

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

MODUL X

MATERI



Oleh:

Destia Ananda Putra

2311102176

S1IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Struktur Program GO

- Dalam kerangka program yang ditulis dalam Bahasa pemograman 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.
Komentar, bukan bagian dari kode program, dan dapat ditulis di mana saja di dalam program:
 - Satu baris teks yang diawali dengan garis miring ganda (“//”) s.d. akhir baris, atau
 - Beberapa baris teks yang dimulai dengan pasangan karakter ‘/*’ dan diakhiri dengan ‘/*’.
- 1) Koding, Kompilasi, dan Eksekusi Go Koding
- Tidak berbeda dengan penulisan program sumber dalam Bahasa lain, program go harus dibuat menggunakan penyuting teks dan disimpan dalam format teks, bukan dalam format dokumen.
 - Setiap program go disimpan dalam file teks dengan ekstensi “nama_file.go” dengan nama bebas. Sebaiknya nama file adalah nama untuk program tersebut.
 - Setiap satu program lengkap Go disimpan dalam satu folder tersendiri. Nama folder merupakan nama program tersebut. Karena itu secara prinsip, satu program Go dapat dipecah dalam beberapa file dengan ekstensi “nama_file.go” selama disimpan dalam folder yang sama.

Komplikasi

Beberapa Bahasa pemograman 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 konsisten penulisan (seperti penggunaan tipe data) sudah diperiksa sebelum eksekusi. Selain itu karena program dibuat menjadi eksekutabel lebih dahulu, proses optimasi dapat dilakukan sehingga program menjadi sangat efisien.

Go diimplementasikan sebagai kompilator. Berikut adalah contoh sesi yang biasa dilakukan saat mengkompilasi dan mengeksekusi program dalam Bahasa Go:

- Panggil shell atau terminal (program utiliti cmd.exe di Windows)
- Masuk ke dalam (cd) folder program
- Kemudian panggil perintah `GO build` atau `GO build file.go` untuk mengkompilasi file.go
- Jika gagal, akan muncul pesan error yang sesuai, pelajari dengan baik pesan tersebut, perbaiki teks program sumber, kemudian ulangi proses build-nya.
- Jika berhasil maka pada folder tersebut akan dibuat program dengan nama yang sama dan diakhiri dengan .exe (untuk Windows)
- Panggil program eksekutabel tersebut dari terminal yang sama. Jangan memanggil program tersebut dengan mengklik eksekutabel tersebut dari folder karena program kalian hanya berbasis teks, bukan/belum dirancang dengan tampilan Windows.

2). Latihan

1. Selidiki bahasa-bahasa pemrograman berikut, apakah termasuk diinterpretasi, dikompilasi, dikompilasi (ke instruksi perantara) kemudian diinterpretasi:

- Pascal
- C dan C++
- Java
- Python

2. Unduh kompilator GO di perangkat yang anda gunakan. Kemudian salin contoh program di atas ke dalam folder `C:\user\userid\go\hello\hello.go` yaitu buat folder go dalam direktori home anda, kemudian buat subfolder hello dan taruh file hello.go di dalamnya. Hidupkan terminal (cmd.exe), dan panggil `go build` ke dalam folder tersebut. Periksa apakah hello.exe muncul di folder tersebut, jika iya coba eksekusi program tersebut, juga melalui terminal cmd.exe tersebut.

2.2 Tipe Data dan Instruksi Dasar

Data Variabel

- Nama Variabel
- Tipe Data
- Nilai Data
- Informasi Alamat
- Operasi yang dapat dilakukan pada tipe data di atas
- Bahasa Go memiliki kesesuaian tipe data yang ketat
- Variabel harus dideklarasikan dahulu sebelum digunakan

II. Guided-1

Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data yang sesuai sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut?

Source Code

```
package main

import "fmt"

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

    fmt.Print("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

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Guided\tempCodeRunnerFile.go"
Masukkan input string: Destia
Masukkan input string: Ananda
Masukkan input string: Putra
Output awal = Destia Ananda Putra
Output akhir = Ananda Putra Destia
PS C:\Users\Lenovo\Documents\Guided>

```

Penjelasan:

Program ini berfungsi untuk menerima tiga input string dari pengguna, lalu menampilkan hasil sebelum dan sesudah melakukan pertukaran nilai string. Awalnya, program meminta pengguna memasukkan tiga string dan menampilkan hasil awalnya. Setelah pertukaran, program menampilkan hasil akhir dengan string yang telah ditukar posisinya.

