

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 2
I/O, TIPE DATA DAN VARIABEL



Disusun oleh:
MUHAMMAD FAIZ MAULANA
109082500124
S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

LATIHAN KELAS – GUIDED

1. Guided 1 Source Code

```
package main

import "fmt"

func main() {

    var a, b, c, d, e int

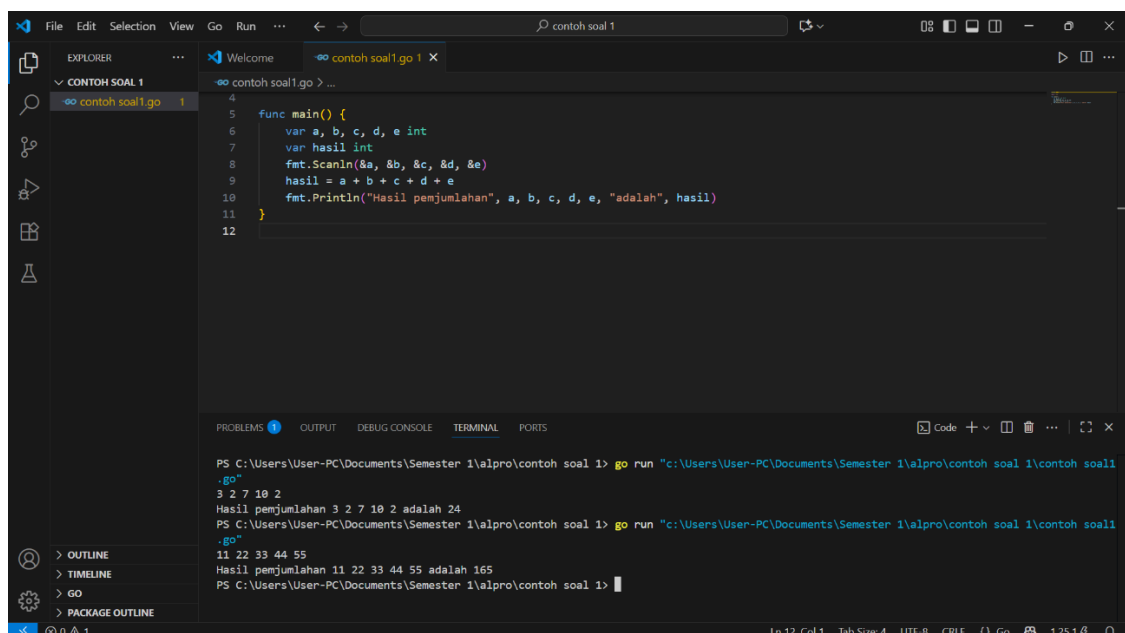
    var hasil int

    fmt.Scanln(&a, &b, &c, &d, &e)

    hasil = a + b + c + d + e

    fmt.Println("Hasil penjumlahan", a, b, c, d, e, "adalah",
hasil)
}
```

Screenshoot program



Deskripsi program

- **package main**
Menandai berkas sebagai paket eksekusi utama; program akan mulai dari fungsi main pada paket ini.

- **import "fmt"**
Mengimpor paket fmt untuk fasilitas input/output terformat di terminal seperti Scanln, Println, dan Printf.
- **func main() { ... }**
Mendefinisikan fungsi utama tempat eksekusi program dimulai dan diakhiri.
- **var a, b, c, d, e int**
Mendeklarasikan lima variabel bilangan bulat untuk menampung lima nilai masukan.
- **var hasil int**
Mendeklarasikan variabel bilangan bulat untuk menyimpan total penjumlahan.
- **fmt.Scanln(&a, &b, &c, &d, &e)**
Membaca lima nilai integer dari input standar dalam satu baris, dipisahkan spasi, lalu mengisi masing-masing variabel; tanda & memberi alamat variabel karena fungsi pembaca membutuhkan pointer untuk menuliskan nilai.
- **hasil = a + b + c + d + e**
Menjumlahkan kelima nilai masukan dan menyimpannya ke variabel hasil.
- **fmt.Println("Hasil penjumlahan", a, b, c, d, e, "adalah", hasil)**
Menampilkan kalimat laporan berisi kelima angka yang dimasukkan dan nilai totalnya, diakhiri baris baru.

