

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL IV
“TIPE DATA & VARIABEL”



Disusun oleh :
NAMA : Felix Pedrosa Valentino
NIM : 103112400056
S1 IF-12-01
Dosen Pengampu :
Yohani Setiya Rafika Nur, M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

DASAR TEORI

Tipe Bentukan :

A.) Tipe Bentukan Alias

1. Deklarasi tipe bentukan sebagai alias.
2. Salah satu cara pemberian nama tipe data dengan nama baru sehingga lebih mudah untuk dikenali atau dipahami oleh pemrogram.

type nama_alias tipe_data

B.)Tipe Bentukan: Structure/Record

1. Memungkinkan kita untuk mengelompokkan beberapa variabel yang mungkin berelasi menjadi suatu kesatuan.
2. Sebagai contoh, data buku biasanya terdiri dari judul buku, nama penulis, tahun terbit, nama penerbit, edisi, dan data-data lainnya.
3. Tipe data dari field bisa berupa tipe data dasar ataupun tipe bentukan lainnya.

Contoh :

- Tipe Waktu (jam, menit, dan detik)
- Tipe Orang (nama, gender, alamat, agama dan biodata lainnya)
- Tipe Buku (judul, penulis, tahun terbit, penerbit, dan data lainnya)

Konstanta :

Pada pemrograman memungkinkan adanya konstanta, berbeda dengan variabel yang nilainya bisa diganti-ganti selama program berjalan. Konstanta bersifat fix dan tidak dapat diganti nilainya.

Pseudocode :

constant nama_konstanta: tipe_data = value

Golang :

1 **cons** nama_konstanta = value

2 **cons** nama_konstanta tipe_data = value

Penggunaan konstanta sama halnya dengan variabel pada umumnya, perbedaannya adalah nilainya yang tidak bisa diganti.

CONTOH SOAL

1.) Contoh Soal 1

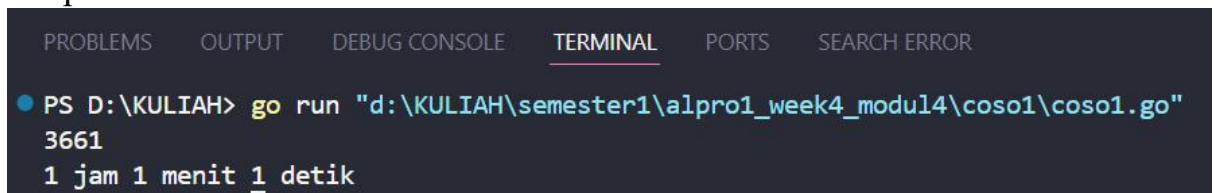
Source Code :

```
package main

import "fmt"

func main() {
    var detik, jam, menit int
    fmt.Scan(&detik)
    jam = detik / 3600
    menit = (detik % 3600) / 60
    detik = detik % 60
    fmt.Println(jam, "jam", menit, "menit", detik, "detik")
}
```

Output :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week4_modul4\coso1\coso1.go"
3661
1 jam 1 menit 1 detik
```

Deskripsi Program :

Program ini adalah program dalam bahasa Go yang mengonversi waktu dalam detik menjadi format yang lebih mudah dipahami, yaitu jam, menit, dan detik.

Penjelasan :

1. Input :

- Program meminta pengguna untuk memasukkan jumlah detik yang akan dikonversi.

2. Proses Konversi :

- Program menghitung jumlah jam dengan cara membagi total detik dengan 3600 (jumlah detik dalam satu jam).

- Sisa dari hasil pembagian untuk jam dihitung menjadi menit dengan cara membagi sisa detik setelah pembagian jam dengan 60 (jumlah detik dalam satu menit).
- Sisa detik setelah dikonversi menjadi jam dan menit disimpan sebagai detik yang tersisa.

3. Output :

- Program menampilkan hasil konversi dalam format "X jam Y menit Z detik", di mana X adalah jumlah jam, Y adalah jumlah menit, dan Z adalah sisa detik.

2.) Contoh Soal 2

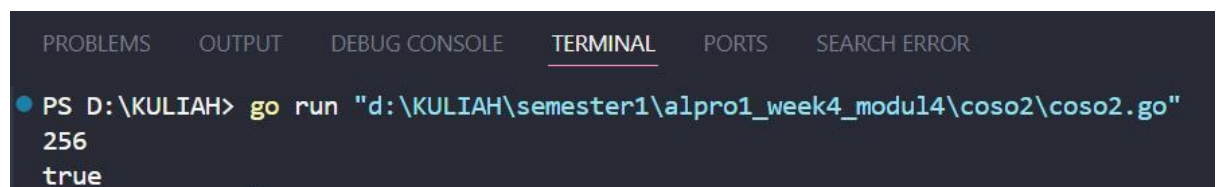
Source Code :

