

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 2
I/O, TIPE DATA & VARIABEL



Disusun oleh:
FIKRI LUQMAN MUKTABAR
109082500103
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 int64

    var hasil int64

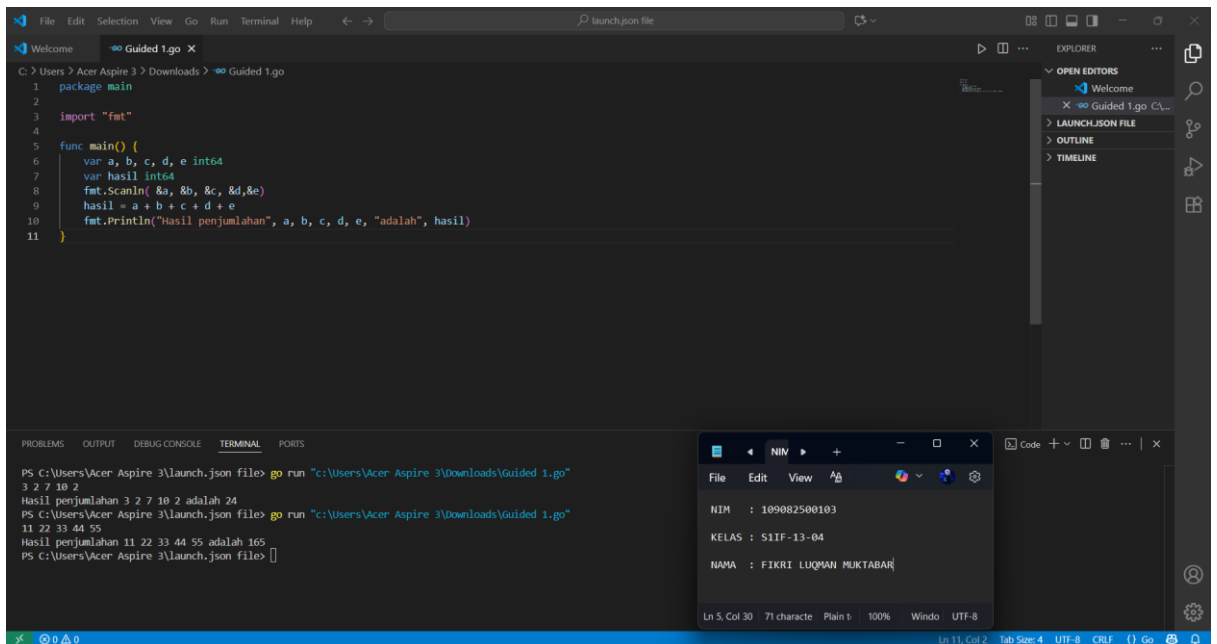
    fmt.Scanln( &a, &b, &c, &d,&e)

    hasil = a + b + c + d + e

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

}
```

Screenshoot Program



Deskripsi Program

1) package main

Menyatakan paket main. Program Go yang dieksekusi harus berada pada paket main. Ini memberitahu compiler bahwa ini adalah program yang bisa dijalankan (bukan library).

2) import "fmt"

Mengimpor paket standar fmt (format). fmt menyediakan fungsi untuk input/output ke terminal seperti Println, Printf, Scanln, dan lain-lain.

3) func main() {

Fungsi main adalah titik masuk (entry point) program. Eksekusi dimulai dari sini.

4) var a, b, c, d, e int64

5) var hasil int64

-Mendeklarasikan enam variabel bertipe int64.

- a, b, c, d, e akan menampung lima bilangan bulat yang dibaca dari input.
- hasil akan menampung hasil penjumlahan.

- Semua variabel bertipe numerik diinisialisasi dengan zero value (0) jika belum diisi.

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

- fmt.Scanln membaca nilai dari standard input (terminal).

- Argumen berupa alamat variabel (operator &) — artinya fungsi akan mengisi nilai ke lokasi memori variabel tersebut.

- Scanln membaca token dipisahkan spasi sampai newline. Ia mengembalikan dua nilai (n, err) — jumlah item yang berhasil dibaca dan error — tetapi kode tersebut tidak menangkap nilai kembalian.

- Jika input tidak sesuai tipe (mis. memasukkan huruf), atau jumlah nilai tidak lengkap sebelum newline, pemanggilan ini bisa menghasilkan error. Karena kode tidak

memeriksa error, program akan tetap lanjut (tetapi beberapa variabel mungkin tidak terisi seperti yang diharapkan).

7) `hasil = a + b + c + d + e`

Menjumlahkan kelima variabel (`int64`) dan menyimpan hasilnya ke variabel `hasil`.

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

`}`

Mencetak output ke terminal. `Println` otomatis menambahkan spasi antar-argumen dan newline di akhir. .

2. Guided 2

Source Code

```
package main

import "fmt"