Guided 2

Tahun kabisat adalah tahun yang habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100. Buatlah sebuah program yang menerima input sebuah bilangan bulat dan memeriksa apakah bilangan tersebut merupakan tahun kabisat (true) atau bukan (false).

Source Code

```

package main

import "fmt"

func main() {
    var tahun int
    fmt.Print("Tahun : ")
    fmt.Scan(&tahun)
    fmt.Printf("Kabisat : %t \n", (tahun%4 == 0 && (tahun%100 != 0

```

```
|| tahun%400 == 0)))  
}
```

Output

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Guided\Guided2.go"  
Tahun : 2000  
Kabisat : true  
PS C:\Users\Lenovo\Documents\Guided> |
```

Penjelasan

Program akan menampilkan "Kabisat true" jika tahun tersebut adalah tahun kabisat, atau "Kabisat: false" jika bukan. Program ini digunakan untuk memeriksa apakah suatu tahun merupakan tahun kabisat atau bukan. Program meminta pengguna memasukkan tahun, kemudian menggunakan fungsi `tahunKabisat` untuk menentukan apakah tahun tersebut memenuhi kriteria tahun kabisat, yaitu habis dibagi 400, atau habis dibagi 4 tetapi tidak habis dibagi 100.

Guided 3

```
package main  
  
import (  
    "fmt"  
    "math"  
)  
  
func main() {  
  
    var jari, volume, luasKulit float64  
    fmt.Print("Jari = ")  
    fmt.Scan(&jari)  
    volume = math.Pi * (4.0 / 3.0) * math.Pow(jari, 3)  
    luasKulit = 4 * math.Pi * math.Pow(jari, 2)  
    fmt.Printf("Bola dengan jari %v memiliki volume %.4f dan luas  
kulit %.4f \n", jari, volume, luasKulit)  
}
```

Output

```
PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Guided\Guided3.go"
Jejari = 6
Bola dengan jejari 6 memiliki volume 904.7787 dan luas kulit 452.3893
PS C:\Users\Lenovo\Documents\Guided> █
```

Penjelasan

Program ini menghitung **volume** dan **luas permukaan** bola berdasarkan jari-jari yang dimasukkan oleh pengguna, lalu menampilkan hasilnya dalam format desimal. Program meminta pengguna memasukkan nilai jari-jari bola, lalu menggunakan dua fungsi, `hitungVolume()` dan `hitungLuasKulit()`, untuk menghitung volume dan luas kulit bola menggunakan rumus matematika yang sesuai. Hasil perhitungan kemudian ditampilkan dengan format dua angka di belakang koma.

III. Unguided 2B

Unguided 1

```
package main

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

func main() {
    // Urutan warna yang benar
    correctOrder := []string{"merah", "kuning", "hijau", "ungu"}
    reader := bufio.NewReader(os.Stdin)

    for {
        var hadError bool

        // Membaca input untuk 5 percobaan
        for i := 1; i <= 5; i++ {
            fmt.Printf("Percobaan %d: ", i)

            // Membaca input dari pengguna
```

```

    input, _ := reader.ReadString('\n')
    input = strings.TrimSpace(input)

    // Memisahkan input berdasarkan spasi
    colors := strings.Split(input, " ")

    // Mengecek apakah urutan warna sesuai
    for j := 0; j < 4; j++ {
        if j >= len(colors) || colors[j] != correctOrder[j]
{
            hadError = true
            break
        }
    }

    // Menampilkan hasil
    if !hadError {
        fmt.Println("BERHASIL: true")
    } else {
        fmt.Println("BERHASIL: false")
    }

    // Menanyakan pengguna apakah ingin mencoba lagi
    fmt.Print("Apakah Anda ingin mencoba lagi? (ya/tidak): ")
    retry, _ := reader.ReadString('\n')
    retry = strings.TrimSpace(retry)

    if strings.ToLower(retry) != "ya" {
        break // Keluar dari perulangan jika pengguna tidak
ingin mencoba lagi
    }
}

fmt.Println("Terima kasih telah bermain!")
}

```


Output

```
PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Destia Ananda Putra 2311102176\Destia Ananda Putra_2311102176
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
Apakah Anda ingin mencoba lagi? (ya/tidak): ya
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
Apakah Anda ingin mencoba lagi? (ya/tidak): tidak
Terima kasih telah bermain!
PS C:\Users\Lenovo\Documents\Guided> █
```

Penjelasan

Program menentukan secara acak sebuah urutan warna tertentu Kode ini digunakan untuk menguji apakah pengguna dapat memasukkan urutan warna merah, kuning, hijau, ungu dalam lima percobaan. Setiap percobaan, pengguna diminta memasukkan empat warna, dan program memeriksa apakah urutannya cocok dengan urutan yang benar. Jika semua lima percobaan berhasil, program menampilkan

Unguided 2

