

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

MODUL II

MATERI



Oleh:

AHMADAN SYARIDIN

2311102038

S1-IF11-02

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dalam kerangka program yang ditulis dalam bahasa pemrograman Go, program utama selalu mempunyai dua komponen berikut:

- package main merupakan penanda bahwa file ini berisi program utama.
- func main() berisi kode utama dari sebuah program Go.

Beberapa bahasa 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 kemudian mengubahnya menjadi program eksekutabel, sehingga konsistensi penulisan sudah diperiksa sebelum eksekusi.

Semua proses terkait bahasa Go dilakukan melalui utilitas `gk`, beberapa utilitas `go` :

- `Go build` : mengkompilasikan program sumber yang ada dalam folder menjadi sebuah program
- `Go build file.go` : mengkompilasi program sumber `file.go` saja
- `Go fmt` : membaca semua program sumber dalam folder dan mereformat penulisannya agar sesuai dengan standar penulisan program sumber `go`
- `Go clean` : membersihkan file-file dalam folder sehingga tersisa program sumber nya saja.

Variabel merupakan nama dari suatu Lokasi di memori, yang data dengan tipe tertentu dapat disimpan.

Nama variabel dimulai dengan huruf dan dapat diikuti dengan sejumlah 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 ⁹ ..10 ⁹ 64 bit: -10 ¹⁹ ..10 ¹⁹ bergantung platform 0..255 0..4294967295 0..(2 ⁶⁴ -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 di atas 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 per-bit 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 boolean AND, OR, dan NOT
* &	variabel apa saja	mendapatkan data dari lokasi memori dan mendapatkan lokasi dari variabel

Contoh

Operasi	Hasil
"non suffi" + "cit mundo"	"non sufficit mundo"
2019.01 + 1.0102	2020.0202
2020 / 20	22.22
20.2 * 1.1	101
2020 % 1999	21
2020 & 1111	2104
2020 ^ 1111	1663
2020 >> 2	505
"minutus" < "magnus"	false
2020 >= 1234	true
! false && true	true

Bahasa Go mempunyai tipe data yang ketat. Sehingga tipe data yang berbeda tidak bisa dicampur Dalam satu ekspresi, bahkan tipe data yang masih sejenis, misalnya nashi sama sama interger (Int dan Int32). Untuk menyesuaikan tipe data, terdapat beberapa cara yang bisa dilakukan.

Seperti casting, tipe data, Mengubah tipe dari data yang diberikan ke tipe yang diinginkan. Memanfaatkan fungsi print dan scan dari paket fmt. Dan memanfaatkan fungsi-fungsi Dalam pake strconv seperti atoi, itoa, dan parsebool.

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(a,"%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

Variabel harus dideklarasikan terlebih dahul sebelum digunakan, variabel juga harus diinisialisasi dulu agar nilai yang tersimpan diketahui dengan jelas dan eksekusi algoritma menjadi terprediksi. Dalam Bahasa Go, variabel yang tidak diinisialisasi lebih dahulu otomatis di isi dengan nilai default yang ekuivalen dengan bit 0

Notasi deklarasi variabel	Penulisan dalam Go	Keterangan
kamus a : tipe	var a tipe	a diinisialisasi dengan nilai default
kamus a : tipe algoritma a <- nilai_awal	var a tipe = nilai_awal var a = (tipe)nilai_awal	a diinisialisasi dengan nilai_awal
	a := nilai_awal a := (tipe)nilai_awal	secara implisit , tipe variabel a ditentukan dari nilai inisialisasinya

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

Konstanta dapat diberi nama untuk memudahkan mengingat maksud dan manfaat dari nilai yang diberi nama tersebut. Seperti PI untuk merepresentasikan konstanta π .

