

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 2
REVIEW STRUKTUR KONTROL**



Oleh:

YAYANG ALYA BILQIS

2311102229

IF 11 02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

- **DATA DAN VARIABEL**

Variabel merupakan nama dari suatu lokasi di memori dengan tipe tertentu Dimana nama variable akan dimulai dengan huruf dan dapat diikuti dengan sejumlah huruf lainnya, maupun angka, atau garis bawah.

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

- Nilai data yang tersimpan dapat diperoleh dengan menyebutkan langsung nama variabelnya
 - Informasi Alamat dapat diperoleh untuk menyimpan data pada found dengan catatan jika variable berisi Alamat memori, prefix* pada variable tersebut akan memberikan nilai yang tersimpan dalam memori yang lokasinya juga disimpan di variable yang sama.
- Bahasa GO menganut kesesuaian tipe data yang ketat Dimana tipe data yang berbeda tidak boleh dicampur dalam satu ekspresi misal; sama sama integer.
 - Variabel harus dideklarasikan dulu sebelum digunakan dan harus diinisialisasi (diisi data terlebih dahulu) agar nilai yang tersimpan diketahui dengan jelas dan eksekusi algoritma menjadi terprediksi. Variabel yang tidak

diinisialisasi terlebih dahulu otomatis akan memperoleh nilai ekuivalen 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

- Intruksi dasar

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.Scanf("%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 Simbolik

Konstanta bisa diberi nama agar memudahkan mengingat maksud dan manfaat nilai yang dimasukkan nama tersebut.

II. GUIDED

1. GUIDED 1

```
package main

import "fmt"

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

```

                                temp                string

    )

    fmt.Print("Masukkan input string: ")
    fmt.ScanIn(&satu)
    fmt.Print("Masukkan input string: ")
    fmt.ScanIn(&dua)
    fmt.Print("Masukkan input string: ")
    fmt.ScanIn(&tiga)
    fmt.Println("Output awal = " + satu + " " + dua + " " +
tiga)

    temp = satu
    satu = dua
    dua = tiga
    tiga = temp

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

```

Deskripsi Program

Program tersebut diminta menginputkan tiga input string dari user dengan menampilkan urutan awal dari input tersebut dan kemudian melakukan pertukaran urutan string Dimana variable satu, dua, tiga digunakan untuk menyimpan input string dari pengguna dan variable temp digunakan sebagai tempat penyimpanan sementara. Lalu program akan meminta user untuk memasukkan string tiga kali dan pertukaran dengan memindahkan nilai

GUIDED 2

```

package main

import "fmt"

```

```

func main() {
    year := 0
    fmt.Print("Tahun: ")
    fmt.Scan(&year)
    fmt.Println("Kabisat: ", year%4 == 0 && year%100
!= 0)
}

```

Deskripsi Program

Program ini berfungsi untuk menentukan tahun kabisat dimana user diminta untuk memasukkan tahun yang ingin dicek kemudian program akan menggunakan rumus sederhana untuk menentukan apakah tahun tersebut habis dibagi 4 namun tidak habis dibagi 100. Jika kondisi ini terpenuhi, maka tahun tersebut merupakan tahun kabisat.

UNGUIDED 3

```

package main

import (
    "fmt"
    "math"
)

func main() {
    var jari float64
    fmt.Print("Jejari = ")
    fmt.Scanln(&jari)
    volume := (4.0 / 3.0) * math.Pi * math.Pow(jari,
3)
    luas := 4 * math.Pi * math.Pow(jari, 2)
}

```

```
        fmt.Printf("Bola dengan jejari %v memiliki volume\n%.4f dan luas kulit %.4f", jari, volume, luas)\n    }\n}
```

Screenshot Program

```
Jejari = 8\nBola dengan jejari 8 memiliki volume 2144.6606 dan luas kulit 804.2477\nPS C:\\Users\\ASUS Vivobook> █
```

Deskripsi Program

Program ini berfungsi untuk menghitung volume dan luas permukaan bola berdasarkan jari-jari yang diberikan oleh user. User diminta untuk memasukkan nilai jari-jari bola, kemudian program akan menghitung volume bola menggunakan rumus $\frac{4}{3} * \pi * r^3$ dan luas permukaan bola menggunakan rumus $4 * \pi * r^2$

III. UNGUIDED

UNGUIDED 1

```
package main\n\nimport (\n    \"fmt\"\n    \"strings\"\n)\n\nfunc cekPercobaan(percobaan [5][4]string) bool {\n    expected := [4]string{\"merah\", \"kuning\", \"hijau\", \"ungu\"}\n\n    for i := 0; i < 5; i++ {\n
```

