

# Tugas Pendahuluan Algoritma dan Pemograman

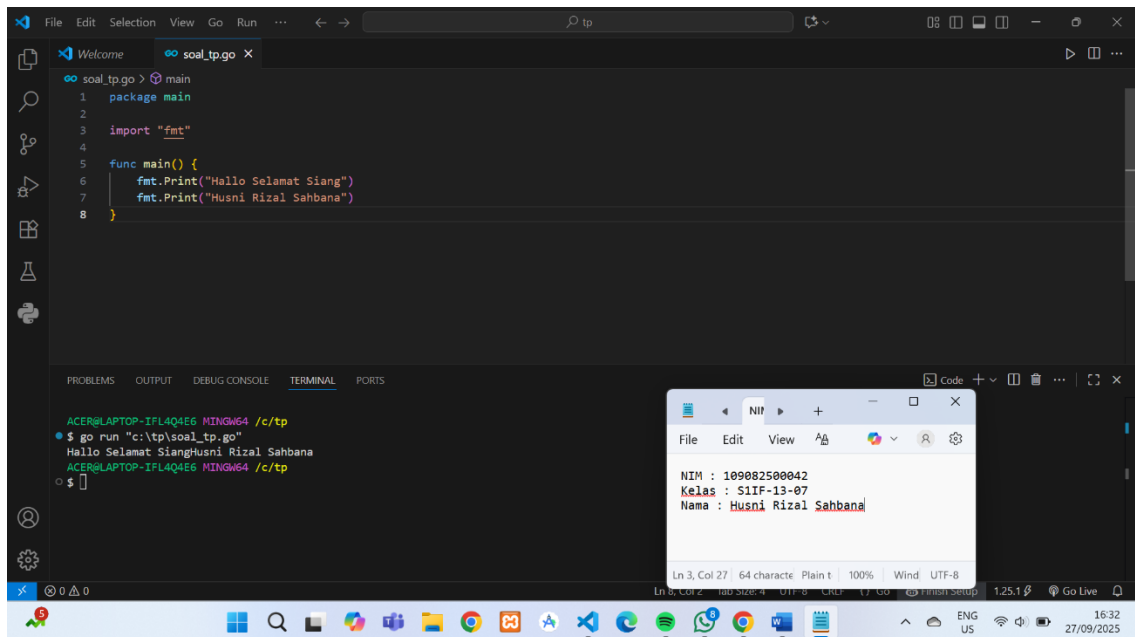
## Modul 2 “Input/Output Tipe Data dan Variable”

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

#### - `fmt.Print()`

`fmt.Print()` berfungsi untuk mencetak teks atau variable apa adanya, tanpa adanya spasi tambahan dan tanpa newline (enter) di akhir, jadi kalo kita mencetak beberapa nilai, maka teks tersebut bakal nempel langsung (kecuali kalo dikasih spasi secara manual).

Contoh :



The screenshot shows a Go program in a code editor and its output in a terminal. The code defines a `main` function that uses `fmt.Print()` to output two strings. The terminal output shows the two strings concatenated without a newline.

