

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL II
REVIEW STRUKTUR KONTROL**



Oleh:

FADHLIN NAURI MAHRIJAR

2211102339

IF11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Struktur Program Go

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.

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 '*/'.

Tipe Data dan Instruksi Dasar

Data dan Variabel Variabel adalah 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 garisbawah. Tipe data yang umum tersedia adalah integer, real, boolean, karakter, dan string. Lihat tabel berikut ini untuk variasi tipe data yang disediakan dalam bahasa Go. Nilai data yang tersimpan dalam variabel dapat diperoleh dengan menyebutkan langsung nama variabelnya. Contoh: Menyebutkan nama found akan mengambil nilai tersimpan dalam memori untuk variabel found, pastinya.

Informasi alamat atau lokasi dari variabel dapat diperoleh dengan menambahkan prefiks & di depan nama variabel tersebut. Contoh: &found akan mendapatkan alamat memori untuk menyimpan data pada found.

Jika variabel berisi alamat memori, prefiks pada variabel tersebut akan memberikan nilai yang tersimpan dalam memori yang lokasinya disimpan dalam variabel tersebut. Variabel harus dideklarasikan dulu sebelum digunakan. Variabel juga harus diinisialisasi dulu (diisi data) agar nilai yang tersimpan diketahui dengan jelas dan eksekusi algoritma menjadi terprediksi.

Dalam bahasa Go, variabel yang tidak diinisialisasi lebih dahulu otomatis diisi dengan nilai default yang ekuivalen dengan bit 0.

- Nilai 0 untuk bilangan Integer
- 0.0E+0 untuk bilangan real
- false untuk Boolean
- Karakter NUL (lihat tabel ASCII) untuk karakter
- "" (string kosong, string dengan panjang 0) untuk string
- nil untuk alamat memori

Struktur Kontrol Perulangan

Go hanya mempunyai kata kunci for untuk semua jenis perulangan yang kita pelajari dalam notasi algoritma. Dua bentuk yang kita gunakan di sini adalah struktur while-loop dan repeat-until. Dalam konsep pemrograman terstruktur, setiap rancangan algoritma harus memenuhi syarat satu pintu masuk dan satu pintu keluar. Karena itu tidaklah diperkenankan untuk membuat program sumber yang mempunyai struktur loop yang mempunyai pintu keluar lebih dari satu, seperti: Satu pintu keluar dari kondisi for dan satu lagi dari instruksi `if-break` Atau mempunyai Instruksi `if-break` yang lebih dari satu.

1) Bentuk While-Loop Format: `while (kondisi) { ... }` Bentuk while-loop memastikan setiap kali memasuki loop, ada kondisi yang harus terpenuhi (benar true). Ini juga berarti saat keluar dari loop, maka nilai kondisi tersebut pasti salah/false!

2) Bentuk Repeat-Until Format: `repeat { ... } until (kondisi)` Bentuk repeat until di perulangan dilakukan terus menerus sampai kondisi keluar terpenuhi. Artinya selama kondisi belum terpenuhi (salah/false) maka perulangan akan terus dilakukan. Pada saat keluar dari loop maka nilai kondisi pasti benar true!

Perhatian: Karena pernyataan kondisi ada di bawah pada bentuk repeat-until, apapun kondisinya, badan loop pasti akan pernah dieksekusi minimum satu kali!

Kode Go di bawah menggunakan algoritma yang sangat mirip dengan algoritma di atas, dengan perbedaan pada digunakannya bentuk while-loop. Umumnya keluaran kedua algoritma sama, kecuali saat `max` diinisialisasi dengan nilai 0 atau lebih kecil!

Struktur Kontrol Percabangan

Untuk analisa kasus, bahasa Go mendukung dua bentuk percabangan, yaitu `if-else` dan `switch-case`.

1) Bentuk H-Else Berikut Ini bentuk-bentuk if-else yang mungkin dilakukan dalam bahasa Go. Semua bentuk di bawah merupakan satu instruksi if-else-endif saja (hanya satu endif). Bentuk if-else yang bersarang (dengan beberapa endif) dapat dibentuk dengan komposisi beberapa if-else-endif tersebut.

