

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



**DISUSUN OLEH:**  
**MUHAMMAD ZAKY MUBAROK**  
**103112400073**  
**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. Konsep Tipe Data**

Tipe data adalah kategori yang digunakan untuk mengklasifikasikan berbagai jenis data yang dapat digunakan dalam program. Tipe data menentukan operasi apa yang valid dan seberapa banyak memori yang diperlukan oleh sebuah nilai yang akan disimpan atau dioperasikan. Tipe data dapat dibagi menjadi dua kategori utama: tipe data primitif dan tipe data non-primitif (referensi).

#### **A. Tipe Data Primitif**

Tipe data primitif adalah tipe data dasar yang disediakan oleh bahasa pemrograman dan memiliki nilai tertentu, bukan referensi kelas atau objek. Contoh tipe data primitif dalam Java termasuk `byte`, `short`, `int`, `long`, `double`, `char`, dan `boolean`. Tipe data ini adalah bawaan dari sebuah bahasa pemrograman dan digunakan untuk menyimpan nilai-nilai sederhana seperti angka dan karakter.

#### **B. Tipe Data Non-Primitif**

Tipe data non-primitif, juga dikenal sebagai tipe data referensi, adalah tipe data yang dibuat berdasarkan kebutuhan programmer. Tipe data ini mencakup `array`, `record`, `file`, dan `set`. Tipe data non-primitif digunakan untuk menyimpan objek yang lebih kompleks dan biasanya didefinisikan oleh programmer.

### **2. Variabel**

Variabel adalah tempat penyimpanan data dalam memori komputer yang mengandung data atau nilai sementara dari sebuah proses pada suatu pemrograman. Setiap variabel harus dideklarasikan dengan tipe data tertentu sebelum dapat digunakan. Deklarasi variabel adalah proses untuk mengenal suatu variabel berikut dengan tipe data yang dipakai pada variabel tersebut.

### **3. Deklarasi dan Penggunaan Variabel**

Deklarasi variabel melibatkan penentuan nama variabel dan tipe data yang akan digunakan. Misalnya, dalam bahasa pemrograman Java, deklarasi variabel dapat dilakukan dengan sintaks seperti `int angka;` yang berarti variabel angka adalah tipe data integer. Variabel digunakan untuk menampung suatu nilai, dan karena itu setiap variabel pasti memiliki tipe data yang harus dideklarasikan terlebih dahulu sebelum dapat digunakan.

#### **4. Konstanta**

Konstanta adalah variabel yang mempunyai nilai tetap dan tidak dapat diubah nilainya setelah dideklarasikan. Konstanta digunakan untuk menyimpan nilai yang tidak berubah sepanjang eksekusi program, seperti nilai pi ( $\pi$ ) dalam perhitungan matematika.

## CONTOH SOAL

### 1. Latihan1

Source Code:

```
package main

import "fmt"

func main() {
    var a, b, c, d, e int
    fmt.Scan(&a, &b, &c, &d, &e)
    var Hasil int
    Hasil = a + b + c + d + e
    fmt.Print(Hasil)
}
```

Output:

3 2 7 10 2  
24

11 22 33 44 55  
165

Deskripsi Program:

Program ini adalah program sederhana dalam bahasa Go yang membaca lima bilangan bulat dari input pengguna, menjumlahkan kelima bilangan tersebut, dan kemudian mencetak hasil penjumlahannya. Berikut adalah penjelasan langkah demi langkah:

**Deklarasi Paket Utama :**  
**package main**

**Impor Paket “fmt”:** Paket ini digunakan untuk fungsi input dan output :  
**import "fmt"**

**Fungsi Utama:** Fungsi main adalah titik masuk dari program :  
**func main() {**

**Deklarasi Variabel:** Lima variabel a, b, c, d, dan e dideklarasikan sebagai bilangan bulat.:  
**var a, b, c, d, e int**

