

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 10

Else-If



Disusun oleh:

DADI MAULANA MUHAMMAD

109082500080

S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

LATIHAN KELAS – GUIDED

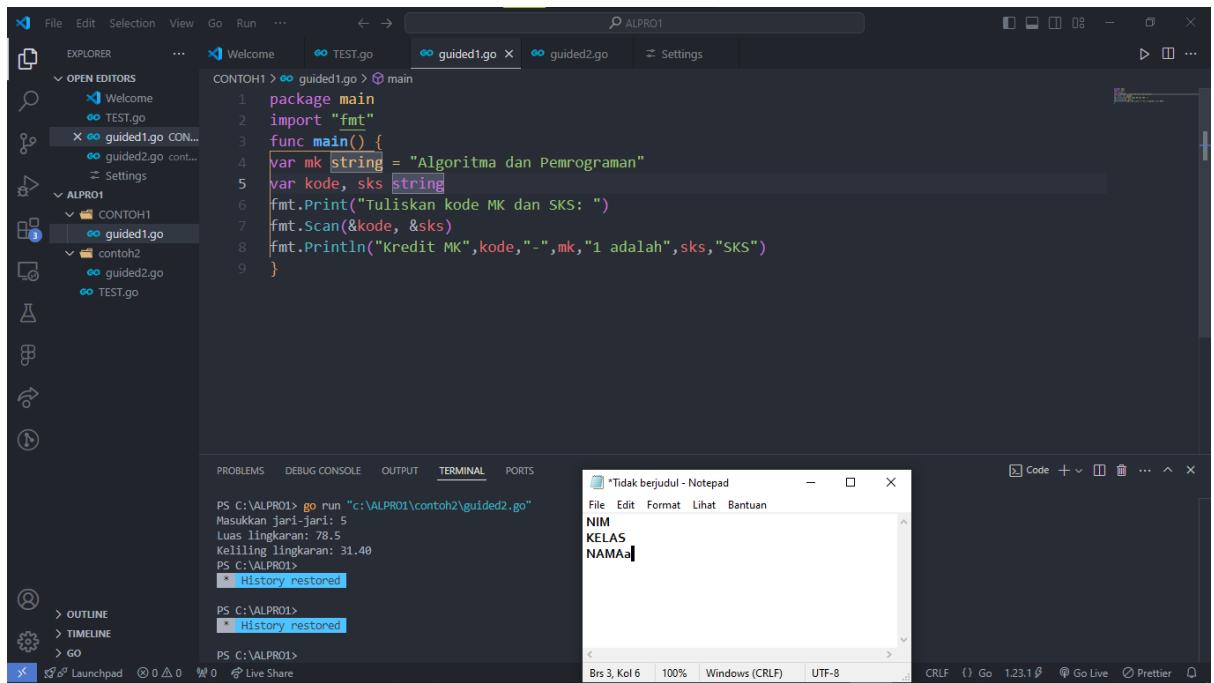
1. Guided 1

Source Code

```
//SALIN KODE KESINI  
  
ATURAN:  
  
PENULISAN SEESUAI MODUL 1  
  
GUNAKAN FONT Courier New ukuran 11pt dengan spasi baris  
dan paragraf 1,5
```

Screenshot program

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)
CONTOH TANGKAPAN LAYAR:



The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar showing a project structure with files like 'Welcome', 'TEST.go', 'guided1.go', 'guided2.go', and 'contoh2'. The main editor area displays the following Go code:

```
package main
import "fmt"
func main() {
    var mk string = "Algoritma dan Pemrograman"
    var kode, sks string
    fmt.Println("Tuliskan kode MK dan SKS: ")
    fmt.Scan(&kode, &sks)
    fmt.Println("Kredit MK", kode, "-", mk, "1 adalah", sks, "SKS")
}
```

Below the editor is a terminal window titled 'ALPRO1' showing the output of a command:

```
PS C:\ALPRO1> go run "c:\ALPRO1\contoh2\guided2.go"
Masukkan jari-jari: 5
Luas lingkaran: 78.5
Keiling lingkaran: 31.40
PS C:\ALPRO1>
* History restored

PS C:\ALPRO1>
* History restored
PS C:\ALPRO1>
```

To the right of the terminal is a small Notepad window titled 'Tidak berjudul - Notepad' containing the text:

```
NIM  
KELAS  
NAMAa|
```