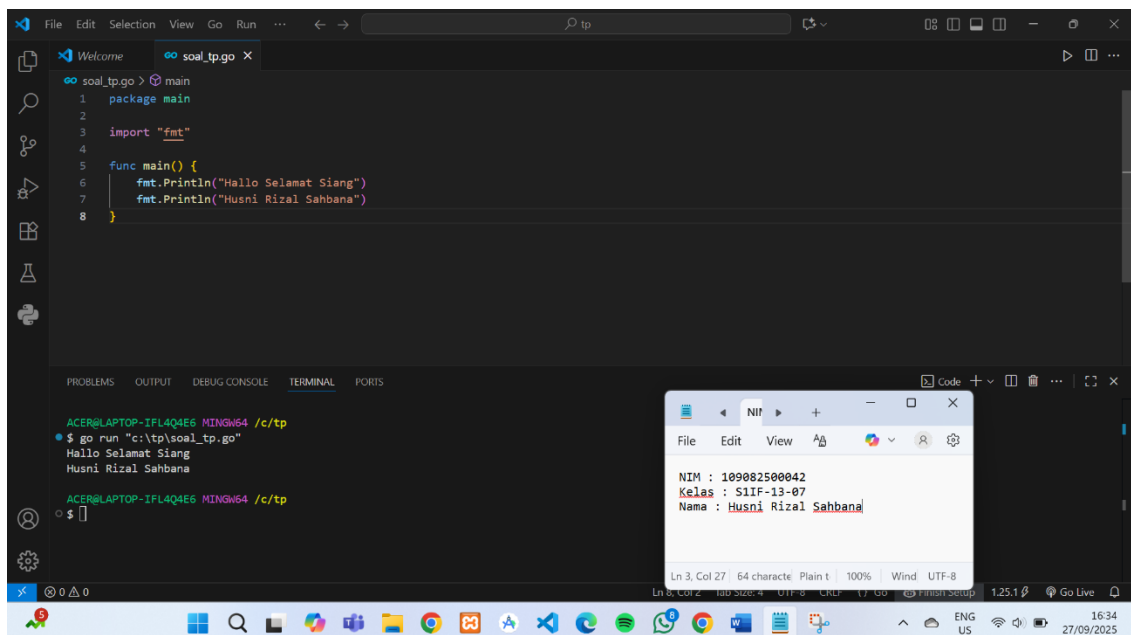
```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Print("Hallo Selamat Siang")
7     fmt.Print("Husni Rizal Sahbana")
8 }
```

```
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp
$ go run "c:\tp\soal_tp.go"
Hallo Selamat SiangHusni Rizal Sahbana
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp
$
```

#### - `fmt.Println()`

Fungsi dari `fmt.Println()` adalah untuk mencetak teks, menambahkan spasi di antara baris kalimat atau menambahkan baris baru di akhir output.

Contoh :



The screenshot shows a Go program in a code editor and its output in a terminal. The code defines a `main` function that uses `fmt.Println()` to output two strings. The terminal output shows the two strings on separate lines, with a newline character added after each string.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("Hallo Selamat Siang")
7     fmt.Println("Husni Rizal Sahbana")
8 }
```

```
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp
$ go run "c:\tp\soal_tp.go"
Hallo Selamat Siang
Husni Rizal Sahbana
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp
$
```

- **fmt.Printf()**

fmt.Printf() itu sebenarnya adalah bawaan Go dari package fmt yang digunakan untuk mencetak output dengan format tertentu. Printf itu sama dengan Print with Format. Dan kita bisa mengatur tampilan outputnya menggunakan format verb seperti :

%s → string

%d → integer

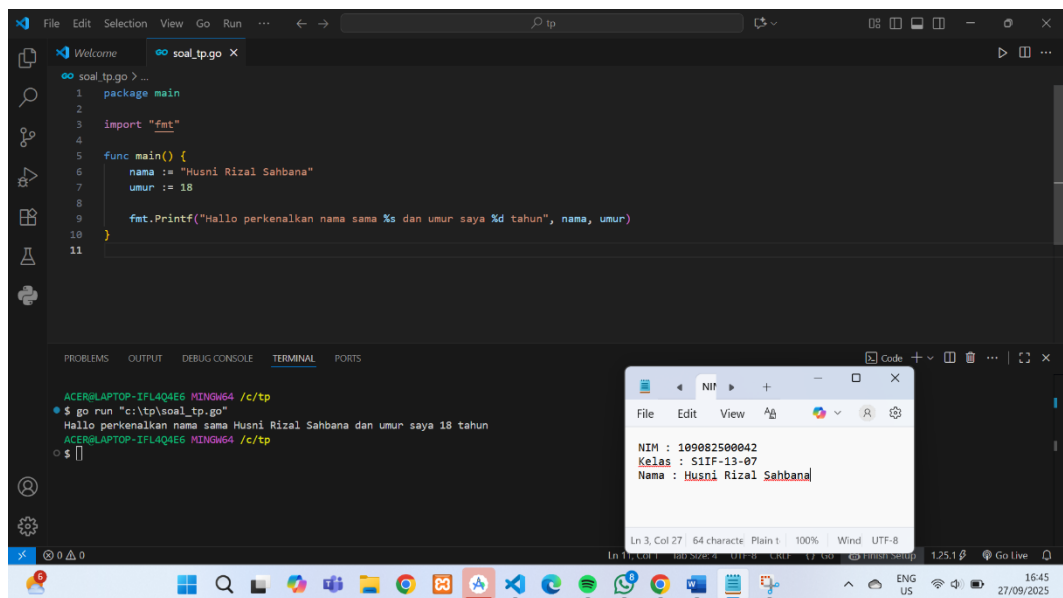
%f → float

%t → boolean

%v → nilai default (general)

Bedanya dengan Print() dan Println(), di Printf() kita harus kasih **format string** biar output sesuai yang diinginkan.

Contoh :



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     nama := "Husni Rizal Sahbana"
7     umur := 18
8
9     fmt.Printf("Halo perkenalkan nama sama %s dan umur saya %d tahun", nama, umur)
10 }
11
```

