disimpan dalam variabel `nmk`, lalu dicetak sebagai output.

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

```
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

Deskripsi program

Program ini membaca nilai akhir mata kuliah dalam bentuk angka dan menentukan nilai huruf berdasarkan rentang nilai yang telah ditetapkan. Struktur if–else if digunakan untuk memastikan bahwa hanya satu kondisi yang dipilih sesuai nilai yang dimasukkan. Variabel nmk menyimpan nilai huruf yang sesuai, yang kemudian dicetak sebagai hasil akhir. Program ini memastikan logika penilaian berjalan berurutan dan tepat tanpa menimpa hasil sebelumnya.

3. Tugas 3-1

Source code

```
package main

import "fmt"

func main() {
    var n, i int
    fmt.Print("masukkan bilangan: ")
    fmt.Scan(&n)

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

```

        if n % i == 0 {
            fmt.Println(" ", i)
        }
    }
}

```

Screenshot program

The screenshot shows a code editor window with a dark theme. The file is named 'soal3-1.go' and contains the following Go code:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var n, i int
7     fmt.Print("masukkan bilangan: ")
8     fmt.Scan(&n)
9
10    fmt.Print("faktor: ")
11    for i = 1; i <= n; i++ {
12        if n % i == 0 {
13            fmt.Print(" ", i)
14        }
15    }
16 }

```

Below the code editor is a terminal window showing the execution of the program. It prompts the user to enter a number (100), then lists all factors of 100 from 1 to 100. To the right of the terminal is a small application window titled 'Nama Ofi' containing student information:

File	Edit	Lihat	A _{AA}	...	⚙️
Nama: Ofi Andre Khoiruniza					
Kelas: S1IF-13-07					
NIM: 109082500061					

Deskripsi program

Program ini meminta pengguna memasukkan sebuah bilangan bulat, kemudian mencari semua faktor dari bilangan tersebut. Pencarian faktor dilakukan dengan melakukan perulangan dari 1 sampai n, lalu setiap nilai i yang dapat membagi n secara sempurna ($n \% i == 0$) dicetak sebagai faktor. Program memanfaatkan perulangan dan pengecekan modulus untuk menentukan faktor secara sederhana dan langsung.

4. Tugas 3-2

Source code

```

package main

import "fmt"

func main() {

```

```
var b int
var jumlahFaktor int = 0

fmt.Print("Bilangan: ")
fmt.Scan(&b)

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

fmt.Print("Prima: ")
if jumlahFaktor == 2 {
    fmt.Println("true")
} else {
    fmt.Println("false")
}
```

Screenshot 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 floating terminal window titled 'Nama Ofi'. The code editor has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and SPELL CHECKER. The terminal window shows command-line output for two runs of the program.

```
soal3-2.go > main
2
3 import "fmt"
4
5 func main() {
6     var b int
7     var jumlahFaktor int = 0
8
9     fmt.Print("Bilangan: ")
10    fmt.Scan(&b)
11
12    fmt.Print("Faktor: ")
13    for i := 1; i <= b; i++ {
14        if b%i == 0 {
15            fmt.Print(i, " ")
16            jumlahFaktor++
17        }
18    }
19    fmt.Println()
20
21    fmt.Print("Prima: ")
22    if jumlahFaktor == 2 {
23        fmt.Println("true")
}
● PS C:\Kuliah\Semester 1\Algoritma Pemrograman\latihan\latihan minggu 10> go run soal3-2.go
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
○ PS C:\Kuliah\Semester 1\Algoritma Pemrograman\latihan\latihan minggu 10> go run soal3-2.go
Bilangan: 5
Faktor: 1 5
Prima: true
```

Deskripsi program

Program ini membaca sebuah bilangan kemudian mencari semua faktornya dengan memeriksa nilai dari 1 hingga b yang dapat membagi bilangan tersebut tanpa sisa. Setiap faktor dicetak, dan jumlah faktor dihitung untuk menentukan apakah bilangan tersebut adalah bilangan prima. Sebuah bilangan dinyatakan prima jika hanya memiliki dua faktor (1 dan dirinya sendiri). Program ini menggabungkan proses pencarian faktor dan pemeriksaan keprimaan dalam satu alur perhitungan.