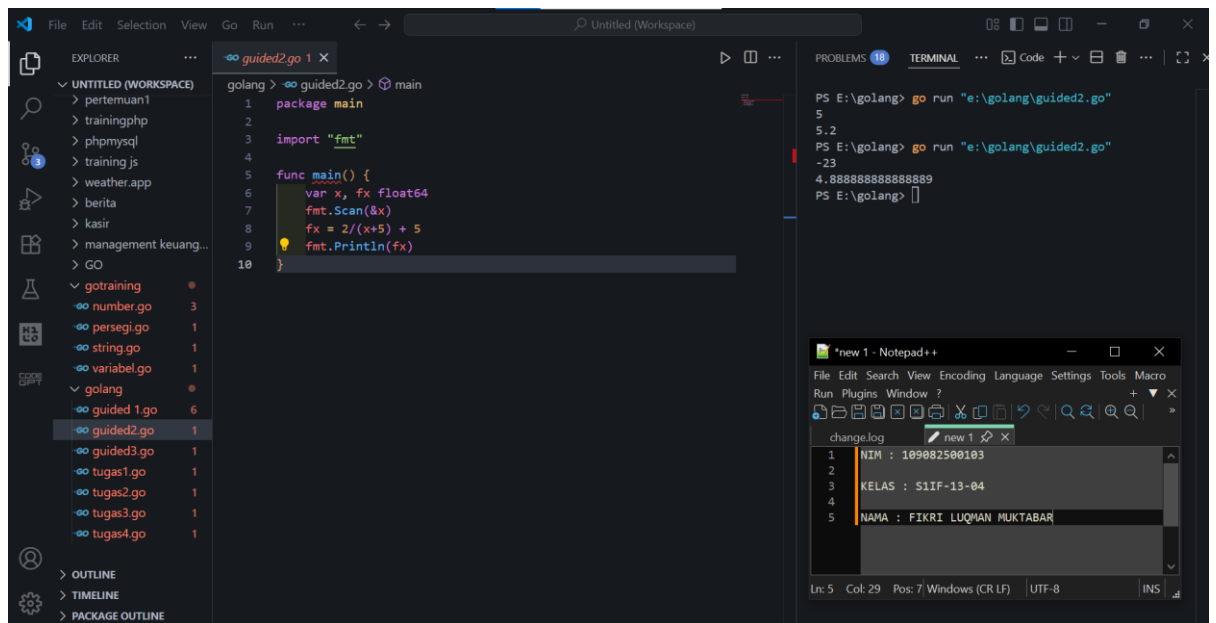
func main(){
    var x, fx float64

    fmt.Scan(&x)

    fx = 2/(x+5) + 5

    fmt.Println(fx)
```

Screenshoot Program



Deskripsi Program

1) package main

Menyatakan paket main. Program Go yang dieksekusi harus berada pada paket main. Ini memberitahu compiler bahwa ini adalah program yang bisa dijalankan (bukan library).

2) import "fmt"

Mengimpor paket standar fmt (format). fmt menyediakan fungsi untuk input/output ke terminal seperti Println, Printf, Scanln, dan lain-lain.

3) func main() {

Fungsi main adalah titik masuk (entry point) program. Eksekusi dimulai dari sini.

4) var x, fx float64

Mendeklarasikan dua variabel x dan fx bertipe float64 (floating point 64-bit). Nilai awal (zero value) keduanya adalah 0.0.

5) fmt.Scan(&x)

- Membaca satu nilai dari input standar (keyboard) dan menyimpannya ke x.

- Argumen &x adalah alamat variabel (pointer) — Scan membutuhkan alamat agar bisa menulis nilai yang dibaca ke variabel tersebut.

6) $fx = 2/(x+5) + 5$

-Menghitung nilai fungsi $f(x)=2x+5+5f(x) = \frac{2}{x+5} + 5f(x)=x+52+5$ dan menyimpan hasilnya ke fx.

-Operasi dilakukan sesuai tanda kurung: pertama hitung $x+5$, lalu $2/(x+5)$, kemudian tambahkan 5.

7) `fmt.Println(fx)`

Mencetak nilai fx ke layar diikuti newline. Format default float akan digunakan.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var c1, c2, c3, c4, c5 byte

    var b1, b2, b3 byte

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

    var str string

    fmt.Scan(&str)

    b1 = str[0]

    b2 = str[1]

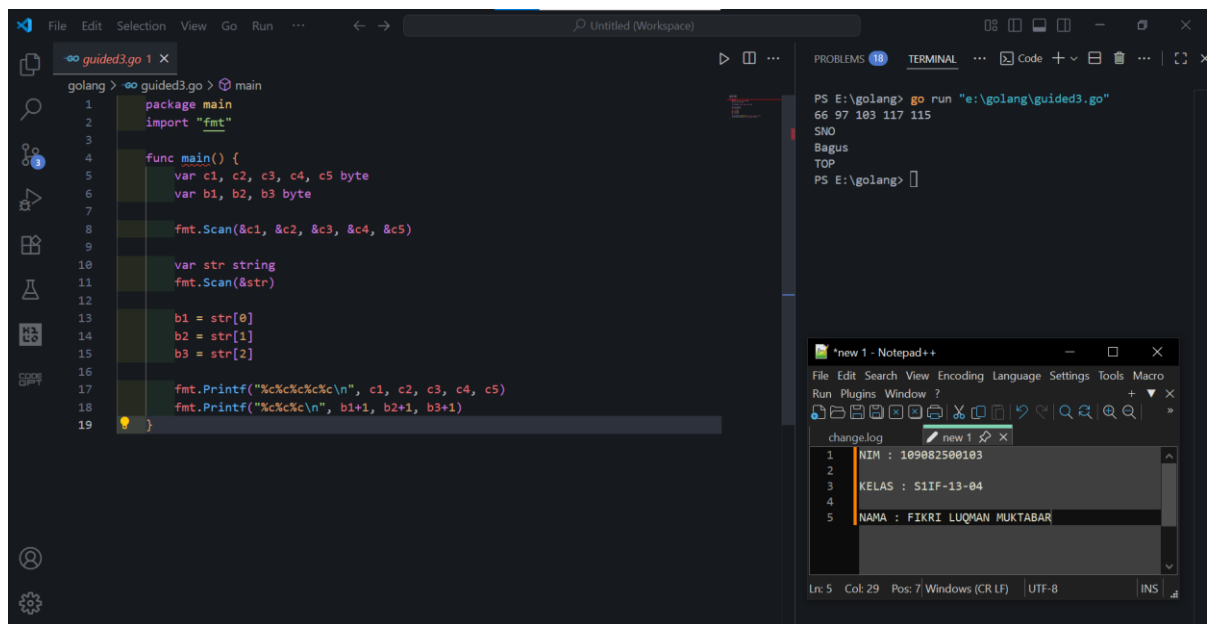
    b3 = str[2]

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

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

}
```

