

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 1&2
MATERI**



Oleh:

DAFFA TSAQIFNA FAUZTSANY

103112400032

S1 IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. DASAR TEORI

1. Struktur Dasar Program Go:

- Wajib menggunakan package main dan func main().
- Gunakan import "fmt" untuk input-output.

2. Tipe Data & Variabel:

- Tipe data: int, float64, bool, string, rune, byte.
- Variabel harus dideklarasikan dan diinisialisasi dengan nilai awal atau default.
- Tipe data tidak bisa dicampur dalam satu ekspresi tanpa konversi eksplisit (casting).

3. Instruksi Dasar & Operasi:

- Assignment: =, +=, ++, dll.
- Input: fmt.Scan, fmt.Scanln.
- Output: fmt.Print, fmt.Println, fmt.Printf.

4. Struktur Kontrol Perulangan:

- Hanya tersedia for untuk semua bentuk loop:
 - for kondisi {} → seperti while-loop.
 - for {...} tanpa kondisi → infinite loop (berhenti dengan break).
 - Gunakan range untuk loop pada array/slice.
- Hindari penggunaan break lebih dari satu atau continue.

5. Struktur Kontrol Percabangan:

- If-Else dan Switch-Case tersedia.
- If-Else bersarang bisa digunakan.
- Switch bisa menggunakan ekspresi atau langsung kondisi boolean.

6. Konstanta Simbolik:

- Dideklarasikan dengan const, misalnya const PI = 3.14.

7. Praktik Program Dasar:

- Soal latihan tentang string, tahun kabisat, volume bola, konversi suhu, ASCII, perulangan, dan percabangan.

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

1. GUIDED 1

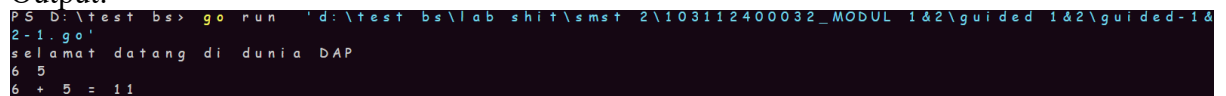
Source Code:

```
package main

import "fmt"

func main() {
    var greetings = "selamat datang di dunia DAP"
    var a, b int
    fmt.Println(greetings)
    fmt.Scanln(&a, &b)
    fmt.Printf("%v + %v = %v\n", a, b, a+b)
}
```

Output:



```
P5 D:\test bs> go run 'd:\test bs\lab shift\smst 2\103112400032_MODUL 1&2\guided 1&2\guided-1&2-1.go'
selamat datang di dunia DAP
6 5
6 + 5 = 11
```

Deskripsi Program:

Program ini mencetak pesan sambutan, lalu meminta pengguna untuk memasukkan dua bilangan bulat. Setelah itu, program menjumlahkan kedua bilangan tersebut dan mencetak hasil penjumlahannya dalam format $a + b = \text{hasil}$.

1. Deklarasi variabel:

```
var greetings = "selamat datang di dunia DAP"
var a, b int
```

- greetings: Variabel bertipe string yang berisi pesan sambutan.
- a, b: Variabel bertipe integer yang akan diisi oleh input pengguna.

2. Menampilkan Pesan Sambutan:

```
fmt.Println(greetings)
```

- Menampilkan teks: selamat datang di dunia DAP.

3. Input Dua Angka:

```
fmt.Scanln(&a, &b)
```

- Pengguna diminta memasukkan dua bilangan integer dalam satu baris, dipisahkan dengan spasi.

4. Menampilkan Hasil Penjumlahan:

```
fmt.Printf("%v + %v = %v\n", a, b, a+b)
```

- Program akan mencetak hasil dari penjumlahan a dan b dalam format yang sesuai.

Contoh:

```
selamat datang di dunia DAP(output)
```

```
4 5 (input)
```

4 + 5 = 9 (output)

2. GUIDED 2

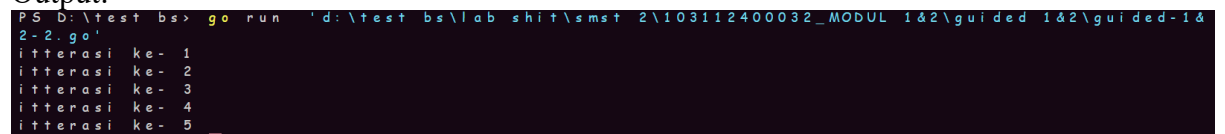
Source Code:

```
package main

import "fmt"

func main() {
    for i := 1; i <= 5; i++ { fmt.Println("itterasi ke-", i) }
}
```

Output:



```
PS D:\test bs> go run 'd:\test bs\lab shift\smst 2\103112400032_MODUL 1&2\guided 1&2\guided-1&2-2.go'
itterasi ke- 1
itterasi ke- 2
itterasi ke- 3
itterasi ke- 4
itterasi ke- 5
```

Deskripsi Program:

Program ini menggunakan perulangan for untuk mencetak teks "itterasi ke-" diikuti dengan angka dari 1 sampai 5. Tujuan dari program ini adalah untuk mendemonstrasikan penggunaan struktur perulangan dalam bahasa Go.

