

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 10

ELSE IF



Disusun oleh:

MOHAMAD ERLANGGA ZEIN

109082500020

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 umur int
    var kk bool
    fmt.Print("Masukkan umur: ")
    fmt.Scan(&umur)
    fmt.Print("punya KK?: ")
    fmt.Scan(&kk)

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

Screenshot program

The screenshot shows a GoLang IDE interface. On the left, the code editor displays the provided Go program. On the right, a terminal window shows the execution of the program and its output.

Code Editor:

```
∞ guided1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {    main redeclared in this block (see details)
6     var umur int
7     var kk bool
8     fmt.Print("Masukkan umur: ")
9     fmt.Scan(&umur)
10    fmt.Print("punya KK?: ")
11    fmt.Scan(&kk)
12
13    if umur ≥ 17 && kk {
14        fmt.Print("bisa membuat KTP")
15    } else {
16        fmt.Print("belum bisa membuat KTP")
17    }
18 }
```

Terminal Output:

```
PS C:\GoLang> go run guided1.go
Masukkan umur: 17
punya KK?: true
bisa membuat KTP
PS C:\GoLang> go run guided1.go
Masukkan umur: 20
punya KK?: false
belum bisa membuat KTP
PS C:\GoLang> go run guided1.go
Masukkan umur: 15
punya KK?: true
belum bisa membuat KTP
PS C:\GoLang>
```

Deskripsi program

Pada program tersebut dijelaskan bahwa ketika user berumur 17 tahun dan memiliki Kartu Keluarga(KK) maka ia bisa membuat KTP, namun apabila user sudah berumur lebih dari 17 tahun namun belum mempunyai KK, maka ia belum bisa membuat KTP, dan bila user berumur kurang dari 17 tahun dan belum mempunyai KK ia juga belum bisa membuat KTP, program ini dibuat dengan algoritma if-else dimana ketika suatu algoritma bernilai true maka output yang dihasilkan true, sedangkan apabila input yang dimasukkan/algoritmanya belum memenuhi maka output yang dihasilkan false.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var x rune

    var huruf, vKecil, vBesar bool

    fmt.Println("masukkan huruf: ")

    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 GoLang IDE interface. On the left, the code editor displays a file named guided2.go with the following content:

```
guided2.go > main
1 package main
2 import "fmt"
3 func main() {    main redeclared in this block
4     var x rune
5     var huruf, vKecil, vBesar bool
6     fmt.Println("masukkan huruf: ")
7     fmt.Scan("%c", &x)
8
9     huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
10    vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o'
11    vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'
12
13    if huruf && (vKecil || vBesar) {
14        fmt.Println("vokal")
15    }else if huruf && !(vKecil || vBesar){
16        fmt.Println("konsonan")
17    }else{
18        fmt.Println("bukan huruf")
19    }
20 }
```

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

```
PS C:\GoLang> go run guided2.go
masukkan huruf: A
vokal
PS C:\GoLang> go run guided2.go
masukkan huruf: f
konsonan
PS C:\GoLang> go run guided2.go
masukkan huruf: 1
bukan huruf
PS C:\GoLang> go run guided2.go
masukkan huruf: $
bukan huruf
PS C:\GoLang>
```

Below the terminal window is a status bar with the following information: Ln 3, Col 16, 61 caractere, Plain t, 160%, Windo, UTF-8.