2. Guided 2

Source Code

```
package main

import "fmt"

func main () {

    var x, fx float64

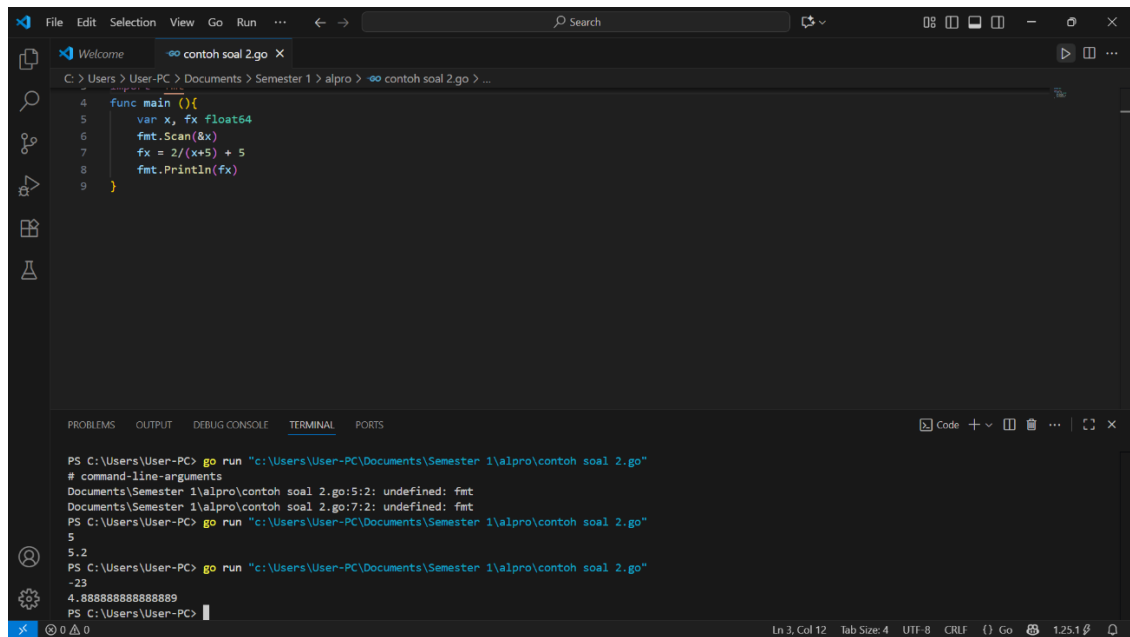
    fmt.Scan(&x)

    fx = 2/(x+5) + 5

    fmt.Println(fx)

}
```

Screenshoot program



The screenshot shows the Visual Studio Code interface. The editor window displays a Go file named `contoh soal 2.go` with the following code:

```
4 func main() {
5     var x, fx float64
6     fmt.Scan(&x)
7     fx = 2/(x+5) + 5
8     fmt.Println(fx)
9 }
```

The terminal window at the bottom shows the execution of the program. It starts with the command `go run "c:\Users\User-PC\Documents\Semester 1\alpro\contoh soal 2.go"`. The output shows the program running successfully and printing the value `5.2` to the terminal.

Deskripsi program

- **package main**
Menandakan bahwa berkas ini adalah paket utama yang dapat dikompilasi menjadi program yang bisa dieksekusi. Program akan mulai berjalan dari fungsi main di paket ini.
- **import "fmt"**
Mengimpor paket `fmt` yang menyediakan fungsi-fungsi input/output terformat seperti `Scan`, `Println`, dan `Printf` untuk berinteraksi lewat terminal.
- **func main () {**
Mendefinisikan fungsi utama tempat eksekusi program dimulai.
- **var x, fx float64**
Mendeklarasikan dua variabel bertipe `float64`.
 - `x`: menampung nilai masukan berupa bilangan real.
 - `fx`: menampung hasil perhitungan fungsi.
- **fmt.Scan(&x)**
Membaca satu nilai dari input standar (keyboard) dan menyimpannya ke variabel `x`. Tanda `&` memberikan alamat variabel karena `Scan` membutuhkan pointer untuk mengisi nilainya.
- **fx = 2/(x+5) + 5**
Menghitung nilai fungsi $f(x) = 2/(x + 5) + 5$ lalu menyimpannya ke `fx`.
Catatan: jika $x = -5$, maka terjadi pembagian dengan nol sehingga sebaiknya dihindari atau ditangani dengan validasi.
- **fmt.Println(fx)**
Mencetak nilai `fx` ke layar diakhiri karakter newline, sehingga hasil perhitungan terlihat pada terminal.
- **}**
Menutup blok fungsi main.
Ringkasnya, program meminta satu angka real, menghitung $f(x) = 2/(x+5) + 5$, lalu menampilkan hasilnya. Untuk peningkatan, bisa ditambah pengecekan error dari `Scan`

serta validasi agar $x \neq -5$, dan menggunakan `fmt.Printf` untuk mengatur jumlah digit desimal jika diperlukan.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var c1, c2, c3, c4, c5 byte

    var b1, b2, b3 int

    fmt.Scan(&c1, &c2, &c3, &c4, &c5)

    fmt.Scanf("%c", &b1)
    fmt.Scanf("%c", &b2)
    fmt.Scanf("%cc", &b3)


    fmt.Printf("%c%c%c%c%c" , c1 , c2 , c3 , c4 , c5)

    fmt.Printf("%c%c%c" , b1+1, b2+1, b3+1)

}
```

Screenshoot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var c1, c2, c3, c4, c5 byte
7     var b1, b2, b3 int
8     fmt.Scan(&c1, &c2, &c3, &c4, &c5)
9     fmt.Scanf("%c", &b1)
10    fmt.Scanf("%c", &b2)
11    fmt.Scanf("%c", &b3)
12
13    fmt.Printf("%c%c%c%c%c", c1, c2, c3, c4, c5)
14    fmt.Printf("%c%c%c", b1+1, b2+1, b3+1)
15 }
```

PS C:\Users\User-PC> go run "c:\Users\User-PC\Documents\Semester 1\alpro\contohsoal3.go"