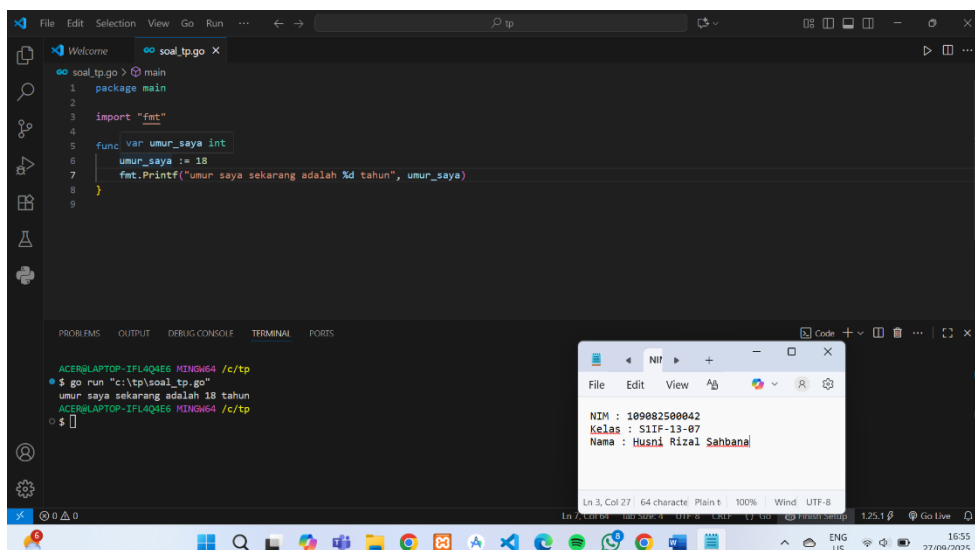
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp  
\$ go run "c:\tp\soal\_tp.go"  
Halo perkenalkan nama sama Husni Rizal Sahbana dan umur saya 18 tahun  
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp  
\$

## 2. Jelaskan Penggunaan Tipe Data int, float64, bool, dan string di Go!

- **Tipe Data Int**

Tipe data int (Integer) adalah tipe data untuk menyimpan bilangan bulat (positif, negative, ataupun nol). Di go integer itu ada banyak jenis nya (int8, int16, int32, int64), tapi paling sering dipakai cukup int.

Contoh :