Screenshoot Program



Deskripsi Program

1) package main

Menyatakan paket main. Program Go yang dieksekusi harus berada pada paket main. Ini memberitahu compiler bahwa ini adalah program yang bisa dijalankan (bukan library).

2) import "fmt"

Mengimpor paket standar fmt (format). fmt menyediakan fungsi untuk input/output ke terminal seperti Println, Printf, Scanln, dan lain-lain.

3) func main() {

Fungsi main adalah titik masuk (entry point) program. Eksekusi dimulai dari sini.

4) var c1, c2, c3, c4, c5 byte

Mendeklarasikan 5 variabel bertipe byte (alias uint8). byte menyimpan nilai 0..255 — sering dipakai untuk menyimpan satu nilai oktet (ASCII).

5) var b1, b2, b3 byte

Mendeklarasikan 3 variabel byte lagi.

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

Membaca sampai 5 token dari input standar (dipisah spasi/newline) dan mencoba mengisi c1..c5

7) var str string

Menyediakan variabel str untuk menyimpan token string berikutnya.

8) fmt.Scan(&str)

Membaca token berikutnya (string tanpa spasi) dan menyimpannya ke str.

9) b1 = str[0] ; b2 = str[1] ; b3 = str[2]

Mengambil byte pada indeks 0,1,2 dari str dan menyimpannya ke b1..b3. str[i] menghasilkan byte (offset ke i dalam representasi UTF-8 string). **Kalau str kurang panjang (<3), program akan panic (index out of range).**

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

Mencetak kelima variabel c1..c5 sebagai karakter (%c). Nilai numerik yang tersimpan di tiap cX ditampilkan sebagai karakter sesuai kode Unicode/ASCIInya.

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

Menambahkan 1 ke setiap byte b1..b3 lalu mencetaknya sebagai karakter — efeknya menggeser tiap kode karakter satu langkah ke depan (mis. 'a' -> 'b').

TUGAS

1. Tugas 1

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)