66 97 103 117 115 SNO

BagusTOP

PS C:\Users\User-PC>

Deskripsi program

- **package main**
Menandai berkas sebagai paket eksekusi utama; program yang dapat dijalankan selalu dimulai dari paket ini.
- **import "fmt"**
Mengimpor paket fmt yang menyediakan fungsi I/O terformat (Scan, Scanf, Printf) untuk membaca dan menulis ke terminal.
- **func main() { ... }**
Mendefinisikan fungsi utama tempat eksekusi program dimulai dan diakhiri.
- **var c1, c2, c3, c4, c5 byte**
Mendeklarasikan lima variabel bertipe byte (alias dari uint8) untuk menyimpan lima karakter pertama yang dibaca dari input. Tipe byte cocok untuk karakter ASCII satu byte.
- **var b1, b2, b3 int**
Mendeklarasikan tiga variabel bilangan bulat untuk menampung karakter tambahan yang akan dibaca dengan Scanf ke dalam bentuk kode numeriknya (akan diperlakukan sebagai kode karakter/ASCII ketika dicetak dengan %c).
- **fmt.Scan(&c1, &c2, &c3, &c4, &c5)**
Membaca lima token input (biasanya dipisahkan spasi atau newline) dan menempatkannya ke c1–c5. Karena variabel bertipe byte, yang efektif dibaca adalah nilai satu byte (misalnya karakter ASCII). Tanda & memberikan alamat variabel karena fungsi pembaca mengisi nilai melalui pointer.
- **fmt.Scanf("%c", &b1)**
Membaca tepat satu karakter mentah, termasuk spasi/newline jika ada, lalu menyimpannya ke b1 (sebagai nilai integer dari kode byte karakter tersebut).
- **fmt.Scanf("%c", &b2)**
Sama seperti sebelumnya, membaca satu karakter lagi ke b2.

- **fmt.Scanf("%cc", &b3)**
Pola format “%cc” berarti: baca satu karakter untuk ditempatkan ke b3, kemudian harapkan literal huruf c pada input. Ini mudah menyebabkan input tidak cocok dan menimbulkan kegagalan pembacaan jika di aliran input tidak ada huruf c setelah karakter yang dibaca. Umumnya yang diinginkan hanyalah “%c” tanpa c tambahan.
- **fmt.Printf("%c%c%c%c%c", c1, c2, c3, c4, c5)**
Mencetak kembali lima karakter yang pertama dibaca. Spesifier %c menampilkan argumen numerik sebagai karakter berdasarkan kodenya.
- **fmt.Printf("%c%c%c", b1+1, b2+1, b3+1)**
Mencetak tiga karakter yang merupakan “pergeseran” satu tingkat pada kode karakter dari b1, b2, b3. Misalnya jika b1 berisi 'A' (65), maka b1+1 akan menjadi 66 yang dicetak sebagai 'B'. Untuk karakter non-huruf atau di luar ASCII, hasil bisa tidak sesuai harapan.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

// Fungsi utama untuk menampilkan biodata mahasiswa

func tampilkanBiodata(nama, nim, kelas string) {

    fmt.Printf("Perkenalkan saya adalah %s, salah satu mahasiswa
Prodi SI-IF dari kelas %s dengan NIM %s.\n",

        nama, kelas, nim)

}

// Fungsi untuk input biodata secara interaktif

func inputBiodata() {

    var nama, nim, kelas string

    fmt.Println("=== Program Biodata Mahasiswa ===")

    fmt.Print("Masukkan Nama: ")

    fmt.Scanln(&nama)
```

```

        fmt.Print("Masukkan NIM: ")

        fmt.Scanln(&nim)

        fmt.Print("Masukkan Kelas: ")

        fmt.Scanln(&kelas)

        fmt.Println("\n=== RESUME BIODATA MAHASISWA ===")

        tampilkanBiodata(nama, nim, kelas)
    }

func main() {

    inputBiodata()

}

```

Screenshoot program

The screenshot shows a Go IDE with the following code in the editor:

```

6 func tampilkanBiodata(nama, nim, kelas string) {
7     fmt.Printf("Perkenalkan saya adalah %s, salah satu mahasiswa Prodi S1-IF dari kelas %s dengan NIM %s.\n",
8         nama, kelas, nim)
9 }
10
11 // Fungsi untuk input biodata secara interaktif
12 func inputBiodata() {
13     var nama, nim, kelas string
14
15     fmt.Println("=== Program Biodata Mahasiswa ===")
16     fmt.Print("Masukkan Nama: ")
17     fmt.Scanln(&nama)
18
19     fmt.Print("Masukkan NIM: ")
20     fmt.Scanln(&nim)
21
22     fmt.Print("Masukkan Kelas: ")
23     fmt.Scanln(&kelas)
24 }

```

The terminal output shows the program's execution:

```

Masukkan Kelas: IF-48-GAB

=== RESUME BIODATA MAHASISWA ===
Perkenalkan saya adalah Bina, salah satu mahasiswa Prodi S1-IF dari kelas IF-48-GAB dengan NIM 1124431414.
PS C:\Users\User-PC> go run "c:\Users\User-PC\Documents\Semester 1\alpro\biodata_clean.go"

=== Program Biodata Mahasiswa ===
Masukkan Nama: Yura
Masukkan NIM: 1324234545
Masukkan Kelas: IF-48-12