2) Bentuk Switch-Case Dalam bahasa Go ada dua variasi bentuk switch-case. Bentuk yang biasa digunakan adalah ekspresi ditulis pada perintah switch dan nilai ditulis dalam setiap label case. Bentuk yang kedua mempunyai switch tanpa ekspresi, tetapi setiap case boleh berisi ekspresi boolean. Tentunya bentuk yang kedua lebih bersifat umum dan merupakan penyederhanaan bentuk (atau alias dari) susunan if-elseif-..-else-endif

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

SOAL 2A

NO 1

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 :

```
PS C:\Users\fadh1> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 2\tempCodeRunnerFile.go"
Masukkan input string: p
Masukkan input string: q
Masukkan input string: r
Output awal = p q r
Output akhir = q r p
```

Penjelasan :

Program ini melakukan tugas sederhana yaitu membaca tiga input string dari pengguna, menampilkan urutan awalnya, lalu menukar urutannya secara melingkar dan menampilkannya kembali.

NO 2

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 :

```
PS C:\Users\fadh1> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 2\Guided2.go"
Tahun : 2004
Kabisat : true
```

Penjelasan :

Program ini untuk menentukan apakah sebuah tahun yang dimasukkan oleh pengguna merupakan tahun kabisat atau tidak,

NO 3

Source code :

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jejari, volume, luas_kulit float64
    fmt.Print("Jejari = ")
    fmt.Scan(&jejari)
    volume = math.Pi * 4.0 / 3.0 *
math.Pow(jejari, 3)
    luas_kulit = math.Pi * 4 * math.Pow(jejari,
2)
    fmt.Printf("Bola dengan jejari %v memiliki
volume %.4f dan luas kulit %.4f", jejari,
volume, luas_kulit)
}
```

Output :

```
PS C:\Users\fadhl> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 2\Guided3.go"
Jejari = 10
Bola dengan jejari 10 memiliki volume 4188.7902 dan luas kulit 1256.6371
```

Penjelasan :

Kesimpulannya, source code ini adalah sebuah program Go yang menerima input berupa jejari (radius) bola dari pengguna, kemudian menghitung dua hal utama menggunakan rumus matematika yaitu Volume bola dan Luas permukaan bola

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

SOAL 2A

No 4

Source code :

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jejari, volume, luas_kulit float64
    fmt.Print("Jejari = ")
    fmt.Scan(&jejari)
    volume = math.Pi * 4.0 / 3.0 *
math.Pow(jejari, 3)
    luas_kulit = math.Pi * 4 * math.Pow(jejari,
2)
    fmt.Printf("Bola dengan jejari %v memiliki
volume %.4f dan luas kulit %.4f", jejari,
volume, luas_kulit)
}
```

Output :

```
PS C:\Users\fadh1> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 2\Guided3.go"
Jejari = 10
Bola dengan jejari 10 memiliki volume 4188.7902 dan luas kulit 1256.6371
```

Penjelasan :

Program ini dirancang untuk menghitung volume dan luas permukaan (luas kulit) sebuah bola berdasarkan jari-jari yang diberikan oleh pengguna.

No 5

Source code :

```
package main

import "fmt"

func main() {
    var numbers [5]int
    var chars [3]byte

    fmt.Println("Masukkan 5 buah data integer (32-127):")
    for i := 0; i < 5; i++ {
        fmt.Scan(&numbers[i])
    }

    fmt.Sprintf("%c%c%c", &chars[0], &chars[1], &chars[2])

    for _, num := range numbers {
        fmt.Printf("%c", num)
    }
    for _, char := range chars {
        fmt.Printf("%c", char+1)
    }
    fmt.Println()
}
```

Output :

```
PS C:\Users\fadh1> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 2\Unguided\Nomor5.go"
Masukkan 5 buah data integer (32-127):
66 97 103 117 115
SNO
Bagus
TO
```

Penjelasan :

Program ini dirancang untuk:

1. Meminta pengguna untuk memasukkan 5 bilangan bulat dalam rentang 32-127.
2. Membaca 3 karakter dari pengguna.
3. Mencetak karakter-karakter yang sesuai dengan bilangan bulat yang diinputkan.
4. Mencetak karakter-karakter berikutnya dalam urutan ASCII berdasarkan karakter yang diinputkan.

