

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 2
REVIEW STRUCTURE CONTROL**



Oleh:

NAMA : CHRIST DANIEL SANTOSO

NIM : 2311102305

KELAS : IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Dasar Teori Pemrograman Go

1. Struktur Program Go

Setiap program yang ditulis dalam bahasa pemrograman Go (Golang) memiliki dua komponen utama:

- Package Main:

- Menandakan bahwa file tersebut adalah program utama yang akan dieksekusi.

- Dituliskan dengan perintah ``package main``

- Func Main():

- Merupakan fungsi utama di mana eksekusi program dimulai.

- Semua kode yang akan dijalankan berada dalam blok ini.

2. Komentar

Komentar adalah teks yang tidak termasuk dalam eksekusi kode. Di Go, komentar dapat dituliskan dalam dua bentuk:

- Komentar Satu Baris:

- Dimulai dengan garis miring ganda (`//`) hingga akhir baris.

- Komentar Beberapa Baris:

- Dimulai dengan pasangan karakter ``/*` dan diakhiri dengan `*/`.

3. Tipe Data dan Instruksi Dasar

Data dan Variabel

- Variabel:

- Merupakan nama untuk lokasi dalam memori di mana data dengan tipe tertentu disimpan.

- Nama variabel harus dimulai dengan huruf dan dapat diikuti oleh huruf, angka, atau garis bawah.

- Tipe Data Umum:

- Integer: Bilangan bulat.
- Real: Bilangan pecahan (floating-point).
- Boolean: Nilai benar (true) atau salah (false).
- Karakter: Representasi karakter.
- String: Kumpulan karakter (teks).
- Akses Data:
 - Untuk mengakses nilai dari variabel, cukup sebutkan nama variabelnya.
 - Untuk mendapatkan alamat memori dari variabel, gunakan prefiks `&`.
Contoh: `&found`.
- Inisialisasi Variabel:
 - Variabel harus dideklarasikan dan diinisialisasi sebelum digunakan.
 - Jika tidak diinisialisasi, variabel akan memiliki nilai default sebagai berikut:
 - Integer: 0
 - Real: 0.0E+0
 - Boolean: false
 - Karakter: NUL (null)
 - String: string kosong ("")
 - Pointer: nil

4. Struktur Kontrol Perulangan

Go hanya memiliki satu kata kunci untuk semua jenis perulangan, yaitu `for`. Dua bentuk umum dari `for` adalah:

1. Bentuk While-Loop:

- Memastikan bahwa setiap kali memasuki loop, ada kondisi yang harus terpenuhi (true).

- Saat keluar dari loop, nilai kondisi tersebut akan menjadi salah (false)

2. Bentuk Repeat-Until:

- Perulangan dilakukan hingga kondisi terpenuhi.
- Selama kondisi belum terpenuhi (false), perulangan terus dilakukan.
- Karena kondisi diperiksa setelah blok perulangan, minimal satu kali eksekusi akan terjadi.

5. Struktur Kontrol Percabangan

Go mendukung dua bentuk percabangan:

1. Bentuk If-Else:

- Struktur ``if`` digunakan untuk menguji kondisi. Jika kondisi benar, blok kode yang terkait dieksekusi. Jika tidak, blok ``else`` (jika ada) akan dijalankan.

- Bentuk bersarang memungkinkan penggunaan beberapa ``if-else`` dalam satu blok.

2. Bentuk Switch-Case:

- Menyediakan cara untuk menangani beberapa kondisi.
- Ekspresi dituliskan pada pernyataan ``switch``, dan nilai-nilai pada setiap ``case`` diperiksa.
- Dapat menggunakan ``switch`` tanpa ekspresi dengan setiap ``case`` berisi ekspresi boolean, yang memperluas fleksibilitas penggunaannya.

