

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 2
I/O TIPE DATA & VARIABEL



Disusun oleh:

ALIF RIZKI PANGESTU

109082530004

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"
)

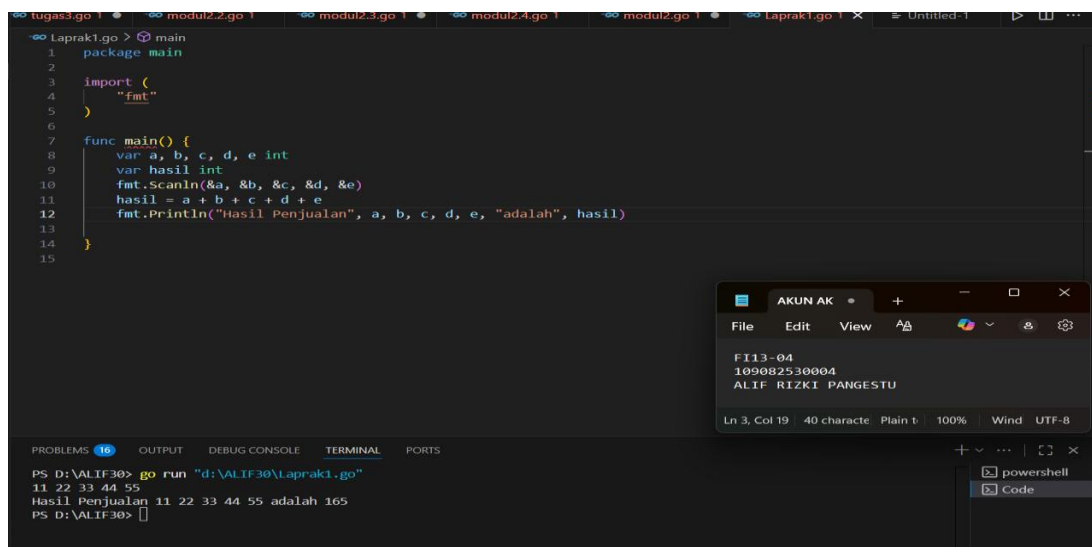
func main() {
    var a, b, c, d, e int
    var hasil int

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

    hasil = a + b + c + d + e

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

Screenshoot program



The screenshot shows a code editor with a Go program and a terminal window. The code is as follows:

```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var a, b, c, d, e int
9     var hasil int
10    fmt.Scanln(&a, &b, &c, &d, &e)
11    hasil = a + b + c + d + e
12    fmt.Println("Hasil Penjualan", a, b, c, d, e, "adalah", hasil)
13 }
14
15
```

The terminal window shows the output of the program:

```
PS D:\ALIF30> go run "d:\ALIF30\Laprak1.go"
11 22 33 44 55
Hasil Penjualan 11 22 33 44 55 adalah 165
PS D:\ALIF30>
```

Deskripsi program

1. package main
 - Menandakan program utama yang bisa dieksekusi.
2. import "fmt"
 - Mengimpor paket fmt untuk fungsi input/output.
3. func main()
 - Titik masuk program.
4. var a, b, c, d, e int
 - Mendeklarasikan lima variabel bertipe int. Nilai awalnya 0 (default).
5. var hasil int
 - Variabel untuk menyimpan jumlah akhir.
6. fmt.Scanln(&a, &b, &c, &d, &e)
 - Program menunggu input dari keyboard. Pengguna harus memasukkan 5 angka (dipisah spasi) lalu tekan Enter.
 - Memberi alamat variabel agar Scanln bisa mengisi nilainya.
 - Catatan penting: Scanln mengembalikan (n int, err error) tetapi kode asli tidak memeriksa error. Jika pengguna memasukkan kurang dari 5 nilai atau memasukkan teks non-angka, Scanln akan menghasilkan error → perilaku menjadi tidak sesuai harapan.
7. hasil = a + b + c + d + e
 - Menjumlahkan kelima variabel (jika ada nilai belum terbaca, nilainya tetap 0).
8. fmt.Println("Hasil Penjualan", a, b, c, d, e, "adalah", hasil)
 - Mencetak semua nilai dan hasil penjumlahan. Println otomatis menambahkan spasi antar argumen dan newline di akhir.