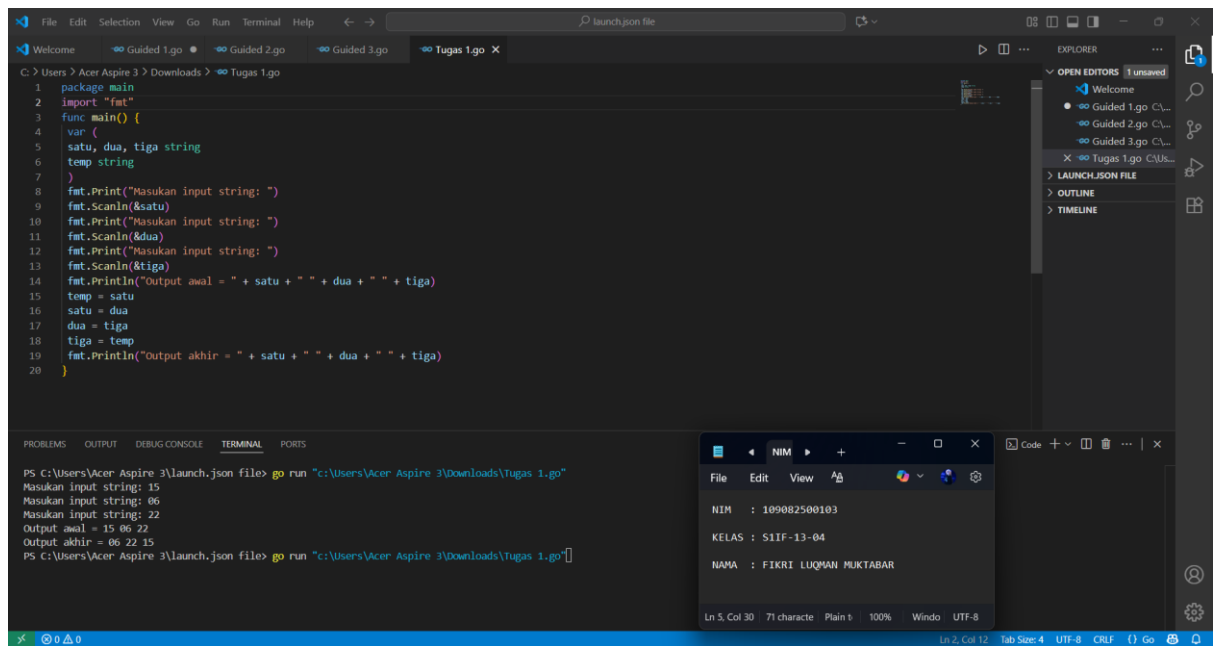
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&tiga)

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

    temp = satu
    satu = dua
    dua = tiga
    tiga = temp

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

Screenshoot Program



The screenshot shows a Go program in a text editor and its execution output in a terminal. The program is a simple input/output exercise. The terminal shows the program being run from a PowerShell prompt, followed by three input strings (15, 06, 22) and the corresponding formatted output strings.

```
1 package main
2 import "fmt"
3 func main() {
4     var (
5         satu, dua, tiga string
6         temp string
7     )
8     fmt.Print("Masukan input string: ")
9     fmt.Scanln(&satu)
10    fmt.Print("Masukan input string: ")
11    fmt.Scanln(&dua)
12    fmt.Print("Masukan input string: ")
13    fmt.Scanln(&tiga)
14    fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)
15    temp = satu
16    satu = dua
17    dua = tiga
18    tiga = temp
19    fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)
20 }
```

```
PS C:\Users\Acer Aspire 3\launch.json file> go run "c:\Users\Acer Aspire 3\Downloads\tugas 1.go"
Masukan input string: 15
Masukan input string: 06
Masukan input string: 22
Output awal = 15 06 22
Output akhir = 06 22 15
PS C:\Users\Acer Aspire 3\launch.json file> go run "c:\Users\Acer Aspire 3\Downloads\tugas 1.go"
```

Deskripsi Program

1) package main

Menyatakan paket main. Program Go yang dieksekusi harus berada pada paket main. Ini memberitahu compiler bahwa ini adalah program yang bisa dijalankan (bukan library).

2) import "fmt"

Mengimpor paket standar fmt (format). fmt menyediakan fungsi untuk input/output ke terminal seperti Println, Printf, Scanln, dan lain-lain.

3) func main() {

Fungsi main adalah titik masuk (entry point) program. Eksekusi dimulai dari sini.

4) var (satu, dua, tiga string; temp string)

Mendeklarasikan empat variabel bertipe string. Nilai awal (zero value) setiap string adalah "" (empty string).

5) fmt.Print("Masukan input string: ")

Menampilkan prompt tanpa newline. (Pengguna akan melihat prompt, lalu mengetik input di baris yang sama.)

6) `fmt.Scanln(&satu)`

Membaca satu token dari `stdin` dan menyimpannya ke `satu`. `Scanln` memisah berdasarkan whitespace dan berhenti pada newline.

7) `fmt.Print("Masukan input string: ")`

Menampilkan prompt tanpa newline. (Pengguna akan melihat prompt, lalu mengetik input di baris yang sama.)

8) `fmt.Scanln(&dua)`

Membaca satu token dari `stdin` dan menyimpannya ke `dua`. `Scanln` memisah berdasarkan whitespace dan berhenti pada newline.

9) `fmt.Print("Masukan input string: ")`

Menampilkan prompt tanpa newline. (Pengguna akan melihat prompt, lalu mengetik input di baris yang sama.)

10) `fmt.Scanln(&tiga)`

Membaca satu token dari `stdin` dan menyimpannya ke `tiga`. `Scanln` memisah berdasarkan whitespace dan berhenti pada newline.

11) `fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)`

Mencetak gabungan string (spasi disisipkan manual).

12) `temp = satu; satu = dua; dua = tiga; tiga = temp`

Merotasi kiri satu langkah menggunakan variabel sementara.

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

Mencetak hasil setelah rotasi.

