**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN 1**

**MODUL 4**

**“TIPE DATA & VARIABEL”**



**DISUSUN OLEH:**

**RIZKINA AZIZAH**

**103112400082**

**S1 IF-12-01**

**DOSEN:**

**Yohani Setiya Rafika Nur, M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024/2025**

**DASAR TEORI**

1. **Tipe Data Numerik Non-Desimal**

Tipe data numerik non-desimal atau **non floating point** di Go ada beberapa jenis. Secara umum ada 2 tipe data kategori ini yang perlu diketahui.

* uint, tipe data untuk bilangan cacah (bilangan positif).
* int, tipe data untuk bilangan bulat (bilangan negatif dan positif).

Kedua tipe data di atas kemudian dibagi lagi menjadi beberapa jenis, dengan pembagian berdasarkan lebar cakupan nilainya, detailnya bisa dilihat di tabel berikut.

| **Tipe data** | **Cakupan bilangan** |
| --- | --- |
| uint8 | 0 ↔ 255 |
| uint16 | 0 ↔ 65535 |
| uint32 | 0 ↔ 4294967295 |
| uint64 | 0 ↔ 18446744073709551615 |
| uint | sama dengan uint32 atau uint64 (tergantung nilai) |
| byte | sama dengan uint8 |
| int8 | -128 ↔ 127 |
| int16 | -32768 ↔ 32767 |
| int32 | -2147483648 ↔ 2147483647 |
| int64 | -9223372036854775808 ↔ 9223372036854775807 |
| int | sama dengan int32 atau int64 (tergantung nilai) |
| rune | sama dengan int32 |

Dianjurkan untuk tidak sembarangan dalam menentukan tipe data variabel, sebisa mungkin tipe yang dipilih harus disesuaikan dengan nilainya, karena efeknya adalah ke alokasi memori variabel. Pemilihan tipe data yang tepat akan membuat pemakaian memori lebih optimal, tidak berlebihan.

var positiveNumber uint8 = 89

var positiveNumber uint8 = 89

var negativeNumber = -1243423644

fmt.Printf("bilangan positif: %d\n", positiveNumber)

fmt.Printf("bilangan negatif: %d\n", negativeNumber)

var negativeNumber = -1243423644

fmt.Printf("bilangan positif: %d\n", positiveNumber)

fmt.Printf("bilangan negatif: %d\n", negativeNumber)

Variabel positiveNumber bertipe uint8 dengan nilai awal 89. Sedangkan variabel negativeNumber dideklarasikan dengan nilai awal -1243423644. Compiler secara cerdas akan menentukan tipe data variabel tersebut sebagai int32 (karena angka tersebut masuk ke cakupan tipe data int32).

String format %d pada fmt.Printf() digunakan untuk memformat data numerik non-desimal.

1. **Tipe Data Numerik Desimal**

Tipe data numerik desimal yang perlu diketahui ada 2, float32 dan float64. Perbedaan kedua tipe data tersebut berada di lebar cakupan nilai desimal yang bisa ditampung. Untuk lebih jelasnya bisa merujuk ke spesifikasi [IEEE-754 32-bit floating-point numbers](http://www.h-schmidt.net/FloatConverter/IEEE754.html).

var decimalNumber = 2.62

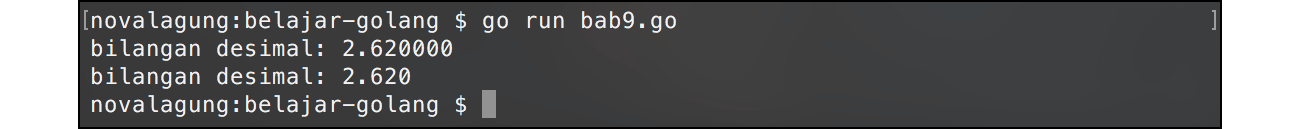
fmt.Printf("bilangan desimal: %f\n", decimalNumber)

fmt.Printf("bilangan desimal: %.3f\n", decimalNumber)

fmt.Printf("bilangan desimal: %f\n", decimalNumber)

fmt.Printf("bilangan desimal: %.3f\n", decimalNumber)

Pada kode di atas, variabel decimalNumber akan memiliki tipe data float32, karena nilainya berada di cakupan tipe data tersebut.