II. GUIDED

Guided 1

Source Code :

```

C: > CHRIST DANIEL SANTOSO_2311102305 > GUIDED MODUL 2 > -∞ GUIDED 1.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var (
7          satu, dua, tiga string
8          string
9      )
10     fmt.Print("masukan input string: ")
11     fmt.Scanln(&tempsatu)
12     fmt.Print("masukan input string: ")
13     fmt.Scanln(&dua)
14     fmt.Print("masukan input string: ")
15     fmt.Scanln(&tiga)
16     fmt.Println("output awal = " + satu + " " + dua + " " + tiga)
17     temp = satu
18     satu = dua
19     dua = tiga
20     tiga = temp
21     fmt.Println("output akhir = " + satu + " " + dua + " " + tiga)
22
23 }
24

```

```

PS C:\Users\cynzw> go run "c:\Users\cynzw\Downloads\tempCodeRunnerFile.go"
masukan input string: 1
masukan input string: 2
masukan input string: 3
output awal = 1 2 3
output akhir = 2 3 1

```

Output :

Penjelasan :

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

Guided 2

Source code:

```

package main

import "fmt"

func main() {
    year := 0
    fmt.Scan(&year)
    fmt.Println((year%4 == 0 && year%100 != 0) || (year%400 == 0))
}

```

Output:

```

PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\GUIDED MODUL 2\GUIDED 2.go"
1
false
PS C:\Users\cynzw>

```

Penjelasan :

program ini menggunakan logika sederhana untuk menentukan apakah suatu tahun adalah tahun kabisat berdasarkan aturan yang telah ditetapkan. Program ini sangat berguna untuk memeriksa apakah tahun tertentu adalah tahun kabisat tanpa perlu menghitung secara manual.

Guided 3

Source code :

```

C: > CHRIST DANIEL SANTOSO_2311102305 > GUIDED MODUL 2 > tempCodeRunnerFile.go > ...
1  package main
2
3  import (
4      "fmt"
5      "math"
6  )
7
8  func main() {
9      var r float64
10     fmt.Print("Jejari = ")
11     fmt.Scan(&r)
12     volume := 4.0 / 3.0 * math.Pi * math.Pow(r, 3)
13     luas := 4 * math.Pi * math.Pow(r, 2)
14     fmt.Println("Bola dengan jejari ", r, "memiliki volume", volume, "dan luas kulit", luas)
15 }
16

```

Output :

```
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\GUIDED MODUL 2\tempCodeRunnerFile.go"
Jejari = 5
Bola dengan jejari 5 memiliki volume 523.598775598299 dan luas kulit 314.1592653589793
PS C:\Users\cynzw> 
```

Penjelasan :

Program Go ini dirancang untuk menghitung volume dan luas permukaan sebuah bola secara otomatis. Pengguna hanya perlu memasukkan nilai jari-jari bola, kemudian program akan menggunakan rumus matematika yang sudah terdefinisi untuk menghitung volume dan luas permukaan bola tersebut. Setelah perhitungan selesai, program akan menampilkan hasil perhitungan volume dan luas permukaan bola ke layar. Program ini mengilustrasikan bagaimana bahasa pemrograman Go dapat digunakan untuk menyelesaikan masalah matematika sederhana seperti perhitungan geometri. Konsep-konsep dasar pemrograman seperti variabel, input/output, dan penggunaan fungsi matematika diterapkan dalam program.

III. UNGUIDED 2B1

Source code:

```
C: > CHRIST DANIEL SANTOSO_2311102305 > UNGUIDED MODUL 2 > -go 2B1.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var warna1, warna2, warna3, warna4 string
7      attempt := 1
8      success := true
9
10     for attempt <= 5 {
11         fmt.Printf("Percobaan %d: ", attempt)
12         fmt.Scanln(&warna1, &warna2, &warna3, &warna4)
13
14         if warna1 != "merah" || warna2 != "kuning" || warna3 != "hijau" || warna4 != "ungu" {
15             success = false
16         }
17
18         attempt++
19     }
20
21     fmt.Println(success)
22 }
23
```

Output :

```
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\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\cynzw>
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\2B1.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
false
PS C:\Users\cynzw>
```


Penjelasan :

Program Go ini dirancang untuk melakukan serangkaian percobaan di mana pengguna diminta untuk memasukkan empat warna dalam urutan tertentu (merah, kuning, hijau, ungu). Program akan terus meminta input hingga pengguna memasukkan urutan warna yang benar atau mencapai jumlah percobaan maksimum. Setelah setiap percobaan, program akan menampilkan apakah urutan warna yang dimasukkan sudah benar atau salah. Jika pengguna berhasil memasukkan urutan warna yang benar dalam salah satu percobaan, program akan mencetak "true" sebagai hasil akhir, yang menandakan bahwa urutan warna yang benar telah ditemukan. Sebaliknya, jika dalam semua percobaan urutan warna yang dimasukkan selalu salah, program akan mencetak "false". Program ini mengilustrasikan penggunaan perulangan (loop) dan kondisi (if-else) dalam bahasa pemrograman Go untuk menyelesaikan suatu tugas yang berulang.