Kode contoh:

1 const PI = 3.1415926535897932384626433

2 const MARKER = "AKHIR"

II. GUIDED

Source Code

```
package main

import "fmt"

func main() {

    var (

        satu, dua, tiga string
```

```

        temp        string
    )

    fmt.Print("Masukan input string: ")

    fmt.Scanln(&satu)

    fmt.Print("Masukan input string: ")

    fmt.Scanln(&dua)

    fmt.Print("Masukan input string: ")

    fmt.Scanln(&tiga)

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

    temp = satu

    satu = dua

    dua = tiga

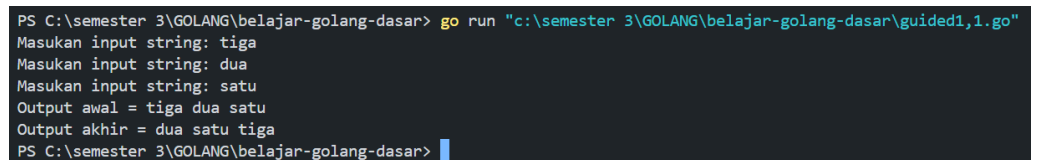
    tiga = temp

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

}

```

Screenshot hasil program



```

PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\guided1,1.go"
Masukan input string: tiga
Masukan input string: dua
Masukan input string: satu
Output awal = tiga dua satu
Output akhir = dua satu tiga
PS C:\semester 3\GOLANG\belajar-golang-dasar>

```

Deskripsi Program

2. Guided 2

```
package main

import "fmt"

func main() {

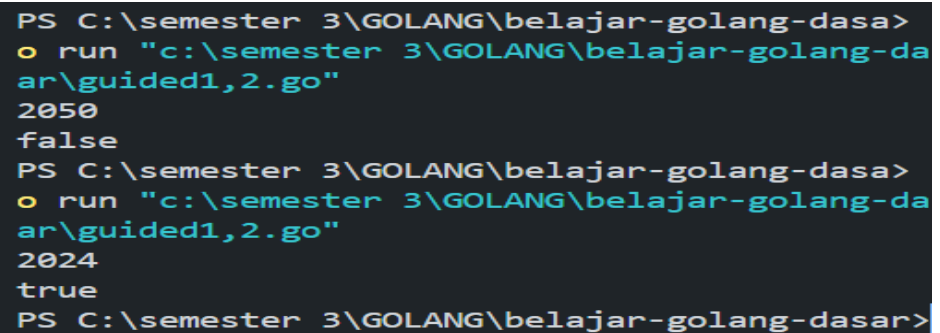
    year := 0

    fmt.Scan(&year)

    fmt.Println((year%4 == 0 && year%100 != 0) || (year%400 == 0))

}
```

Screenshoot



```
PS C:\semester 3\GOLANG\belajar-golang-dasa>
o run "c:\semester 3\GOLANG\belajar-golang-da
ar\guided1,2.go"
2050
false
PS C:\semester 3\GOLANG\belajar-golang-dasa>
o run "c:\semester 3\GOLANG\belajar-golang-da
ar\guided1,2.go"
2024
true
PS C:\semester 3\GOLANG\belajar-golang-dasar>
```

3. Guided 3

Source Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var r float64

    fmt.Print("Jejari = ")

    fmt.Scan(&r)

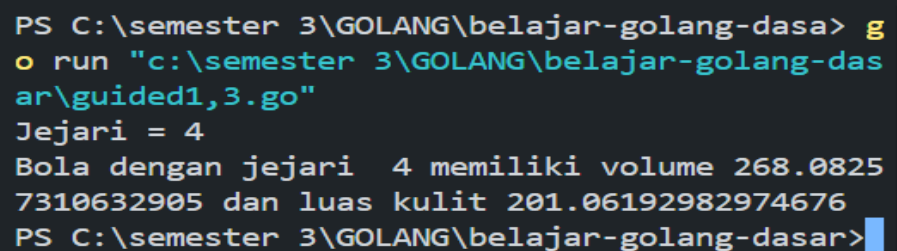
    volume := 4.0 / 3.0 * math.Pi * math.Pow(r, 3)

    luas := 4 * math.Pi * math.Pow(r, 2)

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

}
```

Srceenshot Program



```
PS C:\semester 3\GOLANG\belajar-golang-dasa> go
o run "c:\semester 3\GOLANG\belajar-golang-das
ar\guided1,3.go"
Jejari = 4
Bola dengan jejari 4 memiliki volume 268.0825
7310632905 dan luas kulit 201.06192982974676
PS C:\semester 3\GOLANG\belajar-golang-dasar>
```


III. UNGUIDED

1.2B1

Source Code

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    expectedColors := []string{ "merah", "kuning", "hijau", "ungu" }
    sanner := bufio.NewScanner(os.Stdin)
    totalTrials := 5
    var allTrials [][]string
    for i := 1; i <= totalTrials; i++ {
        fmt.Printf("Percobaan %d: ", i)
        scanner.Scan()
        inputColors := strings.Split(scanner.Text(), " ")
        allTrials = append(allTrials, inputColors) }
    success := true
    for _, trial := range allTrials {
        if !compareSlices(trial, expectedColors) {
```

```
    success = false
        break  }
    }
    if success {
        fmt.Println("BERHASIL: true")
    } else {
        fmt.Println("BERHASIL: false") }
}

func compareSlices(a, b []string) bool {
    if len(a) != len(b) {
        return false
    }
    for i := range a {
        if a[i] != b[i] {
            return false
        }
    }
    return true }
```

