

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL X**

**MATERI**



Oleh:

**YASVIN SYAHGANA**

**2311102065**

**IF-11-02**

**PROGRAM STUDI S1 INFORMATIKA  
FAKULTAS INFORMATIKA  
UNIVERSITAS TELKOM PURWOKERTO  
2024**

# I. DASAR TEORI

## Dasar Teori

### 1. Struktur GO

Dalam Bahasa pemrograman golang, program mempunyai dua komponen utama yaitu:

- **Package main** merupakan penanda bahwa file ini berisi program utama
- **Func main()** berisi kode utama dari sebuah file program go

### 2. Koding dan Kompilasi

#### a. Koding

- Tidak berbeda dengan penulisan program sumber dalam Bahasa lain, program Go harus dibuat dengan menggunakan penyunting teks dan disimpan dalam format teks, bukan dalam format dokumen
- Setiap program Go disimpan dalam file teks dengan ekstensi \*.go, dengan nama bebas.
- Setiap satu program Go disimpan dalam bentuk satu folder tersendiri.

#### b. Kompilasi

Pemrograman dirancang untuk diimplementasikan sebagai interpreter dan lainnya sebagai kompilator. Interpreter akan membaca setiap baris instruksi dan kemudian langsung mengeksekusinya, dengan hanya sedikit pemeriksaan apakah penulisan keseluruhan program sudah benar atau belum. Kompilator akan memeriksa keseluruhan program sumber dan mengubahnya menjadi program eksekutabel. Sehingga penulisan sudah diperiksa sebelum dieksekusi.

### 3. Tipe Data dan Instruksi Dasar

#### a) Data dan Variabel

Variable adalah nama dari suatu Lokasi di memori, data dengan tipe tertentu dapat disimpan

- Nama variable dimulai dengan huruf dan dapat diikuti dengan jumlah huruf, angka, atau garis bawah

Notasi tipe dasar	Tipe dalam Go	Keterangan
integer	int int8 int32 //rune int64 uint uint8 //byte uint32 uint64	bergantung platform 8 bit: -128..127 32 bit: -10 <sup>9</sup> ..10 <sup>9</sup> 64 bit: -10 <sup>19</sup> ..10 <sup>19</sup> bergantung platform 0..255 0..4294967295 0..(2 <sup>64</sup> -1)
real	float32 float64	32bit: -3.4E+38 .. 3.4E+38 64bit: -1.7E+308 .. 1.7E+308
boolean (atau logikal)	bool	false dan true
karakter	byte //uint8 rune //int32	tabel ASCII/UTF-8 tabel UTF-16
string	string	

- Operasi yang dapat dilakukan terhadap tipe data adalah

Operator dalam Go	Tipe data terkait	Keterangan
+	string integer dan real	konkatenasi 2 string operasi penjumlahan
- * /	integer dan real	operasi pengurangan, perkalian, dan pembagian
%	integer	operasi sisa pembagian integer (modulo)
&   ^ &^	integer	operasi <b>per-bit</b> AND, OR, XOR, AND-NOT
<< >>	integer dan unsigned integer	operasi geser bit kiri/kanan sebanyak unsigned integer yang diberikan
< <= > >= !=	selain boolean	komparasi menghasilkan nilai boolean komparasi karakter sesuai dengan posisi karakter tersebut dalam tabel ASCII/UTF-16 komparasi string sesuai dengan operasi karakter per karakter, dimulai dari karakter paling kiri (awal)
&&    !	boolean	operasi <b>boolean</b> AND, OR, dan NOT
* &	variabel apasaja	mendapatkan data dari lokasi memori dan mendapatkan lokasi dari variabel

- Tipe data yang berbeda tidak boleh dicampur dalam satu ekspresi, bahkan tipe data yang satu jenis, contohnya int yang sama (**Int** dan **Int32**)

Konversi tipe	Data	Tipe baru	Keterangan
tipe(data)	integer	integer	format data tidak berubah, hanya penyesuaian jumlah bit. Kekurangan bit diisi bit 0 di sebelah kiri (MSB)
	real	real	format data tidak berubah, hanya penyesuaian jumlah bit. Kekurangan bit, maka bit mantisa diisi bit 0.
	real	integer	format data disesuaikan dengan tipe data tujuan
	integer	real	format data disesuaikan dengan tipe data tujuan
fmt.Sprintf("%v",v)	any type	string	tulis output ke string
fmt.Sprintf("%c",v)	karakter	string	tulis karakter ke string
fmt.Sscanf(s,"%v",&v)	string	any type	baca string ke variabel dengan tipe tertentu
fmt.Sscanf(s,"%c",&v)	string	karakter	baca string ke variabel bertipe karakter

