

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1**

MODUL 4

IO, Tipe Data dan Variabel



Disusun Oleh :

Rizki Agung Prasetya (103112400263)

IF-12-05

Asisten Praktikum :

Ayu Susilowati

Noviana Rizki Anisa Putri

Dosen Pengampu :

Yudha Islami Sulistya

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

TUGAS PENDAHULUAN

A. PRAKTIKUM Menghitung Luas dan Keliling Lingkaran

Soal Studi Case

Buatlah program yang meminta pengguna memasukkan jari-jari sebuah lingkaran, kemudian menghitung dan menampilkan luas serta keliling lingkaran. Instruksi:

- a. Formula untuk luas lingkaran:

$$\text{Luas} = \pi \times r^2$$

- b. Formula untuk keliling lingkaran = $2 \times \pi \times r$

- c. Cetak hasil luas dan keliling.

Sourcecode

```
package main

import "fmt"

func main() {
    var radius float64
    const pi = 3.14159
    // Meminta input dari pengguna untuk jari-jari lingkaran
    fmt.Print("Masukkan jari-jari lingkaran: ")
    fmt.Scanln(&radius)
    // Menghitung luas lingkaran
    area := pi * radius * radius
    // Menghitung keliling lingkaran
    circumference := 2 * pi * radius
    // Menampilkan hasil
    fmt.Printf("Luas lingkaran: %.2f\n", area)
    fmt.Printf("Keliling lingkaran: %.2f\n", circumference)
}
```

Screenshoot Output

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS	SEARCH ERROR	COMMENTS
PS C:\Users\Rizki Agung Prasetya> go run "d:\Pemrograman\04_TipeData_Variabel\luasdankelilinglingkaran.go"						
Masukkan jari-jari lingkaran: 20						
Luas lingkaran: 1256.64						
Keliling lingkaran: 125.66						
PS C:\Users\Rizki Agung Prasetya>						

Deskripsi Program

Program tersebut adalah program sederhana dalam bahasa Go (Golang) yang menghitung luas dan keliling lingkaran berdasarkan input jari-jari yang diberikan oleh pengguna. Berikut adalah deskripsi dari program tersebut :

1. Mendefinisikan variabel:
 - o Program ini memiliki variabel radius yang bertipe float64 untuk menyimpan nilai jari-jari lingkaran yang akan dimasukkan oleh pengguna.
 - o Selain itu, ada konstanta pi yang didefinisikan sebagai 3.14, yang merupakan nilai dari pi (π) yang digunakan untuk perhitungan lingkaran.
2. Meminta input pengguna:
 - o Program meminta pengguna untuk memasukkan nilai jari-jari lingkaran. Setelah itu, nilai jari-jari disimpan dalam variabel radius.
3. Menghitung luas lingkaran:
 - o Rumus untuk menghitung luas lingkaran adalah $\pi * r^2$. Program menghitung luas lingkaran dengan menggunakan rumus ini dan menyimpannya dalam variabel area.
4. Menghitung keliling lingkaran:
 - o Rumus untuk menghitung keliling lingkaran adalah $2 * \pi * r$. Program menghitung keliling lingkaran dengan menggunakan rumus ini dan menyimpannya dalam variabel circumference.
5. Menampilkan hasil:
 - o Program kemudian menampilkan hasil perhitungan luas dan keliling lingkaran dengan format dua angka desimal.

B. PRAKTIKUM Menghitung Gaji

Bulanan Soal Studi Case

Tulis program Go yang menghitung gaji bulanan karyawan berdasarkan jam kerja. Program harus meminta input jumlah jam kerja dalam seminggu dan upah per jam. Instruksi:

a) Jika jam kerja lebih dari 40 jam per minggu, hitung lembur dengan bayaran 1,5 kali lipat untuk jam lembur.

b) Total gaji dihitung sebagai:

$$\text{Total Gaji} = \text{Jam Normal} \times \text{Upah per Jam} + \text{Lembur} \times 1.5 \times \text{Upah per Jam}$$

c) Tampilkan total gaji bulanan.