```

        for j := 0; j < 4; j++ {
            if percobaan[i][j] != expected[j] {
                return false
            }
        }
    }
    return true
}

func main() {
    percobaan1 := [5][4]string{
        {"merah", "kuning", "hijau", "ungu"},
        {"merah", "kuning", "hijau", "ungu"},
        {"merah", "kuning", "hijau", "ungu"},
        {"merah", "kuning", "hijau", "ungu"},
        {"merah", "kuning", "hijau", "ungu"},
    }

    percobaan2 := [5][4]string{
        {"merah", "kuning", "hijau", "ungu"},
        {"merah", "kuning", "hijau", "ungu"},
        {"merah", "kuning", "hijau", "ungu"},
        {"kuning", "merah", "hijau", "ungu"},
        {"merah", "kuning", "hijau", "ungu"},
    }

    fmt.Println("Percobaan 1:")
    for i, p := range percobaan1 {
        fmt.Printf("Percobaan %d: %s\n", i+1,
strings.Join(p[:], " "))
    }
}

```

```
    }

    if cekPercobaan(percobaan1) {
        fmt.Println("BERHASIL: true")
    } else {
        fmt.Println("BERHASIL: false")
    }

    fmt.Println("\nPercobaan 2:")
    for i, p := range percobaan2 {
        fmt.Printf("Percobaan %d: %s\n", i+1,
strings.Join(p[:], " "))
    }
    if cekPercobaan(percobaan2) {
        fmt.Println("BERHASIL: true")
    } else {
        fmt.Println("BERHASIL: false")
    }
}
```

Screenshot Program


```
Percobaan 1:
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

Percobaan 2:
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: kuning merah hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: false
```

Deskripsi Program

Program ini akan memeriksa apakah suatu rangkaian percobaan warna sesuai dengan pola yang telah ditentukan atau tidak. Jika semua warna pada semua baris sesuai dengan pola, maka fungsi akan menjadi true (berhasil) dan apabila warna tidak sesuai maka hasilnya akan false (gagal).

UNGUIDED 2

```
package main

import "fmt"

func main() {
    var n int
    var bunga, pita string

    fmt.Print("N: ")
    fmt.Scan(&n)
```

```

        for i := 1; i <= n; i++ {
            fmt.Printf("Bunga %d: ", i)
            fmt.Scan(&bunga)
            pita += bunga + " - "
        }

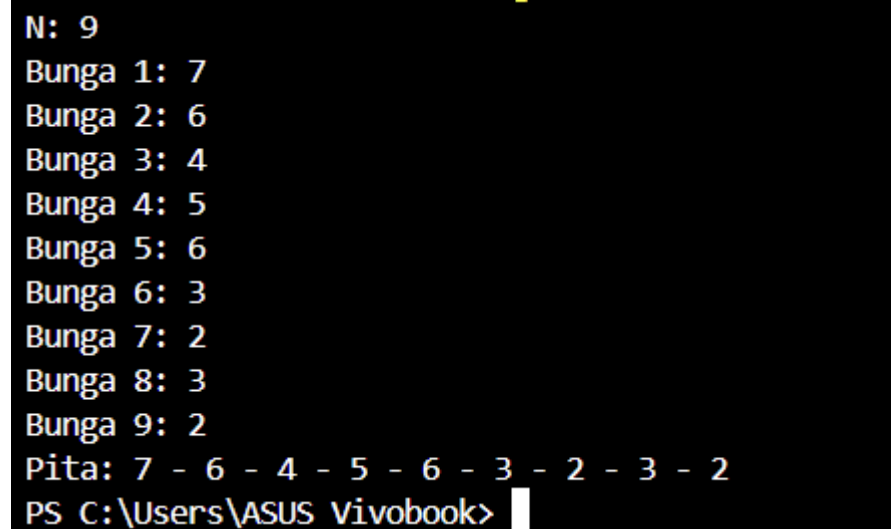
        // Hapus spasi dan tanda "-" berlebih di akhir
pita

        pita = pita[:len(pita)-3]

        fmt.Println("Pita:", pita)
    }

```

Screenshot Program



```

N: 9
Bunga 1: 7
Bunga 2: 6
Bunga 3: 4
Bunga 4: 5
Bunga 5: 6
Bunga 6: 3
Bunga 7: 2
Bunga 8: 3
Bunga 9: 2
Pita: 7 - 6 - 4 - 5 - 6 - 3 - 2 - 3 - 2
PS C:\Users\ASUS Vivobook>

```

Deskripsi Program

Pada program ini, user akan diminta untuk memasukkan jumlah bunga yang ingin dimasukkan (nilai N). kemudian, program ini akan meminta user memasukkan nama setiap bunga secara berurutan. Program ini

bertujuan sebagai pengumpulan data nama bunga dan menyajikannya dalam bentuk sebuah kalimat yang terhubung dengan tanda penghubung.

UNGUIDED 3