1. Deklarasi variabel:

```
for i := 1; i <= 5; i++ {
```

- i: Variabel penghitung (iterator) bertipe integer, dimulai dari 1 hingga 5.

Penjelasan Proses:

- Program memasuki perulangan for dengan inisialisasi i = 1.
- Selama kondisi i <= 5 terpenuhi, program akan mencetak:

```
fmt.Println("itterasi ke-", i)
```

- Setelah mencetak, nilai i ditambah 1 (i++).
- Ketika nilai i menjadi 6, kondisi tidak lagi terpenuhi dan perulangan berhenti.

Contoh: Jika pengguna memasukkan

```
itterasi ke- 1 (output)
itterasi ke- 2 (output)
itterasi ke- 3 (output)
itterasi ke- 4 (output)
itterasi ke- 5 (output)
```

3. GUIDED 3

Source Code:

```
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, mendapatkan indeks %s\n", nilai, pctHadir*100, adaTubes, indeks)
}
```

Output:

```
PS D:\test bs> go run 'd:\test bs\lab shift\smst 2\103112400032_MODUL 1&2\guided 1&2\guided-1&2-3.go'
Nilai 80 dengan kehadiran 75.00% dan buat tubes =true, mendapatkan indeks A
```

Deskripsi Program:

Program ini bertujuan untuk menentukan indeks nilai akhir seorang mahasiswa berdasarkan nilai ujian (nilai), persentase kehadiran (pctHadir), dan apakah mahasiswa membuat tugas besar (adaTubes). Penilaian menggunakan struktur kontrol if-else untuk mengevaluasi kriteria dan menetapkan indeks dari A sampai F.

1. Deklarasi variabel:

```
nilai := 80
pctHadir := 0.75
adaTubes := true
var indeks string
```

- nilai: Nilai ujian mahasiswa (tipe integer).
- pctHadir: Persentase kehadiran mahasiswa (tipe float64).
- adaTubes: Boolean yang menandakan apakah mahasiswa mengerjakan tugas besar.
- indeks: String kosong yang akan diisi dengan hasil indeks berdasarkan kondisi.

2. Logika Penilaian:

```
if nilai > 75 && adaTubes {
    indeks = "A"
```

```
} else if nilai > 65 {  
    indeks = "B"  
} else if nilai > 50 && pctHadir > 0.7 {  
    indeks = "C"  
} else {  
    indeks = "F"  
}
```

- Jika nilai lebih dari 75 dan mahasiswa mengerjakan tugas besar → indeks = "A"
- Jika nilai lebih dari 65 (tanpa syarat tambahan) → indeks = "B"
- Jika nilai lebih dari 50 dan kehadiran lebih dari 70% → indeks = "C"
- Jika tidak memenuhi semua kondisi di atas → indeks = "F"

3. Menampilkan Hasil:

```
fmt.Printf("Nilai %d dengan kehadiran %.2f%% dan buat tubes =%t, mendapatkan indeks %s\n", nilai, pctHadir*100, adaTubes, indeks)
```

Program mencetak:

- Nilai numerik (nilai)
- Persentase kehadiran ($\text{pctHadir} \times 100$)
- Status pengerjaan tugas besar (adaTubes)
- Indeks akhir (indeks)

Contoh:

Nilai 80 dengan kehadiran 75.00% dan buat tubes =true, mendapatkan indeks A (output)

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

1. UNGUIDED 1

Source Code:

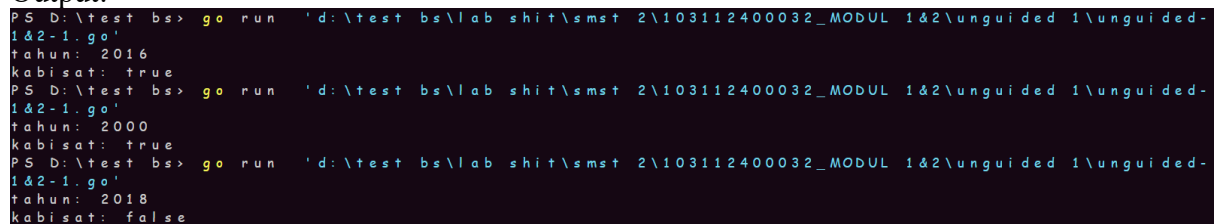
```
package main

import "fmt"

func kabisat(tahun int) bool {
    return (tahun%4 == 0 && tahun%100 != 0) || (tahun%400 == 0)
}

func main() {
    var x int
    fmt.Print("tahun: ")
    fmt.Scan(&x)
    fmt.Print("kabisat: ", kabisat(x))
}
```