```
package main

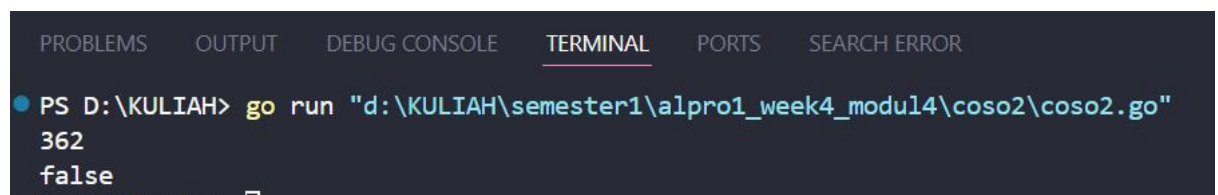
import "fmt"

func main() {
    var bilangan, d1, d2, d3 int
    fmt.Scan(&bilangan)
    d1 = bilangan / 100
    d2 = bilangan % 100 / 10
    d3 = bilangan % 10
    fmt.Println(d1 <= d2 && d2 <= d3)
}
```

Output :



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week4_modul4\coso2\coso2.go"
256
true
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week4_modul4\coso2\coso2.go"
362
false
```

Deskripsi Program :

Program ini adalah program dalam bahasa Go yang mengecek apakah tiga digit dari sebuah bilangan tiga digit berurutan secara tidak menurun (apakah setiap digit lebih kecil atau lebih besar sama dengan digit berikutnya).

Penjelasan :

1. Input :

- Program meminta pengguna untuk memasukkan sebuah bilangan tiga digit (integer).

2. Proses Pemecahan Digit :

- Program membagi bilangan tiga digit tersebut menjadi tiga digit terpisah:
- `d1` untuk digit ratusan: didapat dengan membagi bilangan dengan 100.
- `d2` untuk digit puluhan: didapat dari sisa pembagian bilangan dengan 100, kemudian dibagi dengan 10.
- `d3` untuk digit satuan: didapat dari sisa pembagian bilangan dengan 10.

3. Pemeriksaan :

- Program mengecek apakah digit-digit tersebut berurutan secara non-menurun (dari kiri ke kanan), dengan memeriksa apakah `d1 <= d2` dan `d2 <= d3`.

4. Output :

- Program mencetak nilai `**`true`**` jika digit-digit tersebut berurutan secara non-menurun, atau `**`false`**` jika tidak.

Contoh Eksekusi :

1. Jika pengguna memasukkan angka 123, maka output akan menjadi `true`, karena $1 \leq 2 \leq 3$.

2. Jika pengguna memasukkan angka 321, maka output akan menjadi `false`, karena $3 > 2$.

3.) Contoh Soal 3

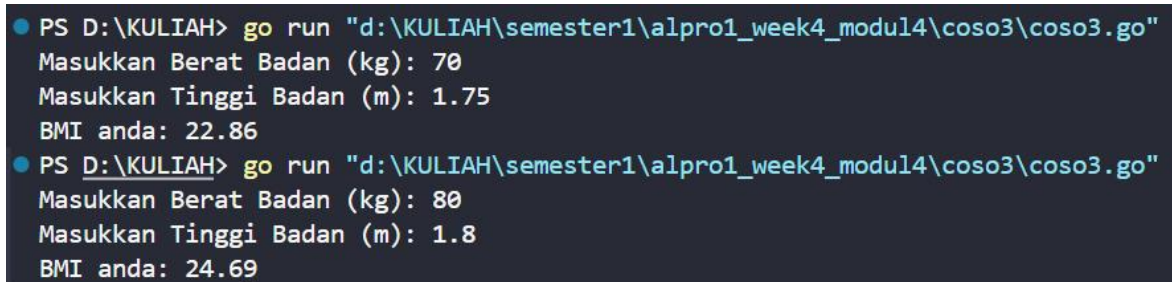
Source Code :

```
package main

import "fmt"

func main() {
    var beratBadan, tinggiBadan, bmi float64
    fmt.Print("Masukkan Berat Badan (kg): ")
    fmt.Scan(&beratBadan)
    fmt.Print("Masukkan Tinggi Badan (m): ")
    fmt.Scan(&tinggiBadan)
    bmi = beratBadan / (tinggiBadan * tinggiBadan)
    fmt.Printf("BMI anda: %.2f", bmi)
}
```

Output :



```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week4_modul4\coso3\coso3.go"
Masukkan Berat Badan (kg): 70
Masukkan Tinggi Badan (m): 1.75
BMI anda: 22.86
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week4_modul4\coso3\coso3.go"
Masukkan Berat Badan (kg): 80
Masukkan Tinggi Badan (m): 1.8
BMI anda: 24.69
```

Deskripsi Program :

Program ini adalah program dalam bahasa Go yang menghitung BMI (Body Mass Index) berdasarkan berat badan dan tinggi badan yang dimasukkan oleh pengguna. Program ini berguna untuk membantu pengguna mengetahui status berat badan mereka berdasarkan standar BMI.