String format %f digunakan untuk memformat data numerik desimal menjadi string. Digit desimal yang akan dihasilkan adalah **6 digit**. Pada contoh di atas, hasil format variabel decimalNumber adalah 2.620000. Jumlah digit yang muncul bisa dikontrol menggunakan %.nf, tinggal ganti n dengan angka yang diinginkan. Contoh: %.3f maka akan menghasilkan 3 digit desimal, %.10f maka akan menghasilkan 10 digit desimal.

**Nb ; Penggunaan Fungsi math.Pow()**

Fungsi math.Pow() digunakan untuk operasi pangkat nilai. math.Pow(2, 3) berarti 2 pangkat 3, hasilnya 8. Fungsi ini berada dalam package math

**CONTOH SOAL**

1. Latihan1

Source Code:

package main

import "fmt"

func main() {

    var detik, jam, menit int

    fmt.Scan(&detik)

    jam = detik / 3600

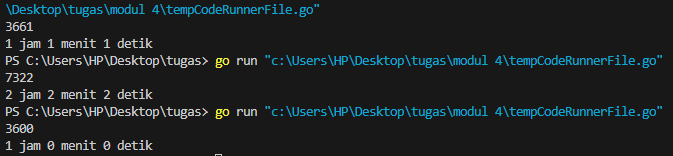
    menit = (detik % 3600) / 60

    detik = detik % 60

    fmt.Println(jam, "jam", menit, "menit", detik, "detik")

}

Output:



Deskripsi Program:

Program diatas adalah program digunakan untuk mengkonversi detik ke jam, menit dan detik, Dimana tiga variabel menggunakan satu tipe data yaitu integer. Dan juga menggunakan salah satu operator precedence yaitu modulo

1. Latihan1

Source Code:

package main

import "fmt"

func main() {

    var bilangan, d1, d2, d3 int

    fmt.Scan(&bilangan)

    d1 = bilangan / 100

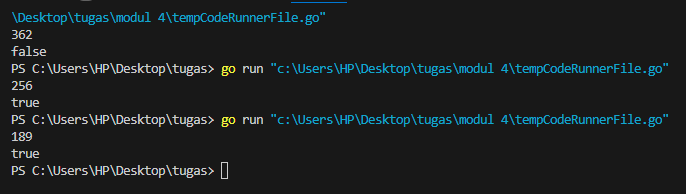
    d2 = bilangan % 100 / 10

    d3 = bilangan % 10

    fmt.Println(d1 <= d2 && d2 <= d3)

}

Output:



Deskripsi Program:

Program ini digunakan untuk menentukan apakah setiap digit pada suatu bilangan terurut membesar atau tidak. Menggunakan tiga variabel Dimana ketiganya menggunakan sama sama menggunakan tipe data integer. Program ini menggunakan salah satu operator precedence yaitu modulo.

1. Latihan1

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)")

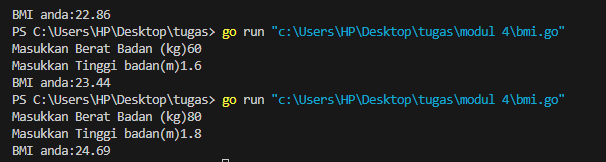
    fmt.Scan(&tinggibadan)

    bmi = beratbadan / (tinggibadan \* tinggibadan)

    fmt.Println("BMI anda:%.2f", bmi)

}

Output:



Deskripsi Program:

Program ini digunakan untuk menghitung BMI atau Body Mass Indeks, yang mana BMI merupakan hasil bagi dari berat badan dengan kuadrat dari tinggi badan. Menggunakan tiga variabel yang bertipe data sama yaitu real. Dan juga penggunaan perintah %.2f yang berfungsi agar output yang dihasilkan terdapat 2 angka dibelakang koma.