```
package main

import "fmt"

func main() {

    var pita string

    var bunga string

    var banyak int

    fmt.Print("Masukkan jumlah (N) : ")

    fmt.Scan(&banyak)

    for i := 0; i < banyak; i++ {
```

```
    fmt.Print("Bunga ", i+1, " : ")

    fmt.Scan(&bunga)

    pita += bunga + " - "

}

fmt.Println("Pita : ", pita)

}
```

Output

```
PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\o.go"
Masukkan jumlah (N) : 3
Bunga 1 : Kertas
Bunga 2 : mawar
Bunga 3 : tulip
Pita : Kertas - mawar - tulip -
```

Penjelasan

Program ini meminta pengguna memasukkan jumlah bunga, lalu nama setiap bunga secara berurutan. Nama-nama bunga tersebut digabungkan menjadi satu string dengan tanda pemisah " - ". Hasil akhir adalah string yang berisi semua nama bunga yang dimasukkan, dipisahkan oleh tanda tersebut. Jika pengguna mengetik "selesai", program akan berhenti menerima input. Setelah semua bunga dimasukkan, program menampilkan daftar bunga yang disusun dalam satu pita serta jumlah total bunga yang dimasukkan.

Unguided 3

```
package main

import "fmt"

func main() {
    for {
        var berat1, berat2 float64

        fmt.Print("Masukkan berat belanjaan di kedua kantong: ")
        _, err := fmt.Scan(&berat1, &berat2)
        if err != nil {
            fmt.Println("Input tidak valid.")
            return
        }

        if berat1+berat2 > 150 || berat1 < 0 || berat2 < 0 {
            fmt.Println("Proses selesai.")
            break
        }

        selisih := berat1 - berat2
        if selisih < 0 {
            selisih = -selisih
        }
        akanOlang := selisih >= 9

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

Output

```
PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Destia And
o"
Masukkan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukkan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai.
PS C:\Users\Lenovo\Documents\Guided> █
```

Penjelasan

Program ini meminta pengguna untuk memasukkan berat belanjaan di dua kantong. Kemudian, program mengecek apakah motor Pak Andi akan oleng berdasarkan perbedaan berat antara kedua kantong. jumlah berat dari kedua kantong lebih dari 150 kg atau salah satu berat negatif, program berhenti dengan menampilkan "Proses perbedaan berat antara dua kantong 9 kg atau lebih, program menampilkan bahwa motor "akan oleng" (True). Jika perbedaannya kurang dari 9 kg, hasilnya "tidak akan oleng.

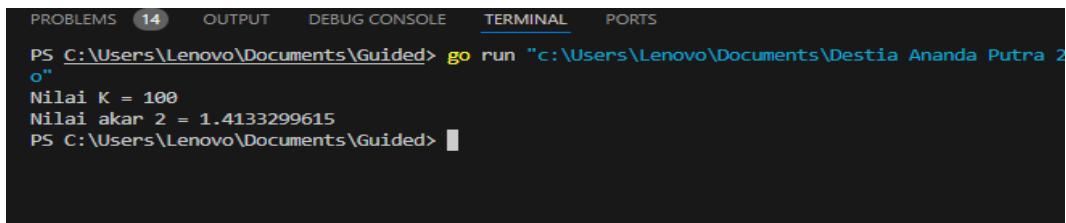
Unguided 4

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var k, hasil float64 = 0, 1
    fmt.Print("Nilai K = ")
    fmt.Scan(&k)
    for i := 0; i < int(k); i++ {
        hasil *= math.Pow(4*float64(i)+2, 2) / ((4*float64(i) + 1) *
(4*float64(i) + 3))
    }
    fmt.Printf("Nilai akar 2 = %.10f", hasil)
}
```

Output



```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Destia Ananda Putra 2 o"
Nilai K = 100
Nilai akar 2 = 1.4133299615
PS C:\Users\Lenovo\Documents\Guided> |
```

Penjelasan

Program ini digunakan untuk menghitung dua hal: nilai fungsi $f(K)$ dan nilai perkiraan akar kuadrat dengan ketelitian hingga 100 angka di belakang koma. Pengguna memasukkan nilai K, lalu program menggunakan rumus tertentu untuk menghitung $f(K)$ dan perkiraan Hasil dari kedua perhitungan tersebut ditampilkan dengan presisi tinggi.

IV. Unguided C2

Unguided 1

