# LAPORAN PRAKTIKUM

**ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 2**

# REVIEW STRUKTUR KONTROL



Oleh:

Raja Muhammad Lufhti

103112400027

12 IF 01

# S1 TEKNIK INFORMATIKA

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

# I. DASAR TEORI

1. **Package Main**  
   package main adalah penanda bahwa file tersebut merupakan program utama dalam bahasa Go. Program yang dapat dieksekusi harus memiliki package main, yang menjadi titik awal eksekusi kode.
2. **Func Main**  
   func main() adalah fungsi utama dalam program Go. Eksekusi program dimulai dari fungsi ini, yang berisi instruksi atau logika utama yang akan dijalankan pertama kali ketika program dijalankan.
3. **Var**  
   var digunakan untuk mendeklarasikan variabel dalam Go. Variabel harus memiliki tipe data tertentu dan dapat diinisialisasi saat deklarasi atau diisi nanti dalam program. Deklarasi bisa dilakukan secara eksplisit dengan menyebutkan tipe data atau secara implisit menggunakan :=.
4. **Float**  
   float adalah tipe data dalam Go yang digunakan untuk menyimpan angka desimal atau bilangan real. Go menyediakan dua jenis float, yaitu float32 dan float64, di mana float64 memiliki presisi yang lebih tinggi dibandingkan float32.
5. **String**  
   string adalah tipe data yang digunakan untuk menyimpan teks atau kumpulan karakter dalam Go. String dalam Go bersifat immutable, yang berarti nilainya tidak dapat diubah setelah dideklarasikan.
6. **For**  
   for adalah satu-satunya kata kunci dalam Go yang digunakan untuk perulangan. Perulangan dapat berupa iterasi dengan batas tertentu, tanpa kondisi (infinite loop), atau dengan range untuk mengakses elemen dalam slice atau array.
7. **If-Else**  
   if-else digunakan untuk pengambilan keputusan berdasarkan kondisi tertentu. Jika kondisi dalam if terpenuhi (bernilai true), maka blok kode dalam if akan dijalankan. Jika tidak, maka blok kode dalam else (jika ada) akan dieksekusi.
8. **Logic (Logika)**  
   Logika dalam Go digunakan untuk mengevaluasi kondisi dan mengontrol alur program. Operasi logika melibatkan operator seperti && (AND), || (OR), dan ! (NOT) untuk mengkombinasikan atau membalikkan nilai kondisi. Logika ini sangat penting dalam kontrol alur program, seperti dalam percabangan dan perulangan.

# II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

Contoh 1

|  |
| --- |
| *package main*    *import "fmt"*    *func main() {*  *var greetings = "Selamat di Dunia DAP" var a, b int*  *fmt.Println(greetings) fmt.Scan(&a, &b)*  *fmt.Printf("%v + %v = %v\n", a, b, a+b)*    *}* |

Screenshots Output

A black screen with white text

AI-generated content may be incorrect.

// Foto hasil dari menjalankan code

Deskripsi: Program ini adalah aplikasi Go dasar yang mencetak "Selamat di Dunia DAP" ke terminal dan meminta dua bilangan bulat dari pengguna, menambahkannya saat memasukkan kode, dan mencetak hasilnya dalam bentuk {a} + {b} = {hasil}. Satu-satunya bug adalah pada deklarasi a dan b; keduanya harus dideklarasikan pada barisnya sendiri atau harus diberi nilai awal agar kode dapat dikompilasi.

Contoh 2

*package main*

*import "fmt"*

*func main() {*

*for i := 1; i <= 5; i++ { fmt.Println("Iterasi ke-", 1)*

*}*

*}*

Screenshots Output

A screen shot of a computer

AI-generated content may be incorrect.

// Foto hasil dari menjalankan code

Deskripsi: Program ini adalah implementasi **for** Perulangan ini akan menghasilkan output 1 - 5 yang di tandai angka di depan iterasi

Contoh 3

|  |
| --- |
| *package main*    *import "fmt"*    *func main() { nilai := 80 pctHadir := 0.75 adaTubes := true var indeks string*  *if nilai > 75 && adaTubes {*  *indeks = "A" } else if nilai > 65 { indeks = "B"*  *} else if nilai > 50 && pctHadir > 0.7 { indeks = "C"*  *} else {*  *indeks = "F"*    *}*  *fmt.Printf("Nilai %d dengan kehadiran %.2f%% dan buat tubes %t, mendapat indeks %s\n", nilai, pctHadir\*100, adaTubes, indeks)*    *}* |

Screenshots Output

A black screen with white text

AI-generated content may be incorrect.