2. Tugas 2

Source Code

```
package main

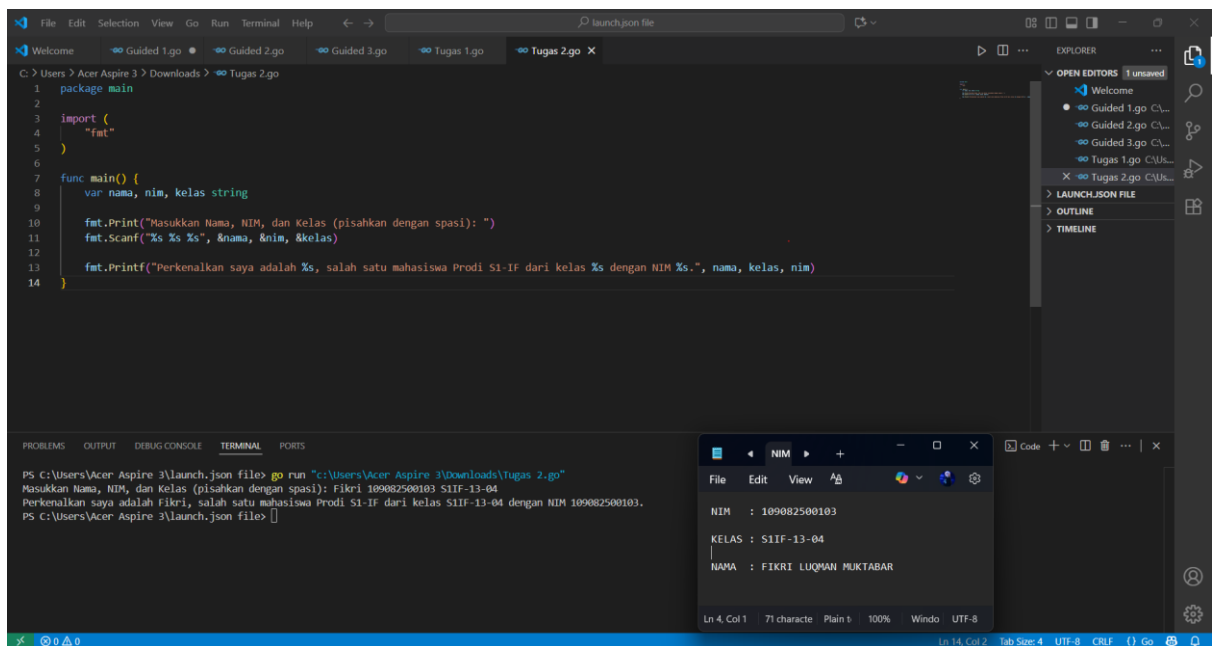
import (
    "fmt"
)

func main() {
    var nama, nim, kelas string

    fmt.Println("Masukkan Nama, NIM, dan Kelas (pisahkan dengan spasi): ")
    fmt.Scanf("%s %s %s", &nama, &nim, &kelas)

    fmt.Printf("Perkenalkan saya adalah %s, salah satu mahasiswa Prodi S1-IF dari kelas %s dengan NIM %s.", nama, kelas, nim)
}
```

Screenshoot Program



Deskripsi Program

1) package main

Menyatakan paket main. Program Go yang dieksekusi harus berada pada paket main. Ini memberitahu compiler bahwa ini adalah program yang bisa dijalankan (bukan library).

2) import "fmt"

Mengimpor paket standar fmt (format). fmt menyediakan fungsi untuk input/output ke terminal seperti Println, Printf, Scanln, dan lain-lain.

3) func main() {

Fungsi main adalah titik masuk (entry point) program. Eksekusi dimulai dari sini.

4) var nama, nim, kelas string

Mendeklarasikan tiga variabel bertipe string. Nilai awalnya "" (kosong).

5) fmt.Print("Masukkan Nama, NIM, dan Kelas (pisahkan dengan spasi): ")

Menampilkan prompt ke pengguna tanpa menambahkan newline (cursor berada di baris yang sama).

6) fmt.Scanf("%s %s %s", &nama, &nim, &kelas)

- Membaca input terformat dari stdin menurut format "%s %s %s"%s membaca **satu token** yang dipisah oleh whitespace (spasi, tab, newline). Jadi setiap %s akan mengambil sampai whitespace berikutnya.

- Argumen harus berupa **pointer** (&nama, &nim, &kelas) supaya fungsi bisa menulis nilai ke variabel

- Scanf sebenarnya mengembalikan dua nilai: n int (jumlah nilai yang berhasil dibaca) dan err error (error jika ada). Kode ini **mengabaikan** kedua nilai tersebut — artinya error tidak ditangani.

- Jika jumlah token di input kurang dari 3, Scanf akan mengembalikan error. Jika ada lebih banyak token, token sisanya **tetap berada di buffer** untuk pembacaan berikutnya.

```
7) fmt.Printf("Perkenalkan saya adalah %, salah satu mahasiswa Prodi S1-IF dari kelas  
%, dengan NIM %s.", nama, kelas, nim)
```

Mencetak string format dengan mengganti setiap %s dengan nilai variabel yang diberikan (urutannya: nama, kelas, nim).

