

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 10

ELSE-IF



Disusun oleh:

JEREMY CHRISTIAN SIHOMBING

109082500178

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.Print("Masukkan usia : ")

    fmt.Scan(&usia)

    fmt.Print("Apakah sudah punya kk (isi dengan
true/false) : ")

    fmt.Scan(&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 interface with a dark theme. On the left, the Explorer sidebar shows a project structure under 'MODUL 10' with files 'Guide1', 'Guide2', 'Guide3', and 'guide1.go'. The 'guide1.go' file is selected and shown in the main editor area. The code is a simple Go program that prompts the user for age and whether they have a KTP, then prints a message based on the input. Below the editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, displaying command-line output from running the program. To the right of the terminal is a small window titled 'NIM 10908' containing student information: NIM 109082500178, KELAS S1IF-13-07, and NAME JEREMY CHRISTIAN SIHOMBING.

```
package main
import "fmt"
func main() {
    var usia int
    var kk bool
    fmt.Println("Masukkan usia : ")
    fmt.Scan(&usia)
    fmt.Println("Apakah sudah punya kk (isi dengan true/false) : ")
    fmt.Scan(&kk)

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

```
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide1\guide1.go
Masukkan usia : 17
Apakah sudah punya kk (isi dengan true/false) : true
bisa membuat ktp
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide1\guide1.go
Masukkan usia : 20
Apakah sudah punya kk (isi dengan true/false) : false
belum bisa membuat ktp
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide1\guide1.go
Masukkan usia : 15
Apakah sudah punya kk (isi dengan true/false) : true
belum bisa membuat ktp
PS C:\Users\LENOVO\Downloads\Modul 10>
```

NIM 109082500178
KELAS S1IF-13-07
NAME JEREMY CHRISTIAN SIHOMBING

Deskripsi program

Program ini digunakan untuk menentukan apakah seseorang sudah memenuhi syarat untuk membuat KTP. Ketika program dijalankan, pengguna diminta memasukkan usia dan menjawab apakah ia sudah memiliki Kartu Keluarga (KK) dengan nilai boolean *true* atau *false*. Setelah kedua input diterima, program memeriksa dua kondisi sekaligus: apakah usia pengguna minimal 17 tahun dan apakah ia sudah memiliki KK. Jika kedua syarat tersebut terpenuhi, program menampilkan pesan "bisa membuat ktp". Namun, jika salah satu atau kedua kondisi tidak terpenuhi, program memberikan output "belum bisa membuat ktp". Dengan demikian, program memastikan kelayakan pembuatan KTP berdasarkan usia dan kepemilikan KK.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var huruf rune

    fmt.Print("Masukkan sebuah 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")

    }

}
```

Screenshoot program

The screenshot shows a code editor interface with several tabs open. The active tab contains Go code for determining if a character is a vowel or consonant. The terminal window below shows the execution of the program and its output for different inputs.

```
package main
import "fmt"
func main() {
    var huruf rune
    fmt.Print("Masukkan sebuah 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")
    }
}
```

TERMINAL

```
PS C:\Users\LENOVO\Downloads\Modul 10> go run ./Guide2\guide2.go
Masukkan sebuah huruf : A
Vokal
PS C:\Users\LENOVO\Downloads\Modul 10> go run ./Guide2\guide2.go
Masukkan sebuah huruf : f
Konsonan
PS C:\Users\LENOVO\Downloads\Modul 10> go run ./Guide2\guide2.go
Masukkan sebuah huruf : 1
Bukan huruf
PS C:\Users\LENOVO\Downloads\Modul 10> go run ./Guide2\guide2.go
Masukkan sebuah huruf : $
Bukan huruf
PS C:\Users\LENOVO\Downloads\Modul 10>
```

Deskripsi program

Program ini digunakan untuk menentukan apakah karakter yang dimasukkan pengguna merupakan huruf vokal, huruf konsonan, atau bukan huruf. Ketika program dijalankan, pengguna diminta memasukkan satu karakter. Nilai yang diterima disimpan dalam variabel bertipe `rune` agar dapat diproses sebagai karakter tunggal.