Deskripsi program

Program ini mengeluarkan output berupa huruf vokal, konsonan, dan bukan huruf, dengan mendeklarasikan variable dengan tipe data rune, yang dimana tipe data rune ini mengkonversikan suatu bilangan/number berubah menjadi Unicode, yang dimana Unicode ini dapat berupa huruf alphabet, sehingga dalam deklarasinya, menggunakan tanda petik 1 saja ('...') kemudian deklarasikan Unicode tersebut ke dalam pengkondisian if else if dan else, yang pertama sebelum memasukkan ke dalam pengkondisian, baiknya dibuat dulu variable lain untuk mendeklarasi nilai dari apa yang ingin di buat perbandingan, dimana disini variable yang akan dibuat menggunakan tipe data Boolean, di dalam program tersebut variable tersebut antara lain variable huruf, vKecil, vBesar, dimana setiap variable berbeda beda perbandingannya, dimana pada variable huruf digunakan sebagai perbandingan dari huruf alphabet, dari alphabet yang besar, dimulai dari $x \geq 'A' \&& x \leq 'Z'$ || $x \geq 'a' \&& x \leq 'z'$ kemudian pada variable vKecil dan vBesar dideklarasikan huruf vokal(A,I,U,E,O) dengan huruf besar dan kecil, kemudian setelah itu di buat pengkondisian seperti pada program di gambar tersebut, sehingga outputnya adalah bila user memasukkan huruf A,I,U,E,O dengan huruf besar maupun huruf kecil, maka output an programnya adalah vokal, ketika huruf selain A,I,U,E,O maka output nya konsonan, dan bila symbol, maka output nya bukan huruf.

3. Guided 3

Source Code

```
package main
```

```

import "fmt"

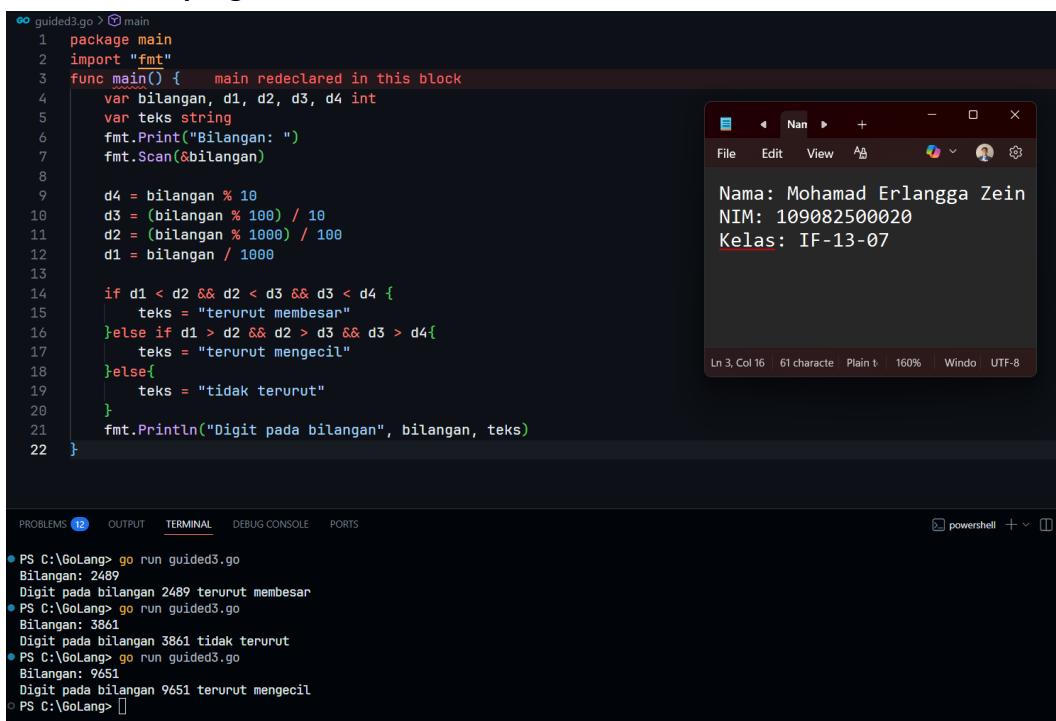
func main() {
    var bilangan, d1, d2, d3, d4 int
    var teks string
    fmt.Print("Bilangan: ")
    fmt.Scan(&bilangan)

    d4 = bilangan % 10
    d3 = (bilangan % 100) / 10
    d2 = (bilangan % 1000) / 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)
}

```

Screenshoot program



The screenshot shows a GoLang development environment with two panes. The left pane displays the source code of `guided3.go`. The right pane shows the terminal output of running the program.