**Membaca Input Pengguna:** Fungsi **fmt.Scan** digunakan untuk membaca lima bilangan bulat dari input pengguna dan menyimpannya ke dalam variabel a, b, c, d, dan e :  
**fmt.Scan(&a, &b, &c, &d, &e)**

**Deklarasi dan Inisialisasi Variabel Hasil:** Variabel Hasil dideklarasikan sebagai bilangan bulat dan diinisialisasi dengan jumlah dari kelima variabel yang dibaca sebelumnya.:

```
var Hasil int
```

```
Hasil = a + b + c + d + e
```

**Mencetak Hasil:** Fungsi `fmt.Print` digunakan untuk mencetak nilai dari variabel Hasil:  
`fmt.Print(Hasil)`

Jadi, program ini pada dasarnya menjumlahkan lima bilangan bulat yang dimasukkan oleh pengguna dan menampilkan hasilnya.

## CONTOH SOAL

### 2. Latihan2

Source Code:

```
package main

import "fmt"

func main() {
    // fx = 2 / (x + 5) + 5
    // masukan input x
    var x, fx float32
    fmt.Scan(&x)
    fx = 2/(x+5) + 5
    fmt.Print(fx)
}
```

Output:

5  
5.2

-23  
4.88888

Deskripsi Program:

#### **Deklarasi Paket Utama:**

package main

**Impor Paket “fmt”:** Paket ini digunakan untuk fungsi input dan output :

import "fmt"

**Fungsi Utama:** Fungsi main adalah titik masuk dari program :

func main() {

**Deklarasi Variabel:** Dua variabel x dan fx dideklarasikan sebagai bilangan desimal (float32) :

var x, fx float32

**Membaca Input Pengguna:** Fungsi `fmt.Scan` digunakan untuk membaca satu bilangan desimal dari input pengguna dan menyimpannya ke dalam variabel `x` :  
`fmt.Scan(&x)`

**Menghitung Nilai Fungsi:** Variabel `fx` diinisialisasi dengan hasil perhitungan fungsi  $f(x) = 2/(x+5) + 5$  menggunakan nilai `x` yang dibaca sebelumnya :  
 $fx = 2/(x+5) + 5$

**Mencetak Hasil:** Fungsi `fmt.Print` digunakan untuk mencetak nilai dari variabel `fx`:  
`fmt.Print(fx)`

Jadi, program ini pada dasarnya meminta pengguna untuk memasukkan nilai `x`, kemudian menghitung nilai fungsi  $f(x)$  berdasarkan rumus yang diberikan, dan menampilkan hasilnya

## CONTOH SOAL

### 3. Latihan3

Source Code:

```
package main

import "fmt"

func main() {
    // 1. input 5 buah byte, 5 buah karakter
    var c1, c2, c3, c4, c5 byte
    fmt.Scan(&c1, &c2, &c3, &c4, &c5)
    fmt.Printf("%c%c%c%c%c", c1, c2, c3, c4, c5)
    fmt.Println()
    // 2. input 3 buah rune, output karakter setelahnya
    var b1, b2, b3 rune
    fmt.Scanf("%c%c%c", &b1, &b2, &b3)
    fmt.Printf("%c%c%c", b1+1, b2+1, b3+1)
}
```

Output:

66 97 103 117 115

Bagus

SNO

TOP

Deskripsi Program:

#### Deklarasi Paket Utama:

package main

**Impor Paket “fmt”:** Paket ini digunakan untuk fungsi input dan output :

import "fmt"

**Fungsi Utama:** Fungsi main adalah titik masuk dari program :

func main() {

#### Operasi Pertama: Input dan Output 5 Karakter:

- **Deklarasi Variabel:** Lima variabel c1, c2, c3, c4, dan c5 dideklarasikan sebagai tipe byte :  
var c1, c2, c3, c4, c5 byte

**Membaca Input Pengguna:** Fungsi fmt.Scan digunakan untuk membaca lima karakter dari input pengguna dan menyimpannya ke dalam variabel c1, c2, c3, c4, dan c5 :

fmt.Scan(&c1, &c2, &c3, &c4, &c5)