Output:



```
PS D:\test bs> go run 'd:\test bs\lab shit\smsr 2\103112400032_MODUL 1&2\unguided 1\unguided-1&2-1.go'
tahun: 2016
kabisat: true
PS D:\test bs> go run 'd:\test bs\lab shit\smsr 2\103112400032_MODUL 1&2\unguided 1\unguided-1&2-1.go'
tahun: 2000
kabisat: true
PS D:\test bs> go run 'd:\test bs\lab shit\smsr 2\103112400032_MODUL 1&2\unguided 1\unguided-1&2-1.go'
tahun: 2018
kabisat: false
```

Deskripsi Program:

Program ini digunakan untuk mengecek apakah suatu tahun merupakan tahun kabisat atau bukan. Program menggunakan fungsi khusus bernama kabisat yang menerima satu parameter bertipe integer (tahun) dan mengembalikan nilai boolean (true jika kabisat, false jika tidak). Pengguna diminta memasukkan tahun, lalu program mencetak hasil pengecekan.

1. Deklarasi Fungsi:

```
func kabisat(tahun int) bool {
    return (tahun%4 == 0 && tahun%100 != 0) ||
    (tahun%400 == 0)
}
```

- Fungsi kabisat menerima satu parameter: tahun (int).
- Mengembalikan true jika:
 - Tahun habis dibagi 4 dan tidak habis dibagi 100, atau
 - Tahun habis dibagi 400.

Ini adalah aturan standar penentuan tahun kabisat.

2. Deklarasi dan Input Variabel di Fungsi main:

```
var x int
fmt.Print("tahun: ")
```

```
fmt.Scan(&x)
```

- x: Variabel integer untuk menyimpan input tahun dari pengguna.
3. Pemanggilan Fungsi dan Output:

```
fmt.Print("kabisat: ", kabisat(x))
```

- Program memanggil fungsi kabisat dengan argumen x, lalu mencetak hasilnya (true atau false).

Contoh:

tahun: 2016 (input)

kabisat: true (output)

2. UNGUIDED 2

Source Code:

```
package main

import "fmt"

func fx(k int) float64 {
    var x float64
    x = 1.0
    for i := 0; i < k; i++ {
        y := float64((4*i + 2) * (4*i + 2))
        z := float64((4*i + 1) * (4*i + 3))
        x *= y / z
    }
    return x
}

func main() {
    var x int
    fmt.Print("nilai K = ")
    fmt.Scan(&x)
    fmt.Printf("nilai f(k) = %.10f", fx(x))
}
```

Output:

```
PS D:\test bs> go run 'd:\test bs\lab shit\smst 2\103112400032_MODUL 1&2\unguided 1\unguided-1&2-2.go'
nilai K = 10
nilai f(k) = 1.4054086752
PS D:\test bs> go run 'd:\test bs\lab shit\smst 2\103112400032_MODUL 1&2\unguided 1\unguided-1&2-2.go'
nilai K = 100
nilai f(k) = 1.4133299615
PS D:\test bs> go run 'd:\test bs\lab shit\smst 2\103112400032_MODUL 1&2\unguided 1\unguided-1&2-2.go'
nilai K = 1000
nilai f(k) = 1.4141251768
```

Deskripsi Program:

Program ini menghitung nilai dari suatu fungsi matematika $f(k)$ dengan menggunakan pendekatan perkalian berurutan berdasarkan nilai integer k yang dimasukkan oleh pengguna.

Perhitungan dilakukan menggunakan sebuah fungsi fx yang akan memproses nilai k dan mengembalikan hasil sebagai `float64`.

1. Deklarasi Fungsi:

```
func fx(k int) float64 {  
    var x float64  
    x = 1.0  
    for i := 0; i < k; i++ {  
        y := float64((4*i + 2) * (4*i + 2))  
        z := float64((4*i + 1) * (4*i + 3))  
        x *= y / z  
    }  
    return x  
}
```

Penjelasan:

- fx adalah fungsi dengan parameter k (integer) dan return value bertipe `float64`.
- x diinisialisasi dengan 1.0, dan akan dikalikan secara bertahap dalam perulangan.
- Pada setiap iterasi ke- i :
 - y adalah hasil kuadrat dari $(4i + 2)$
 - z adalah hasil perkalian $(4i + 1) * (4i + 3)$
 - x dikalikan dengan hasil pembagian y / z
- Nilai akhir dari x dikembalikan sebagai hasil fungsi $f(k)$.