Contoh alur runtime

- Jalankan go run Laprak1.go
- Ketik: 11 22 33 44 55 lalu Enter
- Output:
Hasil Penjualan 11 22 33 44 55 adalah 165

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 a Go IDE with the following components:

- Editor:** Displays the source code of `tugas2.go`, which matches the code provided in the previous block.
- Terminal:** Shows the execution of the program. The first run with input `5` produces the output `5.2`. The second run with input `-23` produces the output `-23`.
- File Explorer:** Shows a file named `FI13-04` with the content `109082530004` and `ALIF RIZKI PANGESTU`.

Deskripsi program

1. `package main`
 - Menandakan bahwa berkas ini adalah program eksekusi utama, bukan sekadar library.
 - File dengan package main harus memiliki fungsi `main()` sebagai titik masuk.
2. `import "fmt"`
 - Mengimpor paket `fmt` (format) yang berisi fungsi-fungsi untuk input/output standar, misalnya `Scan` untuk membaca input dan `Println` untuk mencetak output.
3. `func main() { ... }`
 - Fungsi `main` adalah titik awal eksekusi program Go.
4. `var x, fx float64`
 - Mendeklarasikan dua variabel `x` dan `fx` dengan tipe `float64` (bilangan desimal dengan presisi double).

- Nilai awalnya otomatis 0.0.
- 5. `fmt.Scan(&x)`
 - Menunggu input dari user di standard input (keyboard).
 - `&x` memberikan alamat memori dari variabel `x` supaya fungsi `Scan` bisa menuliskan nilai yang diketik user ke variabel tersebut.
 - Ketika user menekan Enter, Go akan mencoba mem-parsing input menjadi `float64`.
 - Jika user mengetik angka desimal, gunakan titik (misal 5.2) karena Go memakai notasi desimal internasional.
- 6. `fx = 2/(x+5) + 5`
 - Melakukan perhitungan sesuai formula:

$$f(x) = \frac{2}{x+5} + 5$$
 - Operator `/` di Go pada tipe `float64` adalah pembagian floating point.
 - `x` diambil dari input user.
- 7. `fmt.Println(fx)`
 - Mencetak hasil `fx` ke layar, diikuti newline (baris baru).
 - Hasil akan ditampilkan dengan representasi default float (bisa muncul banyak angka di belakang desimal jika tidak diformat).
- 8. Source Code
 1. Program dijalankan
 Contoh: `go run tugas2.go`

 Terminal akan “diam” menunggu input.
 2. User memasukkan angka
 - Misal user mengetik 5 lalu Enter.
 - Nilai `x` menjadi 5.
 3. Perhitungan
 - Komputer menghitung $2/(5+5) + 5$
 - $5+5 = 10$
 - $2/10 = 0.2$
 - $0.2 + 5 = 5.2$
 4. Output ditampilkan 5.2

3. Guided 3

Source Code

```
package main
```

```
import "fmt"

func main() {
    var a, b, c, d, e int

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

    fmt.Printf("%c%c%c%c%c\n", a, b, c, d, e)

    var x, y, z byte

    fmt.Scanf("%c%c%c", &x, &y, &z)

    fmt.Printf("%c%c%c\n", x+1, y+1, z+1)
}
```

Screenshoot program

```

tugas3.go 1
1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, b, c, d, e int
7     fmt.Scan(&a, &b, &c, &d, &e)
8     fmt.Printf("%c%c%c%c%c\n", a, b, c, d, e)
9     var x, y, z byte
10    fmt.Scanf("%c%c%c", &x, &y, &z)
11    fmt.Printf("%c%c%c\n", x+1, y+1, z+1)
12 }
13

PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALIF30> go run "d:\ALIF30\tugas3.go"
66 97 103 117 115 SNO
Bagus
TOP
PS D:\ALIF30>

```

Deskripsi program

- package main

Menandakan program ini adalah program utama yang bisa dieksekusi langsung.

- `import "fmt"`

Menggunakan package `fmt` untuk keperluan input dan output (`Scan`, `Scanf`, `Printf`).

- Fungsi `main`

Semua proses dijalankan di dalam fungsi ini.

- Deklarasi variabel `a`, `b`, `c`, `d`, `e` bertipe `int`.

- `fmt.Scan(&a, &b, &c, &d, &e)`

Meminta pengguna memasukkan 5 bilangan bulat.

- `fmt.Printf("%c%c%c%c%c\n", a, b, c, d, e)`

Mencetak angka-angka tadi sebagai karakter ASCII.

- Contoh: jika input adalah 66 97 103 117 115 maka output menjadi Bagus (karena 66 = 'B', 97 = 'a', 103 = 'g', 117 = 'u', 115 = 's').

- Deklarasi variabel `x`, `y`, `z` bertipe `byte` (setara dengan `uint8`).

