

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 4
I/O, TIPE DATA & VARIABEL



Disusun oleh:
LEONARDO FARRIZ GARCYA
109082530036
S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

Tugas Pendahuluan

Soal 1

Pemberi Soal: Muhammad Tegar Wahyu Hida –

Deskripsi Soal: Menghitung Biaya Parkir

Sebuah parkiran mengenakan biaya:

Rp 3.000 per jam pertama

Rp 2.000 per jam berikutnya

Buat program Go untuk menghitung total biaya parkir berdasarkan lama parkir (jam),

dengan pembulatan ke atas (misal: 2 jam 30 menit → dianggap 3 jam).

Input	Output
20 jam	Rp 41000
2.5 jam	Rp 7000
8.5 jam	Rp 19000

Source Code :

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var lama float64

    fmt.Print("Masukkan lama parkir (jam): ")

    fmt.Scan(&lama)

    jam := math.Ceil(lama)

    boolToFloat := func(b bool) float64 {
        return map[bool]float64{true: 1, false: 0}[b]
    }

    biaya := 3000 + (jam-1)*2000*boolToFloat(jam > 1)

    fmt.Printf("Total biaya parkir: Rp %.0f\n", biaya)
}
```

Screenshot Hasil Eksekusi :

The screenshot shows a code editor with a Go file named `pendahuluan 1.1.go`. The code defines a `main` package with a `main` function. It prompts the user for parking duration in minutes, calculates the total fee based on a rate of 3000 per minute plus a flat fee of 2000 for each hour, and prints the result. Below the code editor is a terminal window showing the command `go run "c:\code\pendahuluan 1.1.go"` and its output, which includes sample inputs and outputs for different durations.

```

Go Run Terminal Help ← → code
-∞ tester 2.go 1 -∞ pendahuluan 1.1.go 1 -∞ pendahuluan 1.2.go 1 -∞ guided 2.go 1 -∞ tempCodeRunnerFile.go 1 -∞ guided 3.go 1 -∞ Tugas 1.go 1 -∞ Tugas 2.go 1 ⌂ ...
-∞ pendahuluan 1.1.go > ⌂ main
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var lama float64
10    fmt.Println("Masukkan lama parkir (jam): ")
11    fmt.Scan(&lama)
12
13    jam := math.Ceil(lama)
14
15    boolToFloat := func(b bool) float64 {
16        return map[bool]float64{true: 1, false: 0}[b]
17    }
18
19    biaya := 3000 + (jam-1)*2000*boolToFloat(jam > 1)
20
21    fmt.Printf("Total biaya parkir: Rp %.0f\n", biaya)
22 }

```

PROBLEMS 09 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\code> go run "c:\code\pendahuluan 1.1.go"
Masukkan lama parkir (jam): 20
Total biaya parkir: Rp 41000
PS C:\code> go run "c:\code\pendahuluan 1.1.go"
Masukkan lama parkir (jam): 2.5
Total biaya parkir: Rp 7000
PS C:\code> go run "c:\code\pendahuluan 1.1.go"
Masukkan lama parkir (jam): 8.5
Total biaya parkir: Rp 19000
PS C:\code>

```

F1 13-04
109082530036
LEONARDO FARRIZ GARCYA

Soal 2

Deskripsi Soal:

Kasir menerima pembayaran dan menghitung kembalian otomatis dalam pecahan.

Pecahan uang: 50.000, 20.000, 10.000, 5.000, 2.000, 1.000.

Gunakan / dan %.

Input	Output
total harga: 73000 uang dibayar: 100000	Kembalian: 27000 50000 = 0 lembar 20000 = 1 lembar 10000 = 0 lembar 5000 = 1 lembar 2000 = 1 lembar 1000 = 0 lembar
total harga: 125000 uang dibayar: 200000	Kembalian: 75000 50000 = 1 lembar 20000 = 1 lembar 10000 = 0 lembar

	5000 = 1 lembar 2000 = 0 lembar 1000 = 0 lembar
total harga: 231000 uang dibayar: 300000	Kembalian: 69000 50000 = 1 lembar 20000 = 0 lembar 10000 = 1 lembar 5000 = 1 lembar 2000 = 2 lembar 1000 = 0 lembar