**SOAL LATIHAN**

**Statement perulangan**

1. **Latihan 1**

**Source Code:**

package main

import "fmt"

func main() {

    var belnja\_awal, besar\_diskon int

    fmt.Scan(&belnja\_awal, &besar\_diskon)

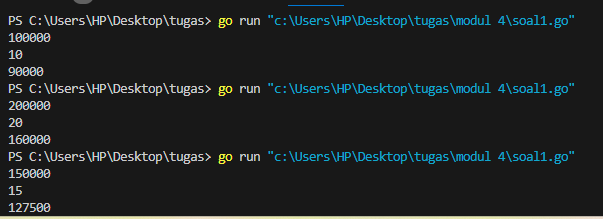
    fmt.Scanln()

    belanja\_akhir := belnja\_awal - (belnja\_awal \* besar\_diskon/100)

    fmt.Print(belanja\_akhir)

}

Output:



Deskripsi Program:

Program ini digunakan untuk menghitung total harga setelah memperoleh diskon dengan besaran tertentu. Menggunakan tiga variabel dengan tipe data sama yaitu integer. Dan juga penggunaan rumus total setelah diskon yaitu :

Total belanja – (total belanja x diskon/100)

1. **Latihan 2**

**Source Code:**

package main

import "fmt"

func main() {

    var bmi, tb float64

    fmt.Print("Masukkan nilai BMI:")

    fmt.Scan(&bmi)

    fmt.Print("Masukkan tinggi badan (meter):")

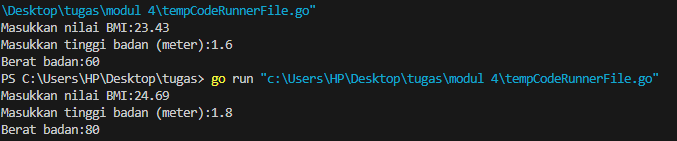
    fmt.Scan(&tb)

    bb := (tb\*tb) \* bmi

    fmt.Printf("Berat badan:%.f",bb)

}

Output:



Deskripsi Program:

Program ini digunakan untuk menentukan berat badan seseorang apabila diketahui nilai BMI dan tinggi badannya. Menggunakan tiga variabel yang ketiganya menggunakan tipe data sama yaitu real. Dan pada output menggunakan %.f agar hasilnya dibulatkan

1. **Latihan 3**

**Source Code:**

package main

import (

    "fmt"

    "math"

)

func main() {

    var ax, ay, bx, by, cx, cy float64

    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)

    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))

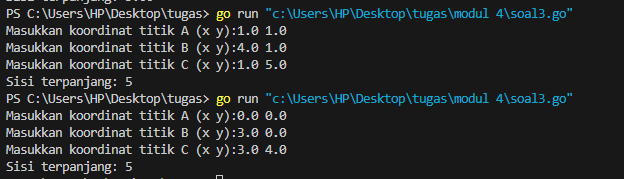
    ac := math.Sqrt(math.Pow(cx-ax, 2) + math.Pow(cy-ay, 2))

    maxSide := math.Max(ab, math.Max(bc, ac))

    fmt.Printf("Sisi terpanjang: %.f\n", maxSide)

}

Output:



Deskripsi Program:

Program ini digunakan untuk menentukan sisi terpanjang segitiga. Menggunakan enam variabel yang bertipe data real. Didalam program ini untuk menentukan sisi terpanjang segitiga kita menggunakan rumus teorema Phytagoras. Dan juga menggunakan perintah math.Pow untuk pemangkatan dan math.Sqrt untuk menghitung akar kuadrat.

**DAFTAR PUSTAKA**

novalagung(2024).A.19.FungsiMultipleReturn **(**<https://dasarpemrogramangolang.novalagung.com/A-fungsi-multiple-return.html>,2024)

novalagung(2024).A.10. Tipe Data (<https://dasarpemrogramangolang.novalagung.com/A-tipe-data.html>, 2024)