**Mencetak Karakter:** Fungsi `fmt.Printf` digunakan untuk mencetak lima karakter yang telah dibaca :

```
fmt.Printf("%c%c%c%c%c", c1, c2, c3, c4, c5)
fmt.Println()
```

**Operasi Kedua: Input 3 Rune dan Output Karakter Setelahnnya:**

- **Deklarasi Variabel:** Tiga variabel `b1`, `b2`, dan `b3` dideklarasikan sebagai tipe rune :  
`var b1, b2, b3 rune`

**Membaca Input Pengguna:** Fungsi `fmt.Scanf` digunakan untuk membaca tiga karakter dari input pengguna dan menyimpannya ke dalam variabel `b1`, `b2`, dan `b3` :

```
fmt.Scanf("%c%c%c", &b1, &b2, &b3)
```

**Mencetak Karakter Setelahnnya:** Fungsi `fmt.Printf` digunakan untuk mencetak karakter yang berada setelah karakter yang dibaca (dengan menambahkan 1 pada nilai rune) :

```
fmt.Printf("%c%c%c", b1+1, b2+1, b3+1)
```

Jadi, program ini membaca lima karakter dari input pengguna dan mencetaknya kembali. Kemudian, program membaca tiga karakter lagi dan mencetak karakter yang berada setelah masing-masing karakter yang dibaca.

## SOAL LATIHAN

### Statement perulangan

1.

#### Source Code:

```
package main
import "fmt"
func main() {
    var (
        satu, dua, tiga string
        temp string
    )
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&satu)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&dua)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&tiga)
    fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)
    temp = satu
    satu = dua
    dua = tiga
    tiga = temp
    fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)
}
```

Output:

Masukan input string: 1 2 3

Masukan input string: Masukan input string: 1 2 3

Output awal = 1 3 1

Output akhir = 3 1 1

Deskripsi Program:

#### Deklarasi Paket Utama:

```
package main
```

**Impor Paket “fmt”:** Paket ini digunakan untuk fungsi input dan output :

```
import "fmt"
```

**Fungsi Utama:** Fungsi main adalah titik masuk dari program :

```
func main() {
```

**Deklarasi Variabel:** Tiga variabel satu, dua, dan tiga dideklarasikan sebagai tipe string untuk menyimpan input pengguna. Variabel temp digunakan untuk membantu dalam proses penukaran :

```
var (  
    satu, dua, tiga string  
    temp string  
)
```

**Membaca Input Pengguna:** Fungsi `fmt.Scanln` digunakan untuk membaca tiga string dari input pengguna dan menyimpannya ke dalam variabel satu, dua, dan tiga :

```
fmt.Print("Masukan input string: ")  
fmt.Scanln(&satu)  
fmt.Print("Masukan input string: ")  
fmt.Scanln(&dua)  
fmt.Print("Masukan input string: ")  
fmt.Scanln(&tiga)
```

**Mencetak Output Awal:** Fungsi `fmt.Println` digunakan untuk mencetak string yang telah dibaca dalam urutan awal :

```
fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)
```

**Menukar Posisi String:** Variabel temp digunakan untuk menyimpan sementara nilai satu, kemudian nilai satu diisi dengan nilai dua, nilai dua diisi dengan nilai tiga, dan nilai tiga diisi dengan nilai temp :

```
temp = satu  
satu = dua  
dua = tiga  
tiga = temp
```

**Mencetak Output Akhir:** Fungsi `fmt.Println` digunakan untuk mencetak string setelah posisi mereka ditukar :

```
fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)
```

jadi, program ini pada dasarnya meminta pengguna untuk memasukkan tiga string, mencetak string tersebut dalam urutan awal, menukar posisi string, dan kemudian mencetak string dalam urutan baru

## SOAL LATIHAN

### Statement perulangan

2.

