

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



**DISUSUN OLEH:
FEROS PEDROSA VALENTINO
103112400055
S1 IF-12-01**

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

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

DASAR TEORI

1. Integer Division dan Modulo

Pembagian pada tipe data integer (integer division atau div) sedikit berbeda dengan pembagian yang sudah sering kita pelajari dari sekolah dasar. Hasil pembagian akan bertipe data integer, artinya kita akan mengabaikan bilangan yang muncul setelah tanda koma atau floating point. Hasil dari operasi div ini dikenal dengan istilah quotient.

Sebagai contoh hasil dari 10 (dividend) dibagi dengan 3 (divisor) adalah 3.333, sedangkan hasil div adalah 3. Artinya kita mengabaikan floating point dari hasil pembagian tersebut. Operator "/" di dalam bahasa pemrograman Go akan melakukan operasi integer division apabila tipe data dari kedua operan adalah integer dan pembagian biasa apabila operan bertipe float.

Modulo atau modulus "mod" adalah operasi untuk mencari nilai integer dari sisa pembagian pada integer division, biasanya dikenal juga dengan istilah remainder. Misalnya untuk operasi 10 dibagi dengan 3, maka sisa pembagian adalah 1. Operator modulo menggunakan "%" di dalam bahasa pemrograman Go.

Semua bilangan apabila di modulus dengan 10 akan didapat satuan atau digit terakhir. Misalnya: 1234 mod 10 akan diperoleh nilai 4, 5677 mod 10 akan diperoleh nilai 7. Apabila 1234 div 10 akan diperoleh 123, sedangkan 5677 div 10 akan didapat nilai 567. Pola ini bisa dimanfaatkan untuk mencari nilai digit-digit yang lain.

2. Casting atau Konversi Tipe Data

Pada bahasa pemrograman Go, tipe data bersifat statis, artinya tipe data yang sudah didefinisikan tidak dapat diganti selama program berjalan. Casting merupakan salah satu teknik konversi antar tipe data di dalam bahasa pemrograman.

Package strconv berisi banyak fungsi yang sangat membantu kita untuk melakukan konversi. Berikut merupakan beberapa fungsi yang dalam package tersebut:

- a. Fungsi `strconv.Atoi()`
Fungsi ini digunakan untuk konversi data dari tipe string ke int. `strconv.Atoi()` menghasilkan 2 buah nilai kembalian, yaitu hasil konversi dan error (jika konversi sukses, maka error berisi nil).
- b. Fungsi `strconv.Itoa()`
Merupakan kebalikan dari `strconv.Atoi()`, berguna untuk konversi int ke string.
- c. Fungsi `strconv.ParseInt()`
Digunakan untuk konversi string berbentuk numerik dengan basis tertentu ke tipe numerik non-desimal dengan lebar data bisa ditentukan.
- d. Fungsi `strconv.FormatInt()`
Berguna untuk konversi data numerik int64 ke string dengan basis numerik bisa ditentukan sendiri.
- e. Fungsi `strconv.ParseFloat()`
Digunakan untuk konversi string ke numerik desimal dengan lebar data bisa ditentukan.
- f. Fungsi `strconv.FormatFloat()`
Berguna untuk konversi data bertipe float64 ke string dengan format eksponen, lebar digit desimal, dan lebar tipe data bisa ditentukan.
- g. Fungsi `strconv.ParseBool()`
Digunakan untuk konversi string ke bool.

h. Fungsi `strconv.FormatBool()`

Digunakan untuk konversi bool ke string.

Konversi Data Menggunakan Teknik Casting

Cara penerapannya adalah dengan menggunakan keyword tipe data sebagai nama fungsi, kemudian argument pemanggilannya diisi dengan data yang ingin dikonversi tipenya.

Casting string ↔ byte

String sebenarnya adalah slice/array byte. Di Go sebuah karakter biasa (bukan unicode) direpresentasikan oleh sebuah elemen slice byte. Tiap elemen slice berisi data int dengan basis desimal, yang merupakan kode ASCII dari karakter dalam string. Cara mendapatkan slice byte dari sebuah data string adalah dengan meng-casting-nya ke tipe `[]byte`.