Screenshot hasil program

```
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run
"c:\semester 3\GOLANG\belajar-golang-dasar\unguided
1,1.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
Berhasil : true
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run
"c:\semester 3\GOLANG\belajar-golang-dasar\unguided
1,1.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
Percobaan 5: merah kuning hijau ungu
Berhasil : false
PS C:\semester 3\GOLANG\belajar-golang-dasar>
```

Deskripsi

Kode tersebut meminta Pengguna untuk memasukkan 4 string sebanyak 5 kali, untuk mencocokkan dengan hasil true jika urutan warna yang diberikan “merah kuning hijau ungu” , dengan program tidak langsung memeriksa saatu-satu melainkan akan memeriksa setelah semua hasil imputan selesai. Kalua semuanya benar, hasilnya true, Namun jika ada inputan yang salah maka hasilnya false.

2. Unguided 2B2

```
package main

import ( "bufio" "fmt" "os" )

func main() {

    scanner := bufio.NewScanner(os.Stdin)

    fmt.Print("N: ")

    var N int

    fmt.Scan(&N)

    if N == 0 {

        fmt.Println("Pita:")

        return }

    scanner.Scan()

    pita := ""

    for i := 1; i <= N; i++ {

        fmt.Printf("Bunga %d: ", i)

        scanner.Scan()

        namaBunga := scanner.Text()

        if i == 1 {

            pita += namaBunga } else {

                pita += " - " + namaBunga }

        }

        fmt.Println("Pita:", pita)

    }
```

Screenshot Program

```
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguided1,2.go"
N: 3
Bunga 1: kertas
Bunga 2: mawar
Bunga 3: tulip
Pita: kertas - mawar - tulip
PS C:\semester 3\GOLANG\belajar-golang-dasar> |
```

```
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguided1,2.go"
N: 0
Pita:
PS C:\semester 3\GOLANG\belajar-golang-dasar> |
```

Deskripsi

Program di atas berfungsi untuk meminta Pengguna menginputkan nama- nama bunga, dan menggabungkannya menjadi satu string yang disebut pita. Nama bunga akan di simpan pada variabel `namBunga`, lalu bunga pertama di masukan ke Dalam pita, dan bunga kedua atau lebih dimasukan ke pita dengan tanda pemisah “-“ pada antara nama bunga

Modifikasi program sebelumnya agar bisa input lebih dari 3 dan berhenti ketika mengetikkan selesai

Source code

```
package main

import ("bufio" "fmt" "os" "strings")

func main() {
    scanner := bufio.NewScanner(os.Stdin)

    pita := ""

    jumlahBunga := 0

    for {
        fmt.Printf("Bunga %d: ", jumlahBunga+1)
```

```
scanner.Scan()

    namaBunga := scanner.Text()

    if strings.ToUpper(namaBunga) == "SELESAI" {

        break

    } if jumlahBunga == 0 {

        pita += namaBunga

    } else {

        pita += " - " + namaBunga

    } jumlahBunga++

}

if jumlahBunga == 0 {

    fmt.Println("Pita:")

} else {

    fmt.Println("Pita:", pita)

    fmt.Println("Bunga:", jumlahBunga)

}

}
```

Screenshot program

```
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguided1,2.go"
Bunga 1: kertas
Bunga 2: mawar
Bunga 3: tulip
Bunga 4: selesai
Pita: kertas - mawar - tulip
Bunga: 3
PS C:\semester 3\GOLANG\belajar-golang-dasar>
```

Program di atas di modifikasi untuk meminta Pengguna berhenti jika Memasukkan “selesai” dengan namahbahkan fungsi `strings.ToUpper()` yang memastikan input di ubah menjadi kapital sebelum di bandingkan dengan string “selesai”