#### Source Code:

```
package main

import (
    "bufio"
    "fmt"
    "os"
)

func main() {
    reader := bufio.NewReader(os.Stdin)

    var nama, nim, kelas string

    // Membaca input pengguna
    fmt.Print("Masukkan Nama: ")
    nama, _ = reader.ReadString('\n')
    fmt.Print("Masukkan NIM: ")
    nim, _ = reader.ReadString('\n')
    fmt.Print("Masukkan Kelas: ")
    kelas, _ = reader.ReadString('\n')

    // Menghapus karakter newline dari input
    nama = nama[:len(nama)-1]
    nim = nim[:len(nim)-1]
    kelas = kelas[:len(kelas)-1]

    // Menampilkan resume singkat
    fmt.Println("\n--- Resume Mahasiswa ---")
    fmt.Println("Nama :", nama)
    fmt.Println("NIM  :", nim)
    fmt.Println("Kelas :", kelas)
}
```

Output :

Masukkan Nama: Bima  
Masukkan NIM: 1124431414  
Masukkan Kelas: IF-48-GAB

--- Resume Mahasiswa ---  
Nama : Bima  
NIM : 1124431414  
Kelas : IF-48-GAB

Masukkan Nama: Yura  
Masukkan NIM: 1324234545  
Masukkan Kelas: IFX-48-12

--- Resume Mahasiswa ---  
Nama : Yura  
NIM : 1324234545  
Kelas : IFX-48-12

Deskripsi program :

**Deklarasi Paket Utama :**  
package main

**Impor Paket:** Paket fmt digunakan untuk fungsi input dan output, sedangkan bufio dan os digunakan untuk membaca input dari pengguna dengan lebih andal :

```
import (  
    "bufio"  
    "fmt"  
    "os"  
)
```

**Fungsi Utama:** Fungsi main adalah titik masuk dari program :  
func main() {

**Membuat Reader:** Membuat objek reader dari bufio.NewReader untuk membaca input dari pengguna :  
reader := bufio.NewReader(os.Stdin)

**Deklarasi Variabel:** Tiga variabel nama, nim, dan kelas dideklarasikan sebagai tipe string untuk menyimpan input pengguna :  
var nama, nim, kelas string

**Membaca Input Pengguna:** Fungsi `reader.ReadString('\n')` digunakan untuk membaca input hingga karakter newline (`\n`). Input ini disimpan dalam variabel `nama`, `nim`, dan `kelas` :

```
fmt.Print("Masukkan Nama: ")
nama, _ = reader.ReadString('\n')
fmt.Print("Masukkan NIM: ")
nim, _ = reader.ReadString('\n')
fmt.Print("Masukkan Kelas: ")
kelas, _ = reader.ReadString('\n')
```

**Menghapus Karakter Newline:** Setelah membaca input, karakter newline di akhir string dihapus dengan menggunakan slicing :

```
nama = nama[:len(nama)-1]
nim = nim[:len(nim)-1]
kelas = kelas[:len(kelas)-1]
```

**Menampilkan Resume Singkat:** Fungsi `fmt.Println` digunakan untuk mencetak resume singkat mahasiswa berdasarkan input yang diberikan :

```
fmt.Println("\n--- Resume Mahasiswa ---")
fmt.Println("Nama :", nama)
fmt.Println("NIM :", nim)
fmt.Println("Kelas :", kelas)
```

Jadi, program ini meminta pengguna untuk memasukkan nama, NIM, dan kelas, kemudian menampilkan data tersebut dalam format resume singkat.

## SOAL LATIHAN

### Statement perulangan

3.

#### Source Code:

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jariJari float64

    // Membaca input pengguna
    fmt.Print("Masukkan jari-jari lingkaran: ")
    fmt.Scanln(&jariJari)

    // Menghitung luas lingkaran
    luas := math.Pi * jariJari * jariJari

    // Menampilkan hasil perhitungan
    fmt.Printf("Luas lingkaran dengan jari-jari %.2f adalah %.2f\n", jariJari,
luas)
}
```

Output :

Masukkan jari-jari lingkaran: 7  
Luas lingkaran dengan jari-jari 7.00 adalah 153.94

Masukkan jari-jari lingkaran: 14  
Luas lingkaran dengan jari-jari 14.00 adalah 615.75

Masukkan jari-jari lingkaran: 20  
Luas lingkaran dengan jari-jari 20.00 adalah 1256.64

Deskripsi program :

**Deklarasi Paket Utama:**