## b) Instruksi Dasar

Notasi instruksi dasar	Penulisan dalam bahasa Go	Keterangan
<code>v1 &lt;- e1</code> <code>v1 &lt;- v1 + e1</code> <code>v1 &lt;- v1 - e1</code> <code>v1 &lt;- v1 + 1</code> <code>v1 &lt;- v1 - 1</code>	<code>v1 = e1</code> <code>v1 += e1 // atau v1 = v1 + e1</code> <code>v1 -= e1 // atau v1 = v1 - e1</code> <code>v1++ // atau v1 = v1 + 1</code> <code>v1-- // atau v1 = v1 - 1</code>	operasi assignment, mengisi data ke lokasi memori (variabel)
<code>input(v1, v2)</code>	<code>fmt.Scan( &amp;v1, &amp;v2 )</code> <code>fmt.Scanln( &amp;v1, &amp;v2 )</code> <code>fmt.Scanf( "%v %v", &amp;v1, &amp;v2 )</code>	Pembacaan data memerlukan alamat memori ke mana data akan disimpan.
<code>output(e1, e2)</code>	<code>fmt.Print( e1, e2 )</code> <code>fmt.Println( e1, e2 )</code> <code>fmt.Printf( "%v %v\n", e1, e2 )</code>	Penulisan data memerlukan nilai data yang akan ditulis.

#### 4. Struktur perulangan Go

Dalam pemrograman struktur, setiap rancangan algoritma tidak harus memenuhi syarat satu pintu masuk dan pintu keluar. Karena itu tidak boleh membuat program sumber yang mempunyai struktur loop yang mempunyai pintu keluar lebih dari satu.

##### • Bentuk While – Loop

	Notasi algoritma	Penulisan dalam bahasa Go
1 2 3 4	<pre>while (kondisi) do     ... kode yang diulang endwhile</pre>	<pre>for kondisi {     .. kode yang diulang }</pre>

##### • Bentuk Repeat – Until

	Notasi Algoritma	Penulisan dalam bahasa Go
1 2 3 4 5 6 7 8 9	<pre>repeat     .. kode yang diulang until (kondisi)</pre>	<pre>for selesai:=false; !selesai; {     .. kode yang diulang     selesai = kondisi }</pre> <pre>for selesai:=false; !selesai; selesai=kondisi {     .. kode yang diulang }</pre>

##### • Bentuk if – else

	Notasi algoritma	Penulisan dalam bahasa Go
1 2 3	<pre>if (kondisi) then     .. kode untuk kondisi true endif</pre>	<pre>if kondisi {     .. kode untuk kondisi true }</pre>
4 5 6 7 8	<pre>if (kondisi) then     .. kode untuk kondisi true else     .. kode untuk kondisi false endif</pre>	<pre>if kondisi {     .. kode untuk kondisi true } else {     .. kode untuk kondisi false }</pre>
9 10 11 12 13 14 15 16 17	<pre>if (kondisi-1) then     .. kode untuk kondisi-1 true else if (kondisi-2) then     .. kode untuk kondisi-2 true .. dst. dst. else     .. kode jika semua kondisi     .. di atas false endif</pre>	<pre>if kondisi_1 {     .. kode untuk kondisi_1 true } else if kondisi_2 {     .. kode untuk kondisi_2 true .. dst. dst. } else {     .. kode jika semua kondisi     .. di atas false }</pre>

##### • Bentuk Switch – Case

	Notasi algoritma	Penulisan dalam bahasa Go
1 2 3 4 5 6 7 8 9 10	<pre> depend on expresi   nilai_1:   .. kode jika ekspresi bernilai_1   nilai_2:   .. kode jika ekspresi bernilai_2   .. dst. dst.   } </pre>	<pre> switch ekspresi { case nilai_1:   .. kode jika ekspresi bernilai_1 case nilai_2:   .. kode jika ekspresi bernilai_2   .. dst. dst. default:   .. kode jika tidak ada nilai   .. yang cocok dengan ekspresi } </pre>
11 12 13 14 15 16 17 18 19 20	<pre> depend on (daftar variabel)   kondisi_1:   .. kode jika ekspresi_1 true   kondisi_2:   .. kode jika ekspresi_2 true   .. dst. dst.   } </pre>	<pre> switch { case kondisi_1:   .. kode jika ekspresi_1 true case kondisi_2:   .. kode jika ekspresi_2 true   .. dst. dst. default:   .. jika tidak ada ekspresi   .. yang bernilai true } </pre>