Code (guided3.go):

```

1 package main
2 import "fmt"
3 func main() { main redeclared in this block
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 terurut"
18    }
19    fmt.Println("Digit pada bilangan", bilangan, teks)
20
21 }
22 }
```

Terminal Output:

- PS C:\GoLang> go run guided3.go

Bilangan: 2489

Digit pada bilangan 2489 terurut membesar
- PS C:\GoLang> go run guided3.go

Bilangan: 3861

Digit pada bilangan 3861 tidak terurut
- PS C:\GoLang> go run guided3.go

Bilangan: 9651

Digit pada bilangan 9651 terurut mengecil

Deskripsi program

Program ini menghasilkan output dimana ketika user menginputkan angka dengan 4 digit, kemudian algoritmanya akan memproses apakah bilangan yang di inputkan oleh user itu terurut membesar, terurut mengecil, atau tidak terurut, sehingga dalam pemrogramannya, perlu mendeklarasikan operasi matematika terhadap d1, d2, d3, d4 dengan membaginya dengan membaginya dengan 1000, lalu moduluskan dengan 1000, 100, dan 10 dengan urutan ribuan, ratusan, puluhan, satuan, sehingga dalam pengkondisianya adalah $d1 < d2 \&& d2 < d3 \&& d3 < d4$ yang mendeklarasikan terurut membesar, lalu else if $d1 > d2 \&& d2 > d3 \&& d3 > d4$ yang mendeklarasikan terurut mengecil lalu ketika digitnya acak maka tidak terurut.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var parsel, kilogram, gram, harga, biaya, total int
    fmt.Print("Berat parsel(gram): ")
    fmt.Scan(&parsel)

    kilogram = parsel / 1000
    gram = parsel % 1000
    harga = kilogram * 10000

    if parsel > 10000 {
        biaya = gram * 5
        total = harga
    } else if gram >= 500 {
        biaya = gram * 5
        total = harga + biaya
    } else if gram < 500 {
        biaya = gram * 15
        total = harga + biaya
    }
}
```

```

        }

        fmt.Printf("detail berat: %d kg + %d gr\n", kilogram,
gram)

        fmt.Printf("detail biaya: Rp. %d + Rp. %d\n", harga,
biaya)

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

}

```

Screenshot program

The screenshot shows a code editor with a dark theme. On the left, the code for `soal1.go` is displayed:

```

1 package main
2 import "fmt"
3
4 func main() {
5     var parsel, kilogram, gram, harga, biaya, total int
6     fmt.Print("Berat parsel(gram): ")
7     fmt.Scan(&parsel)
8
9     kilogram = parsel / 1000
10    gram = parsel % 1000
11    harga = kilogram * 10000
12
13    if parsel > 10000 {
14        biaya = gram * 5
15        total = harga
16    } else if gram ≥ 500 {
17        biaya = gram * 5
18        total = harga + biaya
19    } else if gram < 500 {
20        biaya = gram * 15
21        total = harga + biaya
22    }
23
24    fmt.Printf("detail berat: %d kg + %d gr\n", kilogram, gram)
25    fmt.Printf("detail biaya: Rp. %d + Rp. %d\n", harga, biaya)
26    fmt.Printf("total biaya: Rp. %d", total)
27 }

```

On the right, a terminal window shows the output of running the program:

```

Nama: Mohamad Erlangga Zein
NIM: 109082500020
Kelas: IF-13-07

Ln 3, Col 16 61 character Plain t 160% Windo UTF-8

```

The terminal also lists previous command history:

- PS C:\GoLang> go run soal1.go

Deskripsi program

Dalam program ini, seluruh variabel operasional yaitu `parsel`, `kilogram`, `gram`, `harga`, `biaya`, dan `total` dideklarasikan menggunakan tipe data `int` (integer) karena program ini hanya memproses bilangan bulat untuk satuan berat dan nilai rupiah tanpa desimal. Fungsi utama `main()` berperan sebagai titik awal eksekusi logika, yang di dalamnya memanfaatkan pustaka standar `fmt` untuk menangani input dan output. Secara spesifik, fungsi `fmt.Print` digunakan untuk menampilkan teks *prompt* tanpa baris baru, `fmt.Scan` berfungsi menangkap input angka dari pengguna ke alamat memori variabel `parsel`, dan `fmt.Printf` digunakan untuk menampilkan hasil perhitungan akhir dengan format khusus (menggunakan `%d`) agar data angka dapat disisipkan rapi ke dalam kalimat output. Dan program ini juga menggunakan logika penentuan harga di mana setiap 1 kilogram dikenakan biaya Rp10.000, dan menerapkan aturan kondisional untuk sisa gram: jika total berat melebihi 10 kg maka

sisa gram tidak dikenakan biaya tambahan, namun jika berat standar, sisa gram dikenakan biaya Rp5 per gram (jika ≥ 500 g) atau Rp15 per gram (jika < 500 g), yang kemudian hasil akhirnya ditampilkan secara terperinci meliputi detail berat, kalkulasi biaya, dan total harga.

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 GoLang IDE interface. On the left, the code editor displays a file named 'soal2.go' with several syntax errors highlighted in red. The errors are related to assignments where string constants are used instead of float64 values. On the right, a terminal window shows the command 'go run soal2.go' being run, followed by a stack trace of errors. The errors are identical to those in the code editor.

```

soal2.go:3: main redeclared in this block
soal2.go:4: var nam float64
soal2.go:5: var nmk string
soal2.go:6: fmt.Println("Nilai akhir mata kuliah: ")
soal2.go:7: fmt.Scan(&nam)
soal2.go:8: if nam > 80 {
soal2.go:9:     nam = "A"    cannot use "A" (untyped string constant) as float64 value in assignment
soal2.go:10: }
soal2.go:11: if nam > 72.5 {
soal2.go:12:     nam = "AB"   cannot use "AB" (untyped string constant) as float64 value in assignment
soal2.go:13: }
soal2.go:14: if nam > 65 {
soal2.go:15:     nam = "B"    cannot use "B" (untyped string constant) as float64 value in assignment
soal2.go:16: }
soal2.go:17: if nam > 57.5 {
soal2.go:18:     nam = "BC"   cannot use "BC" (untyped string constant) as float64 value in assignment
soal2.go:19: }
soal2.go:20: if nam > 50 {
soal2.go:21:     nam = "C"    cannot use "C" (untyped string constant) as float64 value in assignment
soal2.go:22: }
soal2.go:23: if nam > 40 {
soal2.go:24:     nam = "D"    cannot use "D" (untyped string constant) as float64 value in assignment
soal2.go:25: } else if nam <= 40 {
soal2.go:26:     nam = "E"    cannot use "E" (untyped string constant) as float64 value in assignment
soal2.go:27: }
soal2.go:29: fmt.Println("Nilai mata kuliah: ", nmk)
soal2.go:30: }

File Edit View AA File Edit View AA
Nama: Mohamad Erlangga Zein
NIM: 109082500020
Kelas: IF-13-07
Ln 3, Col 16 | 61 caracte | Plain t | 160% | Windo | UTF-8

```

• SOAL A:

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

Jawaban:

Jawabannya program tidak dapat dieksekusi, karena prosedur dari program tersebut salah, dimana harusnya pada pengkondisian if then nya harusnya if nam = > 80 { nmk = "A" } bukan pada program sebelumnya yang dimana mendeklarasikannya nam = "A", karena tipe data pada nam dan nmk berbeda, nam dengan tipe data float64, sedangkan nmk tipe data string, sehingga programnya tidak dapat tereksekusi/error.

• SOAL B:

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

Jawaban:

Kesalahan dalam program tersebut adalah, kesalahan dalam mendeklarasikan variable, dimana seharusnya pada pengkondisiannya seharusnya menggunakan if nam > 80 { nmk = "A" }, karena perbedaan tipe data, dimana nam adalah float64 dan nmk adalah string, sehingga mengalami error, dan setelah diubah juga output an dari 80.1 belum benar, hasil output an nya adalah D, seharusnya adalah A, karena dalam program tersebut hanya mendeklarasikan if then, bukan if – else if, sehingga dalam pemrosesannya hanya memproses pengkondisian yang memiliki aksi, serta dalam pengkondisiannya juga mengapa output an nya D karena pada 2 pengkondisian if then terakhir itu menggunakan if – else if, sehingga dalam pemrosesannya, ketika user menginput 80.1 maka hasilnya akan D, bukan A.

• SOAL C:

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

Jawaban:

```
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 {
        nmk = "D"
    }
    fmt.Println(nmk)
}
```

```

        } 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)
    }
}

```

Deskripsi program

Program ini berfungsi sebagai sistem konversi nilai akademik sederhana yang mengklasifikasikan nilai angka menjadi indeks huruf mutu (A hingga E) menggunakan bahasa Go. Variabel input nam dideklarasikan dengan tipe data float64 agar dapat memproses nilai desimal presisi ganda, sedangkan variabel output nmk bertipe string digunakan untuk menampung hasil konversi berupa teks huruf. Di dalam blok fungsi main(), interaksi input-output dikelola oleh paket fmt—di mana fmt.Print menampilkan *prompt*, fmt.Scan menangkap input ke alamat memori variabel nam, dan fmt.Println mencetak hasil akhir—sementara logika penentuan nilai dijalankan melalui struktur percabangan if-else if berjenjang yang mengevaluasi rentang nilai secara menurun, mulai dari batas atas (> 80 untuk A) hingga batas bawah (<= 40 untuk E).

Sehingga dalam soal ini, ketika diuji memasukkan input 93.5, 70.6, dan 49.5 hasilnya A, B, D

BUKTI:

```

soal2.go > main
1 package main
2 import "fmt"
3 func main() { main redeclared in this block
4 var nam float64
5 var nmk string
6
7     fmt.Println("Nilai akhir mata kuliah: ")
8     fmt.Scan(&nam)
9
10    if nam > 80 {
11        nmk = "A"
12    } else if nam > 72.5 {
13        nmk = "AB"
14    } else if nam > 65 {
15        nmk = "B"
16    } else if nam > 57.5 {
17        nmk = "BC"
18    } else if nam > 50 {
19        nmk = "C"
20    } else if nam > 40 {
21        nmk = "D"
22    } else if nam <= 40 {
23        nmk = "E"
24    }
25    fmt.Println("Nilai mata kuliah: ", nmk)
26
}

```

PS C:\GoLang> go run soal2.go
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\GoLang> go run soal2.go
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\GoLang> go run soal2.go
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\GoLang> []

3. Tugas 3A

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.Print(" ", i)
        }
    }
}
```

Screenshoot program

The screenshot shows a GoLang IDE interface. On the left, the code editor displays the source code for 'soal3_1.go'. On the right, a terminal window shows the execution of the program. The terminal output includes the user input '12' and the program's response 'faktor: 1 2 3 4 6 12'. Below the terminal, the status bar indicates the file is 'soal3_1.go', it has 12 problems, and the encoding is UTF-8.

```
soal3_1.go > main
1 package main
2 import "fmt"
3
4 func main() {    main redeclared in this block
5     var n, i int
6     fmt.Print("masukkan bilangan: ")
7     fmt.Scan(&n)
8
9     fmt.Print("faktor: ")
10    for i = 1; i <= n; i++ {
11        if n % i == 0 {
12            fmt.Print(" ", i)
13        }
14    }
15 }
```

```
Nama: Mohamad Erlangga Zein
NIM: 109082500020
Kelas: IF-13-07
```

PROBLEMS 12 OUTPUT TERMINAL DEBUG CONSOLE PORTS

- PS C:\GoLang> go run soal3_1.go
masukkan bilangan: 12
faktor: 1 2 3 4 6 12
- PS C:\GoLang> go run soal3_1.go
masukkan bilangan: 7
faktor: 1 7
- PS C:\GoLang> []