SOAL 2B

No 1

Source code :

```
package main

import "fmt"

func main() {
    var a, b, c, d string
    i := 1
    berhasil := true
    for {
        fmt.Print("percobaan ", i, ":")
        fmt.Scanln(&a, &b, &c, &d)
        if a != "merah" && b != "kuning" && c
!= "hijau" && d != "unggu" {
            berhasil = false
        }
        i++
        if i == 6 {
            break
        }
    }
    fmt.Print(berhasil)
}
```

Output :

```
PS C:\Users\fadh1> go run "C:\Users\fadh1\AppData\Local\Temp\tempCodeRunnerFile.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
true
```

Penjelasan :

Program ini dirancang untuk melakukan percobaan memasukkan empat warna (merah, kuning, hijau, ungu) sebanyak lima kali. Jika salah satu percobaan tidak mengandung keempat warna tersebut, maka program akan dianggap gagal.

No 2

Source code :

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Jumlah bunga: ")
    fmt.Scan(&n)
    total := 0
    var bunga = make([]string, n)
    for i := 0; i < n; i++ {
        fmt.Print("Nama Bunga ", i+1, ": ")
        fmt.Scan(&bunga[i])
        total++
        if bunga[i] == "selesai" {
            break
        }
    }
    fmt.Print("pita: ")
    for i := 0; i < n; i++ {
        fmt.Print(bunga[i], " - ")
    }
    fmt.Println(" ")
    fmt.Print("Bunga: ", total)
}
```

Output :

```
PS C:\Users\fadh1> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 2\Unguided\tempCodeRunnerFile.go"
Jumlah bunga: 5
Nama Bunga 1: teratai
Nama Bunga 2: lotus
Nama Bunga 3: lavender
Nama Bunga 4: mawar
Nama Bunga 5: edelweis
pita: teratai - lotus - lavender - mawar - edelweis -
Bunga: 5
```

Penjelasan :

Program ini dirancang untuk mengumpulkan nama-nama bunga dari pengguna dan menampilkannya dalam format yang diinginkan.

No 3

Source code :

```
package main

import "fmt"

func main() {
    var kantong1, kantong2 float64
    var totalBerat float64

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

        totalBerat = kantong1 +
kantong2

        if kantong1 < 0 || kantong2 < 0
{
            fmt.Println("Berat
tidak boleh negatif.")
            break
        } else if totalBerat > 150 {
            fmt.Println("Proses
selesai.")
            break
        } else if totalBerat >= 9 {
            fmt.Println("Sepeda
motor Pak Andi akan oleng: true")
        } else {
            fmt.Println("Sepeda
motor Pak Andi akan oleng: false")
        }
    }
}
```

Output :

```
PS C:\Users\fadhl> go run "C:\Users\fadhl\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan berat belanjaan di kedua kantong: 60 65
Sepeda motor Pak Andi akan oleng: true
Masukkan berat belanjaan di kedua kantong: 50 40
Sepeda motor Pak Andi akan oleng: true
```

Penjelasan :

Program ini dirancang untuk mensimulasikan proses belanja Pak Andi dan menentukan apakah sepeda motornya akan oleng berdasarkan total berat belanjaan.

No 4

Source code :

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var k int
    var hasil float64 = 1.0
    for {
        fmt.Print("Nilai K = ")
        fmt.Scan(&k)
        if k < 0 {
            break
        }
        hasil *= math.Pow(4*float64(k)+2, 2) /
            ((4*float64(k) + 1) * (4*float64(k)
+ 3))
        fmt.Printf("Nilai Akar 2 = %.10f \n",
hasil)
    }
}
```

Output :

```
PS C:\Users\fadh1> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 2\Unguided\2B_4.go"
Nilai K = 100
Nilai Akar 2 = 1.0000061880
```

Penjelasan :

Program ini dirancang untuk menghitung nilai pendekatan dari akar kuadrat 2 menggunakan suatu metode numerik yang dikenal sebagai formula Vieta. Metode ini memberikan hasil yang semakin akurat seiring dengan bertambahnya iterasi.

