

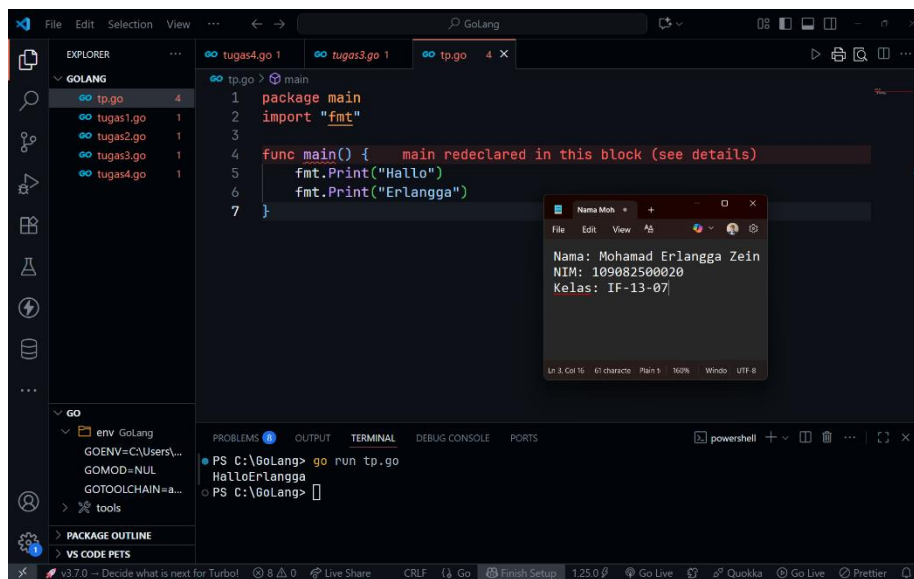
Tugas Pendahuluan Algoritma dan Pemrograman Minggu ke-2: “Input/Output Tipe Data dan Variabel”

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

- **`fmt.Print()`**

Fungsi `fmt.Print()` adalah untuk menampilkan program dengan output an nya menggunakan tipe data string, dan ketika membuat perintah `fmt.Print()` lagi di baris selanjutnya, maka output nya akan menyambung, tidak ada spasi.

Contoh:



The screenshot shows the VS Code editor with a Go file named `tp.go`. The code is as follows:

```
1 package main
2 import "fmt"
3
4 func main() {
5     fmt.Print("Hallo")
6     fmt.Print("Erlangga")
7 }
```

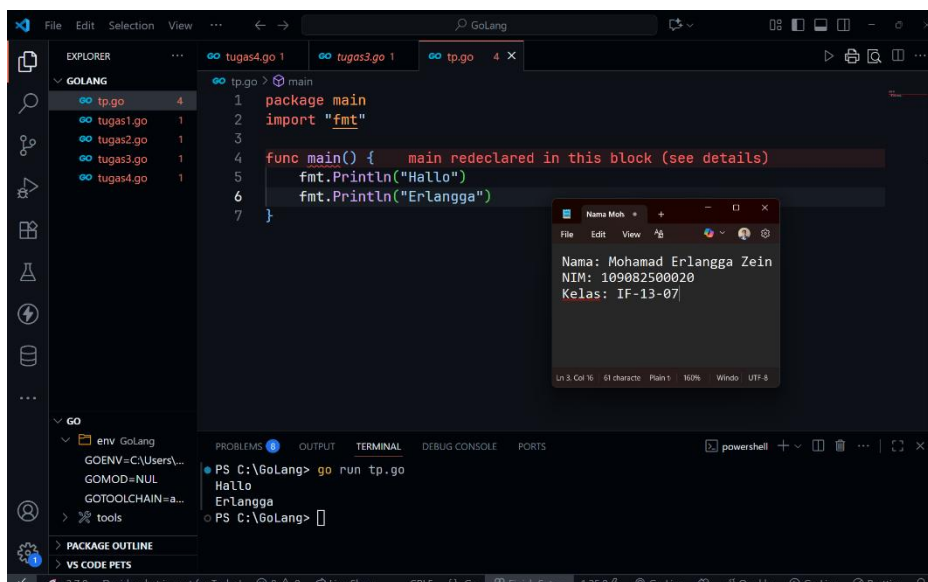
The output window shows the result of running the program:

```
Nama: Mohamad Erlangga Zein
NIM: 109082500020
Kelas: IF-13-07
```

- **`fmt.Println()`**

Fungsi `fmt.Println()` sama saja seperti pada `fmt.Print()`, namun dalam output an nya, Ketika anda membuat `fmt.Println()` baru maka hasil output an nya akan berjarak satu baris/line.

