

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 10

ELSE - IF



Disusun oleh:

BENING PUTRI NARESWARI SUKARNO

109082500211

S1IF-13-02

Asisten Praktikum

Adithana Dharma Putra

Alfin Ilham Berlianto

**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")

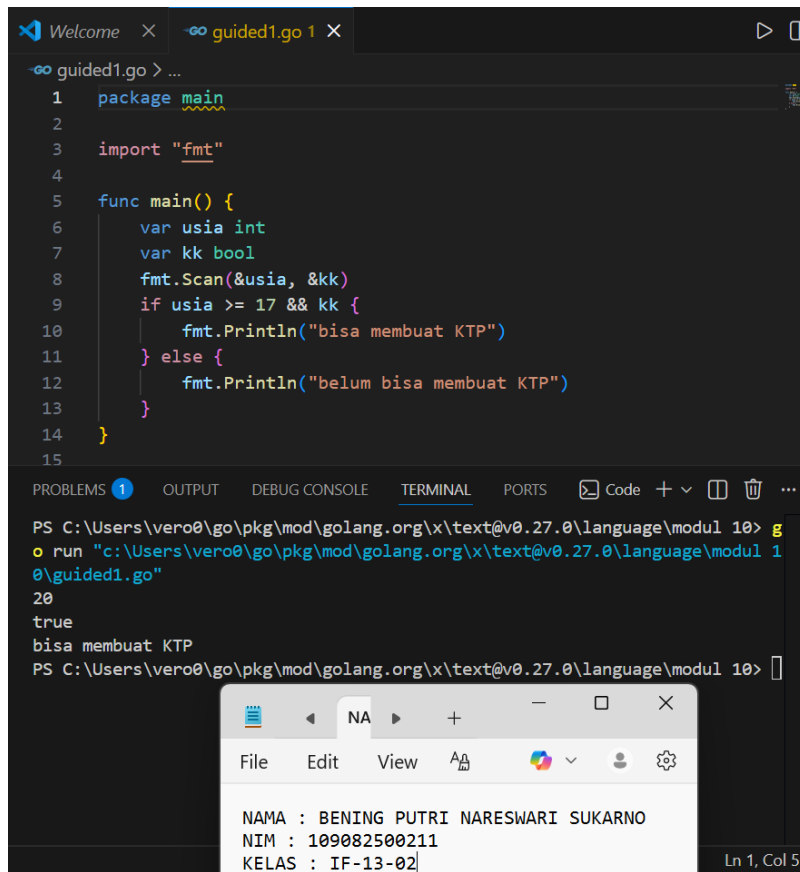
    } else {

        fmt.Println("belum bisa membuat KTP")

    }

}
```

Screenshoot program



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