Penjelasan :

1. Input :

- Program meminta pengguna untuk memasukkan berat badan dalam kilogram (kg).
- Program juga meminta pengguna untuk memasukkan tinggi badan dalam meter (m).

2. Proses Perhitungan :

- Program menghitung BMI menggunakan rumus :

$$\text{BMI} = \frac{\text{berat badan (kg)}}{\text{tinggi badan (m)}^2}$$

- Berat badan dibagi dengan kuadrat tinggi badan.

3. Output :

- Program menampilkan hasil perhitungan BMI dengan dua angka desimal menggunakan fungsi ``fmt.Printf``.

4. Tujuan :

- Program ini bertujuan untuk menghitung dan memberikan informasi BMI kepada pengguna. Nilai BMI ini sering digunakan untuk menilai apakah seseorang memiliki berat badan normal, kurang, atau berlebih.

SOAL LATIHAN

1.) Latihan Soal 1

Source code :

```
package main

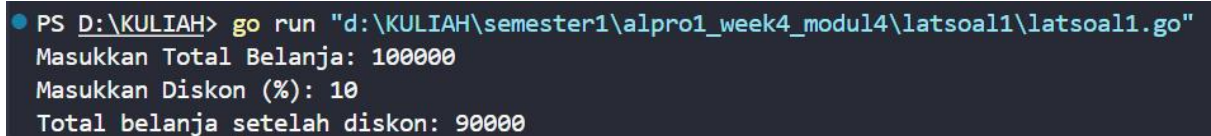
import "fmt"

func main() {
    var totalBelanja, diskon, totalAkhir int
    fmt.Print("Masukkan Total Belanja: ")
    fmt.Scan(&totalBelanja)
    fmt.Print("Masukkan Diskon (%): ")
    fmt.Scan(&diskon)

    // Menghitung total harga setelah diskon
    totalAkhir = totalBelanja - (totalBelanja * diskon / 100)

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

Output :



```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week4_modul4\latsoal1\latsoal1.go"
Masukkan Total Belanja: 100000
Masukkan Diskon (%): 10
Total belanja setelah diskon: 90000
```

Deskripsi Program :

Program ini berguna untuk menghitung total harga setelah diskon, seperti pada skenario belanja di mana diskon tertentu diberikan atas total pembelian.

Penjelasan :

1. Input:

- Program menerima dua input dari pengguna :

- Total belanja awal : Bilangan bulat yang mewakili total belanja sebelum diskon.

- Diskon : Bilangan bulat yang menyatakan besarnya diskon dalam satuan persen.

2. Proses Perhitungan:

- Program menghitung nilai diskon dengan rumus :

$$\text{diskon} = \text{total belanja} \times \frac{\text{diskon persen}}{100}$$

Setelah itu, program menghitung total belanja akhir dengan mengurangi total belanja awal dengan nilai diskon yang diperoleh :

$$\text{total belanja akhir} = \text{total belanja awal} - \text{diskon}$$

3. Output:

- Program menampilkan total belanja akhir setelah dikurangi diskon.

2.) Latihan Soal 2

Source code :

```
package main

import "fmt"

func main() {
    var bmi, tinggiBadan, beratBadan float64
    fmt.Print("Masukkan nilai BMI: ")
    fmt.Scan(&bmi)
    fmt.Print("Masukkan tinggi badan (dalam meter): ")
    fmt.Scan(&tinggiBadan)

    // Menghitung berat badan berdasarkan BMI dan tinggi badan
    beratBadan = bmi * (tinggiBadan * tinggiBadan)

    fmt.Printf("Berat badan anda adalah: %.2f kg\n", beratBadan)
}
```

Output :