Setelah input diterima, program melakukan pengecekan pertama dengan membandingkan karakter tersebut dengan seluruh huruf vokal, baik huruf besar maupun huruf kecil (A, I, U, E, O, dan versi kecilnya). Jika karakter cocok dengan salah satu di antaranya, program menampilkan output "Vokal".

Jika bukan vokal, program melanjutkan pengecekan kedua untuk melihat apakah karakter tersebut masih termasuk dalam rentang huruf alfabet, baik huruf besar maupun kecil. Apabila karakter berada dalam rentang tersebut tetapi bukan vokal, maka program menyatakan karakter tersebut sebagai "Konsonan".

Jika kedua kondisi sebelumnya tidak terpenuhi, berarti karakter yang dimasukkan bukan huruf, sehingga program menampilkan pesan "Bukan huruf".

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var bilangan,d1, d2, d3, d4 int

    var urutan string

    fmt.Print("masukkan bilangan 4 digit :")

    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.Print("digit pada bilangan ", bilangan, " ", urutan)

}
```

Screenshot program

The screenshot shows a Go development environment with the following components:

- EXPLORER**: Shows a tree view of files under "MODUL 10". The file "guide3.go" is selected.
- Code Editor**: Displays the content of "guide3.go". The code reads a 4-digit number, extracts its digits, and prints them along with their order (terurut membesar, terurut mengecil, or tidak terurut).

```
package main
import "fmt"
func main(){
    var bilangan,d1, d2, d3, d4 int
    var urutan string

    fmt.Println("masukkan bilangan 4 digit :")
    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)
}
```
- TERMINAL**: Shows the command-line interface output for three test cases.

```
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide3\guide3.go
masukkan bilangan 4 digit :2489
digit pada bilangan 2489 terurut membesar
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide3\guide3.go
masukkan bilangan 4 digit :3861
digit pada bilangan 3861 tidak terurut
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Guide3\guide3.go
masukkan bilangan 4 digit :9651
digit pada bilangan 9651 terurut mengecil
PS C:\Users\LENOVO\Downloads\Modul 10>
```
- Status Bar**: Shows the file name "NIM 10908", line count "Ln 3, Col 32", character count "65", encoding "UTF-8", and zoom level "100%".

Deskripsi program

Program ini digunakan untuk mengevaluasi apakah susunan empat digit pada sebuah bilangan membentuk urutan yang membesar, mengecil, atau tidak terurut. Ketika program dijalankan, pengguna diminta memasukkan sebuah bilangan yang terdiri dari empat digit. Setelah input diperoleh, program memisahkan setiap digit dengan membagi dan mengambil sisa pembagian dari bilangan tersebut. Digit pertama diambil dengan membagi bilangan dengan 1000, digit kedua diperoleh dari sisa 1000 dibagi 100, digit ketiga diambil dari sisa 100 dibagi 10, dan digit keempat diperoleh dari sisa pembagian 10.

Setelah seluruh digit berhasil diambil, program membandingkan digit-digit tersebut. Jika setiap digit lebih kecil dari digit berikutnya ($d_1 < d_2 < d_3 < d_4$), program menentukan bahwa bilangan tersebut memiliki pola "terurut membesar". Jika setiap digit justru lebih besar dari digit berikutnya ($d_1 > d_2 > d_3 > d_4$), maka pola yang ditampilkan adalah "terurut mengecil". Namun, apabila tidak memenuhi kedua pola tersebut, program menyimpulkan bahwa digit-digit pada bilangan tersebut "tidak terurut". Program kemudian menampilkan hasil akhir berupa pola urutan digit dari bilangan yang dimasukkan.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var beratParsel, kg, gram, biayaKg, biayaSisa, total int

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

    fmt.Scan(&beratParsel)

    kg = beratParsel / 1000

    gram = beratParsel % 1000

    biayaKg = kg * 10000

    if kg >= 10 {

        biayaSisa = 0

    } else if gram >= 500 {

        biayaSisa = gram * 5

    } else {

        biayaSisa = gram * 15

    }

    total = biayaKg + biayaSisa

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

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

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

}
```