3. Unguided 2B3

```
package main

import ("fmt")

func main() {

    var kantong1, kantong2 float64

    for {

        fmt.Print("Masukan berat belanjaan di kedua kantong: ")

        fmt.Scan(&kantong1, &kantong2)

        if kantong1 >= 9 || kantong2 >= 9 {

            fmt.Println("Proses selesai.")

            break }

    } }
```

Screenshot Program

```

go run "c:\semester 3\GOLANG\belajar-golang-dasar\uguided1,3.go"
Masukan berat belanjaan di kedua kantong: 5.5 1.0
Masukan berat belanjaan di kedua kantong: 7.1 8.5
Masukan berat belanjaan di kedua kantong: 2 6
Masukan berat belanjaan di kedua kantong: 9 5.8
Proses selesai.
PS C:\semester 3\GOLANG\belajar-golang-dasar>

```

Deskripsi program

Program tersebut adalah program yang menerima input dua buah bilangan real positif yang menyatakan berat total masing masing isi kantong terpal. Menggunakan loop for yang terus berjalan sampai salah satu dari dua kantong mencapai berat 9kg atau lebih, kemudian Dalam loop meminta Pengguna memasukkan berat belanjaan di kedua kantong menggunakan fmt.Scan, jika lebih dari 9kg atau mencapainya maka akan menampilkan 'proses selesai'

Modifikasi program sebelumnya

```

package main

import (
    "fmt"
    "math"
)

func main() {
    var kantong1, kantong2 float64

    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scan(&kantong1, &kantong2)

        if kantong1 < 0 || kantong2 < 0 {

```



```

        fmt.Println("Proses selesai.")

        break
    } if kantong1+kantong2 > 150 {

        fmt.Println("Proses selesai.")

        break
    }

    selisih := math.Abs(kantong1 - kantong2)

    if selisih >= 9 {

        fmt.Println("Sepeda motor pak Andi akan oleng: true")

    } else { fmt.Println("Sepeda motor pak Andi akan oleng: false")

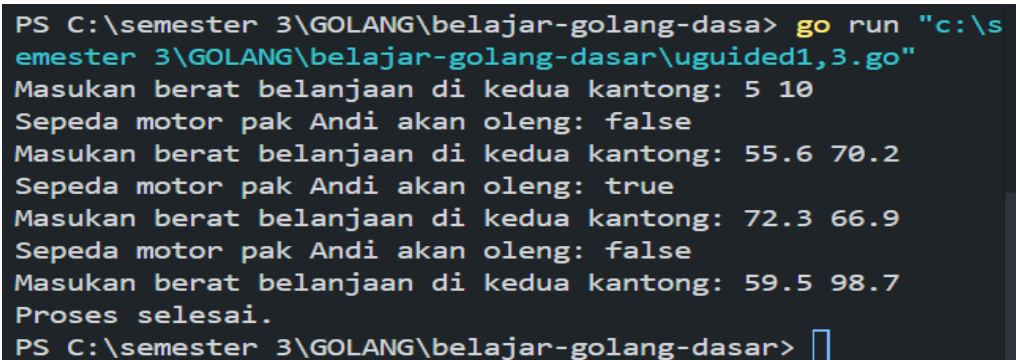
    }

}

}

```

Screenshots Program



```

PS C:\semester 3\GOLANG\belajar-golang-dasa> go run "c:\s
emester 3\GOLANG\belajar-golang-dasar\uguided1,3.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
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai.
PS C:\semester 3\GOLANG\belajar-golang-dasar>

```

Deskripsi

Program dibuat untuk meminta input berat barang yang di masukan ke dua kantong kertpal kanan kiri, yang memeriksa kesetabilan motor pak andi berdasarkan perbedaan berat antara kedua kanton. Dan program

akan berhenti ketika berat kedua kantong melebihi 150 kg atau berat salah satu bernilai negative