Deskripsi program

Jelaskan kode yang ada di source code, semakin detil semakin baik nilainya

2. Guided 2

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 window with the file `guided-01.go` open. The code is identical to the one above. Below the editor is a terminal window showing the execution of the program. The terminal output is as follows:

```
F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-01.go"
17
true
bisa membuat KTP

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-01.go"
20
false
belum bisa membuat KTP
```

To the right of the terminal, there is a small window displaying student information:

Nama : Dadi Maulana Muhammad
NIM : 109082500080
Kelas : S1IF-13-04

Deskripsi program

Program ini membaca dua input dari pengguna, yaitu sebuah bilangan bulat **usia** dan sebuah nilai boolean **kk**. Variabel **usia** merepresentasikan umur seseorang, sedangkan **kk** menunjukkan apakah orang tersebut memiliki Kartu Keluarga (KK) atau tidak.

Setelah kedua nilai dibaca, program melakukan pemeriksaan menggunakan kondisi logika. Pertama, program mengecek apakah **usia ≥ 17** , kemudian memeriksa apakah nilai **kk** bernilai *true*. Kedua syarat ini digabungkan dengan operator logika **&&**, sehingga keduanya harus terpenuhi untuk menghasilkan keputusan positif.

Jika kedua kondisi benar, program menampilkan pesan "*bisa membuat KTP*". Sebaliknya, jika salah satu atau kedua kondisi tidak terpenuhi, program mencetak pesan "*belum bisa membuat KTP*" sebagai hasil evaluasi.

Dengan demikian, program ini menentukan apakah seseorang memenuhi syarat pembuatan KTP berdasarkan kombinasi usia minimal dan kepemilikan Kartu Keluarga.

3. Guided 3

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 code editor with a Go file named `guided-02.go`. The code reads a character from standard input and prints whether it is a vowel or consonant. The interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The DEBUG CONSOLE tab is active, displaying the output of running the program with the letter 'k' as input. The terminal output shows the character 'k' is a consonant.

```
guided-02.go
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")
    }
}

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS

konsonan

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-d-02.go"
1
konsonan

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-d-02.go"
$
bukan huruf

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>
```

Deskripsi program

Program ini membaca satu karakter dari input, yaitu **x**, yang disimpan dalam variabel bertipe rune. Program kemudian menentukan apakah karakter tersebut merupakan huruf, apakah huruf tersebut termasuk vokal kecil, atau vokal besar.

Pertama, program mengevaluasi variabel **huruf**, yang bernilai *true* jika **x** berada dalam rentang 'a' sampai 'z' atau 'A' sampai 'Z'. Setelah itu, program memeriksa apakah karakter merupakan huruf vokal kecil dengan membandingkannya terhadap 'a', 'i', 'u', 'e', dan 'o', dan menyimpan hasilnya dalam variabel **vKecil**. Pemeriksaan serupa dilakukan untuk huruf vokal besar, yang disimpan pada variabel **vBesar**.

Setelah nilai ketiga variabel boolean tersebut ditetapkan, program melakukan serangkaian pengecekan. Jika **huruf** bernilai *true* dan karakter tersebut termasuk salah satu vokal (baik kecil maupun besar), program mencetak "vokal". Jika karakter merupakan huruf tetapi bukan vokal, program mencetak "konsonan". Namun, jika karakter tidak memenuhi kriteria huruf sama sekali, program mencetak "bukan huruf".

Dengan demikian, program ini mengidentifikasi jenis karakter yang dimasukkan pengguna, yaitu apakah termasuk huruf vokal, huruf konsonan, atau bukan huruf sama sekali.

