

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL II
REVIEW STRUKTUR KONTROL**



Oleh:

NAMA : ARNANDA SETYA NOSA PUTRA

NIM : 2311102042

KELAS : IF 11 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Struktur Program Go

Dalam pemrograman menggunakan bahasa Go, program utama selalu memiliki dua komponen inti:

- **package main** menandakan bahwa file tersebut berisi program utama.
- **func main()** adalah tempat utama di mana kode program Go ditulis.

Komentar tidak dianggap sebagai bagian dari kode program dan bisa ditempatkan di mana saja dalam program. Ada dua jenis komentar:

- Komentar satu baris dimulai dengan dua garis miring ("//") hingga akhir baris.
- Komentar beberapa baris dimulai dengan "/*" dan diakhiri dengan "*/".

Tipe Data dan Instruksi Dasar

Data dan Variabel

Variabel adalah nama yang digunakan untuk merujuk pada lokasi di memori, di mana data dengan tipe tertentu dapat disimpan. Nama variabel harus diawali dengan huruf dan bisa diikuti oleh huruf, angka, atau garis bawah. Beberapa tipe data yang umum dalam Go termasuk integer, real, boolean, karakter, dan string. Nilai dari suatu variabel bisa diakses dengan menyebutkan nama variabel tersebut. Misalnya, jika variabel bernama "found", maka menyebut "found" akan mengambil nilai yang disimpan di dalamnya.

Alamat atau lokasi variabel di memori bisa diperoleh dengan menambahkan tanda "&" sebelum nama variabel. Sebagai contoh, "&found" akan mengembalikan alamat memori dari variabel "found". Jika variabel berisi alamat memori, tanda "*" sebelum nama variabel tersebut akan mengakses nilai yang tersimpan di memori yang ditunjuk.

Variabel harus dideklarasikan sebelum digunakan dan harus diinisialisasi dengan nilai awal agar program berjalan dengan benar. Dalam Go, variabel yang tidak diinisialisasi secara otomatis akan diisi dengan nilai default:

- 0 untuk bilangan integer,
- 0.0E+0 untuk bilangan real,
- false untuk boolean,
- karakter NUL (sesuai dengan tabel ASCII) untuk tipe karakter,
- string kosong ("") untuk tipe string,
- nil untuk alamat memori.

Struktur Kontrol Perulangan

Go hanya memiliki satu kata kunci **for** yang digunakan untuk semua jenis perulangan. Dua bentuk umum perulangan yang digunakan adalah while-loop dan repeat-until. Dalam pemrograman terstruktur, setiap algoritma harus memiliki satu titik masuk dan satu titik keluar. Oleh karena itu, penggunaan lebih dari satu pintu keluar dalam sebuah loop, seperti menggunakan break di dalam loop dan di instruksi **for**, tidak disarankan.

1. While-Loop

Pada while-loop, kondisi harus benar (true) agar loop bisa berjalan. Ketika kondisi menjadi salah (false), perulangan akan berhenti.

2. Repeat-Until

Perulangan ini terus dilakukan hingga kondisi keluar terpenuhi (benar/true). Jika kondisi belum terpenuhi (salah/false), perulangan akan tetap berjalan. Berbeda dengan while-loop, di repeat-until, tubuh loop akan dieksekusi setidaknya satu kali karena kondisi diperiksa setelah loop dijalankan.

Struktur Kontrol Percabangan

Go mendukung dua bentuk utama percabangan, yaitu if-else dan switch-case.

1. If-Else

Terdapat berbagai variasi struktur if-else dalam Go. Pada dasarnya, semua bentuk tersebut terdiri dari satu blok if-else-endif. Jika diperlukan, struktur bersarang dapat dibentuk dengan menggabungkan beberapa blok if-else-endif.

2. Switch-Case

Go memiliki dua variasi bentuk switch-case. Yang pertama, switch diikuti oleh ekspresi dan setiap case mengandung nilai untuk dibandingkan. Bentuk kedua tidak memerlukan ekspresi pada switch, namun setiap case bisa berisi ekspresi boolean. Bentuk kedua ini lebih fleksibel dan sering digunakan sebagai alternatif dari if-elseif-else.

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
        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)
}
```

Output :

```
PS C:\Users\Eko Puji Susanto\go\rsc\MODUL2\2A> go run 1.go
masukan input string: 1
masukan input string: 2
masukan input string: 3
output awal = 1 2 3
output akhir = 2 3 1
PS C:\Users\Eko Puji Susanto\go\rsc\MODUL2\2A> █
```

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)
    if tahun%4 == 0 && tahun%100 != 0 {
        fmt.Print("Kabisat : True")
    } else {
        fmt.Print("Kabisat : False")
    }
}
```

Output:

```
PS C:\Users\Eko Puji Susanto\go\rsc\MODUL2\2A> go run 2.go
Tahun : 2016
Kabisat : True
PS C:\Users\Eko Puji Susanto\go\rsc\MODUL2\2A> █
```