4.2B4

Source code

```
package main

import "fmt"

func main() {
    var k int

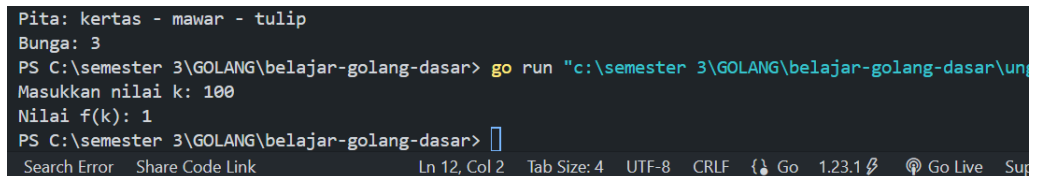
    fmt.Print("Masukkan nilai k: ")

    fmt.Scan(&k)

    hasil := (4*k + 2) * (4*k + 2) / ((4*k + 1) * (4*k + 3))

    fmt.Println("Nilai f(k):", hasil)
}
```

Screenshot program



```
Pita: kertas - mawar - tulip
Bunga: 3
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\ung
Masukkan nilai k: 100
Nilai f(k): 1
PS C:\semester 3\GOLANG\belajar-golang-dasar> 
```

Deskripsi program

Program meminta Pengguna input yang akan dibaca oleh `fmt.Scan(8k)` untuk di simpan nilai input pada variabel `k`. setelah input dan perhitungan selesai akan menampilkan dan mencetak nilai `f(K)`

Modifikasi program

Source code

```
package main

import (
    "fmt"
)

func main() {
    var K int

    fmt.Print("Masukkan nilai K: ")

    fmt.Scan(&K)

    hampiranAkar2 := 1.0

    for k := 0; k <= K; k++ {
        pembilang := (4*float64(k) + 2) * (4*float64(k) + 2)
        penyebut := (4*float64(k) + 1) * (4*float64(k) + 3)
        hampiranAkar2 *= pembilang / penyebut
    }

    fmt.Printf("Nilai hampiran  $\sqrt{2}$  untuk K = %d adalah: %.10f\n", K,
        hampiranAkar2)
}
```

Screenshot

```
Masukkan nilai K: 10
Nilai hampiran v2 untuk K = 10 adalah: 1.4062058441
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguided2C1.go"
Masukkan nilai K: 100
Nilai hampiran v2 untuk K = 100 adalah: 1.4133387072
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguided2C1.go"
exit status 0xc0000005
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguided2C1.go"
Masukkan nilai K: 1000
Nilai hampiran v2 untuk K = 1000 adalah: 1.4141252651
```

Deskripsi program

Program di atas telah di modifikasi dengan Menambahkan variabel hampiranAkar2, yang mulaia dengan nilai 1.0, dan setiap nilai k dai o hingga k, program menghitung nilai. Perhitungan menggunakan float64 untuk menjaga ketelitian oecahan. Dan menampilkan fungsi print fmt

5. Unguided 2C1

```
package main

import (
    "fmt"
)

func hitungBiayaPos(berat int) (int, string, int) {
    kg := berat / 1000
    sisaGram := berat % 1000
    biayaPerKg := 10000
    biayaTambahan := 0
    totalBiaya := kg * biayaPerKg
    if sisaGram > 500 {
        biayaTambahan = sisaGram * 15
    } else if sisaGram > 0 {
        biayaTambahan = sisaGram * 5 /
```

```

    } if kg >= 10 {

        biayaTambahan = 0

    }

    totalBiaya += biayaTambahan

    detailBerat := fmt.Sprintf("%d kg + %d gr", kg, sisaGram)

    return totalBiaya, detailBerat, biayaTambahan

}

func main() {

    var berat int

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

    fmt.Scan(&berat)

    totalBiaya, detailBerat, biayaTambahan := hitungBiayaPos(berat)

    fmt.Println("Detail berat:", detailBerat)

    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", (totalBiaya -
biayaTambahan), biayaTambahan)

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

}

```

Screenshor program

```

PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguidedC1,1.go"
# command-line-arguments
.\unguidedC1,1.go:21:2: syntax error: unexpected }, expected expression
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguidedC1,1.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguidedC1,1.go"
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 1250
Total biaya: Rp. 91250
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguidedC1,1.go"
go: error obtaining buildID for go tool asm: exit status 0xc0000005
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguidedC1,1.go"
Berat parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS C:\semester 3\GOLANG\belajar-golang-dasar>

```

Deskripsi Program

Program ini digunakan untuk menghitung biaya pengiriman parcel berdasarkan berat dalam gram. Program akan mengonversi berat ke dalam kilogram dan gram, menghitung biaya dasar per kilogram, serta biaya tambahan untuk sisa gram yang tidak sampai 1 kg. Ada juga ketentuan diskon untuk parcel yang beratnya 10 kg atau lebih.