Screenshot program

The screenshot shows a Go development environment with two terminal panes. The left pane displays the source code for `soal1.go`:

```
Soal1 > ./soal1.go > main
1 package main
2 import "fmt"
3 func main() {
4     var beratParsel, kg, gram, biayaKg, biayaSisa, total int
5     fmt.Println("Berat parsel (gram): ")
6     fmt.Scan(&beratParsel)
7     kg = beratParsel / 1000
8     gram = beratParsel % 1000
9     biayaKg = kg * 1000
10    if kg >= 10 {
11        biayaSisa = 0
12    } else if gram >= 500 {
13        biayaSisa = gram * 5
14    } else {
15        biayaSisa = gram * 15
16    }
17    total = biayaKg + biayaSisa
18    fmt.Println("Detail berat:", kg, "kg +", gram, "gr")
19    fmt.Println("Detail biaya: Rp.", biayaKg, "+ Rp.", biayaSisa)
20    fmt.Println("Total biaya: Rp.", total)
21 }
```

The right pane shows the output of running the code:

```
NIM 109082500178
KELAS S1IF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING

Ln 3, Col 32 65 character Plain t 100% Wind UTF-8
```

Terminal output:

```
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal1\soal1.go
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal1\soal1.go
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal1\soal1.go
Berat parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
```

Deskripsi program

Program ini digunakan untuk menghitung biaya pengiriman parsel berdasarkan berat total dalam satuan gram. Ketika program dijalankan, pengguna diminta memasukkan berat parsel. Program kemudian memisahkan berat tersebut menjadi dua bagian, yaitu berat dalam kilogram dan sisa berat dalam gram. Nilai kilogram diperoleh dengan membagi total gram dengan 1000, sedangkan sisanya dihitung menggunakan operasi modulo.

Setelah berat terbagi, program menghitung biaya berdasarkan ketentuan tertentu. Biaya dasar dihitung dari jumlah kilogram dikalikan dengan tarif Rp10.000 per kilogram. Untuk sisa berat, program menerapkan aturan tambahan: jika berat kilogram mencapai 10 kg atau lebih, tidak ada biaya tambahan untuk sisa gram. Jika sisa gram kurang dari 500, biaya dihitung dengan tarif Rp15 per gram. Namun, jika sisa gram mencapai 500 gram atau lebih, tarifnya menjadi Rp5 per gram. Seluruh biaya ini kemudian dijumlahkan untuk memperoleh total biaya pengiriman.

Terakhir, program menampilkan rincian berat dalam bentuk kilogram dan gram, rincian biaya berdasarkan perhitungan tarif, serta total biaya keseluruhan yang harus dibayar oleh pengguna.

2. Tugas 2

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

Jika inputnya adalah 80.1, program tidak akan menghasilkan output apa pun karena akan muncul Error (Kegagalan Kompilasi). Alasannya, variabel nam dideklarasikan sebagai float64 (angka desimal), tetapi dalam blok if, kode berusaha memasukkan string ("A", "AB", dan seterusnya) ke dalam variabel tersebut. Di bahasa Go, ini tidak diperbolehkan (Ketidakcocokan Tipe). Kesesuaian Spesifikasinya, program tidak memenuhi spesifikasi karena adanya kesalahan dalam sintaks (aturan bahasa pemrograman) dan logika.

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

1.Kesalahan Tipe Data (Ketidakcocokan Tipe):

- Salah: nam = "A". Penjelasannya, nam merupakan tempat penyimpanan angka (float), sehingga tidak dapat diisi dengan huruf. Seharusnya, huruf grade disimpan di variabel nmk (yang bertipe string).

2.Kesalahan Logika Percabangan (If Mandiri):

- Salah: Memakai beberapa if yang terpisah tanpa else. Penjelasannya, dengan if terpisah, program akan mengevaluasi semua kondisi.

- Contoh skenario: Jika nilai 90.

- if nam > 80: Terpenuhi -> nmk diubah menjadi "A".

- if nam > 72.5: Terpenuhi juga (karena 90 > 72.5) -> nmk diganti lagi menjadi "AB".

- ... dan begitu seterusnya hingga kondisi terakhir.

- Dampaknya, nilai tinggi malah akan mendapat grade terendah yang masih memenuhi syarat (seperti "D").