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\Eko Puji Susanto\go\rsc\MODUL2\2A> go run 3.go
Jejari = 2
Bola dengan jejari 2 memiliki volume 33.5103 dan luas kulit 50.2655
PS C:\Users\Eko Puji Susanto\go\rsc\MODUL2\2A> █
```

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"

func main() {
    var celcius, reamur, fahrenheit, kelvin float64
    fmt.Print("Temperatur Celcius : ")
    fmt.Scan(&celcius)

    fahrenheit = celcius*9.0/5.0 + 32
    reamur = celcius * 4.0 / 5.0
    kelvin = 5.0 / 9.0 * (fahrenheit + 273.15)

    fmt.Println("Derajat Reamur: ", reamur)
    fmt.Println("Derajat Fahrenheit: ", fahrenheit)
    fmt.Println("Derajat Kelvin: ", int(kelvin))
}
```

Output :

```
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2A> go run 4.go
Temperatur Celcius : 50
Derajat Reamur: 40
Derajat Fahrenheit: 122
Derajat Kelvin: 323
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2A> █
```

Penjelasan :

Kesimpulan:

Program ini menerima input temperatur dalam skala Celsius dan mengkonversinya ke tiga skala temperatur lainnya: Reamur, Fahrenheit, dan Kelvin dan membantu pengguna untuk dengan cepat mengonversi temperatur dari Celsius ke beberapa skala yang berbeda, yaitu Reamur, Fahrenheit, dan Kelvin.

NO 5.

Source Code :

```
package main

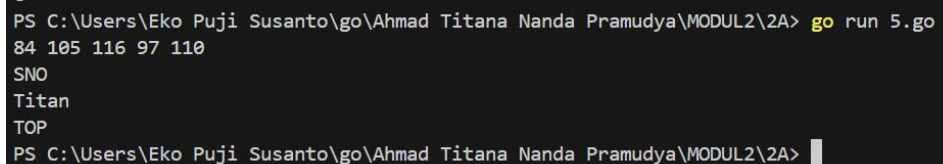
import "fmt"

func main() {
    var angka1, angka2, angka3, angka4, angka5 int
    var char1, char2, char3 byte

    fmt.Scanln(&angka1, &angka2, &angka3, &angka4,
    &angka5)
    fmt.Scanf("%c%c%c", &char1, &char2, &char3)

    fmt.Printf("%c%c%c%c%c\n", angka1, angka2, angka3,
    angka4, angka5)
    fmt.Printf("%c%c%c\n", char1+1, char2+1, char3+1)
}
```

Output:



```
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2A> go run 5.go
84 105 116 97 110
SNO
Titan
TOP
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2A>
```

Penjelasan :

Program ini menggunakan konversi nilai integer menjadi karakter ASCII dan pergeseran karakter berdasarkan nilai ASCII untuk menampilkan hasil yang menarik dari input pengguna.

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\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2B> go run 2b1.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
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2B>
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2B> |
```

Penjelasan :

Program ini menggunakan loop untuk menerima input dari pengguna sebanyak lima kali. Setelah lima kali percobaan, program akan memeriksa apakah input yang diberikan tidak sesuai dengan nilai yang diharapkan ("merah", "kuning", "hijau", "unggu"). Jika ada input yang tidak sesuai, variabel True diubah menjadi false.

NO.2

Source code :

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("masukan berapa bunga: ")
    fmt.Scan(&n)
    total := 0

    var bunga = make([]string, n)
    for i := 0; i < n; i++ {
        fmt.Print("masukan 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\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2B> go run 2B2.go
masukan berapa bunga: 3
masukan bunga 1: kertas
masukan bunga 2: mawar
masukan bunga 3: tulip
pita: kertas - mawar - tulip -
Bunga: 3
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2B> █
```

Penjelasan :

Program ini meminta pengguna untuk memasukkan jumlah bunga, kemudian menerima input berupa nama bunga. Setelah itu, program menampilkan pita (urutan nama bunga) dan total jumlah bunga yang dimasukkan.

NO.3

Source code :

```
package main
import "fmt"
func main() {
    var a, b, val float32
    var isTrue bool

    for {
        fmt.Print("masukan berat di kedua kantong: ")
        fmt.Scan(&a, &b)
        val = a + b
        if val > 150 || a < 0 || b < 0 {
            break
        }

        if a <= b-9.0 || b <= a-9.0 {
            isTrue = true
            fmt.Println("Sepeda motor pak andi akan oleng: ", isTrue)

        } else {
            isTrue = false
            fmt.Println("Sepeda motor pak andi akan oleng: ", isTrue)
        }
    }

    fmt.Print("program selesai")
}
```

Output :

```
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2B> go run 2b3.go
masukan berat di kedua kantong: 60 65
Sepeda motor pak andi akan oleng: false
masukan berat di kedua kantong: 50 40
Sepeda motor pak andi akan oleng: true
```

Penjelasan :

Program ini membaca berat dari dua kantong yang ditempatkan di sepeda motor Pak Andi dan memeriksa apakah berat kantong tidak seimbang (oleng). Program juga menghentikan eksekusi jika berat total kedua kantong lebih dari 150 atau jika salah satu berat bernilai negatif.