2B2

Source code :

```
C: > CHRIST DANIEL SANTOSO_2311102305 > UNGUIDED MODUL 2 > -go 2B2.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var jumlahBunga int
7      fmt.Print("Masukkan jumlah bunga: ")
8      fmt.Scan(&jumlahBunga)
9
10     bungalist := make([]string, 0, jumlahBunga)
11     total := 0
12
13     for i := 0; i < jumlahBunga; i++ {
14         var bunga string
15         fmt.Print("Masukkan bunga ", i+1, ": ")
16         fmt.Scan(&bunga)
17
18         if bunga == "selesai" {
19             break
20         }
21         bungalist = append(bungalist, bunga)
22         total++
23     }
24
25     fmt.Print("Pita: ")
26     for _, bunga := range bungalist {
27         fmt.Print(bunga, " - ")
28     }
29     fmt.Println()
30     fmt.Print("Bunga: ", total)
31 }
32
```

Output:

```
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\2B2.go"
Masukkan jumlah bunga: 3
Masukkan bunga 1: kertas
Masukkan bunga 2: mawar
Masukkan bunga 3: tulip
Pita: kertas - mawar - tulip -
Bunga: 3
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\2B2.go"
Masukkan jumlah bunga: 0
Pita:
Bunga: 0
PS C:\Users\cynzw> █
```

Penjelasan :

Program Go ini dirancang untuk membuat daftar bunga. Program ini meminta pengguna untuk memasukkan jumlah bunga yang ingin didaftarkan, lalu meminta nama setiap bunga satu per satu. Nama-nama bunga yang dimasukkan akan disimpan dalam sebuah list (daftar). Setelah semua nama bunga dimasukkan, program akan menampilkan daftar bunga yang telah dibuat beserta jumlah total bunganya. Program ini juga memiliki fitur untuk menghentikan penambahan bunga dengan mengetikkan kata "selesai". Fitur ini berguna jika pengguna ingin mengakhiri proses input sebelum mencapai jumlah bunga yang ditentukan sebelumnya. Singkatnya, program ini merupakan contoh sederhana dari penggunaan list dalam bahasa Go untuk menyimpan dan mengelola data berupa teks (dalam hal ini, nama-nama bunga). Konsep-konsep dasar pemrograman seperti variabel, perulangan, kondisi, dan input/output diimplementasikan dalam program ini.

2B3

Source code :

```

C: > CHRIST DANIEL SANTOSO_2311102305 > UNGUIDED MODUL 2 > -go 2B3.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var kantongA, kantongB, total float32
7      var oleng bool
8
9      for {
10         fmt.Print("Masukkan berat di kedua kantong: ")
11         fmt.Scan(&kantongA, &kantongB)
12         total = kantongA + kantongB
13
14         if total > 150 || kantongA < 0 || kantongB < 0 {
15             break
16         }
17
18         if kantongA <= kantongB-9.0 || kantongB <= kantongA-9.0 {
19             oleng = true
20         } else {
21             oleng = false
22         }
23         fmt.Println("Sepeda motor pak Andi akan oleng: ", oleng)
24     }
25
26     fmt.Println("Program selesai")
27 }
28

```

Output :

```

PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\2B3.go"
Masukkan berat di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukkan berat di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukkan berat di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukkan berat di kedua kantong: 59.5 98.7
Program selesai
PS C:\Users\cynzw>

```

Penjelasan :

Program Go ini dirancang untuk menentukan apakah sepeda motor Pak Andi akan oleng berdasarkan berat beban yang diletakkan di kedua kantongnya. Program ini akan terus meminta pengguna untuk memasukkan berat beban di kedua kantong hingga total berat melebihi 150 atau salah satu kantong memiliki berat kurang dari 0. Selain itu, program juga akan memeriksa apakah selisih berat antara kedua kantong lebih dari 9 kg. Jika selisih berat melebihi 9 kg, maka program akan menyimpulkan bahwa sepeda motor Pak Andi akan oleng. Hasil pemeriksaan ini

akan terus ditampilkan hingga pengguna menghentikan program dengan memasukkan nilai berat yang tidak valid. Singkatnya, program ini merupakan simulasi sederhana untuk menganalisis kestabilan sepeda motor berdasarkan distribusi beban pada kedua kantongnya.