- `fmt.Scanf("%c%c%c", &x, &y, &z)`

Meminta input 3 huruf/karakter secara berurutan.

- `fmt.Printf("%c%c%c\n", x+1, y+1, z+1)`

Mencetak karakter berikutnya dari masing-masing huruf.

- Contoh: jika input adalah S N O maka:

- 'S' (ASCII 83) → +1 = 84 → 'T'
- 'N' (ASCII 78) → +1 = 79 → 'O'
- 'O' (ASCII 79) → +1 = 80 → 'P'
- Output: TOP

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

```

modul2.3.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var (
7          satu, dua, tiga string
8          temp          string
9      )
10
11     fmt.Print("Masukan input string : ")
12     fmt.Scanln(&satu)
13     fmt.Print("Masukan input string : ")
14     fmt.Scanln(&dua)
15     fmt.Print("Masukan input string : ")
16     fmt.Scanln(&tiga)
17
18     fmt.Println("Output awal  = ", satu, dua, tiga)
19
20     temp = satu
21     satu = dua
22     dua = tiga
23     tiga = temp
24
25     fmt.Println("Output akhir = ", satu, dua, tiga)
26 }

```

```

PS D:\VALIF30> go run "d:\VALIF30\modul2.3.go"
Masukan input string : A
Masukan input string : B
Masukan input string : C
Output awal = A B C
Output akhir = B C A
PS D:\VALIF30>

```

Deskripsi program

a) package main

- Menandakan bahwa file ini adalah program utama (bukan sekadar library).
- Program dengan package main dapat dijalankan langsung menggunakan go run atau dikompilasi menjadi file binary executable.

b) import ("fmt" "math")

- fmt → paket standar untuk input/output (misalnya Print, Scan, Printf).
- math → paket matematika yang menyediakan konstanta dan fungsi, termasuk math.Pi untuk nilai π (3.1415926535...).

- c) `func main() { ... }`
- Fungsi `main` adalah entry point. Saat program dijalankan, eksekusi dimulai dari sini.
- d) `var r float64`
- Deklarasi variabel `r` dengan tipe `float64`.
- Menyimpan nilai jari-jari lingkaran yang akan dimasukkan oleh pengguna.
- `float64` dipilih karena perhitungan melibatkan bilangan pecahan (desimal).
- e) `fmt.Print("Masukkan jari-jari: ")`
- Menampilkan teks ke layar tanpa menambahkan newline (`\n`).
- Digunakan sebagai prompt agar user tahu apa yang harus diinput.
- f) `fmt.Scan(&r)`
- Membaca input dari keyboard dan menyimpannya ke variabel `r`.
- Tanda `&` (ampersand) berarti memberikan alamat memori variabel `r` agar `Scan` dapat mengisi nilainya.
- User kemudian mengetik angka (misalnya 7) dan menekan Enter.
- g) `luas := math.Pi * r * r`
- Menghitung luas lingkaran dengan rumus $\pi \times r \times r$.
- `math.Pi` adalah konstanta nilai π .
- Operator `*` digunakan untuk perkalian.
- `:=` adalah short variable declaration (deklarasi singkat) untuk membuat variabel `luas` sekaligus memberi nilainya.
- h) `fmt.Printf("%.1f\n", luas)`
- Menampilkan nilai `luas` ke layar dengan format angka desimal 1 digit di belakang koma.
- `%.1f` → format floating point dengan 1 angka setelah titik.
- `\n` → membuat baris baru setelah mencetak.

2. Tugas 2

Source code

```
package main

import "fmt"

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

```

    fmt.Scan(&nama, &nim, &kelas)

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

```

Screenshoot program

```

modul2.2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var nama, nim, kelas string
7
8     fmt.Scan(&nama, &nim, &kelas)
9
10    fmt.Printf("Perkenalkan saya adalah %s, salah satu mahasiswa Prodi S1-IF dari kelas %s dengan NIM %s.\n",
11        nama, kelas, nim)
12 }
13

```

AKUN AK

File Edit View

FI13-04
109082530004
ALIF RIZKI PANGESTU

Ln 3, Col 19 40 character Plain t 100% Wind UTF-8

PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\ALIF30> go run "d:\ALIF30\modul2.2.go"
Bima 1124431414 IF-48-GAB
Perkenalkan saya adalah Bima, salah satu mahasiswa Prodi S1-IF dari kelas IF-48-GAB dengan NIM 1124431414.
PS D:\ALIF30> go run "d:\ALIF30\modul2.2.go"
Yura 1324234545 IFX-48-12
Perkenalkan saya adalah Yura, salah satu mahasiswa Prodi S1-IF dari kelas IFX-48-12 dengan NIM 1324234545.
PS D:\ALIF30>