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\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2B> go run 2b4.go
Nilai K = 100
Nilai Akar 2 = 1.0000061880
```

Penjelasan :

Program ini menghitung nilai yang mirip dengan akar kuadrat dua menggunakan serangkaian rumus matematis yang dinyatakan dalam bentuk loop. Namun, hasil dari perhitungan ini bukanlah akar dua dari k dalam arti konvensional, tetapi lebih kepada estimasi menggunakan formula yang telah ditentukan.

Modifikasi program sebelumnya yang menerima input integer K dan menghitung $\sqrt{2}$ untuk K tersebut. Hampiran $\sqrt{2}$ dituliskan dalam ketelitian 10 angka di belakang koma.

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var k int
    var hasil float64

    for {
        fmt.Print("Nilai K = ")
        fmt.Scan(&k)
        if k < 0 {
            break
        }

        hasil = math.Sqrt(2)

        fmt.Printf("Akar 2 ( $\sqrt{2}$ ) = %.10f \n", hasil)
    }
}
```

Output :

```
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2B> go run 2b4new.go
Nilai K = 10
Akar 2 (√2) = 1.4142135624
Nilai K = 100
Akar 2 (√2) = 1.4142135624
Nilai K = 1000
Akar 2 (√2) = 1.4142135624
```

Penjelasan :

Program ini sederhana dan efisien untuk menghitung nilai $\sqrt{2}$ secara konstan tanpa memanfaatkan nilai K yang dimasukkan, yang berarti input tersebut tidak berpengaruh pada hasil akhir. Dan Setiap kali pengguna memberikan nilai K yang valid (non-negatif), program akan menghitung dan menampilkan hasil $\sqrt{2}$.

SOAL 2C

NO.1

Source code :

```
package main

import (
    "fmt"
)

func main() {
    var berat, biaya, val, tambahan int

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

    kg := berat / 1000
    sisaBerat := berat % 1000
    biaya = kg * 10000

    if sisaBerat >= 500 && sisaBerat < 1000 {
        for i := 0; i < sisaBerat; i++ {
            tambahan = tambahan + 5
        }
    } else if sisaBerat < 500 && sisaBerat > 0 {
        for i := 0; i < sisaBerat; i++ {
            tambahan = tambahan + 15
        }
    }
}
```

```

    val = biaya + tambahan
    if kg > 10 {

        fmt.Printf("Total biaya pengiriman: Rp %d\n", biaya)
    } else {
        fmt.Printf("Total biaya pengiriman: Rp %d\n", val)
    }
}

```

Output :

```

PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2C> go run 2C1.go
Masukkan berat parcel dalam gram: 8500
Total biaya pengiriman: Rp 82500
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2C>

```

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 nam diberikan adalah 80.1, keluaran dari program tersebut adalah A. Eksekusi program sesuai dengan spesifikasi soal, karena nilai 80.1 lebih besar dari 80, yang berarti memenuhi syarat untuk mendapatkan grade 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 :

```
package main

import "fmt"

func main() {
    var nilaiAkhir float64
    var grade string

    fmt.Print("Masukkan nilai akhir mata kuliah: ")
    fmt.Scanln(&nilaiAkhir)

    if nilaiAkhir < 0 || nilaiAkhir > 100 {
        fmt.Println("Nilai harus antara 0 dan
100.")
        return
    }

    if nilaiAkhir >= 80 {
        grade = "A"
    } else if nilaiAkhir >= 72.5 {
        grade = "AB"
    } else if nilaiAkhir >= 65 {
        grade = "B"
    } else if nilaiAkhir >= 57.5 {
        grade = "BC"
    } else if nilaiAkhir >= 50 {
        grade = "C"
    } else if nilaiAkhir >= 40 {
        grade = "D"
    } else {
        grade = "E"
    }

    fmt.Println("Nilai mata kuliah: ", grade)
}
```

Output :

```
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2C> go run perbaikan.go
Masukkan nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2C> go run perbaikan.go
Masukkan nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2C> go run perbaikan.go
Masukkan nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2C> |
```

Penjelasan :

Program akan menerima inputan nilai dan disimpan pada variable nam. Alurnya dengan melakukan pengecekan nam

NO.3

Source code :

```
package main

import "fmt"

func main() {
    var x int

    fmt.Print("masukan x:")
    fmt.Scan(&x)

    for i := 1; i <= x; i++ {
        if x%i == 0 {
            fmt.Print(i, " ")
        }
    }
}
```

Output :

```
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2C> go run 2c3.go
masukan x:12
1 2 3 4 6 12
PS C:\Users\Eko Puji Susanto\go\Ahmad Titana Nanda Pramudya\MODUL2\2C> |
```

Penjelasan :

Program ini untuk menampilkan faktor-faktor dari bilangan x. di mana program memeriksa apakah suatu bilangan i adalah pembagi dari x, kemudian mencetak bilangan tersebut jika x habis dibagi oleh i.