SOAL 2C

No 1

Source code :

```
package main

import "fmt"

func main() {
    var beratParsel int

    fmt.Print("Berat parsel (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)
}
```

Output :

```
PS C:\Users\fadhl> go run "C:\Users\fadhl\AppData\Local\Temp\tempCodeRunnerFile.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
```

Penjelasan :

Program ini berfungsi untuk menghitung biaya pengiriman parcel berdasarkan beratnya. Program menghitung biaya pengiriman setiap kilogram dan memberikan tambahan biaya jika ada sisa berat kurang dari 1 kg.

No 2

Source code :

```
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)
}
```

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

Jawab :

Jika kita input nam = 80.1, maka program akan mengecek kondisi:

- nam >= 80: Benar, karena 80.1 lebih besar dari sama dengan 80.
- Oleh karena itu, program akan mencetak A.

Hasil ini sesuai dengan spesifikasi soal karena nilai 80.1 berada dalam rentang nilai A.

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

Jawab :

Kesalahan dalam program ini yaitu :

- Batasan nilai: Dalam kondisi if, program menggunakan > untuk membandingkan dengan 80, 72.5, 65, dll. Seharusnya, grade untuk nilai 80 harus mencakup 80 itu sendiri, sehingga seharusnya menggunakan <=.
- Tidak ada penanganan input yang tidak valid: Program tidak memeriksa apakah nilai yang dimasukkan berada dalam rentang yang valid (0 - 100).
- Kurangnya penanganan untuk nilai negatif atau lebih dari 100: Nilai negatif atau nilai lebih dari 100 tidak diproses dengan benar.

Alur program seharusnya:

- Menerima input dari pengguna.
- Memeriksa apakah input valid (antara 0 dan 100).
- Menentukan grade berdasarkan nilai akhir dengan memeriksa dengan benar apakah nilai tersebut berada pada batas yang ditentukan.
- Menampilkan grade.

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

Jawab :

```
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)
}
```

Output :

```
PS C:\Users\fadhl> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 2\Unguided\2C_2.go"
Masukkan nilai akhir mata kuliah: 90
Nilai mata kuliah: A
PS C:\Users\fadhl> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 2\Unguided\2C_2.go"
Masukkan nilai akhir mata kuliah: 70
Nilai mata kuliah: B
PS C:\Users\fadhl> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 2\Unguided\2C_2.go"
Masukkan nilai akhir mata kuliah: 40
Nilai mata kuliah: D
```

Penjelasan :

Program ini dirancang untuk mengubah nilai numerik menjadi nilai huruf berdasarkan rentang nilai tertentu. Biasanya, ini digunakan untuk mengkonversi nilai ujian atau tugas menjadi nilai huruf seperti A, B, C, dan seterusnya.

No 3

Source code :

```
package main

import "fmt"

func main() {
    var bilangan int

    fmt.Print("Bilangan: ")
    fmt.Scan(&bilangan)

    fmt.Print("Faktor: ")
    var faktor []int
    for i := 1; i <= bilangan; i++ {
        if bilangan%i == 0 {
            faktor = append(faktor, i)
            fmt.Printf("%d ", i)
        }
    }
    fmt.Println()

    // Cek bilangan prima
    prima := len(faktor) == 2
    fmt.Printf("Prima: %t\n", prima)
}
```

Output :

```
PS C:\Users\fadhl> go run "C:\Users\fadhl\AppData\Local\Temp\tempCodeRunnerFile.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS C:\Users\fadhl> go run "C:\Users\fadhl\AppData\Local\Temp\tempCodeRunnerFile.go"
Bilangan: 7
Faktor: 1 7
Prima: true
```

Program ini dirancang untuk:

1. **Meminta input bilangan bulat:** Pengguna diminta untuk memasukkan sebuah bilangan bulat.
2. **Mencari faktor:** Program akan mencari semua bilangan yang habis membagi bilangan yang diinputkan (faktor).
3. **Menentukan bilangan prima:** Program akan menentukan apakah bilangan yang diinputkan adalah bilangan prima atau bukan.