=== RESUME BIODATA MAHASISWA ===
Perkenalkan saya adalah Yura, salah satu mahasiswa Prodi S1-IF dari kelas IF-48-12 dengan NIM 1324234545.
PS C:\Users\User-PC>

```


Deskripsi program

- **package main**
Menandai berkas sebagai paket eksekusi utama sehingga dapat dikompilasi menjadi program yang bisa dijalankan; eksekusi dimulai dari fungsi main di paket ini.
- **import "fmt"**
Mengimpor paket fmt yang menyediakan fungsi I/O terformat di terminal, seperti Print, Println, Printf, Scan, dan Scanln, yang dipakai untuk membaca input dan menampilkan output.
- **// Fungsi utama untuk menampilkan biodata mahasiswa**
Komentar satu baris untuk menjelaskan tujuan fungsi di bawahnya; komentar tidak dieksekusi dan hanya untuk dokumentasi.
- **func tampilkanBiodata(nama, nim, kelas string) { ... }**
Mendefinisikan fungsi bernama tampilkanBiodata dengan tiga parameter bertipe string: nama, nim, dan kelas; fungsi ini bertanggung jawab merangkai dan mencetak kalimat biodata ke layar.
- **fmt.Printf("Perkenalkan saya adalah %s, salah satu mahasiswa Prodi S1-IF dari kelas %s dengan NIM %s.\n", nama, kelas, nim)**
Mencetak kalimat terformat menggunakan format specifier %s untuk menyisipkan nilai string dari parameter nama, kelas, dan nim; karakter \n menambah baris baru setelah cetak.
- **// Fungsi untuk input biodata secara interaktif**
Komentar yang menjelaskan bahwa fungsi berikutnya berperan membaca data dari pengguna.
- **func inputBiodata() { ... }**
Mendefinisikan fungsi tanpa parameter yang mengelola alur input biodata dan kemudian memanggil tampilkanBiodata untuk menampilkan resume.
- **var nama, nim, kelas string**
Mendeklarasikan tiga variabel lokal bertipe string untuk menampung input: nama, nim, kelas.
- **fmt.Println("=== Program Biodata Mahasiswa ===")**
Menampilkan judul program diikuti baris baru agar antarmuka terminal lebih informatif.
- **fmt.Print("Masukkan Nama: ")**
Menampilkan prompt tanpa baris baru sehingga kursor tetap di baris yang sama menunggu masukan.
- **fmt.Scanln(&nama)**
Membaca satu baris input hingga newline lalu menyimpannya ke variabel nama; tanda & mengoper alamat variabel karena fungsi pembaca akan menuliskan nilai ke sana.
- **fmt.Print("Masukkan NIM: "); fmt.Scanln(&nim)**
Menampilkan prompt dan membaca NIM dengan cara yang sama; NIM disimpan sebagai string agar tidak bermasalah dengan angka yang diawali nol atau berisi huruf.
- **fmt.Print("Masukkan Kelas: "); fmt.Scanln(&kelas)**
Menampilkan prompt dan membaca kelas, misalnya "IF-1A".

- **fmt.Println("\n=== RESUME BIODATA MAHASISWA ===")**
Mencetak baris kosong melalui \n lalu menampilkan header ringkasan biodata agar output rapi.
- **tampilkanBiodata(nama, nim, kelas)**
Memanggil fungsi presentasi dengan tiga nilai yang baru saja diinput, memisahkan logika input dan logika output agar kode lebih modular dan mudah diuji.
- **func main() { inputBiodata() }**
Fungsi utama program; saat dijalankan, program masuk ke main lalu memanggil inputBiodata sehingga seluruh alur input–output berjalan.

2. Tugas 2

Source code

```
package main

import (
    "fmt"
    "math"
)

func hitungLuasLingkaran(jariJari float64) float64 {
    return math.Pi * jariJari * jariJari
}

func main() {
    var jariJari float64

    fmt.Println("Program Menghitung Luas Lingkaran")

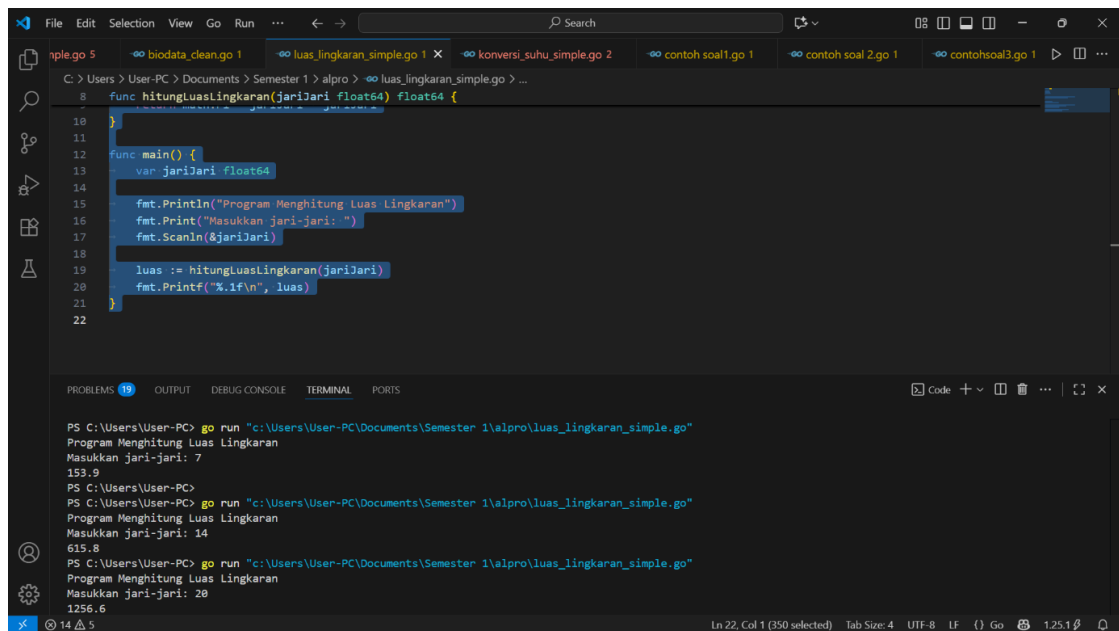
    fmt.Print("Masukkan jari-jari: ")