Source Code :

```
package main

import "fmt"

func main() {
    var harga, bayar, kembalian int
    fmt.Print("Masukkan total harga: ")
    fmt.Scan(&harga)
    fmt.Print("Masukkan uang dibayar: ")
    fmt.Scan(&bayar)

    kembalian = bayar - harga
    fmt.Println("\nKembalian:", kembalian)

    fmt.Println("50000 =", kembalian/50000, "lembar")
    kembalian %= 50000

    fmt.Println("20000 =", kembalian/20000, "lembar")
    kembalian %= 20000

    fmt.Println("10000 =", kembalian/10000, "lembar")
    kembalian %= 10000
}
```

```
fmt.Println("5000 =", kembalian/5000, "lembar")
```

kembalian % = 5000

```
fmt.Println("2000 =", kembalian/2000, "lembar")
```

kembalian % = 2000

```
fmt.Println("1000 =", kembalian/1000, "lembar")
```

}

Screenshot Hasil Eksekusi :

The screenshot shows a Microsoft Visual Studio Code (VS Code) window with several tabs open at the top: "go 9+", "pendahuluan 3.2.go 1", "teori 2.go 1", "teori 4.go 1", "guided 1.go 1", "tesster.go 1", "tester 2.go 1", "pendahuluan 1.1.go 1", "pendahuluan 1.2.go 1". The "pendahuluan 1.2.go 1" tab is active, displaying the following Go code:

```
package main
import "fmt"
func main() {
    var harga, bayar, kembalian int
    fmt.Print("Masukkan total harga: ")
    fmt.Scan(&harga)
    fmt.Print("Masukkan uang dibayar: ")
    fmt.Scan(&bayar)

    kembalian = bayar - harga
    fmt.Println("\nkembalian:", kembalian)

    fmt.Println("50000 =", kembalian/50000, "lembar")
    kembalian %= 50000

    fmt.Println("20000 =", kembalian/20000, "lembar")
    kembalian %= 20000

    fmt.Println("10000 =", kembalian/10000, "lembar")
    kembalian %= 10000

    fmt.Println("5000 =", kembalian/5000, "lembar")
    kembalian %= 5000

    fmt.Println("2000 =", kembalian/2000, "lembar")
    kembalian %= 2000

    fmt.Println("1000 =", kembalian/1000, "lembar")
}
```

Below the code, the terminal pane shows the output of running the program:

```
PS C:\code> go run "c:\code\pendahuluan 1.2.go"
Masukkan total harga: 73000
Masukkan uang dibayar: 100000

Kembalian: 27000
50000 = 0 lembar
20000 = 1 lembar
10000 = 0 lembar
5000 = 1 lembar
2000 = 1 lembar
1000 = 0 lembar
PS C:\code> go run "c:\code\pendahuluan 1.2.go"
Masukkan total harga: 125000
Masukkan uang dibayar: 200000

Kembalian: 75000
50000 = 1 lembar
20000 = 1 lembar
10000 = 0 lembar
5000 = 1 lembar
2000 = 0 lembar
1000 = 0 lembar
PS C:\code>
```

At the bottom right, there is a small floating window with the text "F1 13-04", "109082530036", and "LEONARDO FARRIZ GARCYA".

LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {
    var detik int
    fmt.Print("Masukkan jumlah detik: ")
    fmt.Scan(&detik)

    jam := detik / 3600
    menit := (detik % 3600) / 60
    sisaDetik := detik % 60

    fmt.Printf("%d jam %d menit dan %d detik\n", jam,
    menit, sisaDetik)
}
```

Screenshoot program

The screenshot shows a Windows desktop with the Visual Studio Code application open. The code editor displays a Go file named 'laprak 4.1.go' with the following content:

```
package main
import "fmt"

func main() {
    var detik int
    fmt.Println("Masukkan jumlah detik:")
    fmt.Scan(&detik)

    jam := detik / 3600
    menit := (detik % 3600) / 60
    sisaDetik := detik % 60

    fmt.Printf("%d jam %d menit dan %d detik\n", jam, menit, sisaDetik)
}
```

The terminal tab shows the command `go run "c:\code\laprak 4.1.go"` being run, followed by user input and the program's output:

```
PS C:\code> go run "c:\code\laprak 4.1.go"
Masukkan jumlah detik: 3661
1 jam 1 menit dan 1 detik
PS C:\code> go run "c:\code\laprak 4.1.go"
Masukkan jumlah detik: 7322
2 jam 2 menit dan 2 detik
PS C:\code> go run "c:\code\laprak 4.1.go"
Masukkan jumlah detik: 3600
1 jam 0 menit dan 0 detik
PS C:\code>
```

The status bar at the bottom right indicates the date and time as 11/10/2025.

Deskripsi program

1. package main
 - Bawa program ini adalah program utama yang bisa langsung dijalankan.
 - Tanpa main, kode tidak bisa dieksekusi sebagai aplikasi.
2. .import "fmt"
 - Format (fmt), digunakan untuk input/output standar.
 - Seperti "fmt.Scanln" : untuk membaca input dari user.
"fmt.Println" : menampilkan output ke layar.
3. func main () { ... }
 - Fungsi main adalah untuk titik awal eksekusi program Go.
 - Semua intruksi akan berjalan dari fungsi ini.
4. var detik, jam, menit int
 - Menyatakan tiga variabel bertipe bilangan bulat
5. fmt.Scan(&detik)
 - Membaca input dari pengguna dan menyimpannya ke variabel detik.
 - Simbol & artinya program mengambil alamat memori variabel agar nilainya bisa diubah langsung.
6. jam = detik / 3600
 - Menghitung berapa jam dalam jumlah detik yang dimasukkan.
 - Karena 1 jam = 3600 detik, maka total detik dibagi 3600.
7. menit = (detik % 3600) / 60
 - Menghitung jumlah menit setelah jam dihitung.

8. `detik = detik % 60`
 - Mengambil sisa detik setelah menit dihitung.
9. `fmt.Println (jam, "jam", menit, "menit dan", detik, "detik")`
 - Menampilkan hasil akhir ke layar dalam format yang mudah dibaca.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan bilangan 3 digit: ")
    fmt.Scan(&n)

    d1 := n / 100
    d2 := (n / 10) % 10
    d3 := n % 10

    hasil := d1 < d2 && d2 < d3

    fmt.Println(hasil)
}
```

Screenshoot program

```
File Edit Selection View Go Run Terminal Help ← → 🔍 code
EXPLORER ... laprak 1.go 1 Tugas 2.go 1 Tugas pendahuluan.go 1 teori 3.go 1 laprak 4.1.go 1 laprak 4.2.go 1 laprak 4.3.go 1 laprak 4.4.go 1 ...
CODE
Guided 3.2.go 1 laprak 4.2.go > main
Guided 3.3.go 1
Guided 3.4.go 1
Guided 3.5.go 1
Guided 3.6.go 1
Guided 3.7.go 1
guided 3.go 1
laprak 4.1.go 1
laprak 4.2.go 1
laprak 4.3.go 1
laprak 4.4.go 1
laprak 4.5.go 1
laprak 4.6.go 1
Modul 4.2.go 1
Modul 4.3.go 1
Modul 4.3.go 1
tempCodeKunne... 1
teori 2.go 1
teori 3.go 1
teori 3.2.go 1
teori 3.3.go 1
OUTLINE
TIMELINE
GO
env code
GOENV=C:\Users\...
GOMOD=NUL
GOTOOCHAIN=a...
tools
PACKAGE OUTLINE
ZTC Berawan
PROBLEMS 0 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\code> go run "c:\code\laprak 4.2.go"
Masukkan bilangan 3 digit: 362
false
PS C:\code> go run "c:\code\laprak 4.2.go"
Masukkan bilangan 3 digit: 256
true
PS C:\code> go run "c:\code\laprak 4.2.go"
Masukkan bilangan 3 digit: 189
true
PS C:\code>
F1 13-04
109082530036
LEONARDO FARRIZ GARYA
File Edit View H1 ...
Ln 3, Col 17 44 character ⌂ For 100% Window UTF-8
Ln 12, Col 18 Tab Size: 4 UTF-8 CR/LF ⌂ Go ⌂ 1.25.1 ⌂
ENG US ⌂ 11/10/2025
```

Deskripsi program

1. package main
 - Program ini adalah program utama yang bisa langsung dijalankan.
 - Tanpa main, kode tidak bisa dieksekusi sebagai aplikasi.
2. import "fmt"
 - Format (fmt), digunakan untuk input/output standar.
 - Seperti "fmt.Scanln" : untuk membaca input dari user.
"fmt.Println" : menampilkan output ke layar.
3. func main () { ... }
 - Fungsi main adalah titik awal eksekusi program Go.
 - Semua intruksi akan berjalan dari fungsi ini.
4. var n int
 - Menyatakan variabel n bertipe integer (bilangan bulat)
5. fmt.Print("Masukkan bilangan 3 digit: ")
 - Menampilkan teks di layar untuk meminta input dari pengguna.
6. fmt.Scan(&n)
 - Membaca input bilangan dari pengguna dan menyimpannya ke variabel n.
7. d1 := n / 100
 - Mengambil digit pertama (ratusan) dari bilangan.