## II. GUIDED

### GUIDED 2A\_1

#### Source code

```
package main

import
"fmt"

func main(){      var(
satu, dua, tiga string
temp string

)      fmt.Println("Masukkan input
string: ")      fmt.Scanln(&satu)
fmt.Print("masukkan input string: ")
fmt.Scanln(&dua)      fmt.Print("masukkan
input string: ")      fmt.Scanln(&tiga)

      fmt.Println("Output awal = " + satu + " " + dua + " " +
tiga)      temp = satu      satu = dua      dua = tiga      tiga
= temp

      fmt.Println("Output akhir = " + satu + " " + dua + " " +
tiga)
}
```

#### Output

```
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan 1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan 1\tempCodeRunnerFile.go"
Masukkan input string:
yaswin
masukkan input string: syah
masukkan input string: gana
Output awal = yaswin syah gana
Output akhir = syah gana yaswin
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan 1> |
```

## Deskripsi program

Program ini adalah program yang meminta user untuk memasukan 3buah string, kemudian menukar nilai ketiga string tersebut. Program ini menggunakan Teknik pertukaran nilai untuk menukar nilai antara dua variable tanpa menggunakan operator tambahan.

## GUIDED 2A\_2

```
package main

import

"fmt"

func main() {      var

tahun int

fmt.Print("Tahun: ")

fmt.Scanln(&tahun)

    kabisat := cekKabisat(tahun)

fmt.Println("Kabisat: ", kabisat)

} func cekKabisat(tahun int)

bool {      if tahun%400 == 0 {

return true

    } else if tahun%4 == 0 {

return true

    } else if tahun%100 == 0 {
```

```

        return false
    }
    return false
}

```

### Output

```

rtemuan 1\guided2a_2.go"
Tahun: 2016
Kabisat: true
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\guided2a_2.go"
Tahun: 2000
Kabisat: true
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\guided2a_2.go"
Tahun: 2018
Kabisat: false
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1>

```

### Deskripsi program

Program ini menjelaskan apakah sebuah tahun yang diinput oleh user tahun kabisat atau tidak, jika iya maka output akan muncul "true" dan jika tidak maka output akan muncul "tidak". Program ini menggunakan pemrograman yang sederhana dan efektif untuk memeriksa apakah sebuah tahun adalah tahun kabisat atau tidak.

### GUIDED 2A\_3

```

package main

import

(

    "fmt"

    "math"

) func
main() {

```



```
var jejari int
fmt.Print("Jejari: ")
fmt.Scanf("%d", &jejari)

volume :=
hitungVolumeBola(float64(jejari))    luas :=
hitungLuasKulitBola(float64(jejari))

    fmt.Println("Bola dengan jejari", jejari, "memiliki volume",
volume, "dan luas kulit", luas)

} func hitungVolumeBola(jejari float64) float64
{    const pi = 3.1415926535    return (4.0 /
3.0) * pi * math.Pow(jejari, 3)
} func hitungLuasKulitBola(jejari float64)
float64 {    const pi = 3.1415926535    return
4 * pi * math.Pow(jejari, 2)
}
```

## Output

```
PROBLEMS 27 OUTPUT DEBUG CONSOLE TERMINAL ...

PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\tempCodeRunnerFile.go"
Jejari: 5
Bola dengan jejari 5 memiliki volume 523.5987755833333
dan luas kulit 314.15926535
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> █
```

### Deskripsi program

Program ini menjelaskan program tentang menghitung volume dan luas kulit bola berdasarkan jejari yang diberikan oleh user. Jika user menginput jejari 5 maka output akan menghasilkan output sesuai yang diinput oleh user. Fungsi-fungsi yang digunakan.