    fmt.Scanln(&jariJari)

    luas := hitungLuasLingkaran(jariJari)

    fmt.Printf("%.1f\n", luas)
}
```

Screenshoot program



```
8 func hitungLuasLingkaran(jariJari float64) float64 {
10 }
11
12 func main() {
13     var jariJari float64
14
15     fmt.Println("Program Menghitung Luas Lingkaran")
16     fmt.Print("Masukkan jari-jari: ")
17     fmt.Scanln(&jariJari)
18
19     luas := hitungLuasLingkaran(jariJari)
20     fmt.Printf("%.1f\n", luas)
21 }
22
```

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\User-PC> go run "c:\Users\User-PC\Documents\Semester 1\alpro\luas_lingkaran_simpel.go"
Program Menghitung Luas Lingkaran
Masukkan jari-jari: 7
153.9
PS C:\Users\User-PC> go run "c:\Users\User-PC\Documents\Semester 1\alpro\luas_lingkaran_simpel.go"
Program Menghitung Luas Lingkaran
Masukkan jari-jari: 14
615.8
PS C:\Users\User-PC> go run "c:\Users\User-PC\Documents\Semester 1\alpro\luas_lingkaran_simpel.go"
Program Menghitung Luas Lingkaran
Masukkan jari-jari: 20
1256.6
```

Deskripsi program

- **package main**
Menandai berkas sebagai paket eksekusi utama; semua program yang bisa dijalankan di Go harus berada di paket ini agar fungsi main dieksekusi saat program dimulai.
- **import ("fmt" "math")**
Mengimpor dua paket standar:
 - **fmt** untuk input/output terformat di terminal (Print, Println, Printf, Scan/Scanln).
 - **math** untuk fungsi dan konstanta matematika, termasuk konstanta π yang tersedia sebagai **math.Pi**.
- **func hitungLuasLingkaran(jariJari float64) float64 { return math.Pi * jariJari * jariJari }**
Mendefinisikan fungsi bernama **hitungLuasLingkaran** dengan parameter **jariJari** bertipe **float64** dan nilai balik **float64**. Rumus di dalamnya adalah luas lingkaran, yaitu $L = \pi r^2$, diimplementasikan sebagai **math.Pi * r * r**. Tipe **float64** dipilih agar perhitungan pecahan presisi.
- **func main() { ... }**
Mendefinisikan fungsi utama tempat eksekusi program dimulai.
- **var jariJari float64**
Mendeklarasikan variabel untuk menampung nilai jari-jari yang akan dimasukkan pengguna sebagai bilangan real (**float64**).
- **fmt.Println("Program Menghitung Luas Lingkaran")**
Menampilkan judul program diikuti baris baru agar antarmuka terminal lebih jelas.
- **fmt.Print("Masukkan jari-jari: ")**
Menampilkan prompt tanpa baris baru sehingga kursor tetap di baris yang sama ketika menunggu input.

- **fmt.Scanln(&jariJari)**
Membaca satu baris input dari pengguna sampai newline lalu menyimpannya ke variabel jariJari. Simbol & mengoper alamat variabel karena fungsi pembaca akan menuliskan nilai ke memori yang ditunjuk.
- **luas := hitungLuasLingkaran(jariJari)**
Memanggil fungsi hitungLuasLingkaran dengan argumen jariJari, menyimpan hasilnya pada variabel lokal luas menggunakan deklarasi singkat :=.
- **fmt.Printf("%.1f\n", luas)**
Mencetak hasil perhitungan dengan format angka pecahan satu digit di belakang koma karena specifier %.1f, lalu menambahkan baris baru (\n). Jika ingin lebih presisi, ubah menjadi %.2f atau sesuai kebutuhan.