CONTOH SOAL

1. Coso 1

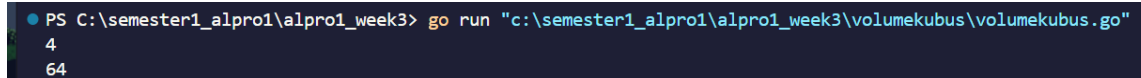
Source code:

```
package main

import "fmt"

func main() {
    var sisi, volume float64
    fmt.Scan(&sisi)
    volume = (sisi * sisi * sisi)
    fmt.Print(volume)
}
```

Output:



```
PS C:\semester1_alpro1\alpro1_week3> go run "c:\semester1_alpro1\alpro1_week3\volumekubus\volumekubus.go"
4
64
```

Deskripsi program:

Program di atas adalah program sederhana dalam bahasa Go yang digunakan untuk menghitung volume sebuah kubus berdasarkan panjang sisi yang dimasukkan oleh pengguna. Berikut adalah deskripsi program tersebut:

1. Paket utama: Program ini menggunakan paket main sebagai entry point dari aplikasi.
2. Import Paket: Program mengimpor paket fmt, yang digunakan untuk menangani input dan output di konsol.
3. Fungsi utama (main):
 - o Program mendeklarasikan dua variabel bertipe float64, yaitu sisi dan volume.
 - o Program meminta pengguna untuk memasukkan nilai panjang sisi kubus melalui fungsi fmt.Scan(), yang kemudian disimpan ke dalam variabel sisi.
 - o Program menghitung volume kubus menggunakan rumus volume kubus, yaitu sisi*sisi*sisi dan hasilnya disimpan ke dalam variabel volume.
 - o Program menampilkan hasil volume yang telah dihitung ke layar menggunakan fungsi fmt.Print().

2. Coso 2

Source code:

```
package main

import "fmt"

func main() {
    var alas, tinggi, luas float64
    fmt.Print("Masukkan alas : ")
    fmt.Scan(&alas)
    fmt.Print("Masukkan tinggi : ")
    fmt.Scan(&tinggi)
    luas = (alas * tinggi / 2)
    fmt.Print(luas)
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week3> go run "c:\semester1_alpro1\alpro1_week3\luassegitiga\luassegitiga.go"
Masukkan alas : 6
Masukkan tinggi : 7
21
```

Deskripsi program:

Program di atas adalah program sederhana dalam bahasa Go yang digunakan untuk menghitung luas segitiga berdasarkan input alas dan tinggi yang dimasukkan oleh pengguna. Berikut deskripsi detail program tersebut:

1. Paket utama: Program menggunakan paket main sebagai entry point dari aplikasi.
2. Import Paket: Program mengimpor paket fmt untuk menangani input dan output dari konsol.
3. Fungsi utama (main):
 - o Tiga variabel bertipe float64 dideklarasikan: alas, tinggi, dan luas.
 - o Program menampilkan teks "Masukkan alas : " di layar menggunakan fungsi fmt.Print() untuk meminta pengguna memasukkan nilai alas.
 - o Nilai alas yang dimasukkan pengguna disimpan dalam variabel alas melalui fungsi fmt.Scan().
 - o Setelah itu, program menampilkan teks "Masukkan tinggi : " untuk meminta pengguna memasukkan nilai tinggi segitiga.
 - o Nilai tinggi yang dimasukkan disimpan dalam variabel tinggi.
 - o Program menghitung luas segitiga dengan rumus $luas = alas \times tinggi / 2$ dan menyimpan hasilnya ke dalam variabel luas.
 - o Program menampilkan nilai luas yang telah dihitung menggunakan fungsi fmt.Print().

3. Coso 3

Source code

```
package main

import "fmt"

func main() {
    var rupiah, dollar float64
    fmt.Print("Masukkan Nominal Rupiah : ")
    fmt.Scan(&rupiah)
    dollar = (rupiah / 15000)
    fmt.Print("Jadi ", rupiah, " rupiah = ", dollar, " dollar ")
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week3> go run "c:\semester1_alpro1\alpro1_week3\matauang\matauang.go"
● Masukkan Nominal Rupiah : 15000
  Jadi 15000 rupiah = 1 dollar
```

Deskripsi program:

Program di atas adalah program sederhana dalam bahasa Go yang digunakan untuk mengonversi nilai mata uang dari Rupiah ke Dollar berdasarkan kurs yang ditetapkan di dalam program. Berikut adalah deskripsi dari program tersebut:

1. Paket utama: Program menggunakan paket main sebagai titik awal eksekusi aplikasi.
2. Import Paket: Program mengimpor paket fmt yang digunakan untuk menangani input dan output di konsol.
3. Fungsi utama (main):
 - Dua variabel bertipe float64 dideklarasikan: rupiah dan dollar.
 - Program menampilkan teks "Masukkan Nominal Rupiah : " di layar menggunakan fungsi fmt.Print(), untuk meminta pengguna memasukkan jumlah uang dalam Rupiah.
 - Nilai Rupiah yang dimasukkan oleh pengguna disimpan dalam variabel rupiah melalui fungsi fmt.Scan().
 - Program mengonversi nilai Rupiah menjadi Dollar menggunakan kurs konversi tetap sebesar 1 Dollar = 15.000 Rupiah. Perhitungan dilakukan dengan rumus $dollar = \text{rupiah} / 15000$.
 - Hasil konversi ditampilkan ke layar dengan pesan, misalnya: "Jadi 150000 rupiah = 10 dollar".

LATIHAN SOAL MANDIRI

1. Nilai x

Source code:

```
package main

import (
    "fmt"
    "log"
)

func calculateX(fx float64) (float64, error) {
    // Cek apakah  $fx + 5 = 0$ , yang akan membuat pembagian menjadi tidak valid
    if fx == -5 {
        return 0, fmt.Errorf("nilai f(x) tidak boleh sama dengan -5 karena menghasilkan pembagian dengan nol")
    }
    // Hitung nilai x dari persamaan  $f(x)$ 
    x := (2 / (fx + 5)) + 5
    return x, nil
}

func main() {
    var fx float64
    fmt.Print("Masukkan nilai f(x): ")
    _, err := fmt.Scan(&fx)
    if err != nil {
        log.Fatal("Input tidak valid")
    }

    // Hitung nilai x berdasarkan  $f(x)$ 
    x, err := calculateX(fx)
    if err != nil {
        log.Fatal(err)
    }

    fmt.Printf("Nilai x adalah: %.4f\n", x)
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week3> go run "c:\semester1_alpro1\alpro1_week3\nilaix\nilaix.go"
Masukkan nilai f(x): 5
Nilai x adalah: 5.2000
PS C:\semester1_alpro1\alpro1_week3> go run "c:\semester1_alpro1\alpro1_week3\nilaix\nilaix.go"
Masukkan nilai f(x): 11
Nilai x adalah: 5.1250
```

Deskripsi program:

Program ini adalah sebuah program sederhana dalam bahasa Go untuk menghitung nilai x berdasarkan persamaan tertentu. Berikut adalah deskripsi dari program tersebut:

1. Program dimulai dengan package main dan mengimpor dua package standar Go: fmt untuk input/output dan log untuk penanganan error.
2. Fungsi calculateX(fx float64) didefinisikan untuk menghitung nilai x berdasarkan input f(x). Fungsi ini:
 - Memeriksa apakah $fx + 5 = 0$, yang akan menyebabkan pembagian dengan nol.
 - Jika $fx = -5$, fungsi mengembalikan error.
 - Jika tidak, fungsi menghitung x menggunakan rumus: $x = (2 / (fx + 5)) + 5$
 - Mengembalikan nilai x dan nil untuk error jika perhitungan berhasil.
3. Fungsi main() adalah entry point program:
 - Meminta pengguna memasukkan nilai f(x).
 - Membaca input menggunakan fmt.Scan().
 - Jika terjadi error saat membaca input, program berhenti dengan pesan error.
4. Program memanggil calculateX() dengan nilai f(x) yang dimasukkan:
 - Jika terjadi error (misalnya, $f(x) = -5$), program berhenti dan menampilkan pesan error.
 - Jika berhasil, program mencetak nilai x yang dihitung dengan 4 angka desimal.

Program ini mendemonstrasikan penanganan error, penggunaan fungsi, dan perhitungan matematika sederhana dalam Go. Ini juga menunjukkan bagaimana menerima input dari pengguna dan memformat output.