```

PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week4_modul4\latsoal2\latsoal2.go"
Masukkan nilai BMI: 22.85
Masukkan tinggi badan (dalam meter): 1.75
Berat badan anda adalah: 69.98 kg
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week4_modul4\latsoal2\latsoal2.go"
Masukkan nilai BMI: 24.69
Masukkan tinggi badan (dalam meter): 1.8
Berat badan anda adalah: 80.00 kg

```

Deskripsi Program :

Program diatas adalah program dalam bahasa Go yang digunakan untuk menghitung berat badan seseorang jika diketahui nilai BMI dan tinggi badan.

1. Input:

-Program menerima dua input dari pengguna :

-Nilai BMI: Bilangan riil (float64) yang menyatakan Body Mass Index (BMI) seseorang.

-Tinggi badan: Bilangan riil (float64) dalam satuan meter yang menyatakan tinggi badan seseorang.

2. Proses Perhitungan :

- Program menghitung berat badan berdasarkan rumus BMI :

$$\text{BMI} = \frac{\text{berat badan (kg)}}{\text{tinggi badan (m)}^2}$$

- Dari rumus ini, kita bisa menghitung berat badan sebagai :

$$\text{berat badan (kg)} = \text{BMI} \times \text{tinggi badan (m)}^2$$

- Program menggunakan rumus tersebut untuk menghitung berat badan berdasarkan nilai BMI dan tinggi badan yang dimasukkan.

3. Output :

- Program menampilkan berat badan seseorang dalam satuan kilogram dengan dua angka di belakang koma.

3.) Latihan Soal 3

Source code :

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var ax, ay, bx, by, cx, cy float64

    // Input koordinat titik A, B, dan C
    fmt.Print("Masukkan koordinat titik A (x y): ")
    fmt.Scan(&ax, &ay)
    fmt.Print("Masukkan koordinat titik B (x y): ")
    fmt.Scan(&bx, &by)
    fmt.Print("Masukkan koordinat titik C (x y): ")
    fmt.Scan(&cx, &cy)

    // Menghitung panjang sisi AB, BC, dan CA menggunakan
    teorema Pythagoras
    AB := math.Sqrt(math.Pow(bx-ax, 2) + math.Pow(by-ay, 2))
    BC := math.Sqrt(math.Pow(cx-bx, 2) + math.Pow(cy-by, 2))
    CA := math.Sqrt(math.Pow(ax-cx, 2) + math.Pow(ay-cy, 2))

    // Menentukan sisi terpanjang
    sisiTerpanjang := math.Max(AB, math.Max(BC, CA))

    // Output hasil dengan 2 angka di belakang koma
    fmt.Printf("Sisi terpanjang dari segitiga tersebut adalah: %.2f\n",
sisiTerpanjang)
}
```

Output :

```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week4_modul4\latsoal3\latsoal3.go"
Masukkan koordinat titik A (x y): 0.0
0.0
Masukkan koordinat titik B (x y): 3.0
0.0
Masukkan koordinat titik C (x y): 3.0
4.0
Sisi terpanjang dari segitiga tersebut adalah: 5.00
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week4_modul4\latsoal3\latsoal3.go"
Masukkan koordinat titik A (x y): 1.0
1.0
Masukkan koordinat titik B (x y): 4.0
1.0
Masukkan koordinat titik C (x y): 1.0
5.0
Sisi terpanjang dari segitiga tersebut adalah: 5.00
```

Deskripsi Program :

Program diatas adalah program dalam bahasa Go yang digunakan untuk menghitung panjang sisi-sisi segitiga yang dibentuk oleh tiga titik dalam koordinat kartesius 2D dan menentukan sisi terpanjang menggunakan teorema Pythagoras.

Penjelasan :

1. Input :

- Program meminta pengguna memasukkan koordinat dari tiga titik A, B, dan C dalam sistem kartesius 2 dimensi.
 - o Titik A dengan koordinat (x_A, y_A)
 - o Titik B dengan koordinat (x_B, y_B)
 - o Titik C dengan koordinat (x_C, y_C)

2. Proses Perhitungan :

- Program menghitung panjang sisi-sisi segitiga yang dibentuk oleh tiga titik menggunakan teorema Pythagoras :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

$$BC = \sqrt{(x_C - x_B)^2 + (y_C - y_B)^2}$$

$$CA = \sqrt{(x_A - x_C)^2 + (y_A - y_C)^2}$$

- Setelah menghitung panjang sisi-sisi AB, BC, dan CA, program menggunakan fungsi `math.Max` untuk menentukan sisi yang memiliki panjang terbesar.

3. Output :

- Program menampilkan panjang sisi terpanjang dari segitiga dengan dua angka di belakang koma.

DAFTAR PUSTAKA

Salim, H. (2017). Praktikum 3 Variabel, Tipe Data & Operator (lanj. Retrieved from https://www.academia.edu/36204247/Praktikum_3_Variabel_Tipe_Data_and_Operator_lanj

LAPORAN PRAKTIKUM Variabel Dan Tipe Data. (n.d.). Retrieved from <https://id.scribd.com/document/671779921/LAPORAN-PRAKTIKUM-variabel-dan-tipe-data-1-Copy>