3. Tugas 3

Source Code

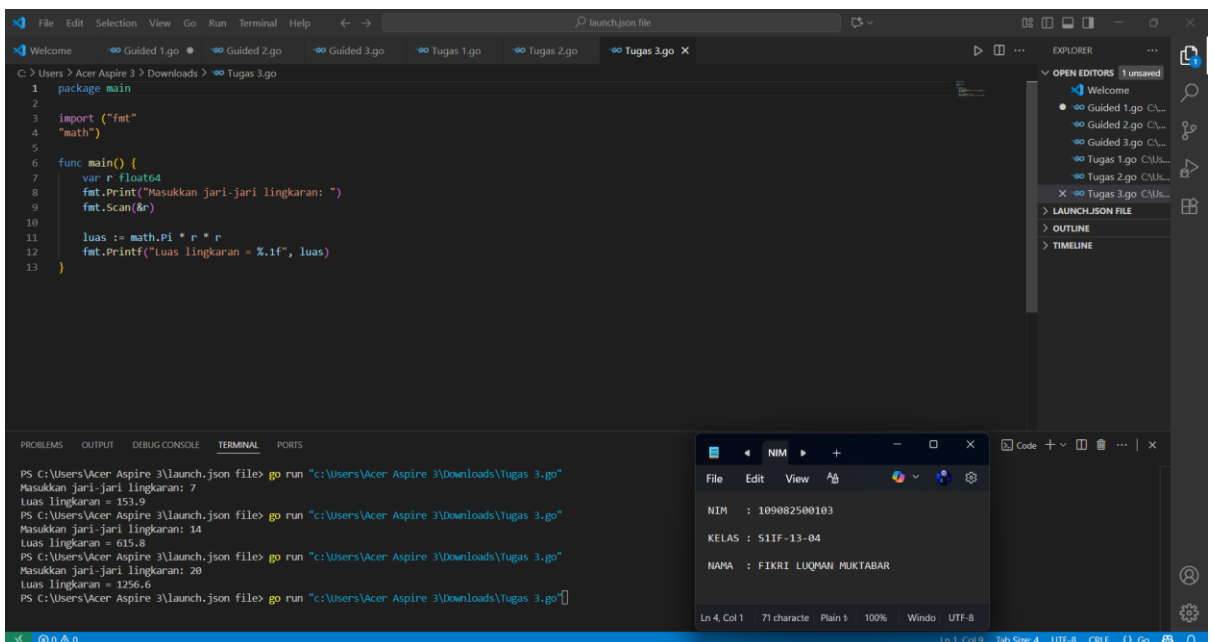
```
package main

import ("fmt"
        "math")

func main() {
    var r float64
    fmt.Print("Masukkan jari-jari lingkaran: ")
    fmt.Scan(&r)

    luas := math.Pi * r * r
    fmt.Printf("Luas lingkaran = %.1f", luas)
}
```

Screenshoot Program



Deskripsi Program

1) package main

Menyatakan paket main. Program Go yang dieksekusi harus berada pada paket main. Ini memberitahu compiler bahwa ini adalah program yang bisa dijalankan (bukan library).

2) import ("fmt" "math")

fmt → digunakan untuk input/output (misalnya Print, Scan, Printf).

math → menyediakan konstanta dan fungsi matematika. Di sini digunakan untuk mendapatkan nilai math.Pi.

3) func main() {

Fungsi main adalah titik masuk (entry point) program. Eksekusi dimulai dari sini.

4) var r float64

Deklarasi variabel r dengan tipe data float64.

r akan menampung **jari-jari lingkaran** yang diinput pengguna.

float64 dipilih karena jari-jari bisa berupa bilangan desimal, bukan hanya integer.

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

Menampilkan pesan ke layar agar pengguna tahu harus memasukkan nilai jari-jari.

6) fmt.Scan(&r)

Membaca input dari keyboard lalu menyimpannya ke variabel r.

&r artinya memberikan **alamat memori** dari r supaya Scan bisa langsung menulis nilai input ke variabel itu.

Scan akan berhenti membaca ketika menemukan whitespace (spasi atau Enter).

7) luas := math.Pi * r * r

Menghitung luas lingkaran menggunakan rumus $\pi \times r^2$.

math.Pi adalah konstanta bawaan Go yang bernilai 3.141592653589793...

Hasil disimpan dalam variabel luas bertipe float64.

```
8) fmt.Printf("Luas lingkaran = %.1f", luas)
```

Printf digunakan untuk mencetak dengan format khusus.

"%.1f" adalah format untuk angka pecahan (floating-point) dengan 1 angka di belakang koma.

Misalnya: 78.539816... akan ditampilkan sebagai 78.5.