Contoh:



The screenshot shows the VS Code editor with a Go file named `tp.go`. The code is as follows:

```
1 package main
2 import "fmt"
3
4 func main() {
5     fmt.Println("Hallo")
6     fmt.Println("Erlangga")
7 }
```

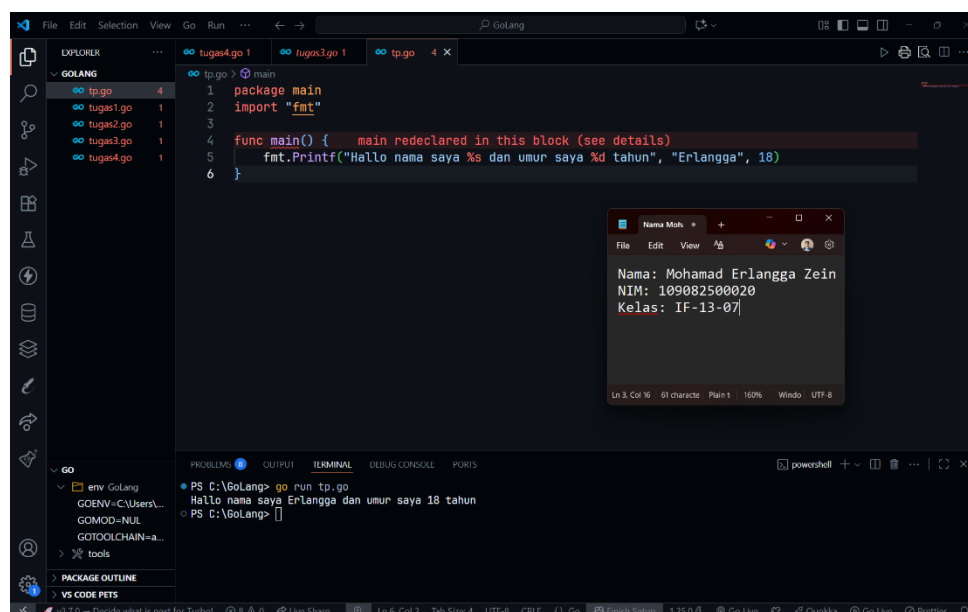
The output window shows the result of running the program:

```
Nama: Mohamad Erlangga Zein
NIM: 109082500020
Kelas: IF-13-07
```

- **fmt.Printf()**

Fungsi `fmt.Printf()` sama seperti fungsi *"print"* pada umumnya, namun pada `fmt.Printf()` Ketika ingin memasukkan tipe data atau ingin memspesifikasikan suatu tipe data yang akan digunakan pada program Go, maka haruslah diberi semacam simbol khusus, dan tidak sembarangan symbol, seperti contohnya ketika ingin menambahkan tipe data `int` pada suatu perintah `fmt.Printf()` maka harus menambahkan symbol dengan *"%d"* yang berartikan *int decimal integer value*, atau biasa disebut digit, dan ada berbagai simbol khusus lainnya seperti *"%s"* untuk tipe data string, *"%f"* untuk tipe data float, *"%t"* untuk tipe data Boolean dan lain lain, dan untuk penempatan atau pengisian data yang ingin kita isi, harus berada setelah *".."* lalu berikan tanda koma(,) lalu isikan datanya sesuai dengan formater yang kita isi di `fmt.Printf()`.

Contoh:



The screenshot shows a Go program in a text editor. The code is as follows:

```
1 package main
2 import "fmt"
3
4 func main() {
5     fmt.Printf("Halo nama saya %s dan umur saya %d tahun", "Erlangga", 18)
6 }
```

The program is executed in a terminal window, showing the output:

```
PS C:\GoLang> go run tp.go
Halo nama saya Erlangga dan umur saya 18 tahun
PS C:\GoLang>
```