8. $d2 := (n / 10) \% 10$
- Mengambil digit kedua (puluhan) dari bilangan.
9. $d3 := n \% 10$
- Mengambil digit ketiga (satuan) dari bilangan.
10. $hasil := d1 < d2 \&& d2 < d3$
- Mengecek apakah digit-digit bilangan naik secara berurutan.
- Jika keduanya benar, maka hasil bernilai true.
- Jika tidak, maka false.
11. `fmt.Println(hasil)`
Menampilkan hasil perbandingan ke layar (true atau false).

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var berat, tinggi float64
    fmt.Print("Masukkan berat (kg) dan tinggi (m): ")
    fmt.Scan(&berat, &tinggi)
    bmi := berat / (tinggi * tinggi)
    fmt.Printf("%.2f", bmi)
}
```

Screenshoot program

```
File Edit Selection View Go Run Terminal Help ← → 🔍 code
EXPLORER ... Tugas 1.go 1 Tugas 2.go 1 Tugas pendahuluan.go 1 teori 3.4.go 1 laprak 4.1.go 1 laprak 4.2.go 1 laprak 4.3.go 1 laprak 4.4.go 1
CODE
Guided 3.2.go 1
Guided 3.3.go 1
Guided 3.4.go 1
Guided 3.5.go 1
Guided 3.6.go 1
Guided 3.7.go 1
guided 3.go 1
laprak 4.1.go 1
laprak 4.2.go 1
laprak 4.3.go 1
laprak 4.4.go 1
laprak 4.5.go 1
laprak 4.6.go 1
Modul 4.2.go 1
Modul 4.3.go 1
Modul 4.3.go 1
tempCodeRunnun... 1
teori 2.go 1
teori 3.1.go 1
teori 3.2.go 1
teori 3.3.go 1
OUTLINE
TIMELINE
GO env GOENV=C:\Users\...
GOMOD=NUL
GOTOOCHAIN=a...
tools
PACKAGE OUTLINE
Hari hujan yang... 27°C
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
File Edit View H1 ...
F1 13-04
109082530036
LEONARDO FARRIZ GARYA
Ln 3, Col 17 44 character ⌂ For 100% Window UTF-8
Ln 12, Col 1 Tab Size: 4 UTF-8 CR/LF ⌂ Go ⌂ 1.25.1 ⌂
ENGLISH ⌂ 11/10/2025
PS C:\code> go run "c:\code\laprak 4.3.go"
Masukkan berat (kg) dan tinggi (m): 70 1.75
22.86
PS C:\code> go run "c:\code\laprak 4.3.go"
Masukkan berat (kg) dan tinggi (m): 60 1.6
23.44
PS C:\code> go run "c:\code\laprak 4.3.go"
Masukkan berat (kg) dan tinggi (m): 80 1.8
24.69
PS C:\code>
```

Deskripsi program

1. package main

- Program ini adalah program utama yang bisa langsung dijalankan.
- Program ini adalah program utama yang bisa langsung dijalankan.

2. import "fmt"

- Format (fmt), digunakan untuk input/output standar.
- Seperti "fmt.Scanln" : untuk membaca input dari user.
"fmt.Println" : menampilkan output ke layar.

3. func main () { ... }

- Fungsi main adalah titik awal eksekusi program Go.
- Semua intruksi akan berjalan dari fungsi ini.

4. var berat, tinggi float64

Menyatakan dua variabel dalam bilangan desimal

berat : untuk menyimpan berat badan dalam kilogram.
tinggi :untuk menyimpan tinggi badan dalam meter.

5. fmt.Println("Masukkan berat (kg) dan tinggi (m): ")

- Menampilkan teks ke layer agar pengguna tahu apa yang akan diinput

6. fmt.Scan (&berat, &tinggi)

- Membaca **dua input angka** dari pengguna, dan menyimpannya ke variabel berat dan tinggi.

7. `bmi := berat / (tinggi * tinggi)`

- Rumus BMI

8. `fmt.Printf("%.2f", bmi)`

- Menampilkan hasil BMI ke layar dengan 2 angka di belakang koma.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var totalAwal, diskon int

    fmt.Print("Masukkan total belanja: ")

    fmt.Scan(&totalAwal)

    fmt.Print("Masukkan besar diskon (%): ")

    fmt.Scan(&diskon)

    totalAkhir := totalAwal - (totalAwal * diskon / 100)

    fmt.Println("Total setelah diskon:", totalAkhir)
}
```