```

Deskripsi program

- Package Main
- Menandakan file ini adalah program utama (bisa langsung dijalankan).
- Import fmt
- Memanggil package fmt yang berisi fungsi-fungsi input/output seperti Scan dan Printf.
- 2. Fungsi main()


```
func main() {
    var nama, nim, kelas string
```
- Varnama,nim,kelastring
Mendeklarasikan tiga variabel (nama, nim, dan kelas) dengan tipe string untuk menampung data yang dimasukkan pengguna.
- 3. Input Data


```
    fmt.Scan(&nama, &nim, &kelas)
```
- fmt.Scan membaca 3 input string dari user secara berurutan:

1. nama
 2. nim
 3. kelas
- Setiap data dipisahkan dengan spasi atau enter saat mengetik di terminal.

Contoh input:

Bima 1124431414 IF-48-GAB

atau

Bima

1124431414

IF-48-GAB

4. Output Data

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

```
}
```

- `fmt.Printf` mencetak teks dengan format tertentu.
- `%s` adalah placeholder untuk string:
 - `%s` pertama → nama
 - `%s` kedua → kelas
 - `%s` ketiga → nim

Hasilnya berupa kalimat:

Perkenalkan saya adalah <nama>, salah satu mahasiswa Prodi S1-IF dari kelas <kelas> dengan NIM <nim>.

Contoh Jalannya Program

Input di Terminal:

Bima 1124431414 IF-48-GAB

Output:

Perkenalkan saya adalah Bima, salah satu mahasiswa Prodi S1-IF dari kelas IF-48-GAB dengan NIM 1124431414.

3. Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
```

```

var r float64

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

fmt.Scan(&r)

luas := math.Pi * r * r

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

}

```

Screenshoot program

The screenshot shows a Go program in an IDE. The source code is as follows:

```

1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var r float64
10    fmt.Print("Masukkan jari-jari: ")
11    fmt.Scan(&r)
12
13    luas := math.Pi * r * r
14    fmt.Printf("%.1f\n", luas)
15 }
16

```

The terminal output shows the program being run three times with different inputs:

```

PS D:\ALIF30> go run "d:\ALIF30\modul2.go"
Masukkan jari-jari: 7
153.9
PS D:\ALIF30> go run "d:\ALIF30\modul2.go"
Masukkan jari-jari: 14
615.8
PS D:\ALIF30> go run "d:\ALIF30\modul2.go"
Masukkan jari-jari: 20
1256.6
PS D:\ALIF30>

```

The IDE interface also shows a file explorer on the right with a file named "AKUN AK" and a status bar at the bottom indicating the current line and column.

Deskripsi program

- package main
Menandakan file ini adalah program utama yang bisa langsung dieksekusi.
- import "fmt"
Memanggil package fmt untuk input/output (misalnya Print dan Scan).
- import "math"
Digunakan untuk mengakses konstanta dan fungsi matematika, seperti math.Pi.
- var r float64
Mendeklarasikan variabel r bertipe float64 untuk menampung jari-jari lingkaran.
- fmt.Print("Masukkan jari-jari: ")
Menampilkan pesan agar pengguna memasukkan nilai jari-jari.
- fmt.Scan(&r)
Membaca input dari pengguna dan menyimpannya ke variabel r.
- luas := math.Pi * r * r
Menghitung luas lingkaran dengan rumus:

$luas = \pi \times r^2$

Konstanta `math.Pi` merepresentasikan nilai π (3.14159...).

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

Menampilkan hasil perhitungan dengan format 1 angka di belakang koma (`%.1f`).

Contoh Jalannya Program

Input/Output di Terminal:

Masukkan jari-jari: 7

153.9

- Luas lingkaran dengan $r = 7$ adalah sekitar 153.9.

Masukkan jari-jari: 14

615.8

Masukkan jari-jari: 20

1256.6

4. Tugas 4

Source code

```
package main

import (
    "fmt"
)

func main() {
    var fahrenheit int
    fmt.Print("Masukkan suhu dalam Fahrenheit: ")
    fmt.Scan(&fahrenheit)
    celcius := (fahrenheit - 32) * 5 / 9

    fmt.Printf("Suhu dalam Celcius: %d\n", celcius)
}
```

Screenshoot program

```

1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var fahrenheit int
9     fmt.Print("Masukkan suhu dalam Fahrenheit: ")
10    fmt.Scan(&fahrenheit)
11    celcius := (fahrenheit - 32) * 5 / 9
12
13    fmt.Printf("Suhu dalam Celcius: %d\n", celcius)
14 }
15

```