```
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10> go run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10\guided1.go"
20
true
bisa membuat KTP
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10>
```

Deskripsi program

Program ini berfungsi untuk menentukan apakah seseorang memenuhi syarat untuk membuat KTP.

Cara Kerja:

1. Program meminta pengguna untuk memasukkan dua nilai secara berurutan:
2. usia (umur), yang harus berupa angka bulat (integer). KK (kepemilikan Kartu Keluarga), yang harus berupa nilai boolean (true atau false).
3. Program akan mencetak "bisa membuat KTP" hanya jika kedua kondisi ini terpenuhi:
 - usia lebih besar atau sama dengan 17 (usia ≥ 17).
 - kk bernilai benar (true).
4. Jika salah satu atau kedua kondisi tidak terpenuhi, program akan mencetak "belum bisa membuat KTP".

Contoh Masukkan:

1. Pengguna memasukkan nilai:
 - 20 (untuk variabel usia)
 - true (untuk variabel kk)
2. Logika program memeriksa:
 - Apakah $20 \geq 17$? (Ya, benar)
 - Apakah kk bernilai true? (Ya, benar)
 - Karena kedua kondisi benar, program menjalankan kode di dalam blok if.
3. Output: Program mencetak hasil akhir "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")

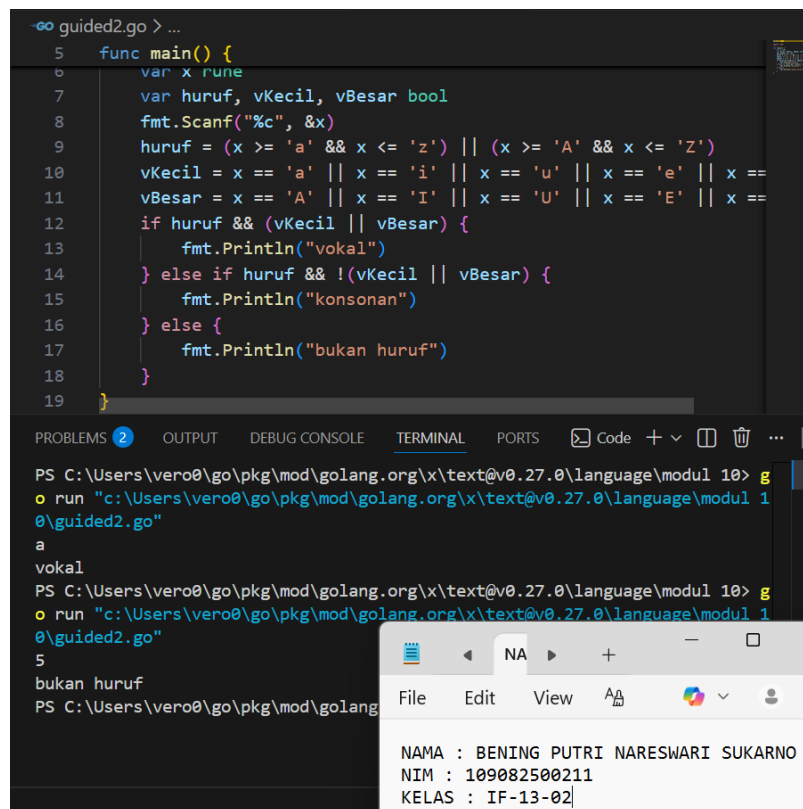
    } else {

        fmt.Println("bukan huruf")

    }

}
```

Screenshoot program



```
guided2.go > ...
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' && x <= 'Z')
10    vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x ==
11    vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x ==
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 }
```

PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10> g
o run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 1
0\guided2.go"
a
vokal
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10> g
o run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 1
0\guided2.go"
5
bukan huruf
PS C:\Users\vero0\go\pkg\mod\golang

NAMA : BENING PUTRI NARESWARI SUKARNO
NIM : 109082500211
KELAS : IF-13-02

Deskripsi program

Program berfungsi untuk mengidentifikasi jenis karakter tunggal yang dimasukkan pengguna. Program ini akan menentukan apakah karakter tersebut adalah huruf vokal, huruf konsonan, atau bukan huruf.

Cara Kerja:

1. Program meminta pengguna untuk memasukkan satu karakter (%c) yang disimpan dalam variabel x.
 - huruf: Bernilai true jika x adalah huruf alfabet (a-z atau A-Z).
 - vKecil: Bernilai true jika x adalah huruf vokal kecil (a, i, u, e, o).
 - vBesar: Bernilai true jika x adalah huruf vokal besar (A, I, U, E, O).
2. Kondisi
 - Jika karakternya adalah huruf DAN (vokal kecil ATAU vokal besar), cetak "vokal".
 - Jika karakternya adalah huruf DAN bukan vokal, cetak "konsonan".
 - Jika kedua kondisi di atas tidak terpenuhi (artinya x bukan huruf sama sekali), cetak "bukan huruf".

Contoh Masukkan:

1. Pengguna memasukkan karakter: a
 - huruf = true (karena 'a' adalah huruf).
 - vKecil = true (karena 'a' adalah vokal kecil).
2. Kondisi Vokal (if) terpenuhi: true && (true || false) adalah true.
3. Output: Program mencetak: vocal.