2. Volume dan luas kulit bola

Source code:

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var radius int
    fmt.Print("Masukkan jari-jari bola: ")
    fmt.Scan(&radius)

    // Konstanta pi
    const pi float64 = 3.1415926535

    // Hitung volume bola
    volume := (4.0 / 3.0) * pi * math.Pow(float64(radius), 3)

    // Hitung luas kulit bola
    area := 4.0 * pi * math.Pow(float64(radius), 2)
```



```

        fmt.Printf("Bola dengan jejari %d memiliki volume: %.2f\n", radius,
volume)
        fmt.Printf("Bola dengan jejari %d memiliki Luas kulit bola: %.2f\n",
radius, area)
    }
}

```

Output:

```

PS C:\semester1_alpro1\alpro1_week3> go run "c:\semester1_alpro1\alpro1_week3\volum&luasbola\volum&luasbola.go"
Masukkan jari-jari bola: 5
Bola dengan jejari 5 memiliki volume: 523.60
Bola dengan jejari 5 memiliki Luas kulit bola: 314.16

```

Deskripsi program:

Program ini adalah sebuah program sederhana dalam bahasa Go untuk menghitung volume dan luas permukaan (kulit) sebuah bola berdasarkan jari-jari yang dimasukkan oleh pengguna. Berikut rincian fungsi program:

1. Program meminta pengguna untuk memasukkan nilai jari-jari bola.
2. Menggunakan konstanta pi dengan nilai 3.1415926535 untuk perhitungan.
3. Menghitung volume bola menggunakan rumus: $V = (4/3) * \pi * r^3$ Di mana r adalah jari-jari bola.
4. Menghitung luas permukaan (kulit) bola menggunakan rumus: $A = 4 * \pi * r^2$
5. Menampilkan hasil perhitungan volume dan luas permukaan bola dengan dua angka desimal.

Program ini memanfaatkan package "fmt" untuk input/output dan package "math" untuk fungsi matematika seperti perpangkatan (Pow). Hasil perhitungan ditampilkan dengan format yang mudah dibaca, menunjukkan jari-jari yang dimasukkan serta hasil perhitungan volume dan luas permukaan.

3. Tahun kabisat

Source code:

```

package main

import "fmt"

func isKabisat(tahun int) bool {
    if tahun%400 == 0 {
        return true
    } else if tahun%100 == 0 {
        return false
    } else if tahun%4 == 0 {
        return true
    }
    return false
}

func main() {
    var tahun int
    fmt.Print("Masukkan tahun: ")
    fmt.Scan(&tahun)
}

```

```

    if isKabisat(tahun) {
        fmt.Printf("%d adalah tahun kabisat: true\n", tahun)
    } else {
        fmt.Printf("%d bukan tahun kabisat: false\n", tahun)
    }
}

```

Output:

```

PS C:\semester1_alpro1\alpro1_week3> go run "c:\semester1_alpro1\alpro1_week3\tahunkabisat\tahunkabisat.go"
Masukkan tahun: 2018
• 2018 bukan tahun kabisat: false
PS C:\semester1_alpro1\alpro1_week3> go run "c:\semester1_alpro1\alpro1_week3\tahunkabisat\tahunkabisat.go"
• Masukkan tahun: 2016
2016 adalah tahun kabisat: true

```

Deskripsi program:

Program di atas adalah program sederhana dalam bahasa Go yang berfungsi untuk menentukan apakah suatu tahun adalah tahun kabisat atau bukan. Berikut adalah deskripsi dari setiap bagian program:

1. Fungsi isKabisat:
 - Fungsi ini menerima satu parameter berupa integer (tahun) dan mengembalikan nilai boolean (true atau false).
 - Logika yang digunakan untuk menentukan apakah suatu tahun adalah tahun kabisat mengikuti aturan berikut:
 - Jika tahun habis dibagi 400, maka itu adalah tahun kabisat.
 - Jika tahun habis dibagi 100 tapi tidak habis dibagi 400, maka itu bukan tahun kabisat.
 - Jika tahun habis dibagi 4 tapi tidak habis dibagi 100, maka itu adalah tahun kabisat.
 - Tahun yang tidak memenuhi kondisi di atas bukan tahun kabisat.
2. Fungsi main:
 - Program meminta pengguna untuk memasukkan tahun yang akan diperiksa.
 - Nilai tahun tersebut diinput oleh pengguna melalui `fmt.Scan(&tahun)`.
 - Setelah tahun dimasukkan, fungsi `isKabisat` dipanggil untuk memeriksa apakah tahun tersebut adalah tahun kabisat.
 - Hasilnya ditampilkan dengan menggunakan `fmt.Printf` dalam bentuk pesan yang menunjukkan apakah tahun tersebut kabisat atau bukan.

4. Temperatur

Source code:

```

package main

import (
    "fmt"
    "log"
)

// Function to convert Celsius to other temperature scales

```

```

func convertTemperature(celsius float64) (fahrenheit, reamur, kelvin float64)
{
    // Convert Celsius to Fahrenheit
    fahrenheit = (celsius * 9 / 5) + 32

    // Convert Celsius to Reamur
    reamur = celsius * 4 / 5

    // Convert Fahrenheit to Kelvin
    kelvin = (fahrenheit + 459.67) * 5 / 9

    return fahrenheit, reamur, kelvin
}

func main() {
    var celsius float64

    // Input temperature in Celsius
    fmt.Print("Masukkan suhu dalam Celsius: ")
    _, err := fmt.Scan(&celsius)
    if err != nil {
        log.Fatal("Input tidak valid. Masukkan angka yang valid.")
    }

    // Convert temperature
    fahrenheit, reamur, kelvin := convertTemperature(celsius)

    // Display results
    fmt.Println("\nHasil Konversi Suhu:")
    fmt.Printf("Derajat reamur dari %.2f°C = %.2f°R\n", celsius, reamur)
    fmt.Printf("Derajat fahrenheit dari %.2f°C = %.2f°F\n", celsius, fahrenheit)
    fmt.Printf("Derajat kelvin dari %.2f°C = %.2f°K\n", celsius, kelvin)
}

```

Output:

```

PS C:\semester1_alpro1\alpro1_week3> go run "c:\semester1_alpro1\alpro1_week3\temperatur\temperatur.go"
Masukkan suhu dalam Celsius: 50

Hasil Konversi Suhu:
Derajat reamur dari 50.00°C = 40.00°R
Derajat fahrenheit dari 50.00°C = 122.00°F
Derajat kelvin dari 50.00°C = 323.15°K

```

Deskripsi program:

Program ini adalah aplikasi konversi suhu sederhana yang ditulis dalam bahasa pemrograman Go. Berikut adalah deskripsi singkat dari program tersebut:

1. Program ini memungkinkan pengguna untuk memasukkan suhu dalam satuan Celsius.
2. Setelah menerima input, program akan mengkonversi suhu Celsius tersebut ke tiga satuan suhu lainnya: Fahrenheit, Reamur, dan Kelvin.
3. Konversi suhu dilakukan menggunakan fungsi `convertTemperature`, yang menerima nilai Celsius sebagai input dan mengembalikan nilai Fahrenheit, Reamur, dan Kelvin.

4. Rumus konversi yang digunakan:
 - Celsius ke Fahrenheit: $(C * 9/5) + 32$
 - Celsius ke Reamur: $C * 4/5$
 - Fahrenheit ke Kelvin: $(F + 459.67) * 5/9$
5. Program menggunakan fungsi `fmt.Scan` untuk menerima input dari pengguna dan menangani kemungkinan kesalahan input.
6. Hasil konversi ditampilkan ke layar menggunakan fungsi `fmt.Printf`, menunjukkan nilai suhu dalam Reamur, Fahrenheit, dan Kelvin, masing-masing dengan dua angka desimal.
7. Program ini menggunakan paket "`fmt`" untuk input/output dan "`log`" untuk penanganan kesalahan.

Secara keseluruhan, program ini menyediakan cara sederhana dan efisien untuk mengkonversi suhu dari Celsius ke tiga satuan suhu lainnya, dengan antarmuka konsol yang mudah digunakan.

DAFTAR PUSTAKA

A.12. Operator. (n.d.). Retrieved from
<https://dasarpemrogramangolang.novalagung.com/A-operator.html>

A.14. Perulangan. (n.d.). Retrieved from
<https://dasarpemrogramangolang.novalagung.com/A-perulangan.html>