```
package main

import "fmt"

func main() {
    var beratKantong1, beratKantong2 float64

    for {
        fmt.Print("Masukkan berat belanjaan di kedua kantong: ")
        fmt.Scan(&beratKantong1, &beratKantong2)

        if beratKantong1 >= 9 || beratKantong2 >= 9 || beratKantong1+beratKantong2 > 150 || beratKantong1 < 0 || beratKantong2 < 0 {
            break
        }

        akanOleng := false
        selisihBerat := beratKantong1 - beratKantong2
        if selisihBerat < 0 {
            selisihBerat = -selisihBerat
        }
        if selisihBerat >= 9 {
            akanOleng = true
        }
    }
}
```

```

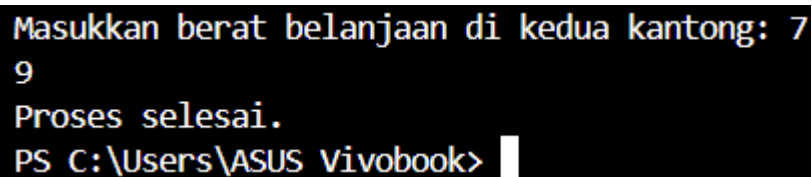
        fmt.Printf("Sepeda motor pak Andi akan
        oleng: %t\n", akanOleng)

    }

    fmt.Println("Proses selesai.")
}

```

Screenshot Program



```

Masukkan berat belanjaan di kedua kantong: 7
9
Proses selesai.
PS C:\Users\ASUS Vivobook>

```

Deskripsi Program

Program akan meminta user memasukkan berat kedua kantong hingga salah satu atau kedua kantong. Program akan menghitung selisih berat antara kedua kantong dan menentukan apakah sepeda motor akan oleng apabila selisih atau lebih besar atau sama dengan 9kg.

UNGUIDED 4

```

package main

import (
    "fmt"
    "math"

)

func f(k int) float64 {
    return float64((4*k+2)*(4*k+2)) /
    float64((4*k+1)*(4*k+3))
}

```

```

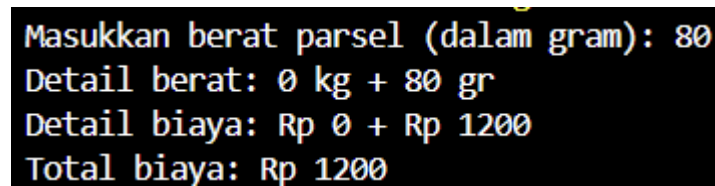
func akar2(k int) float64 {
    var hasil float64 = 1
    for i := 0; i <= k; i++ {
        hasil *= f(i)
    }
    return hasil
}

func main() {
    var k int
    fmt.Print("Masukkan nilai k: ")
    fmt.Scan(&k)

    fmt.Printf("Nilai f(%d) = %.10f\n", k, f(k))
    fmt.Printf("Nilai akar 2 = %.10f\n", akar2(k))
}

```

Screenshot Program



```

Masukkan berat parsel (dalam gram): 80
Detail berat: 0 kg + 80 gr
Detail biaya: Rp 0 + Rp 1200
Total biaya: Rp 1200

```

Deskripsi Program

Program ini berfungsi untuk menghitung nilai fungsi $f(x)$ dan mendekati nilai akar 2 dengan menggunakan rumus tertentu yang didefinisikan sebagai hasil bagi dari dua suku polynomial. Program akan meminta user untuk memasukkan nilai k kemudian menghitung dan menampilkan nilai $f(k)$ dan nilai yang paling mendekati akar 2

UNGUIDED 5

```
package main
```

```
import "fmt"

func main() {
    var beratParsel int

    fmt.Print("Masukkan berat parsel (dalam gram):")
    fmt.Scan(&beratParsel)

    kg := beratParsel / 1000
    gram := beratParsel % 1000

    biayaDasar := kg * 10000

    var biayaTambahan int
    if gram >= 500 {
        biayaTambahan = gram * 5
    } else {
        biayaTambahan = gram * 15
    }