### Guided 2A\_4

#### Source code

```
package main
import
(
    "fmt"
)
func main() {
    var celsius float64

    // Input from user
    fmt.Print("Masukkan suhu dalam derajat Celsius: ")
    fmt.Scan(&celsius)

    // Konversi suhu
    reamur := celsius * 4 / 5
    fahrenheit := (celsius * 9 / 5) + 32
    kelvin := celsius + 273.15

    // Output hasil konversi
    fmt.Println("Temperatur Celsius:", celsius)
    fmt.Println("Derajat Reamur:", reamur)
    fmt.Println("Derajat Fahrenheit:", fahrenheit)
    fmt.Println("Derajat Kelvin:", kelvin)
}
```

### Output

```

PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan 1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan 1\guided2a_4.go"
Masukkan suhu dalam derajat Celsius: 50
Temperatur Celsius: 50
Derajat Reamur: 40
Derajat Fahrenheit: 122
Derajat Kelvin: 323,15
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan 1> 

```

### Deskripsi program

Program ini menjelaskan program untuk mengkonversi suhu dari derajat celcius ke beberapa satuan yaitu Reamur, Fahrenheit, dan Kelvin. User akan menginput angka dan output akan mengeluarkan hasil yang diminta oleh user. Program ini digunakan untuk mengkonversi suhu dalam berbagai situasi, seperti dalam pengukuran suhu lingkungan, pengukuran suhu tubuh dan lain-lain.

### Guided 2A\_5

#### Source code

```

package main
import
(
    "fmt"
)
func main() {
    // Definisikan array untuk menyimpan input integer dan karakter
    var bilangan [5]int
    var karakter [3]rune

    // Input untuk 5 bilangan bulat
    fmt.Println("Masukkan 5 bilangan bulat (nilai antara 32 -
127):")
    for i := 0; i < 5; i++ {
        fmt.Scan(&bilangan[i])
    }

    // Input untuk 3 karakter tanpa spasi
    fmt.Println("Masukkan 3 karakter (tanpa spasi):")
    for i := 0; i < 3; i++ {
        fmt.Scanf("%c", &karakter[i])
    }
    // Menampilkan karakter hasil konversi dari nilai bilangan
    bulat (ASCII)
    fmt.Println("\nKeluaran:")
    for i := 0; i < 5; i++ {
        fmt.Printf("%c", bilangan[i])
    }
    fmt.Println()

    // Menampilkan karakter yang diinput oleh pengguna tanpa spasi
    for i := 0; i < 3; i++ {
        fmt.Printf("%c", karakter[i])
    }

    fmt.Println()
}

```

## Output

```
PS C:\Users\Lenovo> go run "c:\Users\Lenovo\Downloads\Guided5\main.go"
Masukkan 5 angka integer (32-127):
66 97 103 117 115
Masukkan 3 karakter:
SNO
Keluaran:
Bagus
SNO
PS C:\Users\Lenovo> |
```

## Deskripsi Program

Program ini menjelaskan program yang meminta input dari user dan kemudian mengkonversi input tersebut menjadi karakter ASCII. Program ini menggunakan beberapa fungsi Go, seperti `fmt.Println` untuk mencetak teks ke layar, `fmt.Println` untuk meminta input dari pengguna, dan `fmt.Println` untuk mencetak karakter ASCII ke layar.

## III. UNGUIDED

Unguided 2B\_1

Source Code

```

package main
import
(
    "fmt"
)
func main() {
    // Urutan warna yang benar
    expected := [4]string{"merah", "kuning", "hijau",
"ungu"}

    // Data percobaan dari input
    pengguna    var percobaan [5][4]string
    for i := 0; i < 5; i++ {
        fmt.Printf("Masukkan percobaan %d (pisahkan dengan
spasi): ", i+1)      fmt.Scan(&percobaan[i][0],
&percobaan[i][1],
&percobaan[i][2], &percobaan[i][3])
    }
    // Periksa apakah semua percobaan sesuai dengan urutan
warna yang diharapkan    berhasil := true    for i := 0;
i < 5; i++ {        for j := 0; j < 4; j++ {
        if percobaan[i][j] != expected[j] {
berhasil = false        break
        }
    }
    if !berhasil {
break
    }

    // Tampilkan hasil
    if berhasil {
        fmt.Println("BERHASIL: true")
    } else {
        fmt.Println("BERHASIL: false")
    } }
}