Screenshoot program

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. The left sidebar lists several Go source files: Guided 3.2.go, Guided 3.3.go, Guided 3.4.go, Guided 3.5.go, Guided 3.6.go, Guided 3.7.go, guided 3.go, laprak 4.1.go, laprak 4.2.go, laprak 4.3.go, laprak 4.4.go, laprak 4.5.go, laprak 4.6.go, Modul 4.2.go, Modul 4.3.go, Modul 4.3.1.go, tempCodeRunnne..., teori 2.go, teori 3.1.go, teori 3.2.go, and teori 3.3.go. The main editor area displays the content of the laprak 4.4.go file:

```
package main
import "fmt"
func main() {
    var totalAwal, diskon int
    fmt.Println("Masukkan total belanja: ")
    fmt.Scan(&totalAwal)
    fmt.Println("Masukkan besar diskon (%): ")
    fmt.Scan(&diskon)

    totalAkhir := totalAwal - (totalAwal * diskon / 100)
    fmt.Println("Total setelah diskon:", totalAkhir)
}
```

Below the editor is a terminal window titled "F1 13-04" showing the execution of the program:

```
PS C:\code> go run "c:\odelalaprap\4.4.go"
Masukkan total belanja: 100000
Masukkan besar diskon (%): 10
Total setelah diskon: 90000
PS C:\code> go run "c:\odelalaprap\4.4.go"
Masukkan total belanja: 200000
Masukkan besar diskon (%): 20
Total setelah diskon: 160000
PS C:\code> go run "c:\odelalaprap\4.4.go"
Masukkan total belanja: 150000
Masukkan besar diskon (%): 15
Total setelah diskon: 127500
PS C:\code>
```

The taskbar at the bottom shows various pinned icons, including Microsoft Edge, File Explorer, and File History.

Deskripsi program

1. package main

- Program ini adalah program utama yang bisa langsung dijalankan.
- Program ini adalah program utama yang bisa langsung dijalankan.

2. import "fmt"

- Format (fmt), digunakan untuk input/output standar.
- Seperti "fmt.Scanln" : untuk membaca input dari user.
"fmt.Println" : menampilkan output ke layar.

3. func main () { ... }

- Fungsi main adalah untuk titik awal eksekusi program Go.
- Semua intruksi akan berjalan dari fungsi ini.

4. var totalAwal, diskon int

- menyatakan dua variabel bertipe integer (bilangan bulat)
totalAwal : menyimpan total harga belanja sebelum diskon.
Diskon : menyimpan besar diskon dalam persen (%).

5. fmt.Println("Masukkan total belanja: ")

- Menampilkan teks agar pengguna tahu harus menginputkan total belanja.

6. fmt.Scan(&totalAwal)

- Membaca input pengguna dan menyimpannya ke variabel totalAwal

7. fmt.Println("Masukkan besar diskon (%): ")

- Menampilkan teks agar pengguna tahu harus menginputkan diskon dalam persen.
8. `fmt.Scan(&diskon)`
 - Membaca dan menyimpan nilai diskon dari input pengguna
 9. `totalAkhir := totalAwal - (totalAwal * diskon / 100)`
 - Menghitung total setelah diskon dengan rumus:
 10. `fmt.Println("Total setelah diskon:", totalAkhir)`
 - Menampilkan hasil ke layar

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var bmi, tinggi float64
    fmt.Scan(&bmi, &tinggi)
    berat := bmi * (tinggi * tinggi)

    fmt.Printf("%.0f\n", berat)
}
```

Screenshoot program

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. The left sidebar displays a file tree with several Go files (laprak 4.1.go, laprak 4.2.go, laprak 4.3.go, laprak 4.4.go, laprak 4.5.go, etc.) and a 'GOENV' entry. The main editor area contains the following Go code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bmi, tinggi float64
7     fmt.Scan(&bmi, &tinggi)
8     berat := bmi * (tinggi * tinggi)
9
10    fmt.Printf("%.2f\n", berat)
11 }
12
```

Below the editor is a terminal window titled 'F1 13-04' showing command-line output:

```
PS C:\code> go run "c:/code/laprak 4.5.go"
22.85 1.75
70
PS C:\code> go run "c:/code/laprak 4.5.go"
23.43 1.6
60
PS C:\code> go run "c:/code/laprak 4.5.go"
24.69 1.8
80
PS C:\code>
```

The taskbar at the bottom includes icons for File Explorer, Task View, Start, Search, and various system icons. The system tray shows the date and time as 11/10/2025.