3. Tugas 3

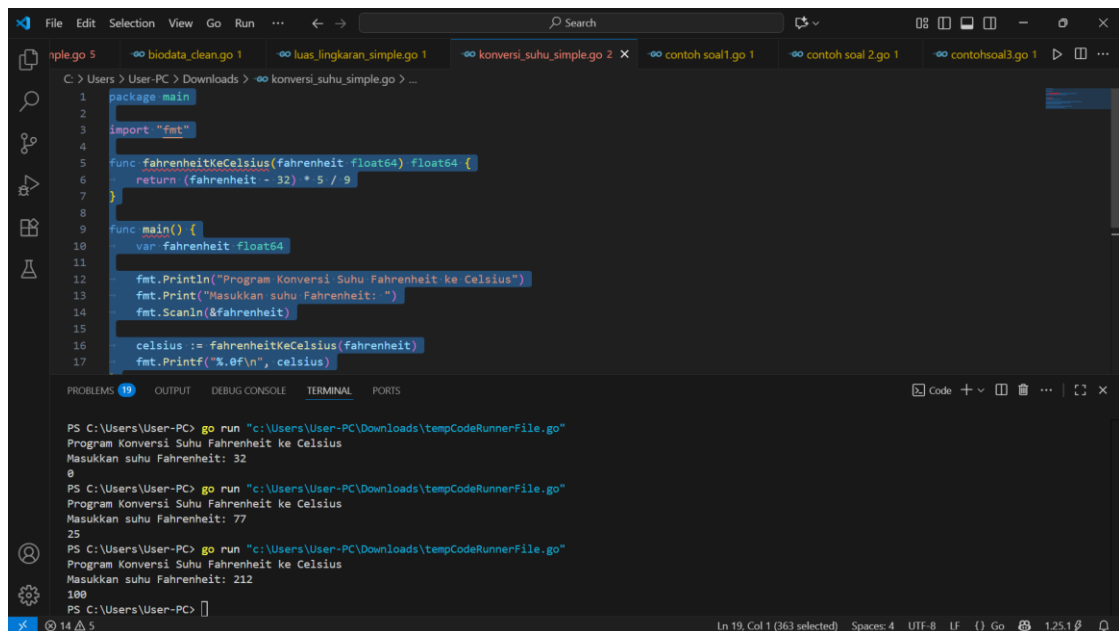
Source code

```
package main
import "fmt"
func fahrenheitKeCelsius(fahrenheit float64) float64 {
    return (fahrenheit - 32) * 5 / 9
}
func main() {
    var fahrenheit float64

    fmt.Println("Program Konversi Suhu Fahrenheit ke
Celsius")
    fmt.Print("Masukkan suhu Fahrenheit: ")
    fmt.Scanln(&fahrenheit)

    celsius := fahrenheitKeCelsius(fahrenheit)
    fmt.Printf("%.0f\n", celsius)
}
```

Screenshoot program



```
1 package main
2
3 import "fmt"
4
5 func fahrenheitKeCelsius(fahrenheit float64) float64 {
6     return (fahrenheit - 32) * 5 / 9
7 }
8
9 func main() {
10     var fahrenheit float64
11
12     fmt.Println("Program Konversi Suhu Fahrenheit ke Celsius")
13     fmt.Print("Masukkan suhu Fahrenheit: ")
14     fmt.Scanln(&fahrenheit)
15
16     celsius := fahrenheitKeCelsius(fahrenheit)
17     fmt.Printf("%.0f\n", celsius)
18 }
```

PS C:\Users\User-PC> go run "c:\Users\User-PC\Downloads\tempCodeRunnerFile.go"

Program Konversi Suhu Fahrenheit ke Celsius

Masukkan suhu Fahrenheit: 32

0

PS C:\Users\User-PC> go run "c:\Users\User-PC\Downloads\tempCodeRunnerFile.go"

Program Konversi Suhu Fahrenheit ke Celsius

Masukkan suhu Fahrenheit: 77

25

PS C:\Users\User-PC> go run "c:\Users\User-PC\Downloads\tempCodeRunnerFile.go"

Program Konversi Suhu Fahrenheit ke Celsius

Masukkan suhu Fahrenheit: 212

100

PS C:\Users\User-PC>

Deskripsi program

- **package main**
Menandai berkas sebagai paket eksekusi utama; semua program Go yang dapat dijalankan harus berada di paket ini agar fungsi main dieksekusi saat program dimulai.
- **import "fmt"**
Mengimpor paket fmt yang menyediakan fungsi input/output terformat di terminal, seperti Print, Println, Printf untuk mencetak, serta Scan/Scanln untuk membaca input.
- **func fahrenheitKeCelsius(fahrenheit float64) float64 { return (fahrenheit - 32) * 5 / 9 }**
Mendefinisikan fungsi bernama fahrenheitKeCelsius dengan satu parameter bertipe float64 dan nilai balik float64. Rumus di dalamnya menerapkan konversi suhu dari Fahrenheit ke Celsius: $C = (F - 32) \times 5/9$. Penggunaan float64 memastikan perhitungan pecahan presisi.
- **func main() { ... }**
Mendefinisikan fungsi utama tempat eksekusi program dimulai.
- **var fahrenheit float64**
Mendeklarasikan variabel untuk menampung input suhu dalam satuan Fahrenheit sebagai bilangan real (float64).
- **fmt.Println("Program Konversi Suhu Fahrenheit ke Celsius")**
Menampilkan judul program diikuti baris baru agar antarmuka terminal informatif.
- **fmt.Print("Masukkan suhu Fahrenheit: ")**
Menampilkan prompt tanpa baris baru agar kursor tetap di baris yang sama saat menunggu input.

- **fmt.Scanln(&fahrenheit)**
Membaca satu baris input hingga newline lalu mengisikan nilainya ke variabel fahrenheit. Tanda & mengoper alamat variabel karena fungsi pembaca akan menuliskan nilai ke memori yang ditunjuk.
- **celsius := fahrenheitKeCelsius(fahrenheit)**
Memanggil fungsi konversi dengan nilai yang sudah dibaca, lalu menyimpan hasilnya ke variabel lokal celsius menggunakan deklarasi singkat :=.
- **fmt.Printf("%.0f\n", celsius)**
Mencetak hasil konversi dalam format angka tanpa desimal (pembulatan ke terdekat) karena spesifikier %.0f, lalu menambahkan baris baru di akhir. Jika ingin menampilkan dua angka desimal, bisa ubah menjadi %.2f.