```

Output

```

PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\per temua
Masukkan percobaan 1 : merah kuning hijau ungu
Masukkan percobaan 2 : merah kuning hijau ungu
Masukkan percobaan 3 : merah kuning hijau ungu
Masukkan percobaan 4 : merah kuning hijau ungu
Masukkan percobaan 5 : merah kuning hijau ungu
BERHASIL: true

```

## Deskripsi Program

Program ini menjelaskan tentang aplikasi yang meminta user untuk memasukan 5 percobaan dengan urutan warna benar. Urutan warna yang benar telah ditentukan sebelumnya sebagai “merah, kuning hijau, ungu”. Program ini memeriksa apakah semua percobaan yang diinput oleh user sesuai dengan urutan warna yang sesuai. Jika sesuai maka program akan menampilkan “BERHASIL: true”.

## Unguided 2B\_2

```
package main
import
(
    "fmt"
)
func main() {
    // Variabel untuk menyimpan pita (kumpulan nama bunga) dan
    jumlah bunga      var pita string      var jumlahBunga int

    // Loop untuk terus meminta input sampai pengguna mengetik
    'SELESAI'
    for {
        var namaBunga string
        fmt.Print("Masukkan nama bunga : ")
        fmt.Scan(&namaBunga)

        // Jika input adalah 'SELESAI', keluar dari
        loop      if namaBunga == "SELESAI" {
        break
            }

        // Jika ini bukan input pertama, tambahkan tanda '-'
        sebelum menambahkan nama bunga      if pita != "" {          pita
        += " - "
            }

        // Menambahkan nama bunga ke dalam
        pita      pita += namaBunga
        jumlahBunga++
    }
    // Menampilkan pita dan jumlah bunga setelah semua input
    selesai
    fmt.Println("\nIsi pita setelah proses input:")
    fmt.Println(pita)
    fmt.Printf("Jumlah bunga dalam pita: %d\n", jumlahBunga)
}
```

## Output

```
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\unguided2b_2.go"
Masukkan nama bunga : kertas
Masukkan nama bunga : mawar
Masukkan nama bunga : tulip
Masukkan nama bunga : SELESAI

Isi pita setelah proses input:
kertas - mawar - tulip
Jumlah bunga dalam pita: 3
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> █
```

## Deskripsi Program

Program ini menjelaskan tentang aplikasi yang meminta user untuk menginput nama-nama bunga secara berulang sampai user mengetik “SELESAI”. Program ini kemudian akan menampilkan semua nama bunga yang telah diinput dalam bentuk pita dan jumlah bunga yang ada didalam pita.

## Unguided 2B\_3

```

package main

import "fmt"

func main() {
    var beratKantong1, beratKantong2 float64
    var totalBerat float64

    for {
        fmt.Print("Masukan berat belanjaan di kedua
kantong: ")
        _, err := fmt.Scan(&beratKantong1, &beratKantong2)

        if err != nil {
            fmt.Println("Input tidak valid. Harap
masukkan bilangan real positif.")
            continue
        }

        totalBerat += beratKantong1 + beratKantong2
        if beratKantong1 < 0 || beratKantong2 < 0 {
            fmt.Println("Berat tidak boleh negatif.")
            break
        }
        if totalBerat > 150 {
            fmt.Println("Total berat melebihi 150 kg. Proses selesai.")
            break
        }

        selisihBerat := beratKantong1 - beratKantong2
        if selisihBerat < 0 {
            selisihBerat = -selisihBerat
        }
        fmt.Printf("Sepeda motor pak Andi akan oleng:
%t\n", selisihBerat >= 9)
    } }

```

## Output

```

PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\unguided2b_3.go"
Masukan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukan berat belanjaan di kedua kantong: 72.3 66.9
Total berat melebihi 150 kg. Proses selesai.
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1>

```

## Deskripsi Program



Program ini menjelaskan user untuk menginput berat belanjaan di kedua kantong sepeda motor secara berulang. Program ini kemudian akan menampilkan apakah sepeda motor akan oleng atau tidak berdasarkan selisih berat antara kedua kantong.