// Foto hasil dari menjalankan code

Deskripsi: Program ini adalahProgram untuk menentukan **indeks** dari mahasiswa berdasarkan presensi kehadiran dan nilai tubes

# III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

Soal 1

|  |
| --- |
| *// RAJA MUHAMMAD LUFHTI || 103112400027*  *package main*  *import "fmt"*  *func main() {*  *var O int*  *var hasil bool*  *fmt.Scan(&O)*  *hasil =  true*  *if O%400 == 0 || O%4 == 0 && O%100 != 0 {*  *fmt.Println("year :", O)*  *fmt.Println("leap :", hasil)*    *}else {*  *fmt.Println("year :", O)*  *fmt.Println("leap :", !hasil)*  *}*  *}* |

Screenshots Output

A screenshot of a computer program

AI-generated content may be incorrect.

// Foto hasil dari menjalankan code

Deskripsi: Kode di atas adalah program dalam bahasa Go yang menentukan apakah suatu tahun merupakan tahun kabisat atau tidak. Program meminta input bilangan integer O yang mewakili tahun, lalu mengevaluasi dengan aturan bahwa tahun kabisat adalah yang habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100. Jika kondisi tersebut terpenuhi, program mencetak bahwa tahun tersebut adalah tahun kabisat (leap: true), jika tidak, program mencetak bahwa tahun tersebut bukan tahun kabisat (leap: false).

Soal 2

|  |
| --- |
| *// RAJA MUHAMMAD LUFHTI || 103112400027*  *package main*  *import "fmt"*  *func main() {*  *var q int*  *var root2 float64*  *fmt.Scan(&q)*  *root2 = 1.0*  *for i := 0; i <= q; i++ {*  *root2 \*= float64((4\*i+2)\*(4\*i+2)) / float64((4\*i+1)\*(4\*i+3))*  *}*  *fmt.Printf("Nilai akar 2 = %.10f\n", root2)*  *}* |

Screenshots Output

A computer screen with white and blue text

AI-generated content may be incorrect.

// Foto hasil dari menjalankan code

Deskripsi: Program ini adalah program dalam bahasa Go yang menghitung hampiran nilai akar dua berdasarkan jumlah iterasi yang diberikan sebagai input. Dengan nilai awal root2 = satu, program menggunakan perulangan for untuk mengalikan root2 dengan ekspresi **"(empat kali i ditambah dua) kuadrat dibagi dengan (empat kali i ditambah satu) dikali (empat kali i ditambah tiga)"** dari i = nol hingga nilai iterasi yang dimasukkan. Setelah perhitungan selesai, program mencetak hasil hampiran akar dua dengan presisi sepuluh angka di belakang koma.

Soal 3

|  |
| --- |
| *// RAJA MUHAMMAD LUFHTI || 103112400027*  *package main*  *import "fmt"*  *func main() {*  *var berat, hasil, hargaperkg, hargapergram int*  *fmt.Scan(&berat)*  *kg := berat / 1000*  *gram := berat % 1000*  *if berat < 10000 {*  *hargaperkg = kg \* 10000*  *if gram > 500 {*  *hargapergram = gram \* 5*  *hasil = hargaperkg + hargapergram*  *} else {*  *hargapergram = gram \* 15*  *hasil = hargaperkg + hargapergram*  *}*  *} else {*  *hargaperkg = kg \* 10000*  *if gram > 500 {*  *hargapergram = gram \* 5*  *hasil = hargaperkg + hargapergram*  *} else {*  *hargapergram = gram \* 15*  *hasil = hargaperkg + hargapergram*  *}*  *hasil = hargaperkg*  *}*  *fmt.Println("Berat Parsel: ", berat)*  *fmt.Println("Detail Berat: ", kg, "kg + ", gram, "gr ")*  *fmt.Println("Detail biaya:", "Rp. ", hargaperkg, " + Rp.", hargapergram)*  *fmt.Print(hasil)*  *}* |

Screenshots Output

A computer screen shot of a black screen

AI-generated content may be incorrect.

// Foto hasil dari menjalankan code

Deskripsi: Program ini adalah program Go yang menghitung harga berdasarkan berat dalam gram. Berat diubah menjadi kilogram dan gram, lalu harga dihitung dengan ketentuan setiap kilogram bernilai 10.000. Jika sisa gram lebih dari 500, harga per gram dikalikan 5 dan ditambahkan ke total harga.

**IV. KESIMPULAN**

Kesimpulannya, program-program yang telah dibahas menggunakan bahasa Go untuk menyelesaikan berbagai perhitungan, seperti menentukan tahun kabisat, menghitung hampiran akar dua, dan menentukan harga berdasarkan berat. Setiap program memanfaatkan struktur dasar pemrograman seperti perulangan (for), percabangan (if-else), serta operasi matematika untuk menghasilkan output sesuai kebutuhan. Dengan pemahaman yang baik terhadap konsep ini, pembuatan program yang efisien dan sesuai logika dapat dilakukan dengan lebih mudah.

# V. REFERENSI

MODUL 2 praktikum alpro