2. Fungsi main dan Input:

```
var x int  
fmt.Print("nilai K = ")  
fmt.Scan(&x)
```

- Variabel x menyimpan input integer dari pengguna sebagai nilai k .

3. Output Hasil Perhitungan:

```
fmt.Printf("nilai f(k) = %.10f", fx(x))
```

- Memanggil fungsi $fx(x)$ dan mencetak hasilnya dengan presisi 10 angka di belakang koma.

Contoh:

```
nilai K = 10 (input)  
nilai f(k) = 1.4054086752 (output)
```

3. UNGUIDED 3

Source Code:

```
package main  
  
import "fmt"  
  
func biaya(rp int) int {
```

```

var x, y int
x = rp / 1000
x *= 10000
if rp%1000 != 0 {
    y = rp - (x / 10)
    if y >= 500 {
        y *= 5
        return x + y
    } else {
        y *= 15
        return x + y
    }
} else {
    return x
}
}
func main() {
    var x, y int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&x)
    y = x % 1000
    if y >= 500 {
        y *= 5
    } else {
        y *= 15
    }
    fmt.Printf("Detail berat: %d kg + %d gr\n", x/1000, x%1000)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", (x/1000)*10000, y)
    fmt.Printf("Total biaya: Rp. %d\n", biaya(x))
}

```

Output:

```

PS D:\test bs> go run 'd:\test bs\lab shit\smst 2\103112400032_MODUL 1&2\unguided 1\unguided-1&2-3.go'
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\test bs> go run 'd:\test bs\lab shit\smst 2\103112400032_MODUL 1&2\unguided 1\unguided-1&2-3.go'
Berat parsel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\test bs> go run 'd:\test bs\lab shit\smst 2\103112400032_MODUL 1&2\unguided 1\unguided-1&2-3.go'
Berat parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 3750
Total biaya: Rp. 113750

```

Deskripsi Program:

Program ini digunakan untuk menghitung biaya pengiriman parsel berdasarkan berat barang dalam satuan gram. Biaya dasar dikenakan Rp10.000 per kilogram, dan sisa berat dalam gram (<1000 gram) dikenakan tarif tambahan berdasarkan nilai:

- Jika sisa ≥ 500 gram \rightarrow tarif Rp5 per gram.
- Jika sisa < 500 gram \rightarrow tarif Rp15 per gram.

Program menghitung dan menampilkan rincian berat dan biaya, serta total biaya pengiriman.

1. Deklarasi dan Logika Fungsi biaya:

```
func biaya(rp int) int {  
    var x, y int  
    x = rp / 1000  
    x *= 10000  
    if rp%1000 != 0 {  
        y = rp - (x / 10)  
        if y >= 500 {  
            y *= 5  
            return x + y  
        } else {  
            y *= 15  
            return x + y  
        }  
    } else {  
        return x  
    }  
}
```

- rp: Parameter fungsi, berat total dalam gram.
- x: Menghitung jumlah kilogram (rp / 1000) lalu dikali Rp10.000.
- y: Menghitung sisa gram dari berat.
 - Jika sisa $\geq 500 \rightarrow$ biaya tambahan Rp5/gr.
 - Jika sisa $< 500 \rightarrow$ biaya tambahan Rp15/gr.
- Fungsi mengembalikan total biaya (x + y), atau hanya x jika beratnya bulat (tanpa sisa gram).

2. Fungsi main dan Alur Program:

```
var x, y int  
fmt.Print("Berat parsel (gram): ")  
fmt.Scan(&x)  
y = x % 1000
```

- Input berat (x) dimasukkan dalam gram.
- y adalah sisa gram (untuk detail biaya tambahan).

3. Menampilkan Detail Berat:

```
fmt.Printf("Detail berat: %d kg + %d gr\n", x/1000, x%1000)
```

- Menampilkan berat dalam bentuk: kg + gram.

4. Menampilkan Detail Biaya:

```
if y >= 500 {  
    y *= 5  
} else {  
    y *= 15  
}  
fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", (x/1000)*10000, y)
```

- Menghitung biaya berdasarkan tarif tambahan, lalu mencetak rincian biaya.

5. Menampilkan Total Biaya:

```
fmt.Printf("Total biaya: Rp. %d\n", biaya(x))
```

- Menggunakan fungsi biaya() untuk menghitung dan menampilkan total akhir.

Contoh:

Berat parcel (gram): 8500 (input)

Detail berat: 8 kg + 500 gr (output)

Detail biaya: Rp. 80000 + Rp. 2500 (output)

Total biaya: Rp. 82500 (output)