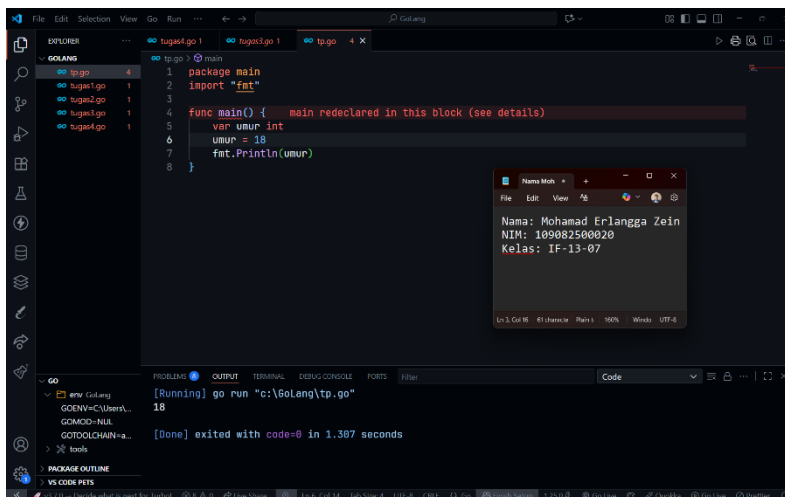
The output is displayed in a separate window titled "Nama Moh.", showing the formatted string and the integer value.

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

- **Tipe data `int`**

Tipe data `int/integer` digunakan untuk menginput data dengan isiannya itu number/angka, tipe data `int` terbagi atas `int32` dan `int64`, bukan tanpa alasan, tetapi itu biasanya dikelompokkan berdasarkan system komputer kita itu berbasis 32bit atau 64bit dan tiap tiap value/nilai nya itu berbeda, disesuaikan dengan kemampuan system yang kita gunakan, namun biasanya kebanyakan orang langsung menggunakan `int` karena ia langsung otomatis mendeteksi systemnya, biasanya `int` digunakan untuk operasi matematika atau pengisian biodata seperti umur, tetapi `int` tidak bisa menginput bilangan decimal, nilai default `int` adalah 0.

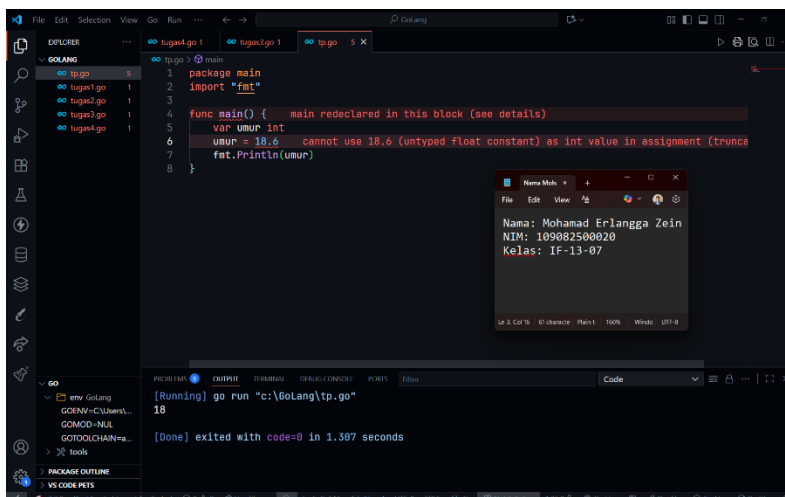
Contoh penggunaan tipe data int:



```
1 package main
2 import "fmt"
3
4 func main() {
5     var umur int
6     umur = 18
7     fmt.Println(umur)
8 }
```

Output: 18

Bukti bahwa int tidak bisa digunakan untuk bilangan decimal:



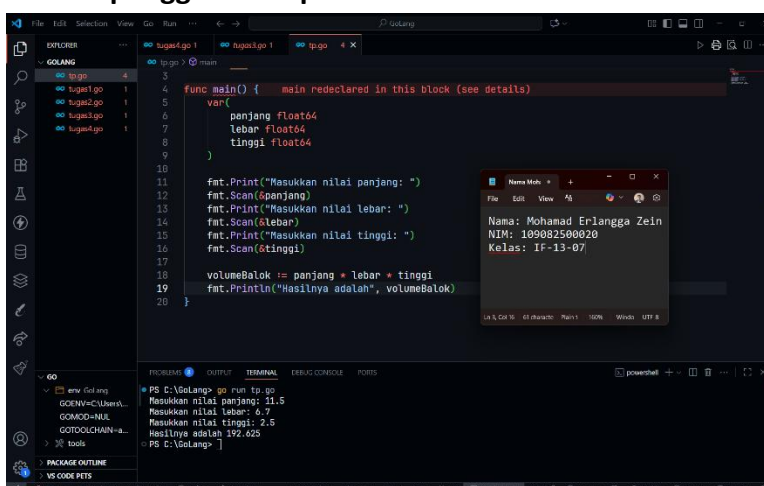
```
1 package main
2 import "fmt"
3
4 func main() {
5     var umur int
6     umur = 18.6
7     fmt.Println(umur)
8 }
```