6. Unguided 2C2

Source code

```
package main

import (
    "fmt"
)

func main() {
    var nam float64
    var nmk string

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

    if nam > 80 {
        nmk = "A"
    } else if nam > 72.5 {
        nmk = "AB"
    } else if nam > 65 {
        nmk = "B"
    }
```

```

    } else if nam > 57.5 {

        nmk = "BC"

    } else if nam > 50 {

        nmk = "C"

    } else if nam > 40 {

        nmk = "D"

    } else {

        nmk = "E"

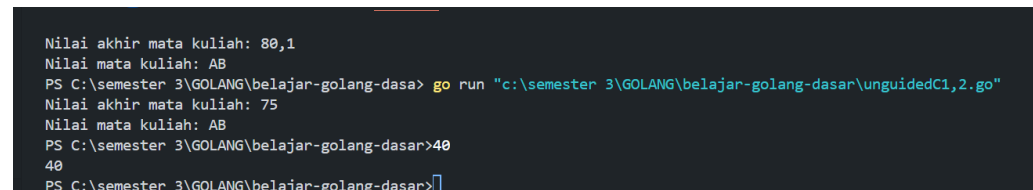
    }

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

}

```

Screenshot



```

Nilai akhir mata kuliah: 80,1
Nilai mata kuliah: AB
PS C:\semester 3\GOLANG\belajar-golang-dasa> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguidedC1,2.go"
Nilai akhir mata kuliah: 75
Nilai mata kuliah: BC
PS C:\semester 3\GOLANG\belajar-golang-dasar> 40
D
PS C:\semester 3\GOLANG\belajar-golang-dasar>

```

Deskripsi

Program menerima input nilai akhir mat kuuliah dalambentuk angka desimal dan engkonversikan nilai menjadi huruf (A, AB, B, BC, C, D, D). Kemudian huruf yang diinputkan akan keluar sesuai nilai yang di inputkan

7.Unguided 3C3

Source code

```
package main

import (
    "fmt"
)

func main() {
    var b int
    var count int

    fmt.Print("Masukkan bilangan bulat: ")

    fmt.Scan(&b)

    fmt.Printf("Bilangan: %d\n", b)

    fmt.Print("Faktor: ")

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

            count++
        }
    }

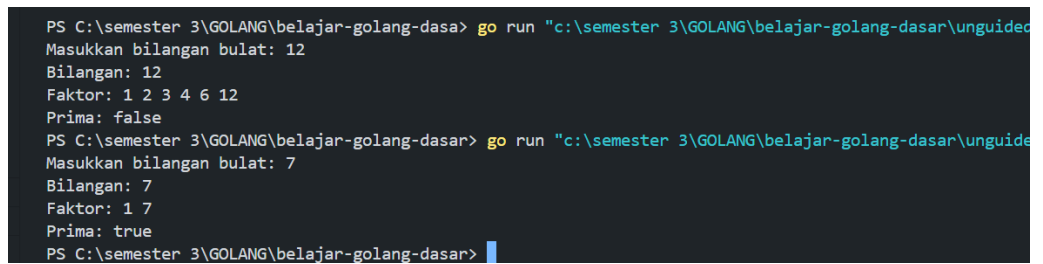
    fmt.Println()

    if count == 2 {
        fmt.Println("Prima: true")
    }
}
```



```
    } else {  
        fmt.Println("Prima: false")  
    }  
}
```

Screenshot



```
PS C:\semester 3\GOLANG\belajar-golang-dasa> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguided  
Masukkan bilangan bulat: 12  
Bilangan: 12  
Faktor: 1 2 3 4 6 12  
Prima: false  
PS C:\semester 3\GOLANG\belajar-golang-dasar> go run "c:\semester 3\GOLANG\belajar-golang-dasar\unguide  
Masukkan bilangan bulat: 7  
Bilangan: 7  
Faktor: 1 7  
Prima: true  
PS C:\semester 3\GOLANG\belajar-golang-dasar>
```

Deskripsi program

Program ini menerima input bilangan bulat dari pengguna, kemudian menampilkan faktor-faktor dari bilangan tersebut serta memeriksa apakah bilangan itu merupakan bilangan prima. Jika bilangan hanya memiliki dua faktor (1 dan dirinya sendiri), maka program menyatakan bahwa bilangan tersebut adalah prima.