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var umur_saya int
7     umur_saya := 18
8     fmt.Printf("umur saya sekarang adalah %d tahun", umur_saya)
9 }

```

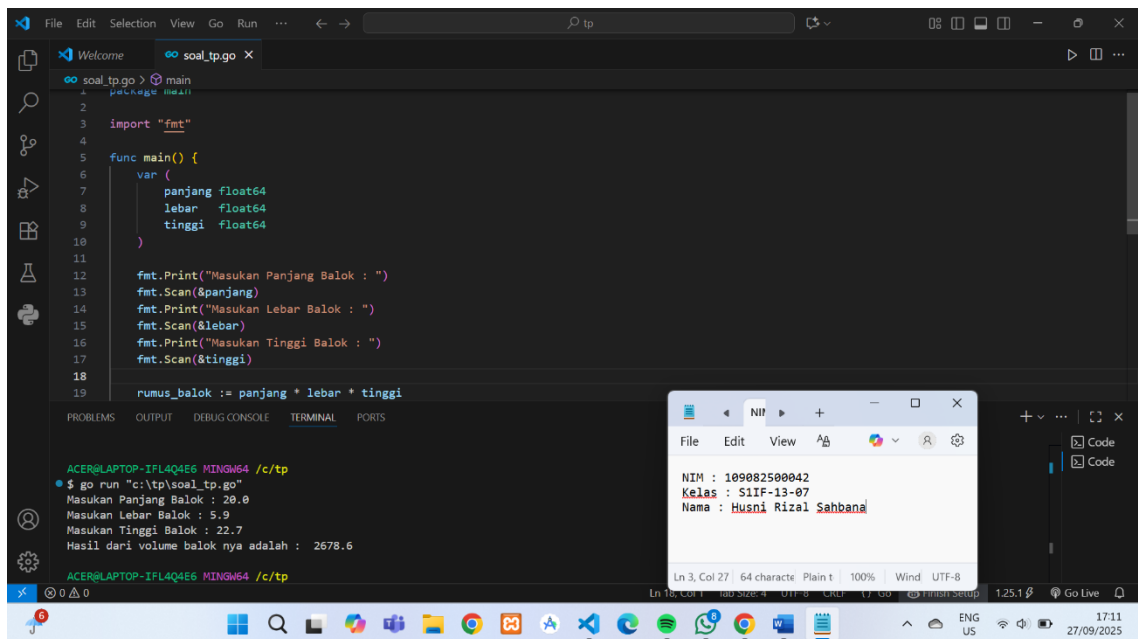
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp  
\$ go run "c:\tp\soal\_tp.go"  
umur saya sekarang adalah 18 tahun  
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp  
\$

## - Tipe Data Float

Jadi tipe data `float` ini adalah untuk menyimpan bilangan pecahan atau decimal, di go ada beberapa jenis float ada float 32 ada pula float 64, namun float64 itu ;lebih umum di pakai karena menawarkan Tingkat presisi yang tinggi dan rentang nilai yang lebih luas dibanding float32.

Tipe float ini cocok untuk tipe jenis angka yang ada koma nya, misalnya rata rata, tinggi badan, dan gaji.

Contoh :

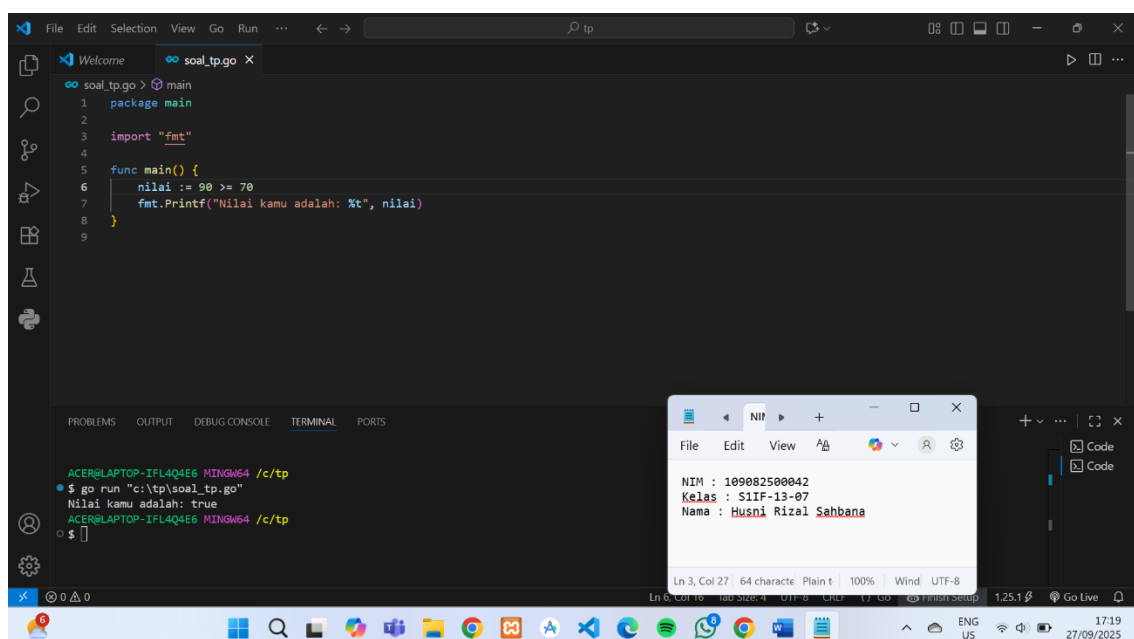