Output: 18

- **Tipe data float64**

Tipe data float64 digunakan untuk menginput suatu data dengan data tersebut menghasilkan bilangan decimal atau bilangan berkoma, biasa digunakan dalam operasi matematika lanjutan, bisa juga digunakan untuk menentukan fungsi dari sebuah rumus bangun ruang atau bangun datar, float64 sangat cocok untuk operasi perhitungan Tingkat lanjut bahkan bisa digunakan untuk perhitungan rumus.

Contoh penggunaan tipe data float64:



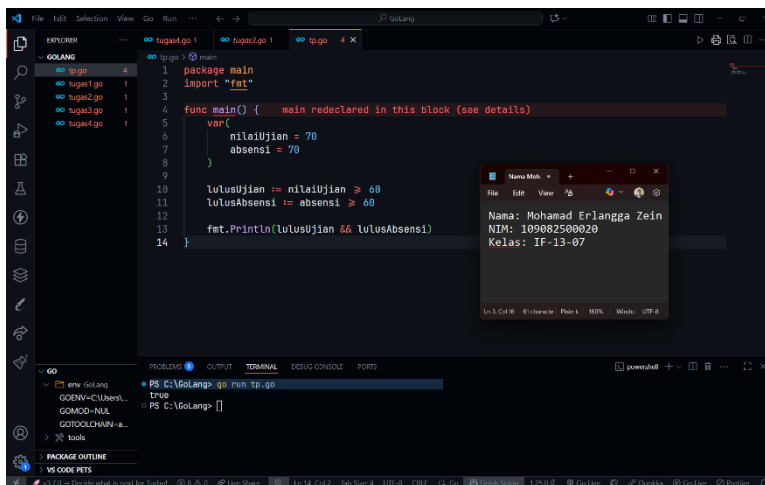
```
1 package main
2 import "fmt"
3
4 func main() {
5     var
6     panjang float64
7     lebar float64
8     tinggi float64
9
10    fmt.Print("Masukkan nilai panjang: ")
11    fmt.Scan(&panjang)
12    fmt.Print("Masukkan nilai lebar: ")
13    fmt.Scan(&lebar)
14    fmt.Print("Masukkan nilai tinggi: ")
15    fmt.Scan(&tinggi)
16
17    volumeBalok := panjang * lebar * tinggi
18    fmt.Println("Hasilnya adalah", volumeBalok)
19 }
20
```

Output: 192.625

- **Tipe data bool/Boolean**

Tipe data bool/Boolean digunakan untuk menyatakan suatu program bernilai benar atau salah, hanya pernyataan untuk menyatakan bahwa program tersebut bernilai true/false, atau dalam bilangan biner 1/0, 1 berartikan benar 0 berartikan salah, tipe data Boolean menggunakan logika and, or, xor pada logika matematika, dan dalam golang kita bisa menggunakan symbol &(untuk and), |(untuk or), ^(untuk xor), dan agar program bisa berjalan diperlukan deklarasi & sebanyak dua kali "&" untuk suatu program yang mengandung unsur perbandingan/menyatakan suatu keputusan antara benar dan salah/true or false.

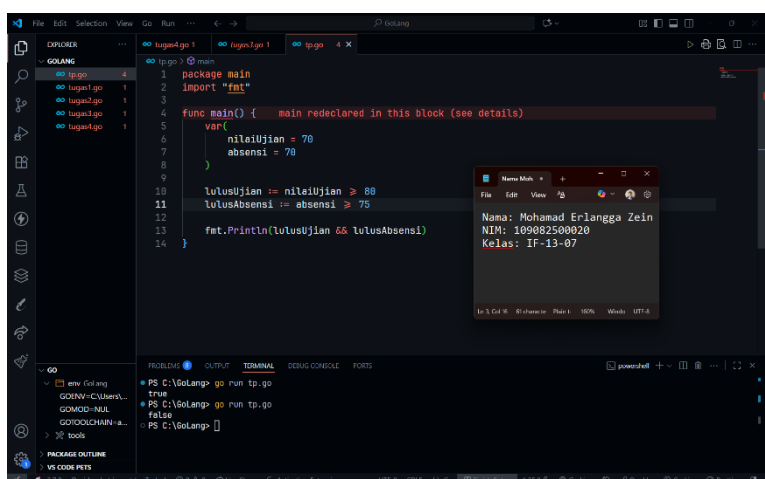
Contoh penggunaan tipe data bool/Boolean:



```
1 package main
2 import "fmt"
3
4 func main() {
5     nilaiUjian = 70
6     absensi = 70
7 }
8
9
10 lulusUjian := nilaiUjian >= 60
11 lulusAbsensi := absensi <= 60
12
13 fmt.Println(lulusUjian && lulusAbsensi)
14 }
```