#### Unguided 2B\_4

```
package main
import
(
    "fmt"
)
func hitungFungsiFK(K int) float64 {
    return float64((4*K+2)*(4*K+2)) / float64((4*K+1)*(4*K+3))
}
func hitungAkar2Hampiran(K int) float64 {
    akar2 := 1.0
    for k := 0; k <= K; k++ {
        akar2 *= float64((4*k+2)*(4*k+2)) /
float64((4*k+1)*(4*k+3))
    }
    return akar2
} func main() {
var pilihan int
var K int

    fmt.Println("Pilih opsi:")
    fmt.Println("1. Hitung f(K)")
    fmt.Println("2. Hitung hampiran akar 2")
    fmt.Print("Masukkan pilihan (1/2): ")    fmt.Scan(&pilihan)

    fmt.Print("Masukkan nilai K: ")
    fmt.Scan(&K)

    if pilihan == 1 {
fK := hitungFungsiFK(K)
        fmt.Printf("Nilai f(K) = %.10f\n", fK)
    } else if pilihan == 2 {
        akar2 := hitungAkar2Hampiran(K)
        fmt.Printf("Nilai hampiran akar 2 = %.10f\n", akar2)
    } else {
        fmt.Println("Pilihan tidak valid!")
    } }
```

#### Output

```

Pilih opsi:
1. Hitung f(K)
2. Hitung hampiran akar 2
Masukkan pilihan (1/2): 1
Masukkan nilai K: 100
Nilai f(K) = 1.0000061880
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "c:\Users\Lenovo\Downloads\UNGUIDED_2B_4.go"

Pilih opsi:
1. Hitung f(K)
2. Hitung hampiran akar 2
Masukkan pilihan (1/2): 2
Masukkan nilai K: 10
Nilai hampiran akar 2 = 1.4062058441
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "c:\Users\Lenovo\Downloads\UNGUIDED_2B_4.go"

Pilih opsi:
1. Hitung f(K)
2. Hitung hampiran akar 2
Masukkan pilihan (1/2): 2
Masukkan nilai K: 100
Nilai hampiran akar 2 = 1.413387072
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "c:\Users\Lenovo\Downloads\UNGUIDED_2B_4.go"

Pilih opsi:
1. Hitung f(K)
2. Hitung hampiran akar 2
Masukkan pilihan (1/2): 2
Masukkan nilai K: 1000
Nilai hampiran akar 2 = 1.4141252651
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1>

```

### Deskripsi Program

Program ini adalah program yang memungkinkan pengguna untuk memilih antara dua pilihan yaitu : menghitung  $f(k)$  atau menghitung hampiran akar 2. Program ini dapat digunakan dengan metode iteratif, metode iteratif adalah metode yang menghasilkan serangkaian perkiraan untuk menyelesaikan sebuah masalah.

### Unguided 2C\_1

```

package main
import
(
    "fmt"
)
func hitungBiayaKirim(beratGram int)
int {
    // Konversi berat ke kg dan sisa
    gram    beratKg := beratGram / 1000
    sisaGram := beratGram % 1000

    // Biaya dasar per kg
    biayaPerKg := 10000
    totalBiaya := beratKg * biayaPerKg

    // Ketentuan tambahan biaya untuk sisa
    berat    var sisaBiaya int    if beratKg > 10
    {
        // Jika total berat lebih dari 10kg, sisa berat digratiskan
        sisaBiaya = 0
    } else {
        if sisaGram >= 500 {
            // Jika sisa lebih dari atau sama dengan 500 gram
            sisaBiaya = sisaGram * 5
        } else {
            // Jika sisa kurang dari 500 gram
            sisaBiaya = sisaGram * 15
        }
    }
    totalBiaya += sisaBiaya
    return totalBiaya
}
func main() {
    var beratParsial int
    fmt.Print("Masukkan berat parsel dalam gram: ")
    _, err := fmt.Scan(&beratParsial)    if err != nil ||
    beratParsial < 0 {
        fmt.Println("Input tidak valid. Harap
        masukkan angka yang benar dan tidak negatif.")    return
    }
    biaya := hitungBiayaKirim(beratParsial)

    fmt.Printf("Biaya pengiriman untuk parsel seberat %d gram
    adalah: Rp. %d\n", beratParsial, biaya)
}

```

Output

```

PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\unguided2c_1.go"
Masukkan berat parcel dalam gram: 8500
Biaya pengiriman untuk parcel seberat 8500 gram adalah
: Rp. 82500
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\unguided2c_1.go"
Masukkan berat parcel dalam gram: 9250
Biaya pengiriman untuk parcel seberat 9250 gram adalah
: Rp. 93750
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\unguided2c_1.go"
Masukkan berat parcel dalam gram: 11750
Biaya pengiriman untuk parcel seberat 11750 gram adala
h: Rp. 110000
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> 

```

### Deskripsi program

Program ini menjelaskan tentang menghitung biaya berat parcel yang diinput oleh user. Jika user menginput berat parcel maka output secara langsung akan memperlihatkan biaya pengiriman pada output.