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 {

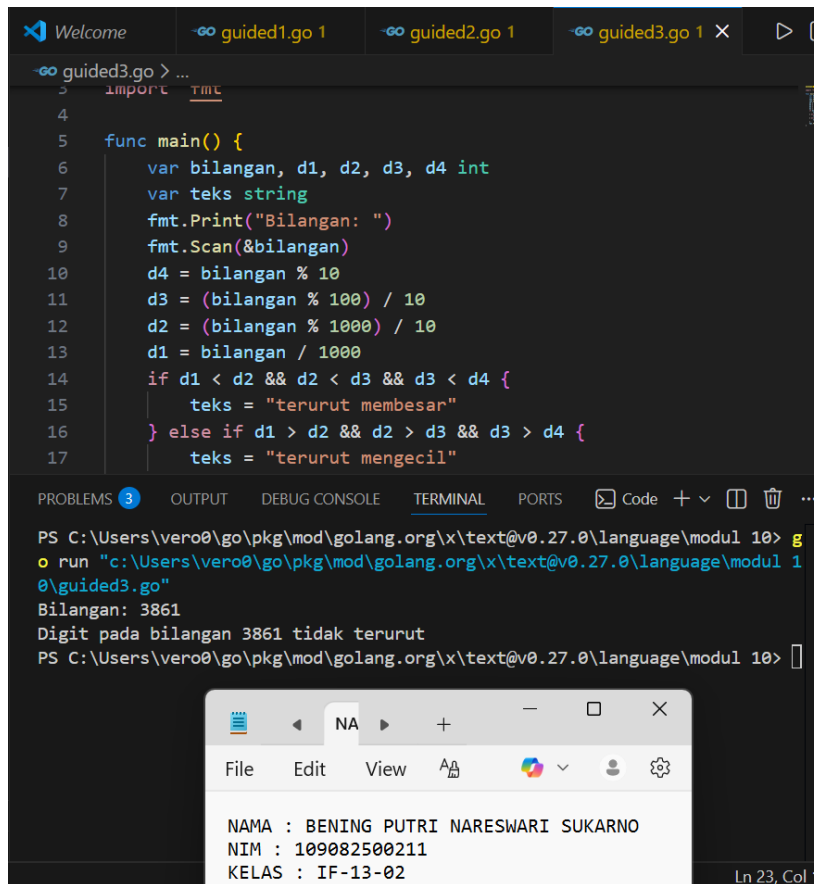
        teks = "tidak terurut"

    }

    fmt.Println("Digit pada bilangan", bilangan, teks)

}
```

Screenshoot program



```
1 import "fmt"
2
3 func main() {
4     var bilangan, d1, d2, d3, d4 int
5     var teks string
6     fmt.Print("Bilangan: ")
7     fmt.Scan(&bilangan)
8     d4 = bilangan % 10
9     d3 = (bilangan % 100) / 10
10    d2 = (bilangan % 1000) / 100
11    d1 = bilangan / 1000
12    if d1 < d2 && d2 < d3 && d3 < d4 {
13        teks = "terurut membesar"
14    } else if d1 > d2 && d2 > d3 && d3 > d4 {
15        teks = "terurut mengecil"
16    } else {
17        teks = "tidak teratur"
18    }
19    fmt.Println(teks)
20 }
```

PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10> go run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10\guided3.go"

Bilangan: 3861
Digit pada bilangan 3861 tidak teratur
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10>

Deskripsi program

Program ini berfungsi untuk memeriksa urutan digit dari sebuah bilangan bulat empat digit yang dimasukkan oleh pengguna, untuk menentukan apakah digit-digit tersebut tersusun secara berurutan menaik, berurutan menurun, atau tidak berurutan.

Cara Kerja:

1. Program meminta pengguna untuk memasukkan suatu bilangan bulat .
2. Program menggunakan operasi matematika (sisa bagi % dan pembagian /) untuk memisahkan bilangan empat digit menjadi digit-digit penyusunnya, dari kiri ke kanan:
d1: Digit pertama (ribuan).
d2: Digit kedua (ratusan).
d3: Digit ketiga (puluhan).
d4: Digit keempat (satuan).
3. Kondisi:
Terurut Membesar: Jika $d1 < d2 < d3 < d4$ (setiap digit lebih kecil dari digit berikutnya).
Terurut Mengecil: Jika $d1 > d2 > d3 > d4$ (setiap digit lebih besar dari digit berikutnya).
Tidak Terurut: Jika tidak memenuhi kedua kondisi di atas.
4. Program akan mencetak status urutan digit tersebut (teks) bersama dengan bilangan aslinya.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var beratGram int

    fmt.Scan(&beratGram)

    beratKg := beratGram / 1000

    sisaGram := beratGram % 1000

    biayaDasar := beratKg * 10000

    biayaSisa := 0

    if beratKg <= 10 {

        if sisaGram >= 500 {

            biayaSisa = sisaGram * 5

        } else {

            biayaSisa = sisaGram * 15

        }

    }

    totalBiaya := biayaDasar + biayaSisa
```



```

    fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg,
sisagram)

    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaDasar,
biayaSisa)

    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}