```

PS D:\ALIF30> go run "d:\ALIF30\modul2.4.go"
Masukkan suhu dalam Fahrenheit: 77
Suhu dalam Celcius: 25
PS D:\ALIF30>

```

Deskripsi program

- Package main
- Menandakan file ini adalah program utama yang bisa langsung dijalankan.
- import "fmt"
- Mengimpor package fmt untuk keperluan input/output seperti Print, Scan, dan Printf.
- Var fahrenheit int
- Mendeklarasikan variabel fahrenheit bertipe integer untuk menampung suhu input dalam Fahrenheit.
- `fmt.Print("Masukkan suhu dalam Fahrenheit: ")`
Menampilkan pesan agar pengguna memasukkan nilai suhu Fahrenheit.
- `fmt.Scan(&fahrenheit)`
Membaca input suhu Fahrenheit dari keyboard dan menyimpannya ke variabel fahrenheit. (&fahrenheit adalah alamat memori variabel, diperlukan agar nilai bisa diubah oleh Scan.)
- `celcius := (fahrenheit - 32) * 5 / 9`
Melakukan konversi suhu dari Fahrenheit ke Celcius dengan rumus:

$$C = (F - 32) \times \frac{5}{9}$$

$$C = \frac{(F - 32) \times 5}{9}$$

$$C = 9(F - 32) \times 5$$
 Karena semua variabel bertipe integer (int), operasi pembagian akan menghasilkan bilangan bulat (pembulatan ke bawah bila ada desimal).
- `fmt.Printf("Suhu dalam Celcius: %d\n", celcius)`
Menampilkan hasil konversi dalam bentuk bilangan bulat (%d).
- Contoh Jalannya Program
Input/Output di Terminal:
Masukkan suhu dalam Fahrenheit: 77
Suhu dalam Celcius: 25
Perhitungan:

$$(77 - 32) \times 5 / 9 = 45 \times 5 / 9 = 225 / 9 = 25$$

TUGAS PENDAHULUAN

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

`fmt.Print()` berfungsi sebagai mencetak teks atau nilai ke layar tanpa baris baru otomatis di akhir. Pada outputnya tidak menambahkan spasi otomatis di antara argumen, kecuali kita menuliskannya sendiri, dan jika kita menginginkan baris baru kita harus menambahkan `\n` secara manual.

`fmt.Printf()` mencetak teks dengan format string serta kita juga bisa atur placeholder untuk string, angka, desimal, dan sebagainya. Jika kita menginginkan baris baru harus menambahkan `\n` secara manual, untuk outputnya spasi atau tidaknya antar argumen diatur di format.

`fmt.Println()` berfungsi sebagai Mencetak teks atau nilai ke layar dengan spasi otomatis antar argumen dan baris baru otomatis di akhir, dan sama seperti `fmt.Print()` tidak pakai format placeholder.

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

Integer merupakan tipe data bilangan bulat untuk ukuran bisa 32 bit atau 64 bit tergantung arsitektur, berguna untuk menyimpan angka seperti umur, jumlah barang, dll

Float64 merupakan tipe data pecahan atau decimal dengan presisi 64 bit. Berguna untuk menyimpan nilai pecahan atau decimal seperti, tinggi badan, nilai rata rata, dll

Bool merupakan tipe data Boolean yang hanya menyimpan `true` dan `false`, kegunaannya seperti logika program, kondisi `if/else`, perbandingan

String merupakan tipe data teks, kegunaannya seperti menyimpan nama, alamat, dll

3. Bagaimana cara mendeklarasikan variable dengan kata kunci `var` dan dengan cara singkat? Berikan contoh! Hint `“:=”`.

Dengan penggunaan `var` caranya yaitu, sebutkan `var`, nama variable, tipe data, nilai (opsional)

Dengan deklarasi singkat menggunakan deklarasi + inisialisasi contoh:

- `nama := "Budi"`
- `umur := 20`

4. Apa perbedaan antara operator `==` dan `=` dalam Bahasa Go?

`“=”` digunakan untuk memberikan nilai ke dalam variable, membaca sebagai “diisi dengan” atau “assign”

`“==”` disebut equality cek karena berfungsi sebagai mengecek apakah dua nilai sama yang nantinya menghasilkan `true/false`

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