4. Guided 4

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 code editor window with a Go file named `guided-03.go`. The code reads a number from the user, extracts its digits, and determines if they are in ascending, descending, or random order. It then prints the original number and the classification. Below the editor is a terminal window showing the execution of the program and its output for three different input numbers: 2489, 3861, and 9651. To the right of the terminal is a small window displaying student information: Name (Dadi Maulana Muhammad), NIM (109082500080), and Class (S1IF-13-04). The bottom status bar of the terminal window shows the current line (Ln 12, Col 30), tab size (Tab Size: 2), and other terminal settings.

```
func main() {
    bilangan := 0
    d1 := 0
    d2 := 0
    d3 := 0
    d4 := 0
    teks := ""

    fmt.Println("Bilangan: ")
    fmt.Scan(&bilangan)
    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)
}
```

```
Digit pada bilangan 2489 tidak terurut
F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10\guided-03.go"
Bilangan: 3861
Digit pada bilangan 3861 tidak terurut
F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10\guided-03.go"
Bilangan: 9651
Digit pada bilangan 9651 tidak terurut
F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>
```

Nama : Dadi Maulana Muhammad
NIM : 109082500080
Kelas : S1IF-13-04

Deskripsi program

Program ini membaca satu bilangan bulat positif, yaitu **bilangan**, lalu memisahkan bilangan tersebut menjadi empat digit: ribuan (**d1**), ratusan (**d2**), puluhan (**d3**), dan satuan (**d4**). Setiap digit dihitung menggunakan operasi pembagian dan modulus. Setelah digit-digit tersebut diperoleh, program menentukan apakah urutan digit dalam bilangan tersebut membentuk pola tertentu. Pertama, program memeriksa apakah digit-digit tersusun secara menaik melalui kondisi **d1 < d2 && d2 < d3 && d3 < d4**. Jika kondisi terpenuhi, variabel **teks** diisi dengan nilai "terurut membesar". Jika kondisi pertama tidak terpenuhi, program kemudian mengecek apakah digit-digit tersebut tersusun secara menurun dengan kondisi **d1 > d2 && d2 > d3 && d3 > d4**. Jika benar, variabel **teks** diberi nilai "terurut mengecil". Apabila kedua kondisi sebelumnya tidak terpenuhi, program menyimpulkan bahwa bilangan tersebut tidak memiliki pola urutan digit tertentu, sehingga **teks** berisi "tidak terurut". Terakhir, program mencetak hasil berupa teks yang menjelaskan apakah digit-digit pada bilangan tersebut tersusun membesar, mengecil, atau tidak terurut sama sekali.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var gram int
    fmt.Print("Masukkan berat parsel (gram): ")
    fmt.Scan(&gram)

    kg := gram / 1000
    sisa := gram % 1000

    biayaKg := kg * 10000
    biayaSisa := 0

    if kg >= 10 {
        biayaSisa = 0
    } else {
        if sisa >= 500 {
            biayaSisa = sisa * 5
        } else {
            biayaSisa = sisa * 15
        }
    }

    total := biayaKg + biayaSisa

    fmt.Println("\n--- Detail Biaya ---")
    fmt.Printf("Berat: %d kg + %d gr\n", kg, sisa)
```

```

        fmt.Printf("Biaya kg: Rp. %d + Rp. %d\n", biayaKg,
biayaSisa)

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

}

```

Screenshot program

The screenshot shows a code editor window for assignment-01.go. The code defines a main function that calculates shipping costs based on weight in grams. It uses integer division and modulus to separate kilograms and grams, then applies different rates for each. The terminal below shows two runs of the program: one for 8 kg (outputting 82500) and one for 9 kg (outputting 93750). The right side of the screen shows a terminal window with student information.

```

10     kg := gram / 1000
11     sisa := gram % 1000
12
13     biayaKg := kg * 10000
14     biayaSisa := 0
15
16     if kg >= 10 {
17         biayaSisa = 0
18     } else {
19         if sisa >= 500 {
20             biayaSisa = sisa * 5
21         } else {
22             biayaSisa = sisa * 15
23         }
24     }
25
26     total := biayaKg + biayaSisa

```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

--- Detail Biaya ---
Berat: 8 kg + 500 gr
Biaya kg: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-dent-01.go"
Masukkan berat parsel (gram): 9250

--- Detail Biaya ---
Berat: 9 kg + 250 gr
Biaya kg: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750

```

Ln 24, Col 4 Tab Size: 2 Ln 3, Col 19 69 character Plain text 130% Wind UTF-8

File Edit View Aa ⚡ 🌐 🌐 🌐 🌐 🌐

Nama : Dadi Maulana Muhammad
NIM : 109082500080
Kelas : S1IF-13-04

Deskripsi program

Program ini membaca satu bilangan bulat positif, yaitu **gram**, yang merepresentasikan berat sebuah parsel dalam satuan gram. Setelah input diterima, program menghitung berapa kilogram utuh dan berapa gram sisa dari berat tersebut. Nilai **kg** diperoleh dengan membagi gram dengan 1000, sedangkan **sisa** dihitung menggunakan operasi modulus.