```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var (
7         panjang float64
8         lebar    float64
9         tinggi   float64
10    )
11
12    fmt.Print("Masukan Panjang Balok : ")
13    fmt.Scan(&panjang)
14    fmt.Print("Masukan Lebar Balok : ")
15    fmt.Scan(&lebar)
16    fmt.Print("Masukan Tinggi Balok : ")
17    fmt.Scan(&tinggi)
18
19    rumus_balok := panjang * lebar * tinggi
20
21    fmt.Println("Hasil dari volume balok nya adalah : ", rumus_balok)
22 }
```

ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp  
\$ go run "c:\tp\soal\_tp.go"  
Masukan Panjang Balok : 20.0  
Masukan Lebar Balok : 5.9  
Masukan Tinggi Balok : 22.7  
Hasil dari volume balok nya adalah : 2678.6

## - Tipe Data Boolean

Tipe data Boolean itu menyimpan nilai logika yang hanya bisa bernilai true = benar atau false = salah, dan biasanya tipe data Boolean itu dipakai untuk kondisi (if, loop, dsb).



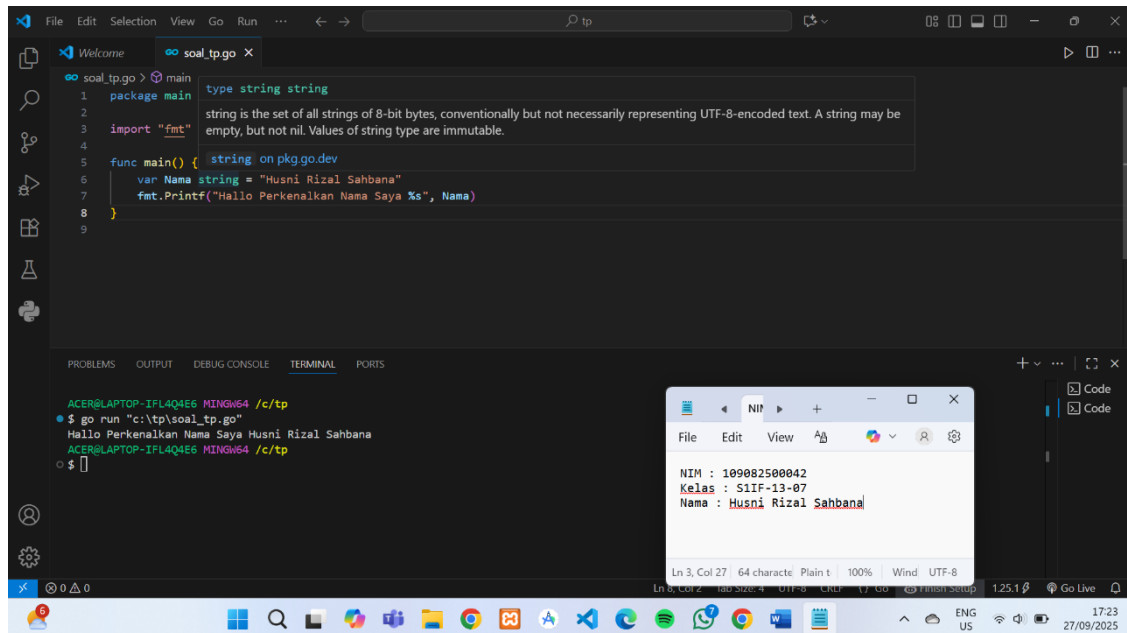
```
1 package main
2
3 import "fmt"
4
5 func main() {
6     nilai := 90 >= 70
7     fmt.Printf("Nilai kamu adalah: %t", nilai)
8 }
9
```

ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp  
\$ go run "c:\tp\soal\_tp.go"  
Nilai kamu adalah: true

### - Tipe Data String

Jadi tipe data string ini berfungsi untuk menyimpan teks/Kumpulan kalimat, dan tipe data string ini ditulis di antara kutip ganda ""

Contoh :



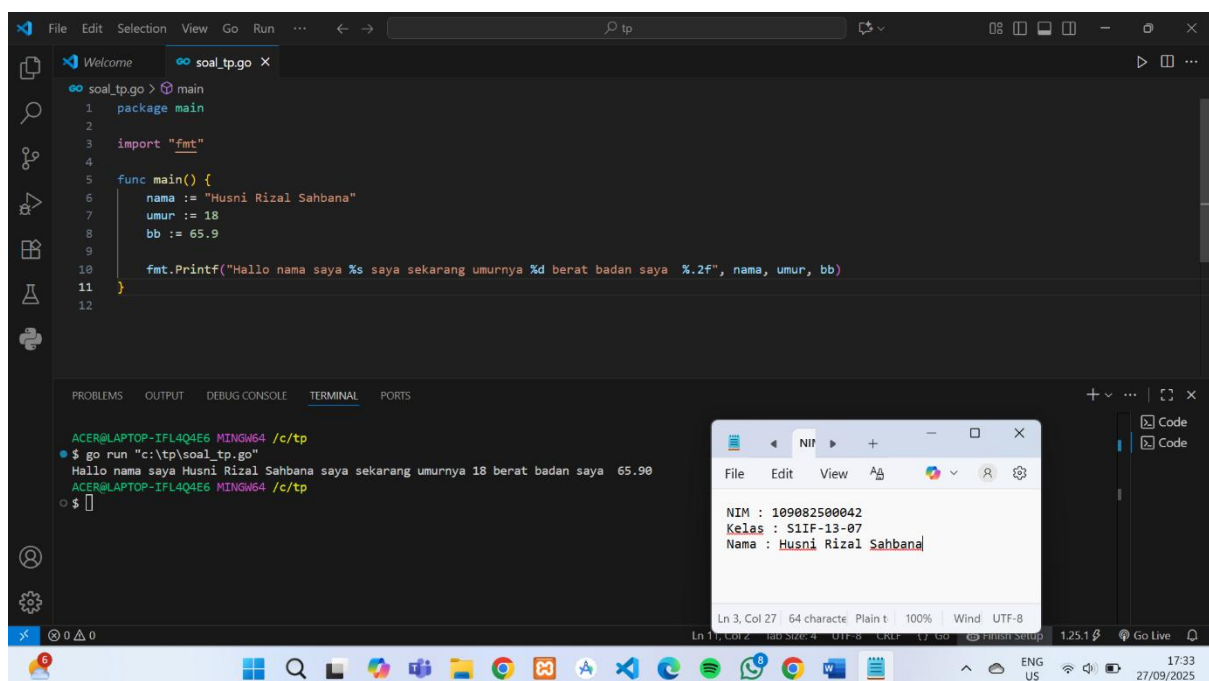