```
package main

import "fmt"

func main() {
    //deklarasi variabel
    var beratGram int
    fmt.Print("Masukkan berat parsel (dalam gram): ")
    fmt.Scan(&beratGram)

    //mengekstrak kg dan gram
    beratKg := beratGram / 1000
    sisaGram := beratGram % 1000
    biaya := beratKg * 10000
    var total int
    var biayagram int

    //hitung dan periksa kondisi
    if beratKg > 10 {
        sisaGram = 0
    }

    if sisaGram >= 500 {
        biayagram = sisaGram * 5
    } else {
        biayagram = sisaGram * 15
    }
    total = biayagram + biaya

    //output
    fmt.Printf("Detail berat %v kg + %v gr\n", beratKg, beratGram)
    fmt.Printf("Detail biaya Rp.%v + Rp.%v\n", biaya, biayagram)
    fmt.Printf("Total biaya pengiriman: Rp %d\n", total)
}
```

Output

```
PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Destia An
Masukkan berat parsel (dalam gram): 8500
Detail berat 8 kg + 8500 gr
Detail biaya Rp.80000 + Rp.2500
Total biaya pengiriman: Rp 82500
PS C:\Users\Lenovo\Documents\Guided> █
```

Penjelasan

Program ini digunakan untuk menghitung biaya pengiriman berdasarkan beratnya dalam gram. Berat tersebut diubah menjadi kg dan sisa gram. Biaya dasar dihitung per kg kemudian biaya tambahan dihitung berdasarkan gram. Jika biaya lebih maka biaya tambahan akan dihapus kemudian program akan langsung menampilkan rincian berat dan total biaya pengiriman.

Unguided 2

```
package main

import "fmt"

func main() {
    for {
        var nam float64
        var nmk string

        // Meminta input nilai akhir dari pengguna
        fmt.Print("Nilai akhir mata kuliah (masukkan -1 untuk
keluar): ")
        fmt.Scanln(&nam)

        // Memeriksa apakah pengguna ingin keluar
        if nam == -1 {
            fmt.Println("Proses selesai.")
            break
        }

        // Menentukan grade berdasarkan nilai akhir
        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"
    }

    // Menampilkan hasil grade
    fmt.Printf("Nilai mata kuliah: %v\n", nmk)
}
}

```

Output

```

PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Des
.go"
Nilai akhir mata kuliah (masukkan -1 untuk keluar): 89
Nilai mata kuliah: A
Nilai akhir mata kuliah (masukkan -1 untuk keluar): 70
Nilai mata kuliah: B
Nilai akhir mata kuliah (masukkan -1 untuk keluar): 50
Nilai mata kuliah: D
Nilai akhir mata kuliah (masukkan -1 untuk keluar): 30
Nilai mata kuliah: E

```

Penjelasan

Kode ini digunakan untuk menentukan nilai huruf (A, AB, B, BC, C, D, atau E) berdasarkan nilai akhir mata kuliah yang dimasukkan oleh pengguna. Pengguna memasukkan nilai akhir dalam bentuk angka, lalu program memeriksa rentang nilai tersebut dan menampilkan nilai huruf yang sesuai berdasarkan standar penilaian yang ditentukan.

Unguided 3

```

package main

import "fmt"

func main() {
    var b int
    var prisma bool
    fmt.Print("Bilangan: ")
    fmt.Scan(&b)
    fmt.Print("Faktor: ")
    for i := 1; i <= b; i++ {
        if b%i == 0 {

```

```

        fmt.Print(i, " ")
    }
}
fmt.Println()
if b < 2 {
    prisma = false
} else {
    prisma = true
}
for i := 2; i*i <= b; i++ {
    if b%i == 0 {
        prisma = false
    }
}
if prisma == true {
    fmt.Println("Prima: true")
} else {
    fmt.Println("Prima: false")
}
}

```

Output

```

PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Destia Ananda Putra 2311102176\Destia
.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS C:\Users\Lenovo\Documents\Guided> go run "c:\Users\Lenovo\Documents\Destia Ananda Putra 2311102176\Destia
erFile.go"
Bilangan: 7
Faktor: 1 7
Prima: true

```

Penjelasan

Program ini menerima input bilangan bulat dan menghitung faktor dari bilangan tersebut. Kemudian, program menentukan apakah bilangan tersebut adalah bilangan prima. Jika bilangan lebih kecil dari 2, dianggap tidak prima. Program mencetak semua faktor dan hasil apakah bilangan tersebut prima atau tidak.