Program kemudian menentukan biaya pengiriman berdasarkan dua komponen, yaitu **biayaKg** dan **biayaSisa**. Biaya untuk kilogram dihitung dengan mengalikan jumlah kilogram dengan 10.000. Variabel **biayaSisa** diinisialisasi dengan nilai 0.

Selanjutnya, program memeriksa apakah berat parsel mencapai 10 kg atau lebih. Jika **kg ≥ 10**, maka biaya sisa gram tidak dikenakan, sehingga **biayaSisa** tetap bernilai 0. Namun, jika berat kurang dari 10 kg, program menghitung biaya berdasarkan jumlah gram sisa. Jika **sisa ≥ 500**, biaya dihitung sebagai **sisa × 5**; jika kurang dari 500, biaya dihitung sebagai **sisa × 15**.

Setelah seluruh perhitungan selesai, program menjumlahkan **biayaKg** dan **biayaSisa** untuk mendapatkan **total** biaya pengiriman. Program kemudian mencetak rincian lengkap berupa berat dalam kg dan gram, biaya masing-masing komponen, serta total biaya yang harus dibayar pengguna.

Dengan demikian, program ini menentukan biaya pengiriman parsel berdasarkan berat dalam kilogram dan sisa gram, dengan aturan tarif yang berbeda tergantung batas 10 kg dan 500 gram.

2. Tugas 2

Source code

```
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 window for a Go program named `assignment-02-false.go`. The code defines a package `main` with a function `main()` that reads a float64 value from the user and prints it as a grade letter (A, AB, B, or BC) based on the value. The terminal window below shows the command `go run` being executed, followed by the output of the program which correctly prints the student's information.

```
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         nam = "A"
10    }
11    if nam > 72.5 {
12        nam = "AB"
13    }
14    if nam > 65 {
15        nam = "B"
16    }
17    if nam > 57.5 {
18        nam = "BC"
}
sion 10.0.2.22631.6060]
(c) Microsoft Corporation. All rights reserved.