Terminal output: true

**contoh Ketika program bernilai true, disini saya menggunakan studi kasus nilai minimum untuk syarat kelulusan, disini bernilai true ketika memenuhi syarat, saya menggunakan operator and(&&).*



```
1 package main
2 import "fmt"
3
4 func main() {
5     nilaiUjian = 70
6     absensi = 70
7 }
8
9
10 lulusUjian := nilaiUjian >= 60
11 lulusAbsensi := absensi <= 75
12
13 fmt.Println(lulusUjian && lulusAbsensi)
14 }
```

Terminal output: false

**ketika program bernilai false karena tidak memenuhi syarat.*

- **Tipe data string**

Tipe data string digunakan untuk menghasilkan sebuah output yang bentuknya sebuah kata/kalimat pada pemrograman golang, tipe data string biasanya diketikkan dengan symbol “...” atau `...`.

Contoh penggunaan tipe data string:

```

1 package main
2 import "fmt"
3
4 func main() {
5     var nama = "erlangga"
6     fmt.Println(nama)
7
8     var nama2 = "zein"
9     fmt.Println(nama2)
10 }

```

Terminal Output:

```

[Running] go run "c:\GoLang\tp.go"
erlangga
zein
[Done] exited with code=0 in 1.171 seconds

```

Output Window:

```

Nama: Mohamad Erlangga Zein
NIM: 109082500020
Kelas: IF-13-07

```

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

Mendaklarasikan variable dengan menggunakan symbol := Adalah cara singkat untuk mendeklarasikan suatu program agar lebih simple dan tidak bertele-tele, dengan menggunakan “:=” kita bisa langsung mengisi nilai tipe datanya tanpa harus di deklarasikan seperti “var nama string” dengan menggunakan “:=” kita bisa langsung memasukkan nilai dari variabelnya dan langsung mendeteksinya tanpa error.

Contoh:

```

1 package main
2 import "fmt"
3
4 func main() {
5     nama := "Erlangga"
6     umur := 18
7     tinggiBadan := 178.5
8
9     fmt.Println("Halo perkenalkan namaku", nama, "Umurku:", umur, "Tahun",
10     "Tinggi Badanku:", tinggiBadan, "cm")
11 }

```

Terminal Output:

```

[Running] go run "c:\GoLang\tp.go"
Halo perkenalkan namaku Erlangga Umurku: 18 Tahun Tinggi Badanku: 178.5 cm
[Done] exited with code=0 in 1.101 seconds

```

Output Window:

```

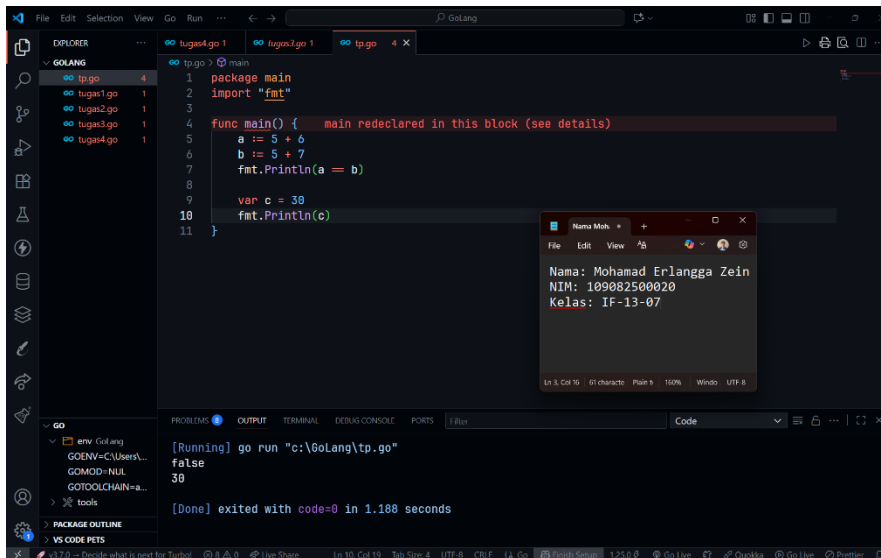
Nama: Mohamad Erlangga Zein
NIM: 109082500020
Kelas: IF-13-07