4. Tugas 4

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
}
```

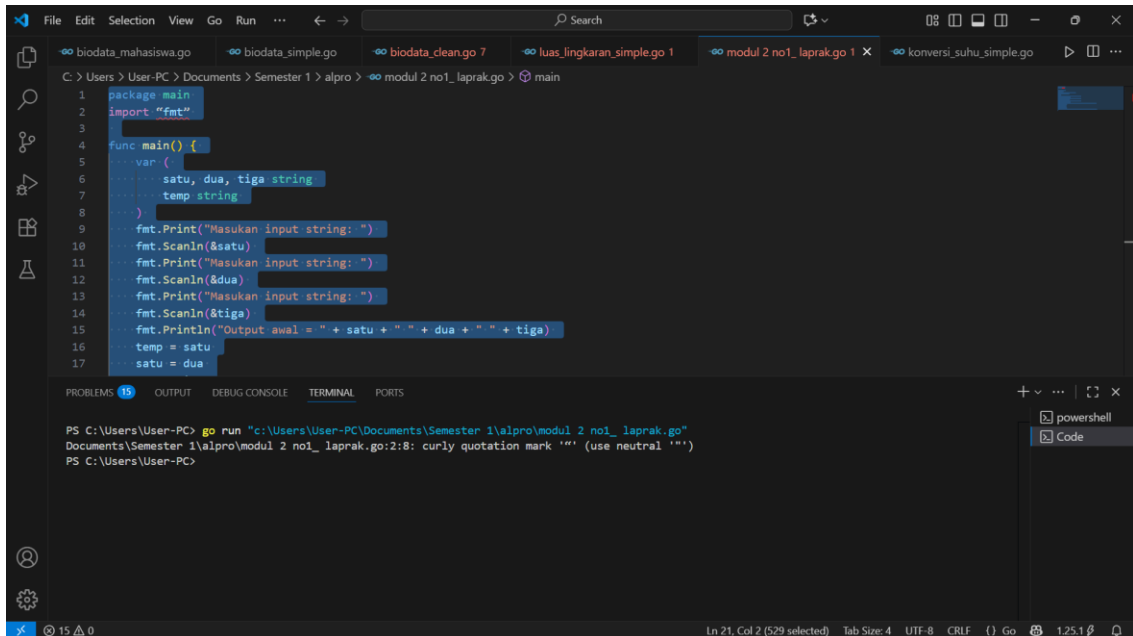
```

    fmt.Println("Output akhir = " + satu + " " + dua + " " +
tiga)