4. Tugas 4

Source Code

```
package main

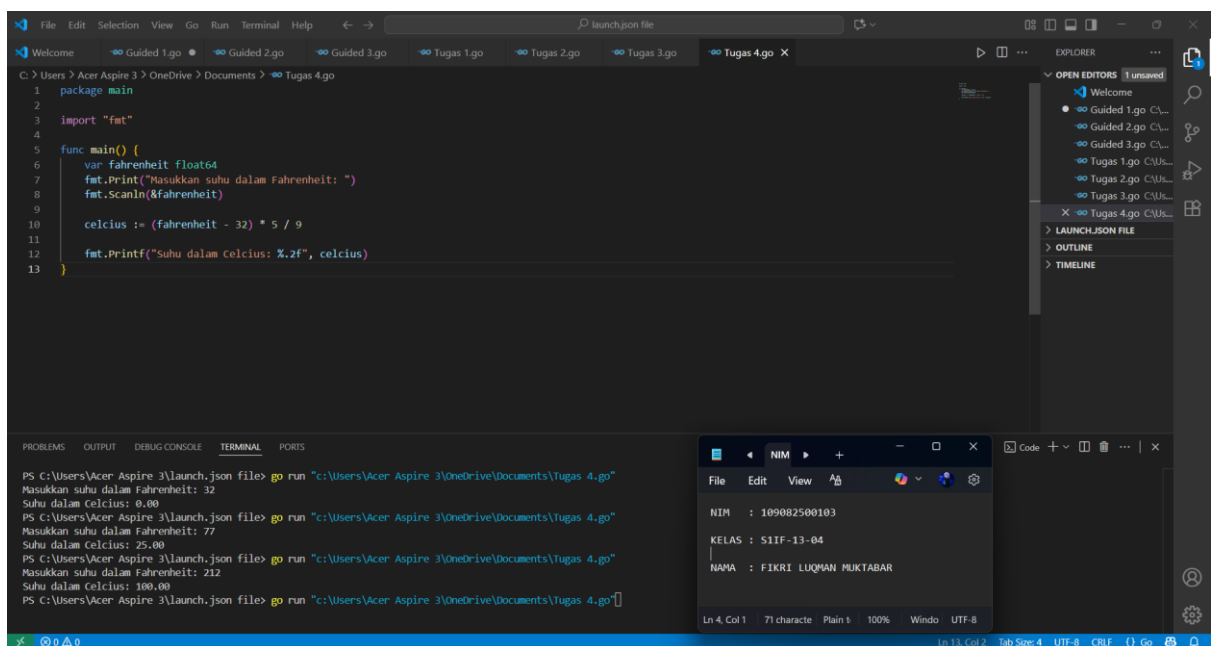
import "fmt"

func main() {
    var fahrenheit float64
    fmt.Print("Masukkan suhu dalam Fahrenheit: ")
    fmt.Scanln(&fahrenheit)

    celcius := (fahrenheit - 32) * 5 / 9

    fmt.Printf("Suhu dalam Celcius: %.2f", celcius)
}
```

Screenshoot Program



Deskripsi Program

1) package main

Menyatakan paket main. Program Go yang dieksekusi harus berada pada paket main. Ini memberitahu compiler bahwa ini adalah program yang bisa dijalankan (bukan library).

2) import "fmt"

Mengimpor paket standar fmt (format). fmt menyediakan fungsi untuk input/output ke terminal seperti Println, Printf, Scanln, dan lain-lain.

3) func main() {

Fungsi main adalah titik masuk (entry point) program. Eksekusi dimulai dari sini

4) var fahrenheit float64

Mendeklarasikan variabel fahrenheit bertipe float64 untuk menampung suhu input (boleh desimal).

5) fmt.Print("Masukkan suhu dalam Fahrenheit: ")

Mencetak prompt ke layar tanpa newline (kursor tetap di baris yang sama).

6) fmt.Scanln(&fahrenheit)

Membaca nilai dari input standar dan menaruhnya ke fahrenheit.

- &fahrenheit adalah pointer; Scanln menulis nilai ke variabel tersebut.
- Scanln mengembalikan (n, err) tetapi di kode ini hasilnya **diabaikan** — jadi jika input bukan angka, program tidak menangani error.

7) celcius := (fahrenheit - 32) * 5 / 9

Menghitung konversi Fahrenheit → Celsius memakai rumus $C = (F - 32) \times \frac{5}{9}$. Karena fahrenheit bertipe float64 dan konstanta numerik di sini bersifat untyped, operasi menghasilkan float64.

8) fmt.Printf("Suhu dalam Celcius: %.2f", celcius)

Mencetak hasil sebagai floating-point dengan **2 angka di belakang koma** (%.2f).

TUGAS PENDAHULUAN

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

- `fmt.Print()`, mencetak apa pun yang masuk ke terminal tanpa menambahkan spasi atau baris baru kecuali jika dikodekan dengan jelas.
- `fmt.Println()`, mirip dengan `fmt.Print()`, perbedaannya adalah menambahkan spasi di antara argumen dan menambahkan baris baru di akhir.
- `fmt.Printf()` menyediakan pemformatan khusus dari string input menggunakan satu atau lebih kata kerja, lalu mencetak string yang diformat ke terminal tanpa menambahkan spasi atau baris baru (kecuali jika dikodekan secara eksplisit).