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-d
ent-02-false.go"
assignment-02-false.go:2:8: curly quotation mark `'' (use neutral '')'
F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>
```

Ln 1, Col 1 Spaces: 4 Ln 3, Col 19 69 caracte Plain t 130% Wind UTF-8

Deskripsi program

Program ini membaca satu nilai desimal, yaitu `nam`, yang merepresentasikan nilai akhir sebuah mata kuliah. Program juga menyiapkan variabel `nmk` untuk menyimpan nilai huruf, meskipun variabel tersebut tidak pernah diisi akibat kesalahan logika dalam program.

Setelah membaca input, program menggunakan beberapa pernyataan `if` terpisah untuk menentukan nilai huruf, namun seluruh kondisi membandingkan variabel `nam` dan mencoba mengisi variabel `nam` dengan string seperti "A" atau "B". Hal ini menimbulkan kesalahan karena `nam` dideklarasikan sebagai bilangan bertipe `float64` tetapi kemudian diberi nilai berupa teks.

Selain itu, karena semua pengecekan menggunakan `if` tanpa `else if`, setiap kondisi dievaluasi secara terpisah. Akibatnya, kondisi yang lebih rendah dapat menimpa kondisi sebelumnya. Misalnya, jika nilai awal lebih besar dari 80, semua kondisi lain yang bernilai true juga akan dijalankan.

Pada bagian akhir, terdapat percabangan `else if` yang hanya terkait dengan kondisi `if nam > 40`, sehingga hanya berfungsi sebagai pasangan dari kondisi tersebut, bukan dari seluruh rangkaian pengecekan sebelumnya.

Setelah seluruh proses, program mencoba mencetak teks "*Nilai mata kuliah:*" diikuti variabel `nmk`. Namun karena variabel `nmk` tidak pernah diubah sepanjang eksekusi, nilai yang tercetak tidak sesuai dengan nilai huruf yang seharusnya dihasilkan.

Dengan demikian, program ini mencoba mengonversi nilai angka menjadi nilai huruf, tetapi mengalami beberapa kesalahan mendasar pada tipe data, struktur percabangan, dan variabel keluaran.

Jawaban:

- A. Program tidak akan menghasilkan keluaran yang benar menurut spesifikasi — bahkan tidak akan berhasil dikompilasi/di-eksekusi dalam bentuk yang ditulis di modul.
- B. Daftar kesalahan dalam kode yang ada:
 1. **Assign string ke float64 (nam = "A").**
nam dideklarasikan sebagai float64. Menugaskan literal string ke variabel bertipe float akan menyebabkan error tipe pada kompilasi di Go.
 2. **Variabel tujuan salah — seharusnya nmk, bukan nam.**
Ada variabel nmk string yang dimaksudkan untuk menampung indeks huruf. Namun kode menulis ke nam. Akibatnya nmk tetap kosong.
 3. **Menggunakan if terpisah, bukan else if berantai.**
Dalam soal menggunakan beberapa if berurutan (if nam > 80 { ... } if nam > 72.5 { ... } ...). Dengan struktur ini, beberapa kondisi bisa terpenuhi berurutan sehingga nilai akhir bisa tertimpa (atau logika ambang menjadi salah). Yang benar: cek ambang tertinggi dulu, lalu else if untuk rentang berikutnya agar hanya satu cabang yang dipilih.
 4. **else hanya terpasang ke if nam > 40 terakhir.**
Karena penulisan if terpisah, else akhir hanya berkaitan dengan if nam > 40 dan bukan dengan seluruh rangkaian, sehingga perilaku else tidak sesuai tujuan.
 5. **Petik/karakter kutip.**
Di salinan modul muncul kutip miring/typographic quotes yang tidak bisa digunakan untuk menyatakan string di Go.

Alur program seharusnya (yang memenuhi spesifikasi):

1. Baca nam sebagai float64.
2. Gunakan struktur percabangan berantai if ... else if ... else untuk memeriksa rentang nilai dari yang tertinggi ke terendah:
 - a) Jika nam > 80 → nmk = "A"
 - b) else if nam > 72.5 → nmk = "AB"
 - c) else if nam > 65 → nmk = "B"
 - d) else if nam > 57.5 → nmk = "BC"
 - e) else if nam > 50 → nmk = "C"
 - f) else if nam > 40 → nmk = "D"
 - g) else → nmk = "E"
3. Cetak nmk.

Dengan alur ini, setiap nilai nam hanya masuk ke satu rentang dan nmk akan berisi huruf sesuai spesifikasi.

C.

```
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

The screenshot shows a code editor window for a Go program named `assignment-02-true.go`. The code defines a function `main()` that reads a float64 value from the user and prints it along with a corresponding grade character based on a switch statement. The terminal below shows three runs of the program with different input values: 93.5, 70.6, and 49.5, each resulting in the output of the grade character.

```
assignment-02-true.go
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 > 40 {
        nmk = "D"
    } else {
        nmk = "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}

```

TERMINAL

```
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10\assignment-02-true.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10\assignment-02-true.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
```

Ln 1, Col 1 Tab Size: 2 Ln 3, Col 19 69 caracte Plain t 130% Wind UTF-8

Deskripsi program

Program ini membaca satu nilai desimal, yaitu **nam**, yang merepresentasikan nilai akhir suatu mata kuliah. Setelah nilai dibaca, program menentukan nilai huruf yang sesuai dan menyimpannya pada variabel **nmk**.

Program menggunakan rangkaian percabangan **if-else if-else** untuk membandingkan nilai **nam** dengan batas-batas kategori nilai. Pertama, program memeriksa apakah **nam > 80**; jika benar, nilai huruf **nmk** diisi dengan "A". Jika tidak, program melanjutkan ke kondisi berikutnya, yaitu **nam > 72.5**, dan jika terpenuhi, **nmk** diisi "AB".

Proses ini berlanjut untuk kategori lainnya: "B" untuk nilai di atas 65, "BC" untuk nilai di atas 57.5, "C" untuk nilai di atas 50, dan "D" untuk nilai di atas 40. Jika tidak ada kondisi yang terpenuhi, maka nilai huruf otomatis diberikan sebagai "E".

Terakhir, program mencetak hasil berupa teks "*Nilai mata kuliah:*" diikuti oleh nilai huruf **nmk**, yang menunjukkan kategori nilai berdasarkan input yang diberikan pengguna.