}

```

Screenshoot program



Deskripsi program

- **package main**
Menandai file ini sebagai program utama yang dapat dieksekusi; setiap aplikasi Go yang berdiri sendiri harus berada di package bernama main.
- **import "fmt"**
Mengimpor paket fmt yang menyediakan fungsi I/O format seperti Print, Println, Printf, Scan, dan Scanln untuk membaca/menulis ke terminal.
- **func main() { ... }**
masuk program; fungsi main akan dieksekusi pertama kali saat program dijalankan.
- **var (satu, dua, tiga string; temp string)**
Mendeklarasikan empat variabel bertipe string; tiga untuk menampung input, satu sebagai penampung sementara saat pertukaran nilai.
- **fmt.Print("Masukan input string: ")**
Menampilkan prompt tanpa newline agar input pengguna diketik di baris yang sama.
- **fmt.Scanln(&satu)**
Membaca satu token/baris input dari pengguna dan menyimpannya ke variabel satu; tanda & mengirim alamat variabel (pass by reference) agar fungsi dapat mengisi nilainya.
- **Pengulangan tiga kali (untuk dua dan tiga)**

Melakukan hal yang sama untuk variabel kedua dan ketiga sehingga didapat tiga string input dari pengguna.

- **fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)**

Menampilkan hasil gabungan ketiga string sebelum ditukar, dengan spasi sebagai pemisah.

- **temp = satu; satu = dua; dua = tiga; tiga = temp**

Melakukan rotasi nilai variabel secara siklik ke kiri menggunakan variabel sementara temp: nilai satu dipindah ke temp, satu diganti dua, dua diganti tiga, tiga diisi dari temp.

- **fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)**

Menampilkan hasil setelah rotasi sehingga urutannya berubah menjadi (awal: satu dua tiga) → (akhir: dua tiga satu).

Tugas pendahuluan

1. Jelaskan perbedaan `fmt.Print()`, `fmt.Println()`, dan `fmt.Printf()` di Go!
2. Jelaskan penggunaan tipe data `int`, `float64`, `bool`, dan `string` di Go!
3. Bagaimana Cara mendeklarasikan variabel dengan kata kunci `var` dan dengan acara singkat? Berikan contoh! Hint `":="`
4. Apa perbedaan antara operator `==` dan `=` dalam bahasa Go?
5. Buatlah program Go sederhana untuk meminta input nama kalian, lalu menampilkan nama.

Jawab

1. Jelaskan perbedaan `fmt.Print()`, `fmt.Println()`, dan `fmt.Printf()` di Go!

- `fmt.Print()`: Mencetak argumen apa adanya, tidak menambah baris baru, dan tidak otomatis menambah spasi di antara argumen; jika ingin baris baru, tambahkan `"\n"` sendiri.
Contoh: `fmt.Print("A", "B")` → output: AB; `fmt.Print("A", " ", "B", "\n")` → output: A B (dengan newline).
- `fmt.Println()`: Mirip Print tapi otomatis menyisipkan spasi antar argumen dan menambahkan baris baru di akhir.
Contoh: `fmt.Println("A", "B")` → output: A B (diakhiri newline).
- `fmt.Printf()`: Mencetak dengan format menggunakan verb/placeholder (mis. `%s`, `%d`, `%v`, `%T`), tidak menambah baris baru otomatis, jadi tambahkan `"\n"` jika perlu.
Contoh: `fmt.Printf("%s berumur %d\n", "Ana", 21)` → output terformat;
`fmt.Printf("%T\n", 3.14)` → menampilkan tipe.

ringkasan:

- Butuh spasi + newline otomatis? Gunakan `Println`.

- Butuh kontrol format (tipe, lebar, presisi)? Gunakan Printf.
- Butuh cetak sederhana tanpa spasi dan tanpa newline? Gunakan Print.

2. Jelaskan penggunaan tipe data int, float64, bool, dan string di Go!

int

- Menyimpan bilangan bulat (positif, negatif, atau nol), lebar mengikuti arsitektur (32/64 bit).
- Cocok untuk perhitungan hitung-bulat, indeks, dan perulangan.

float64

- Menyimpan bilangan desimal floating-point dengan presisi ganda (64-bit).
- Dipakai untuk perhitungan yang memerlukan pecahan dan presisi.

bool

- Menyimpan nilai logika: true atau false.
- Umum digunakan pada kondisi if, for, dan operator logika (&&, ||, !).

string

- Menyimpan teks ber-encoding UTF-8, bersifat immutable (tidak bisa diubah in-place).
- Mendukung literal kutip ganda "..." dan raw string dengan backtick ... untuk multi-baris atau tanpa escape.

Ringkasan:

- int: bilangan bulat umum.
- float64: bilangan desimal presisi ganda.
- bool: logika true/false untuk kontrol alur.
- string: teks UTF-8, immutable, dukung konkatenasi dan raw string literal.

3. Bagaimana Cara mendeklarasikan variabel dengan kata kunci var dan dengan acara singkat? Berikan contoh! Hint ":=

Menggunakan kata kunci var

- Cocok untuk deklarasi variabel dengan atau tanpa inisialisasi nilai.
- Bisa menuliskan tipe data secara eksplisit.

Menggunakan cara singkat (short declaration) :=

- Hanya bisa digunakan di dalam fungsi (bukan di luar fungsi/main package).
- Tidak perlu menuliskan tipe data, Go akan otomatis menentukan tipe berdasarkan nilai yang diberikan (inferensi tipe).

Ringkasan :

"Gunakan var jika ingin deklarasi variabel tanpa langsung memberi nilai, atau di luar fungsi. Gunakan := untuk deklarasi cepat di dalam fungsi saat nilai awal sudah diketahui."

4. Apa perbedaan antara operator == dan = dalam bahasa Go?

- **== (Operator Perbandingan):**
 - Digunakan untuk **membandingkan** dua nilai.
 - Hasilnya adalah nilai boolean (true atau false).
 - Contoh: `a == b` akan bernilai true jika nilai a sama dengan b, dan false jika tidak.
 - Biasanya dipakai dalam kondisi if, for, atau ekspresi logika.
- **= (Operator Penugasan/Assignment):**
 - Digunakan untuk memberikan nilai ke variabel.
 - Tidak menghasilkan nilai boolean, hanya mengisi variabel dengan nilai tertentu.
 - Contoh: `a = 5` berarti variabel a diisi dengan nilai 5.

Ringkasnya:

- `==` untuk membandingkan nilai (apakah sama?), hasilnya true atau false.
- `=` untuk mengisi atau mengubah nilai variabel.

5. . Buatlah program Go sederhana untuk meminta input nama kalian, lalu menampilkan nama.

The image shows a Visual Studio Code editor window with a Go file named `pembuatan_nama_1.go` open. The code is a simple program that prompts the user for their name and prints a greeting. The code is as follows:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     // Deklarasi variabel untuk menyimpan nama
7     var nama string
8
9     // Meminta input nama dari user
10    fmt.Print("Masukkan nama Anda: ")
11    fmt.Scanln(&nama)
12
13    // Menampilkan nama yang diinputkan
14    fmt.Printf("Halo, %s!\n", nama)
15 }
```

The terminal output shows the program being run and the user inputting the name "faiz". The output is:

```
PS C:\Users\User-PC> go run "c:\Users\User-PC\Documents\Semester 1\alpro\tempCodeRunnerFile.go"
Masukkan nama Anda: faiz
Halo, faiz!
PS C:\Users\User-PC>
```

The status bar at the bottom indicates the current line and column (Ln 15, Col 2), the number of tabs (Tab Size: 4), the encoding (UTF-8), the line ending (CRLF), and the zoom level (1.25x).