2B4

Source code :

```
C: > CHRIST DANIEL SANTOSO_2311102305 > UNGUIDED MODUL 2 > -go 2B4.go > main
1  package main
2
3  import (
4      "fmt"
5      "math"
6  )
7
8  func main() {
9      var nilaiK int
10
11      fmt.Print("Masukkan nilai k: ")
12      fmt.Scan(&nilaiK)
13
14      numerator := calculateNumerator(nilaiK)
15      denominator := calculateDenominator(nilaiK)
16
17      fk := float64(numerator) / float64(denominator)
18      fmt.Printf("f(%d) = %.10f\n", nilaiK, fk)
19
20      akarDua := math.Sqrt(2)
21      fmt.Printf("Akar 2 = %.10f\n", akarDua)
22  }
23
24  func calculateNumerator(k int) int {
25      return (4*k + 2) * (4*k + 2)
26  }
27  func calculateDenominator(k int) int {
28      return (4*k + 1) * (4*k + 3)
29  }
30
```

Output :

```
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\2B4.go"
Masukkan nilai k: 10
f(10) = 1.0005672150
Akar 2 = 1.4142135624
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\2B4.go"
Masukkan nilai k: 100
f(100) = 1.0000061880
Akar 2 = 1.4142135624
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\2B4.go"
Masukkan nilai k: 1000
f(1000) = 1.000000624
Akar 2 = 1.4142135624
PS C:\Users\cynzw> █
```

Penjelasan :

Program Go ini dirancang untuk menghitung nilai suatu fungsi matematika tertentu yang melibatkan pembilang dan penyebut berdasarkan nilai input “k” yang diberikan pengguna. Selain itu, program juga menghitung nilai akar kuadrat dari angka 2. Program ini menggunakan fungsi-fungsi untuk memisahkan tugas perhitungan dan menggunakan pustaka “math” untuk menghitung akar kuadrat. Hasil perhitungan nilai fungsi dan akar kuadrat kemudian ditampilkan ke layar. Singkatnya, program ini merupakan kalkulator sederhana yang dapat menghitung nilai fungsi tertentu dan nilai akar kuadrat.

2C1

Source code :

```

C: > CHRIST DANIEL SANTOSO_2311102305 > UNGUIDED MODUL 2 > go 2C1.go > main
1  package main
2
3  import (
4      "fmt"
5  )
6
7  func main() {
8      var beratParsel, biayaPengiriman, biayaTambahan, tambahanBiaya int
9
10     fmt.Print("Masukkan berat parsel dalam gram: ")
11     fmt.Scan(&beratParsel)
12
13     kg := beratParsel / 1000
14     sisaBerat := beratParsel % 1000
15     biayaPengiriman = kg * 10000
16
17     if sisaBerat >= 500 && sisaBerat < 1000 {
18         tambahanBiaya = sisaBerat * 5
19     } else if sisaBerat < 500 && sisaBerat > 0 {
20         tambahanBiaya = sisaBerat * 15
21     }
22
23     biayaTambahan = biayaPengiriman + tambahanBiaya
24     if kg > 10 {
25         fmt.Printf("Total biaya pengiriman: Rp %d\n", biayaPengiriman)
26     } else {
27         fmt.Printf("Total biaya pengiriman: Rp %d\n", biayaTambahan)
28     }
29 }
30

```

Output :

```

PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\C1.go"
Masukkan berat parsel dalam gram: 8500
Total biaya pengiriman: Rp 82500
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\C1.go"
Masukkan berat parsel dalam gram: 9250
Total biaya pengiriman: Rp 93750
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\C1.go"
Masukkan berat parsel dalam gram: 11750
Total biaya pengiriman: Rp 110000
PS C:\Users\cynzw>

```

Penjelasan :

Program Go ini dirancang untuk menghitung biaya pengiriman suatu parsel. Program ini meminta pengguna memasukkan berat parsel dalam gram, kemudian menghitung biaya pengiriman berdasarkan aturan tertentu. Aturan biaya pengiriman ini didasarkan pada berat parsel dalam kilogram dan gram. Jika berat parsel dalam kilogram lebih dari 10 kg, maka akan dikenakan biaya tambahan. Hasil akhir berupa total biaya pengiriman akan ditampilkan ke layar. Program ini

menggunakan struktur percabangan (if-else) untuk menentukan biaya pengiriman berdasarkan berat parcel. Singkatnya, program ini merupakan kalkulator sederhana untuk menghitung biaya pengiriman parcel.