### Unguided 2C\_2

```

package main
import
"fmt"

func main() {
var nam float64
var nmk string

    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&nam)

    // Determine the letter grade based on the numeric
score    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)
}

```

## Output

```

PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\unguided2c_2.go"
Nilai akhir mata kuliah: 80.1
Nilai Mata Kuliah: A
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\unguided2c_2.go"
Nilai akhir mata kuliah: 93.5
Nilai Mata Kuliah: A
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\unguided2c_2.go"
Nilai akhir mata kuliah: 70.6
Nilai Mata Kuliah: B
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pe
rtemuan 1\unguided2c_2.go"
Nilai akhir mata kuliah: 49.5
Nilai Mata Kuliah: D
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemuan
1> 

```

## Deskripsi program

- Jika nilai 80.1 maka nilai mata kuliah yang muncul di output adalah A.
- Pada variabel 'nmk' dideklarasikan tetapi tidak pernah digunakan. Seharusnya variabel tersebut digunakan untuk menampung nilai akhir. Pada variabel 'if' tidak bisa karena tidak ada variabel 'else' seharusnya ada variabel 'else if' agar program bisa dijalankan
- Nilai 93.5 memperoleh A, Nilai 70.6 memperoleh B, Nilai 49.5 memperoleh C

## Unguided 2C\_3

```

package main
import
(
    "fmt"
)

// Fungsi untuk mencari faktor dari bilangan bulat b func
cariFaktor(b int) []int {

```

```

        var faktor []int          // Slice untuk menyimpan faktorfaktor
        for i := 1; i <= b; i++ { // Iterasi dari 1 hingga b          if
b%i == 0 { // Cek apakah i adalah faktor dari b          faktor
= append(faktor, i) // Tambahkan faktor ke slice
        }
    }
    return faktor
}
// Fungsi untuk menentukan apakah bilangan bulat b adalah bilangan
prima
func isBilanganPrima(b int) bool {
if b <= 1 {
    return false // Bilangan kurang dari atau sama dengan 1
    bukan prima    }
    for i := 2; i*i <= b; i++ { // Cek hingga akar kuadrat dari b
        if b%i == 0 {
            return false // Jika ada pembagi lain, bukan prima
        }
    }
    return true // Jika tidak ada pembagi lain, maka bilangan prima
} func
main() {
var b int

    // Input dari pengguna
    fmt.Print("Masukkan bilangan bulat b (b > 1): ")
    _, err := fmt.Scan(&b)
if err != nil || b <= 1 {
    fmt.Println("Input tidak valid. Harap masukkan bilangan
bulat yang lebih besar dari 1.")    return
}    faktorB :=
cariFaktor(b)

    // Menampilkan hasil faktor
    fmt.Printf("Faktor-faktor dari %d adalah: %v\n", b, faktorB)
    // Menentukan dan menampilkan apakah bilangan tersebut adalah
bilangan prima
    if isBilanganPrima(b) {
        fmt.Printf("%d adalah bilangan prima.\n", b)
    } else {
        fmt.Printf("%d bukan bilangan prima.\n", b)
    }
}

```

Output

```
Masukkan bilangan bulat b (b > 1): 12
Faktor-faktor dari 12 adalah: [1 2 3 4 6 12]
12 bukan bilangan prima.
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemu
an 1> go run "d:\SEMESTER 3\DATA FILE PRAKTIKUM ALP
RO2\pertemuan 1\unguided2c_3.go"
Masukkan bilangan bulat b (b > 1): 7
Faktor-faktor dari 7 adalah: [1 7]
7 adalah bilangan prima.
PS D:\SEMESTER 3\DATA FILE PRAKTIKUM ALPRO2\pertemu
an 1> 
```

### Deskripsi Program

Program ini dapat mencari factor-faktor dari sebuah bilangan bulat yang diinput oleh user dan menentukan apakah bilangan bulat tersebut adalah bilangan prima atau tidak.