3.Kesalahan Variabel Output:

- Salah: fmt.Println(..., nmk). Penjelasannya, Variabel nmk yang dicetak di akhir, padahal dalam logika if yang dimodifikasi adalah variabel nam. Akibatnya, nmk akan tetap kosong.

Alur yang Benar: Program seharusnya menerapkan struktur if - else if - else. Setelah satu kondisi terpenuhi (misalnya > 80), program harus menghentikan pemeriksaan kondisi

selanjutnya agar grade "A" tidak digantikan. Selain itu, hasil grade harus disimpan ke variabel nmk.

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

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 {

        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)

    }
}

```

Screenshot program

The screenshot shows a code editor interface with a dark theme. On the left, the Explorer sidebar shows a project structure under 'MODUL 10' with files like 'guide1.go', 'guide2.go', 'guide3.go', 'Soal1', 'Soal2', and 'Soal3'. The 'soal2.go' file is selected. The main editor area contains the following Go code:

```

1 package main
2 import "fmt"
3 func main() {
4     var nam float64
5     var nmk string
6     fmt.Println("Nilai akhir mata kuliah:")
7     fmt.Scan(&nam)
8     if nam > 80 {
9         nmk = "A"
10    } else if nam > 72.5 {
11        nmk = "AB"
12    } else if nam > 65 {
13        nmk = "B"
14    } else if nam > 57.5 {
15        nmk = "BC"
16    } else if nam > 50 {
17        nmk = "C"
18    } else if nam > 40 {
19        nmk = "D"
20    } else {
21        nmk = "E"
22    }
23    fmt.Println("Nilai mata kuliah:", nmk)
24 }

```

To the right of the editor is a terminal window titled 'NIM 10908' showing the output of running the program:

```

NIM 109082500178
KELAS S1IF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING

Ln 3, Col 32 | 65 character Plain t 100% Wind UTF-8

```

The terminal also displays the command-line output for three different scores:

```

PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal2\soal2.go
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal2\soal2.go
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal2\soal2.go
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\Users\LENOVO\Downloads\Modul 10>

```

Deskripsi program

Program ini digunakan untuk menentukan Nilai Mutu Kuliah (NMK) berdasarkan Nilai Akhir Mata Kuliah (NAM) yang dimasukkan oleh pengguna. Program menerima masukan berupa satu bilangan real (float64) yang mewakili nilai akhir mahasiswa.

Setelah nilai NAM dibaca, program membandingkan nilai tersebut dengan rentang standar penilaian. Pemeriksaan dilakukan menggunakan struktur if - else if - else, sehingga hanya satu kondisi yang akan dipilih sebagai hasil akhir. Berdasarkan rentang nilai tersebut, program akan menentukan huruf mutu yang sesuai, mulai dari A untuk nilai yang lebih dari 80, sampai E untuk nilai yang kurang atau sama dengan 40.

Pada akhir program, NMK yang telah ditentukan akan ditampilkan ke layar.

3. Tugas 3a

Source code

```
package main
import "fmt"
func main() {
    var bilangan int
    fmt.Print("bilangan : ")
    fmt.Scan(&bilangan)
    fmt.Print("Faktor : ")
    for i := 1; i <= bilangan; i++ {
        if bilangan%i == 0 {
            fmt.Print(i, " ")
        }
    }
}
```

Screenshot program

The screenshot shows a Go development environment with the following details:

- EXPLORER:** Shows a tree view of files under "MODUL 10". The "Soal3a" folder is selected.
- SOAL3A.GO:** The main file being edited contains the provided Go code.
- TERMINAL:** The terminal window shows the execution of the program and its output.
- Output:** The terminal output is:

```
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal3a\soal3a.go
bilangan : 12
Faktor : 1 2 3 4 6 12
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\Soal3a\soal3a.go
bilangan : 7
Faktor : 1 7
PS C:\Users\LENOVO\Downloads\Modul 10>
```

Deskripsi program

Program ini digunakan untuk menampilkan seluruh faktor dari sebuah bilangan bulat positif. Program diawali dengan meminta pengguna memasukkan satu bilangan bulat. Setelah bilangan tersebut dibaca, program menampilkan teks "Faktor : " sebagai penanda bahwa faktor-faktor bilangan akan ditampilkan.

Untuk menentukan faktor, program menggunakan sebuah perulangan for yang berjalan dari 1 hingga nilai bilangan tersebut. Pada setiap iterasi, program memeriksa apakah nilai pembagi i dapat membagi habis bilangan (dengan memeriksa apakah bilangan % i == 0). Jika kondisi tersebut benar, maka angka i adalah salah satu faktor dari bilangan, dan angka tersebut langsung dicetak ke layar.

Dengan cara ini, seluruh faktor dari bilangan akan ditampilkan secara berurutan mulai dari 1 sampai bilangan itu sendiri.

4. Tugas 3b

Source code

```
package main
import "fmt"
func main() {
    var bilangan int
    var prima bool
    var jmlFaktor int = 0
    fmt.Println("Bilangan : ")
    fmt.Scan(&bilangan)
    fmt.Println("Faktor : ")
    for i := 1; i <= bilangan; i++ {
        if bilangan%i == 0 {
            fmt.Println(i, " ")
            jmlFaktor++
        }
    }
    fmt.Println()
    if jmlFaktor == 2 {
        prima = true
    } else {
        prima = false
    }
    fmt.Println("Prima:", prima)
}
```

Screenshot program

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar showing project structure under 'MODUL 10'. The main area displays a Go file named 'soal3b.go' with the following code:

```
Soal3b > soal3b.go > main
1 package main
2 import "fmt"
3 func main() {
4     var bilangan int
5     var prima bool
6     var jmlFaktor int = 0
7     fmt.Println("Bilangan : ")
8     fmt.Scan(&bilangan)
9     fmt.Println("Faktor : ")
10    for i := 1; i <= bilangan; i++ {
11        if bilangan%i == 0 {
12            fmt.Print(i, " ")
13            jmlFaktor++
14        }
15    }
16    fmt.Println()
17    if jmlFaktor == 2 {
18        prima = true
19    } else {
20        prima = false
21    }
22    fmt.Println("Prima:", prima)
23 }
```

Below the code editor is a terminal window titled 'NIM 10908' showing the output of running the program:

```
NIM 10908
File Edit View Aa + X
NIM 109082500178
KELAS S1IF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING

Ln 3, Col 32 | 65 character Plain t 100% Wind UTF-8

PS C:\Users\LENOVO\Downloads\Modul 10> go run .\soal3b\soal3b.go
Bilangan : 12
Faktor : 1 2 3 4 6 12
Prima: false
PS C:\Users\LENOVO\Downloads\Modul 10> go run .\soal3b\soal3b.go
Bilangan : 7
Faktor : 1 7
Prima: true
PS C:\Users\LENOVO\Downloads\Modul 10>
```

At the bottom left of the code editor are 'OUTLINE' and 'TIMELINE' buttons.

Deskripsi program

Program ini digunakan untuk menentukan faktor-faktor dari sebuah bilangan serta mengecek apakah bilangan tersebut merupakan bilangan prima atau bukan. Pertama, program meminta pengguna memasukkan sebuah bilangan bulat, kemudian menampilkan teks "Faktor :" sebagai penanda bahwa faktor-faktor bilangan akan ditampilkan.

Program menggunakan sebuah perulangan for yang berjalan dari 1 hingga nilai bilangan. Pada setiap nilai i, program memeriksa apakah bilangan % i == 0. Jika kondisi tersebut benar, berarti i adalah salah satu faktor dari bilangan tersebut. Angka ini langsung dicetak, dan variabel penghitung faktor (jmlFaktor) ditambah satu.

Setelah seluruh faktor ditampilkan, program memeriksa apakah jumlah faktor tepat berjumlah 2. Bilangan yang memiliki tepat dua faktor (yaitu 1 dan dirinya sendiri) dikategorikan sebagai bilangan prima, sehingga variabel prima diisi dengan nilai true. Jika tidak, maka prima bernilai false.

Terakhir, program menampilkan hasil pengecekan tersebut melalui tulisan "Prima: true" atau "Prima: false"

.