```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var Nama string = "Husni Rizal Sahbana"
7     fmt.Printf("Halo Perkenalkan Nama Saya %s", Nama)
8 }
9
```

ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp  
\$ go run "c:\tp\soal\_tp.go"  
Halo Perkenalkan Nama Saya Husni Rizal Sahbana  
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp  
\$

### 3. Bagaimana cara mendeklarasikan variabel dengan kata kunci VAR dan dengan acara singkat? Berikan contoh! Hint ":=".

Dengan menggunakan := itu lebih singkat dan praktis, lebih simple juga dibanding menggunakan var, dengan menggunakan := itu kita bisa langsung mengisi value atau nilai nya tanpa harus mendeklarasikan nya terlebih dahulu contohnya seperti "var umur int".

Contoh :



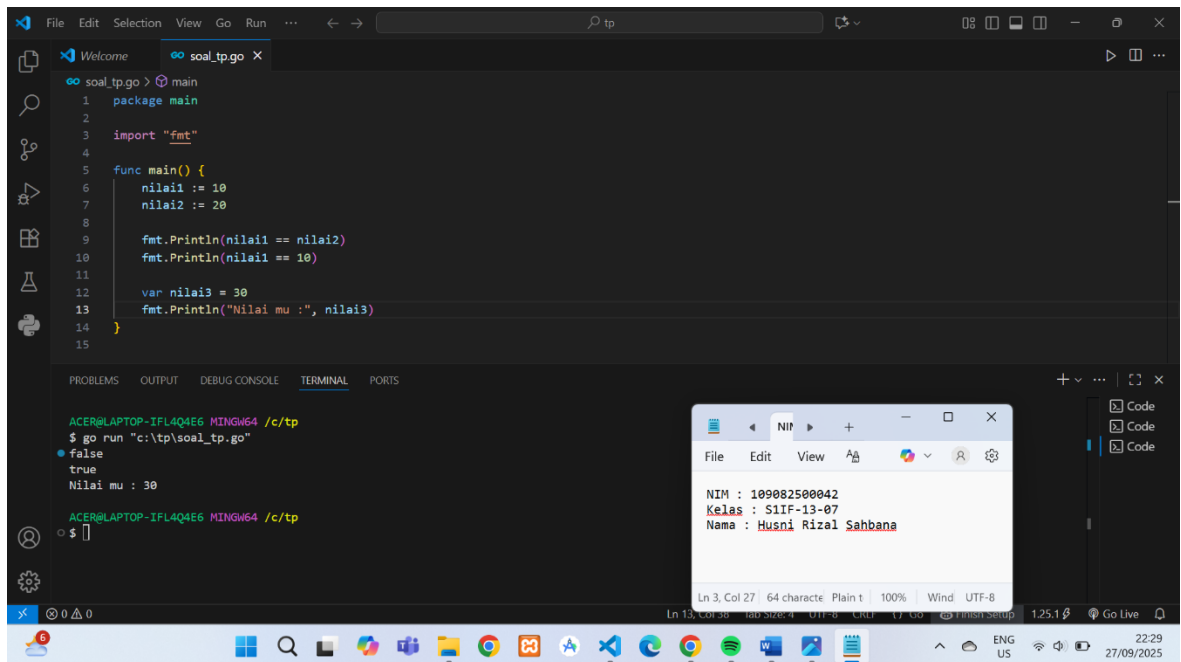
```
1 package main
2
3 import "fmt"
4
5 func main() {
6     nama := "Husni Rizal Sahbana"
7     umur := 18
8     bb := 65.9
9
10    fmt.Printf("Halo nama saya %s saya sekarang umurnya %d berat badan saya %.2f", nama, umur, bb)
11 }
12
```

ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp  
\$ go run "c:\tp\soal\_tp.go"  
Halo nama saya Husni Rizal Sahbana saya sekarang umurnya 18 berat badan saya 65.90  
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp  
\$