3. Tugas 3

Source code

```
package main

import "fmt"

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

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

Screenshot program

The screenshot shows a code editor with a Go file named `assignment-03-01.go`. The code prints a number and lists its factors. Below the editor is a terminal window showing the execution of the program and its output.

```
assignment-03-01.go
...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var b int
7     fmt.Print("Bilangan: ")
8     fmt.Scan(&b)
9
10    fmt.Print("Faktor: ")
11    for f := 1; f <= b; f++ {
12        if b%f == 0 {
13            fmt.Print(f, " ")
14        }
15    }
16 }
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
sion 10.0.22631.6060]
(c) Microsoft Corporation. All rights reserved.

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10\assignment-03-01.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10\assignment-03-01.go"
Bilangan: 7
Faktor: 1 7
F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>
```

File Edit View Aa ☰
Nama : Dadi Maulana Muhammad
NIM : 109082500080
Kelas : S1IF-13-04

Ln 17, Col 1 Tab Size 2 Ln 3, Col 19 69 character Plain text 130% Wind UTF-8

Deskripsi program

Program ini membaca satu bilangan bulat positif, yaitu **b**, lalu menampilkan semua faktor dari bilangan tersebut. Setelah input diterima, program mencetak teks "*Faktor:*" sebagai penanda bahwa daftar faktor akan ditampilkan.

Program kemudian menggunakan sebuah perulangan dari **f = 1** hingga **f = b**. Pada setiap iterasi, program memeriksa apakah **f** merupakan faktor dari **b** dengan mengecek kondisi **b % f == 0**. Jika kondisi tersebut benar, nilai **f** dicetak sebagai salah satu faktor dari bilangan **b**.

Dengan demikian, program ini secara sistematis memeriksa semua bilangan dari 1 hingga **b** dan mencetak setiap bilangan yang memenuhi syarat sebagai faktor dari **b**. Program tidak melakukan pemeriksaan tambahan seperti menentukan apakah bilangan tersebut prima; fungsinya terbatas pada menampilkan daftar faktor saja.

Source code

```
package main

import "fmt"

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

    fmt.Print("Faktor: ")

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

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

Screenshot program

```
assignment-03-02.go ●
assignment-03-02.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var b int
7     fmt.Print("Bilangan: ")
8     fmt.Scan(&b)
9
10    fmt.Print("Faktor: ")
11
12    jumlahFaktor := 0
13    for f := 1; f <= b; f++ {
14        if b%f == 0 {
15            fmt.Print(f, " ")
16            jumlahFaktor++
17        }
18    }

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS
F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10\assignment-03-02.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>go run "f:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10\assignment-03-02.go"
Bilangan: 7
Faktor: 1 7
Prima: true

F:\Me\Others\Kuliah\Algoritma-dan-Pemrograman\week-10>
File Edit View Aa ⚡ ⚙
Nama : Dadi Maulana Muhammad
NIM  : 109082500080
Kelas : S1IF-13-04
```

Deskripsi program

Program ini meminta satu input berupa bilangan bulat positif **b**. Setelah itu, program menampilkan semua faktor dari bilangan tersebut serta menentukan apakah bilangan tersebut merupakan bilangan prima atau bukan.

Pertama, program mencetak teks "*Bilangan:*" lalu membaca nilai **b** dari pengguna. Kemudian program mencetak "*Faktor:*" sebagai penanda bahwa daftar faktor akan ditampilkan.

Program menggunakan perulangan dari 1 hingga **b** untuk memeriksa setiap nilai **f**. Pada setiap iterasi, program mengecek apakah **f** adalah faktor dari **b** dengan kondisi **% f == 0**. Jika benar, nilai **f** dicetak dan penghitung **jumlahFaktor** ditambah satu.

Setelah perulangan selesai, program memeriksa nilai **jumlahFaktor**. Jika jumlah faktor sama dengan 2, berarti bilangan tersebut hanya memiliki dua faktor (1 dan dirinya sendiri), sehingga program mencetak "*Prima: true*". Jika tidak, berarti bilangan memiliki lebih dari dua faktor, dan program mencetak "*Prima: false*".

Dengan demikian, program ini tidak hanya menampilkan semua faktor dari bilangan masukan, tetapi juga menentukan apakah bilangan tersebut termasuk bilangan prima.

.