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

- `int` adalah bilangan bulat yang dapat memiliki nilai nol, positif, dan negatif, tetapi tidak memiliki nilai desimal. Misalnya, 0,5, -10, dan sebagainya. Biasanya kata kunci "`int`" digunakan untuk mendeklarasikan angka integer. Dalam bahasa pemrograman Go, ada dua jenis integer:
 - `Int`: yang terdiri dari bilangan bulat positif dan negatif
 - `Uint`: yang hanya terdiri dari bilangan bulat positif
- `float64`. Digunakan untuk menyimpan nilai dengan titik desimal. Misalnya, 6.7, -7,8, dan sebagainya. Jenis float yang satu ini memiliki ukuran 64 bit (8 byte). Jika kita mendefinisikan variabel float tanpa menentukan ukuran secara jelas, ukuran variabel akan menjadi 64 bit.
- `bool` atau boolean memiliki satu dari dua kemungkinan nilai, yaitu `true` dan `false`. Ditulis dengan menggunakan kata kunci "`bool`".
- `string` adalah serangkaian karakter. Misalnya, "Hello", "Hey there", dan dinyatakan dalam kata kunci "`string`". Dalam bahasa Go, kita menggunakan tanda kutip ganda atau tanda kutip terbalik untuk membuat string.

3. Bagaimana cara mendeklarasikan variabel dengan kata kunci `var` dan dengan cara singkat?

Berikan contoh! Hint "`:=`"

Ketika deklarasi, tipe data yg digunakan harus dituliskan juga. Istilah dari metode deklarasi variabel ini adalah **manifest typing**. Keyword `var` digunakan untuk deklarasi variabel. Berikut skema penggunaan keyword `var`:

```
var <nama-variabel> <tipe-data>
```

```
var <nama-variabel> <tipe-data> = <nilai>
```

Nilai variabel bisa diisi langsung pada saat deklarasi variabel.

Nilai variabel yang pertama diisi langsung ketika deklarasi, berbeda dibanding nama variabel yang nilainya diisi setelah baris kode deklarasi.

Selain *manifest typing*, Go juga mengadopsi konsep **type inference**, yaitu metode deklarasi variabel yang tipe data-nya diketahui secara otomatis dari data/nilai variabel. Cara ini kontradiktif jika dibandingkan dengan cara pertama. Dengan metode jenis ini, keyword `var` dan tipe data tidak perlu ditulis. Pada penggunaan metode ini, `operand =` harus diganti dengan `:=` dan keyword `var` dihilangkan. Diperbolehkan untuk tetap menggunakan keyword `var` pada saat deklarasi meskipun tanpa menuliskan tipe data, dengan ketentuan tidak menggunakan tanda `:=`, melainkan tetap menggunakan `=`. Tanda `:=` hanya digunakan sekali di awal pada saat deklarasi. Untuk assignment nilai selanjutnya harus menggunakan tanda `=`.

4. Apa perbedaan antara operator `==` dan `=` dalam bahasa Go!

Operator `==` (sama dengan) memeriksa apakah kedua operan yang diberikan sama atau tidak. Jika sama, akan mengembalikan nilai `true`. Jika tidak, akan mengembalikan nilai `false`. Misalnya, `5==5` akan mengembalikan `true`. Sedangkan operator `=` (penugasan sederhana) adalah operator penugasan paling sederhana. Operator ini digunakan untuk menetapkan nilai di sebelah kanan ke variabel di sebelah kiri. Misalnya, `go x = 10` yang berarti menetapkan nilai 10 ke variabel `x`.

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

```
package main
```

```
import (  
    "fmt"  
)
```

```
func main() {

    var nama string

    fmt.Print("Masukkan nama Anda: ")
    fmt.Scanln(&nama)

    fmt.Println("Halo,", nama)
}
```

Penjelasannya:

- package main
Menandakan program utama dalam Go. Semua program Go yang bisa dijalankan harus memiliki package main.
- import "fmt"
Mengimpor package fmt yang berisi fungsi untuk input dan output (misalnya Print, Println, Scanln).
- func main() {
Fungsi utama tempat eksekusi program dimulai.
- var nama string
Membuat variabel nama dengan tipe data string untuk menyimpan input dari pengguna.
- fmt.Print("Masukkan nama Anda: ")
Menampilkan teks ke layar agar pengguna tahu harus mengetikkan nama.
- fmt.Scanln(&nama)
Membaca input dari keyboard dan menyimpannya ke variabel nama. Tanda & berarti kita memberikan "alamat memori" dari variabel tersebut supaya fungsi Scanln bisa mengisinya.
- fmt.Scanln(&nama)
Menampilkan hasil output, yaitu teks sapaan beserta nama yang dimasukkan pengguna.