```

Screenshoot program

```

soall1.go > main
4
5 func main() {
6     var beratGram int
7     fmt.Scan(&beratGram)
8
9     beratKg := beratGram / 1000
10    sisaGram := beratGram % 1000
11
12    biayaDasar := beratKg * 10000
13    biayaSisa := 0
14
15    if beratKg <= 10 {
16        if sisaGram >= 500 {
17            biayaSisa = sisaGram * 5
18        } else {

```

```

PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10> g
o run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 1
0\soall1.go"
8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\vero0\go\pkg\mod\gola

```

Deskripsi program

Program ini berfungsi untuk menghitung total biaya pengiriman parcel berdasarkan berat satuan gram.

Cara Kerja:

1. Program meminta pengguna memasukkan berat parcel dalam gram.
2. Program membagi input berat menjadi dua komponen: Kilogram (beratKg) dan Sisa Gram (sisaGram) menggunakan operasi pembagian dan sisa bagi.
3. Perhitungan Biaya:
 - Biaya Dasar: Dihitung dari berat Kilogram (beratKg) dikalikan tarif tetap Rp 10.000 per kg.
 - Biaya Sisa: Dikenakan pada sisa gram, kecuali jika total berat sudah lebih dari 10 kg (beratKg > 10), maka sisa gram digratiskan (biaya Rp 0).

Tarif Sisa Gram: Jika berat ≤ 10 kg, tarif sisa gram ditentukan:

- Rp 5,- per gram, jika sisa ≥ 500 gram.
 - Rp 15,- per gram, jika sisa < 500 gram.
4. Program menampilkan rincian berat (kg dan gram), rincian biaya (dasar + sisa), dan Total Biaya keseluruhan.

Contoh Masukkan:

1. Pengguna memasukkan berat: 8500 gram.
2. Proses Perhitungan:
beratKg = $8500/1000 = 8$ kg
sisaGram = $8500 \pmod{1000} = 500$ gram
biayaDasar = $8 \times 10000 = 80000$
3. Program memeriksa: beratKg ≤ 10 (yaitu $8 \leq 10$), maka tarif sisa berlaku.
Sisa gram adalah 500. Karena $500 \geq 500$, tarif Rp 5,-per gram digunakan.
biayaSisa = $500 \times 5 = 2500$
totalBiaya = $80000 + 2500 = 82500$
4. Program mencetak rincian dan total biaya:
Detail berat: 8 kg 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500

2. Tugas 2

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

Jawaban:

Jika NAM diberikan adalah 80.1, keluaran (output) dari program yang tertera di soal tersebut adalah "D". Terjadi karena program menggunakan serangkaian perintah if yang saling menimpa (bukan if-else).

Saat NAM = 80.1:

- Kondisi if nam > 80 adalah True, sehingga nmk diatur menjadi "A".
- Kondisi if nam > 72.5 adalah True, sehingga nmk ditimpa menjadi "AB". ...dan seterusnya, hingga:
- Kondisi if nam > 40 adalah True, sehingga nmk ditimpa dan berakhir pada nilai "D".

Jadi, eksekusi **program tidak sesuai** dengan spesifikasi soal.

Nilai 80.1 seharusnya masuk dalam rentang $72.5 < \text{NAM} \leq 80$ atau $\text{NAM} > 80$.

Jika NAM = 80.1, nilai yang seharusnya diperoleh adalah "A" (karena $80.1 > 80$).

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

Jawaban:

Kesalahan program adalah menggunakan struktur if yang saling menimpa (bukan if-else if-else) untuk menentukan rentang nilai. Saat sebuah nilai dimasukkan (misalnya 80.1), semua kondisi if yang bernilai true (> 72.5 , > 65 , dll.) akan dieksekusi secara berurutan. Ini menyebabkan variabel nilai mutu (nmk) selalu ditimpa dengan nilai yang lebih rendah, hingga mencapai kondisi terakhir yang benar ($\text{nam} > 40$), sehingga hasilnya menjadi "D" atau "E".

Alur seharusnya menggunakan struktur if-else if-else dan dicek dari nilai tertinggi (A) ke terendah (E). Ini memastikan bahwa hanya satu kondisi yang terpenuhi, dan program akan langsung berhenti setelah menemukan nilai yang benar.