Deskripsi program

Program ini merupakan aplikasi berbasis bahasa Go yang dirancang untuk mencari dan menampilkan faktor-faktor pembentuk dari sebuah bilangan bulat yang diinputkan oleh pengguna. Variabel utama yang digunakan adalah n untuk menampung nilai input dan i sebagai iterator dalam perulangan, di mana keduanya menggunakan tipe data int (integer) karena operasi matematika modulus memerlukan bilangan bulat. Fungsi main() berperan sebagai blok eksekusi utama yang memanggil fungsi-fungsi dari paket fmt: fmt.Print untuk menampilkan teks prompt dan hasil faktor secara menyamping, serta fmt.Scan untuk menangkap input pengguna ke variabel n. Algoritmanya bekerja dengan menggunakan struktur kontrol perulangan for yang mengevaluasi angka dari 1 hingga n, disertai logika percabangan if yang memeriksa apakah sisa bagi (%) antara n dan i bernilai 0; jika kondisinya terpenuhi, maka angka i tersebut ditampilkan sebagai faktor yang valid.

4. Tugas 3B

Source code

```
package main

import "fmt"

func main() {
    var n, i, faktor int
    var prima bool
    fmt.Print("masukkan bilangan: ")
    fmt.Scan(&n)
    faktor = 0

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

    fmt.Println()
}
```

```

fmt.Println("Prima: ")

if faktor == 2 {

    prima = true

}

fmt.Println(prima)

}

```

Screenshot program

The screenshot shows a code editor with a Go file named `soal3_2.go`. The code implements a prime number checker. It prompts for input, finds factors, and checks if the number is prime. The terminal below shows two runs of the program.

```

go soal3_2.go > main
1 package main
2 import "fmt"
3
4 func main() {      main redeclared in this block
5     var n, i, faktor int
6     var prima bool
7     fmt.Print("masukkan bilangan: ")
8     fmt.Scan(&n)
9     faktor = 0
10
11    fmt.Print("faktor: ")
12    for i = 1; i <= n; i++ {
13        if n % i == 0 {
14            fmt.Print(" ", i)
15            faktor++
16        }
17    }
18
19    fmt.Println()
20
21    fmt.Print("Prima: ")
22    if faktor == 2 {
23        prima = true
24    }
25    fmt.Println(prima)
26 }

PROBLEMS 12 OUTPUT TERMINAL DEBUG CONSOLE PORTS
PS C:\GoLang> go run soal3_2.go
masukkan bilangan: 12
faktor: 1 2 3 4 6 12
Prima: false
● PS C:\GoLang> go run soal3_2.go
masukkan bilangan: 7
faktor: 1 7
Prima: true
○ PS C:\GoLang>

```

The terminal output shows two runs of the program. In the first run, the user inputs 12, which has multiple factors (1, 2, 3, 4, 6, 12), so it is determined not to be prime. In the second run, the user inputs 7, which has only two factors (1, 7), so it is determined to be prime.

Deskripsi program

Program ini merupakan pengembangan logika matematika untuk menentukan apakah sebuah bilangan bulat termasuk bilangan prima atau bukan berdasarkan jumlah faktor pembaginya. Variabel `n`, `i`, dan `faktor` dideklarasikan dengan tipe data `int` (integer) untuk menangani input angka, iterasi perulangan, dan pencacahan jumlah faktor, sementara variabel `prima` menggunakan tipe data `bool` (boolean) untuk menyimpan status logis `true` atau `false`. Fungsi utama `main()` mengontrol alur eksekusi yang memanfaatkan paket `fmt` seperti `fmt.Scan` untuk input dan `fmt.Println` untuk output di mana algoritma bekerja dengan mencari faktor-faktor bilangan tersebut melalui perulangan `for`, lalu menggunakan struktur `if` untuk memeriksa apakah jumlah faktor (`faktor`) tepat berjumlah 2; jika ya, maka variabel `prima` diubah menjadi `true` yang menandakan bilangan tersebut adalah prima.