2C2

Source code :

```
C: > CHRIST DANIEL SANTOSO_2311102305 > UNGUIDED MODUL 2 > 2C2.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var nilaiAkhir float64
7      var grade string
8
9      fmt.Print("Masukkan nilai akhir mata kuliah: ")
10     fmt.Scanln(&nilaiAkhir)
11
12     switch {
13     case nilaiAkhir > 80 && nilaiAkhir <= 100:
14         grade = "A"
15     case nilaiAkhir > 72.5 && nilaiAkhir <= 80:
16         grade = "AB"
17     case nilaiAkhir > 65 && nilaiAkhir <= 72.5:
18         grade = "B"
19     case nilaiAkhir > 57.5 && nilaiAkhir <= 65:
20         grade = "BC"
21     case nilaiAkhir > 50 && nilaiAkhir <= 57.5:
22         grade = "C"
23     case nilaiAkhir > 40 && nilaiAkhir <= 50:
24         grade = "D"
25     case nilaiAkhir >= 0 && nilaiAkhir <= 40:
26         grade = "E"
27     default:
28         grade = "Nilai tidak valid"
29     }
30
31     fmt.Println("Grade mata kuliah: ", grade)
32 }
33
```

Output:

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\C2.go"
Masukkan nilai akhir mata kuliah: 93.5
Grade mata kuliah: A
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\C2.go"
Masukkan nilai akhir mata kuliah: 70.7
Grade mata kuliah: B
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\C2.go"
Masukkan nilai akhir mata kuliah: 52.9
Grade mata kuliah: C
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\C2.go"
Masukkan nilai akhir mata kuliah: 45
Grade mata kuliah: D
PS C:\Users\cynzw> 
```

Penjelasan :

Program Go ini dirancang untuk menentukan nilai huruf (grade) berdasarkan nilai akhir mata kuliah yang dimasukkan pengguna. Program ini menggunakan struktur [switch-case] untuk mengecek rentang nilai akhir dan menentukan grade yang sesuai. Pengguna akan diminta memasukkan nilai akhir mata kuliah, kemudian program akan menampilkan nilai huruf (A, B, C, D, atau E) yang sesuai dengan rentang nilai akhir tersebut. Jika nilai akhir yang dimasukkan di luar rentang yang ditentukan, program akan menampilkan pesan "Nilai tidak valid". Singkatnya, program ini merupakan sebuah sistem penilaian sederhana berdasarkan nilai akhir mata kuliah.

2C3

Source code ;

```
C: > CHRIST DANIEL SANTOSO_2311102305 > UNGUIDED MODUL 2 > 2C3.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var angka int
7
8      fmt.Print("Masukkan angka: ")
9      fmt.Scan(&angka)
10     cetakFaktor(angka)
11 }
12 func cetakFaktor(n int) {
13     for i := 1; i <= n; i++ {
14         if n%i == 0 {
15             fmt.Print(i, " ")
16         }
17     }
18     fmt.Println()
19 }
20 
```


Output :

```
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\C3.go"
Masukkan angka: 12
1 2 3 4 6 12
PS C:\Users\cynzw> go run "c:\CHRIST DANIEL SANTOSO_2311102305\UNGUIDED MODUL 2\C3.go"
Masukkan angka: 7
1 7
PS C:\Users\cynzw> █
```

Penjelasan :

Program Go ini dirancang untuk menampilkan semua faktor dari sebuah bilangan bulat yang dimasukkan oleh pengguna. Program ini bekerja dengan cara meminta pengguna untuk memasukkan sebuah angka, kemudian melakukan perulangan untuk memeriksa setiap bilangan dari 1 hingga angka yang dimasukkan. Jika bilangan tersebut habis dibagi oleh angka yang dimasukkan, maka bilangan tersebut adalah faktor dari angka yang dimasukkan dan akan ditampilkan ke layar. Singkatnya, program ini berfungsi sebagai pencari faktor dari sebuah bilangan bulat.