```

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

Perbedaan fungsi antara operator == dan = dalam Bahasa go Adalah dalam kegunaannya, operator “==” berfungsi sebagai operator perbandingan, gunanya untuk membandingkan suatu variable dengan variable lainnya, operator ini juga biasanya untuk menentukan true/false suatu perbandingan antar variable, sedangkan operator “=” berfungsi sebagai operator penugasan, dimana berguna dalam menentukan atau mengisi nilai dalam suatu variable.

Contoh:



The screenshot shows a Go program in VS Code. The code defines a `main` package and imports `fmt`. Inside the `main` function, it declares and assigns values to variables `a`, `b`, and `c`. `a` is assigned 5, `b` is assigned 6, and `c` is assigned 30. The program then prints the value of `a` (5) and the result of the comparison `a == b` (false). Finally, it prints the value of `c` (30). The output window shows the execution results: `5` and `false` on the first line, and `30` on the second line.

```
1 package main
2 import "fmt"
3
4 func main() {
5     a := 5
6     b := 6
7     fmt.Println(a == b)
8
9     var c = 30
10    fmt.Println(c)
11 }
```

Output:

```
5
false
30
```

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

Source Code:

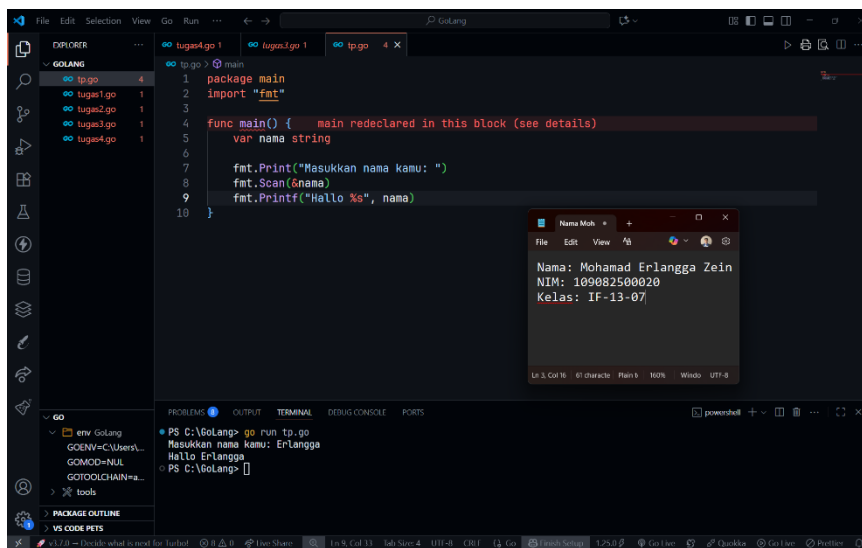
```
package main

import "fmt"

func main() {
    var nama string

    fmt.Print("Masukkan nama kamu: ")
    fmt.Scan(&nama)
    fmt.Printf("Hallo %s", nama)
}
```

Screensshot program:



```
1 package main
2 import "fmt"
3
4 func main() {
5     var nama string
6
7     fmt.Print("Masukkan nama kamu: ")
8     fmt.Scan(&nama)
9     fmt.Printf("Halo %s", nama)
10 }
```

Terminal Output:

```
PS C:\GoLang> go run tp.go
Masukkan nama kamu: Erlangga
Halo Erlangga
PS C:\GoLang>
```

Deskripsi program:

Pada program tersebut mula mula kita membuat variable dengan tipe data string, contoh disini saya mengisi dengan “var nama string”, kemudian untuk menampilkan inputnya, kita deklarasikan dengan `fmt.Print()`, kemudian gunakan `fmt.Scan()` untuk mengambil variable nya, lalu untuk menampilkan output, gunakan `fmt.Printf()` karena variable nya hanya terdeklarasikan “var nama string” sehingga belum ada nilainya, hingga saya menggunakan `fmt.Printf()` sekalian untuk mengisi format variable nya. Dan hasilnya seperti pada screenshot diatas.