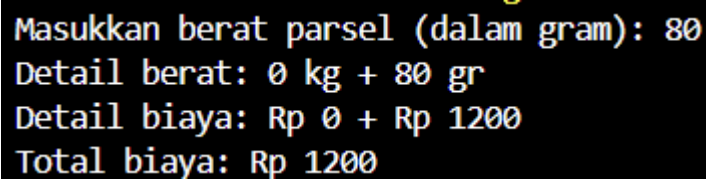
    if kg > 10 {
        biayaTambahan = 0
    }

    totalBiaya := biayaDasar + biayaTambahan

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

```
        fmt.Printf("Detail biaya: Rp %d + Rp %d\n",  
        biayaDasar, biayaTambahan)  
  
        fmt.Printf("Total biaya: Rp %d\n", totalBiaya)  
  
    }
```

Screenshot Program



Masukkan berat parcel (dalam gram): 80
Detail berat: 0 kg + 80 gr
Detail biaya: Rp 0 + Rp 1200
Total biaya: Rp 1200

Deskripsi Program

Program akan meminta user untuk memasukkan berat parcel dalam satuan gram kemudian dikonversi kedalam kilogram dan gram untuk menghitung biaya dasar berdasarkan berat yang diinput oleh user. Program akan menampilkan rincian berat, biaya, dan biaya pengiriman.

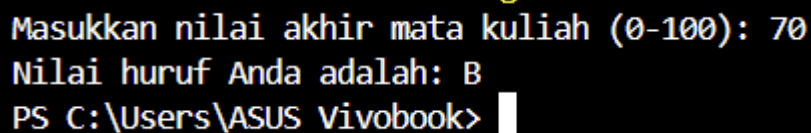
UNGUIDED 6

```
package main  
  
import "fmt"  
  
func main() {  
    var nilaiAkhir float64  
  
    fmt.Print("Masukkan nilai akhir mata kuliah (0-  
100): ")  
    fmt.Scan(&nilaiAkhir)  
  
    var nilaiHuruf string  
  
    switch {
```

```
        case nilaiAkhir >= 80:
            nilaiHuruf = "A"
        case nilaiAkhir >= 72.5:
            nilaiHuruf = "AB"
        case nilaiAkhir >= 65:
            nilaiHuruf = "B"
        case nilaiAkhir >= 57.5:
            nilaiHuruf = "BC"
        case nilaiAkhir >= 50:
            nilaiHuruf = "C"
        case nilaiAkhir >= 40:
            nilaiHuruf = "D"
        default:
            nilaiHuruf = "E"
    }

    fmt.Printf("Nilai huruf Anda adalah: %s\n",
nilaiHuruf)
}
```

Screenshot Program



```
Masukkan nilai akhir mata kuliah (0-100): 70
Nilai huruf Anda adalah: B
PS C:\Users\ASUS Vivobook>
```

Deskripsi Program

Program ini akan mengkonversikan nilai akhir mata kuliah menjadi nilai huruf sesuai dengan skala yang umum. Program akan meminta user untuk memasukkan nilai akhir mata kuliah dalam rentang nilai 0-100. Lalu program akan mengeksekusikan nilai akhir tersebut untuk setiap nilai huru. Program akan menampilkan nilai huruf yang sesuai dengan nilai angka yang diinput oleh user.

UNGUIDED 7

```
package main

import "fmt"

func findFactors(num int) []int {
    var factors []int
    for i := 1; i <= num; i++ {
        if num%i == 0 {
            factors = append(factors, i)
        }
    }
    return factors
}

func isPrime(num int) bool {
    if num <= 1 {
        return false
    }
    for i := 2; i*i <= num; i++ {
        if num%i == 0 {
            return false
        }
    }
    return true
}

func main() {
    var number int
```

```
        for {
            fmt.Print("Masukkan bilangan (0 untuk
berhenti): ")

            fmt.Scan(&number)

            if number == 0 {
                break
            }

            factors := findFactors(number)
            fmt.Println("Faktor:", factors)

            if isPrime(number) {
                fmt.Println("Prima: true")
            } else {
                fmt.Println("Prima: false")
            }
        }
    }
```

Deskripsi Program

Program ini dirancang untuk mencari factor dan juga menentukan apakah sebuah bilangan adalah bilangan prima atau bukan dimana dari bilangan yang diberikan akan memeriksa apakah bilangan tersebut habis dibagi oleh angka dari 1. Program akan meminta user untuk memasukkan bilangan secara berulang hingga pengguna memasukkan 0 untuk berhenti.