Deskripsi program

1. package main

- Program ini adalah program utama yang bisa langsung dijalankan.
- Program ini adalah program utama yang bisa langsung dijalankan.

2. import "fmt"

- Format (fmt), digunakan untuk input/output standar.
- Seperti "fmt.Scanln" : untuk membaca input dari user.
"fmt.Println" : menampilkan output ke layar.

3. func main() {

- Fungsi main adalah titik awal eksekusi program Go.
- Semua intruksi akan berjalan dari fungsi ini.

4. var berat, tinggi float64

- Menyatakan dua variabel dengan float64 (bilangan desimal):
berat = berat badan
tinggi = tinggi badan

5. fmt.Scan(&berat, &tinggi)

- Membaca dua input dari pengguna, dan menyimpannya ke variabel berat dan tinggi.

6. fmt.Printf("%.2f", berat/(tinggi*tinggi))

- Menghitung BMI (Body Mass Index) dengan rumus.
- ("%.2f") : Menampilkan 2 angka di belakang koma

3. Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var x1, y1, x2, y2, x3, y3 float64
    fmt.Scan(&x1, &y1)
    fmt.Scan(&x2, &y2)
    fmt.Scan(&x3, &y3)

    ab := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
    bc := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
    ca := math.Sqrt(math.Pow(x3-x1, 2) + math.Pow(y3-y1, 2))

    max := math.Max(ab, math.Max(bc, ca))
    fmt.Printf("%.2f\n", max)
}
```

Screenshoot program

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. The left sidebar displays a file tree with several Go source files: laprak 4.3.go, laprak 4.4.go, laprak 4.6.go, teori 3.3.go, teori 3.1.go, teori 3.2.go, Tugas 3.go, and Tugas 4.go. The main editor area contains the following Go code:

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var x1, y1, x2, y2, x3, y3 float64
    fmt.Scan(&x1, &y1)
    fmt.Scan(&x2, &y2)
    fmt.Scan(&x3, &y3)

    ab := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
    bc := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
    ca := math.Sqrt(math.Pow(x3-x1, 2) + math.Pow(y3-y1, 2))

    max := math.Max(ab, math.Max(bc, ca))
    fmt.Printf("%.2f\n", max)
}
```

Below the editor is a terminal window titled "F1 13-04" showing the command PS C:\code> go run "c:/code/laprak_4.6.go" and its output: 1.0 1.0, 3.0 5.0, 1.0 1.0, 4.47, 109082530036, LEONARDO FARRIZ GARCIA. The bottom status bar indicates the file is 1.25.1 and the date is 11/10/2025.

Deskripsi program

1. package main

- Program ini adalah program utama yang bisa langsung dijalankan.
- Program ini adalah program utama yang bisa langsung dijalankan.

2. Import "fmt"

- Format (fmt), digunakan untuk input/output standar.
- Seperti "fmt.Scanln" : untuk membaca input dari user.
"fmt.Println" : menampilkan output ke layar.

3. func main () {...}

- Fungsi main adalah untuk titik awal eksekusi program Go.
- Semua intruksi akan berjalan dari fungsi ini.

4. var x1, y1, x2, y2, x3, y3 float64

- Menyatakan enam variabel bertipe float64(bilangan desimal) untuk menyimpan koordinat tiga titik:

Titik A : (x1, y1)

Titik B : (x2, y2)

Titik C : (x3, y3)

5. fmt.Scan(&x1, &y1) dst.

- Membaca input koordinat dari pengguna. Program ini akan meminta tiga pasang nilai dan menyimpan nya sesuai Alamat.

6. ab := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2)) dst.

- Menghitung jarak antara ketiga titik(tergantung berapa yang di inginkan) menggunakan rumus jarak Euclidean.
- Rumus tersebut menghitung jarak dari titik 1 ke titik ke 2 (titik A dan B, titik B dan C, titik C dan A).
- math.Pow : digunakan untuk menghitung pangkat
- math.Sqrt : digunakan untuk akar kuadrat

7. `fmt.Printf("%.2f\n", max)`

- Menampilkan hasil (jarak terpanjang) dengan 2 angka di belakang koma ("`%2f\n`")