- 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.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 {

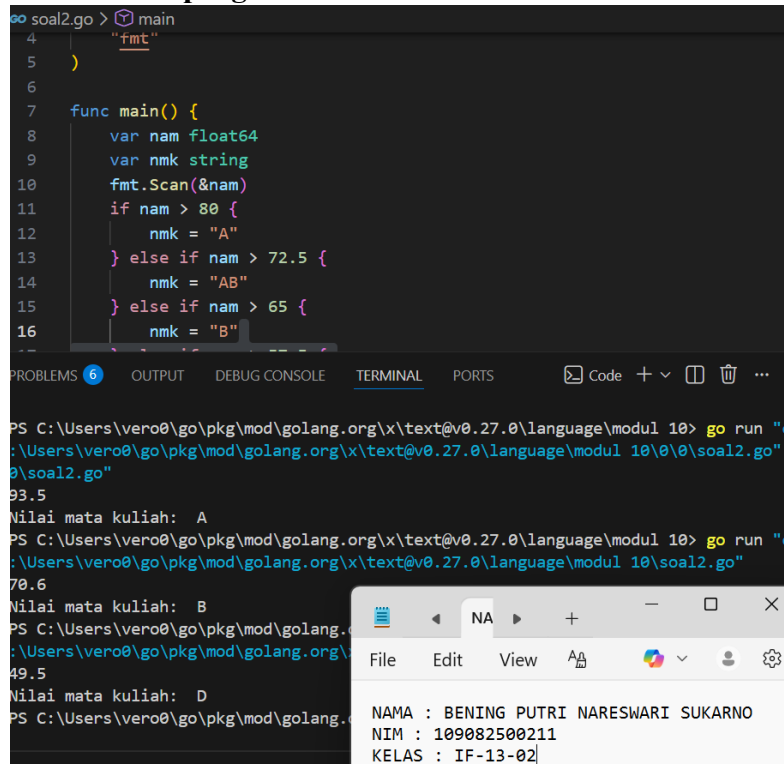
        nmk = "E"

    }

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

}
```

Screenshoot program



```
soal2.go > main
4   "fmt"
5   )
6
7   func main() {
8       var nam float64
9       var nmk string
10      fmt.Scan(&nam)
11      if nam > 80 {
12          nmk = "A"
13      } else if nam > 72.5 {
14          nmk = "AB"
15      } else if nam > 65 {
16          nmk = "B"
17      }
18      fmt.Println(nmk)
19  }
```

PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10> go run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10\soal2.go"

93.5
Nilai mata kuliah: A

PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10> go run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10\soal2.go"

70.6
Nilai mata kuliah: B

PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10> go run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10\soal2.go"

49.5
Nilai mata kuliah: D

PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10> go run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10\soal2.go"

NAMA : BENING PUTRI NARESWARI SUKARNO
NIM : 109082500211
KELAS : IF-13-02

Deskripsi program

Program berfungsi untuk mengkonversi (mengubah) nilai akhir mata kuliah (NAM) yang dimasukkan pengguna menjadi nilai mutu kuliah (NMK) berupa huruf (A, AB, B, BC, C, D, dan E).

Cara Kerja:

1. Program menerima satu bilangan riil sebagai nilai akhir mata kuliah (nam).
2. Program menggunakan struktur if-else
Pengecekan dari kriteria nilai tertinggi (A) ke kriteria nilai terendah (E).
Apabila satu kondisi ($\text{nam} > \text{batas_nilai}$) terpenuhi, program langsung menetapkan nilai mutu (nmk) yang sesuai dan berhenti mengecek kondisi di bawahnya.
3. Program menampilkan hasil nilai mutu kuliah (NMK) berupa huruf.

Contoh Masukkan:

1. Input: 93.5
Hasil: nmk = A (Karena $93.5 > 80$)
 2. Input: 70.6
Hasil: nmk B (Karena $70.6 > 65$)
 3. Input: 49.5
Hasil: nmk = D (Karena $49.5 > 40$)
- Jadi, perbaikan program ini berhasil.

3. Tugas 3

Source code

```
package main

import (
    "fmt"
)