Sourcecode

```
package main

import "fmt"

func main() {
    var hoursPerWeek float64
    var wagePerHour float64
    const normalHours = 40
```

```

const overtimeMultiplier = 1.5

// Meminta input jumlah jam kerja dalam seminggu dan upah per jam
fmt.Print("Masukkan jumlah jam kerja per minggu: ")
fmt.Scanln(&hoursPerWeek)

fmt.Print("Masukkan upah per jam: ")
fmt.Scanln(&wagePerHour)

var normalPay, overtimePay float64

// Jika jam kerja lebih dari 40 jam, hitung lembur
if hoursPerWeek > normalHours {
    normalPay = normalHours * wagePerHour
    overtimeHours := hoursPerWeek - normalHours
    overtimePay = overtimeHours * wagePerHour * overtimeMultiplier
} else {
    normalPay = hoursPerWeek * wagePerHour
    overtimePay = 0
}

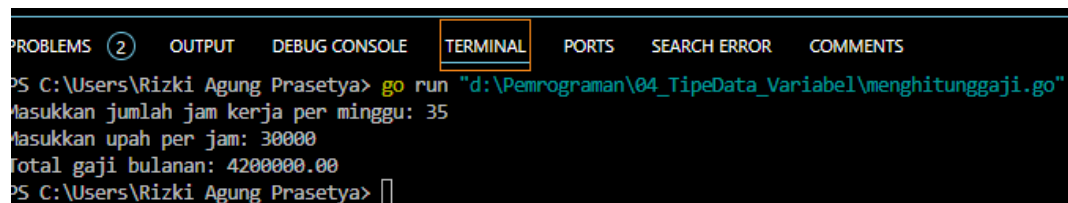
// Menghitung total gaji mingguan
totalWeeklySalary := normalPay + overtimePay

// Menghitung gaji bulanan (4 minggu)
totalMonthlySalary := totalWeeklySalary * 4

// Menampilkan total gaji bulanan
fmt.Printf("Total gaji bulanan: %.2f\n", totalMonthlySalary)
}

```

Screenshoot Output



```

PROBLEMS (2) OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS
PS C:\Users\Rizki Agung Prasetya> go run "d:\Pemrograman\04_TipeData_Variabel\menghitunggaji.go"
Masukkan jumlah jam kerja per minggu: 35
Masukkan upah per jam: 30000
Total gaji bulanan: 4200000.00
PS C:\Users\Rizki Agung Prasetya>

```

Deskripsi Program

Program ini adalah program sederhana dalam bahasa Go (Golang) yang menghitung gaji bulanan berdasarkan jumlah jam kerja per minggu dan upah per jam, termasuk gaji lembur jika jam kerja melebihi 35 jam per minggu. Berikut penjelasannya:

1. Mendefinisikan variabel dan konstanta:
 - Program memiliki dua variabel: `hoursPerWeek` untuk menyimpan jumlah jam kerja per minggu, dan `wagePerHour` untuk menyimpan upah per jam yang akan dimasukkan oleh pengguna.
 - Konstanta `normalHours` bernilai 35 mewakili jumlah jam kerja normal per minggu, dan `overtimeMultiplier` bernilai 1.5 yang digunakan untuk menghitung upah lembur (1,5 kali upah biasa).
2. Meminta input pengguna:
 - Program meminta pengguna memasukkan jumlah jam kerja per minggu dan upah per jam. Nilai-nilai ini disimpan dalam variabel `hoursPerWeek` dan `wagePerHour`.
3. Menghitung gaji normal dan gaji lembur:
 - Jika jam kerja per minggu (`hoursPerWeek`) melebihi 35 jam, program menghitung gaji normal untuk 35 jam pertama, dan sisanya dianggap lembur. Gaji lembur dihitung dengan rumus: $\text{jumlah jam lembur} \times \text{upah per jam} \times 1,5$ (karena lembur dibayar 1,5 kali lipat).
 - Jika jam kerja tidak melebihi 35 jam, maka seluruh gaji dihitung sebagai gaji normal tanpa lembur.
4. Menghitung total gaji mingguan:
 - Program menjumlahkan gaji normal dan gaji lembur untuk mendapatkan total gaji mingguan.
5. Menghitung total gaji bulanan:
 - Program mengalikan total gaji mingguan dengan 4 untuk menghitung total gaji bulanan (diasumsikan ada 4 minggu dalam sebulan).
6. Menampilkan hasil:
 - Program menampilkan total gaji bulanan yang telah dihitung dengan format dua angka desimal.