#### 4. Apa perbedaan antara operator == dan = dalam Bahasa Go!

Jadi dalam bahasa Go, tanda = digunakan untuk **operator assignment** yaitu untuk memberikan atau mengubah nilai pada sebuah variabel, misalnya `x = 10` artinya variabel `x` diberi nilai 10, sedangkan tanda `==` digunakan sebagai **operator perbandingan** untuk mengecek apakah dua nilai sama, dan hasilnya berupa nilai boolean `true` atau `false`, misalnya `x == 10` akan menghasilkan `true` jika nilai `x` memang 10; jadi perbedaan utamanya adalah `=` dipakai untuk menugaskan nilai, sedangkan `==` dipakai untuk membandingkan nilai.

Contoh :



The screenshot shows a Go program in a VS Code editor. The code defines a `main` function with three variables: `nilai1` (10), `nilai2` (20), and `nilai3` (30). It prints the result of `nilai1 == nilai2` (false) and `nilai1 == 10` (true), followed by the value of `nilai3` (30). The terminal output confirms these results. An inset window shows a student ID (NIP) and name.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     nilai1 := 10
7     nilai2 := 20
8
9     fmt.Println(nilai1 == nilai2)
10    fmt.Println(nilai1 == 10)
11
12    var nilai3 = 30
13    fmt.Println("Nilai mu :", nilai3)
14 }
15
```

```
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp
$ go run "c:\tp\soal_tp.go"
false
true
Nilai mu : 30
ACER@LAPTOP-IFL4Q4E6 MINGW64 /c/tp
$
```

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

Source code :

```
package main

import "fmt"

func main() {
    nama := ""
    fmt.Print("Masukan nama kalian : ")
    fmt.Scan(&nama)
    fmt.Printf("Hallo %s", nama)
}
```