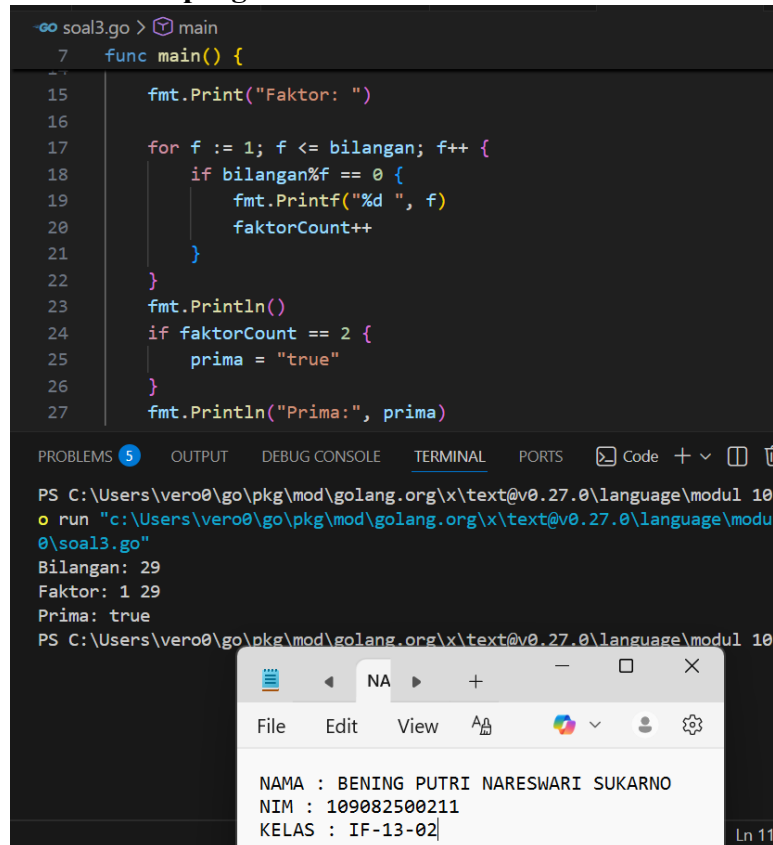
func main() {
    var bilangan int
    fmt.Print("Bilangan: ")
    fmt.Scan(&bilangan)

    faktorCount := 0
    prima := "false"

    fmt.Print("Faktor: ")

    for f := 1; f <= bilangan; f++ {
        if bilangan%f == 0 {
            fmt.Printf("%d ", f)
            faktorCount++
        }
    }
    fmt.Println()
    if faktorCount == 2 {
        prima = "true"
    }
    fmt.Println("Prima:", prima)
}
```

Screenshoot program



```
soal3.go > main
7 func main() {
15     fmt.Print("Faktor: ")
16
17     for f := 1; f <= bilangan; f++ {
18         if bilangan%f == 0 {
19             fmt.Printf("%d ", f)
20             faktorCount++
21         }
22     }
23     fmt.Println()
24     if faktorCount == 2 {
25         prima = "true"
26     }
27     fmt.Println("Prima:", prima)
}

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + v
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10
o run "c:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul
0\soal3.go"
Bilangan: 29
Faktor: 1 29
Prima: true
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\language\modul 10
```

NAMA : BENING PUTRI NARESWARI SUKARNO
NIM : 109082500211
KELAS : IF-13-02

Deskripsi program

Program berfungsi mencari dan menampilkan semua faktor dari bilangan bulat yang dimasukkan pengguna, dan kemudian menentukan apakah bilangan tersebut termasuk bilangan prima atau bukan.

Cara Kerja:

1. Program meminta pengguna memasukkan satu bilangan bulat
2. Mencari Faktor: Program mencoba membagi angka Anda dengan setiap angka dari 1 sampai angka itu sendiri. Setiap angka yang habis membagi (sisa baginya nol) akan dicetak sebagai faktor.
3. Menentukan Prima: Program menghitung berapa banyak faktor yang ia temukan. Jika jumlah faktornya tepat dua (yaitu hanya 1 dan angka itu sendiri), program menyatakan angka itu Bilangan Prima (true).

Contoh Masukkan:

1. Pengguna memasukkan: 29
2. Proses Perhitungan Faktor:
 - Program menguji angka 1 sampai 29.
 - Hanya angka 1 dan 29 yang habis membagi 29.
 - faktor Count bernilai 2.
3. Proses Penentuan Prima:
 - Program memeriksa: faktorCount == 2 (yaitu 2 == 2). (Benar)

4. Keluaran (Output): Program mencetak hasil:

- Bilangan: 29
- Faktor: 1 29
- Prima: true