```
package main
```

**Impor Paket:** Paket fmt digunakan untuk fungsi input dan output, sedangkan paket math digunakan untuk mendapatkan nilai konstanta  $\pi$  (Pi) :

```
import (  
    "fmt"  
    "math"  
)
```

**Fungsi Utama:** Fungsi main adalah titik masuk dari program :

```
func main() {
```

**Deklarasi Variabel:** Variabel jariJari dideklarasikan sebagai tipe float64 untuk menyimpan input jari-jari lingkaran :

```
var jariJari float64
```

**Membaca Input Pengguna:** Fungsi fmt.Scanln digunakan untuk membaca nilai jari-jari dari input pengguna :

```
fmt.Print("Masukkan jari-jari lingkaran: ")  
fmt.Scanln(&jariJari)
```

**Menghitung Luas Lingkaran:** Luas lingkaran dihitung menggunakan rumus

(luas =  $\pi \times \text{jariJari}^2$ ) :

```
luas := math.Pi * jariJari * jariJari
```

**Menampilkan Hasil Perhitungan:** Fungsi fmt.Printf digunakan untuk mencetak hasil perhitungan luas lingkaran dengan format dua angka decimal :

```
fmt.Printf("Luas lingkaran dengan jari-jari %.2f adalah %.2f\n", jariJari, luas)
```

Program ini akan meminta pengguna untuk memasukkan panjang jari-jari lingkaran, menghitung luas lingkaran berdasarkan input tersebut, dan menampilkan hasilnya.



## SOAL LATIHAN

### Statement perulangan

4.

#### Source Code:

```
package main

import "fmt"

func main() {
    var fahrenheit float64

    // Membaca input pengguna
    fmt.Print("Masukkan suhu dalam Fahrenheit: ")
    fmt.Scanln(&fahrenheit)

    // Menghitung suhu dalam Celsius
    celsius := (fahrenheit - 32) * 5 / 9

    // Menampilkan hasil konversi
    fmt.Printf("Suhu dalam Celsius: %.2f\n", celsius)
}
```

Output :

Masukkan suhu dalam Fahrenheit: 32  
Suhu dalam Celsius: 0.00

Masukkan suhu dalam Fahrenheit: 77  
Suhu dalam Celsius: 25.00

Masukkan suhu dalam Fahrenheit: 212  
Suhu dalam Celsius: 100.00

Deskripsi program :

#### Deklarasi Paket Utama:

package main

**Impor Paket “fmt”:** Paket ini digunakan untuk fungsi input dan output :

import "fmt"

**Fungsi Utama:** Fungsi main adalah titik masuk dari program :  
func main() {

**Deklarasi Variabel:** Variabel fahrenheit dideklarasikan sebagai tipe float64 untuk menyimpan input suhu dalam Fahrenheit :  
var fahrenheit float64

**Membaca Input Pengguna:** Fungsi fmt.Scanln digunakan untuk membaca nilai suhu dalam Fahrenheit dari input pengguna :  
fmt.Print("Masukkan suhu dalam Fahrenheit: ")  
fmt.Scanln(&fahrenheit)

**Menghitung Suhu dalam Celsius:** Suhu dalam Celsius dihitung menggunakan rumus  $(C = \frac{5}{9} \times (F - 32))$  :  
celsius := (fahrenheit - 32) \* 5 / 9

**Menampilkan Hasil Konversi:** Fungsi fmt.Printf digunakan untuk mencetak hasil konversi suhu dalam Celsius dengan format dua angka decimal :  
fmt.Printf("Suhu dalam Celsius: %.2f\n", celsius)

Program ini akan meminta pengguna untuk memasukkan suhu dalam Fahrenheit, menghitung suhu dalam Celsius berdasarkan input tersebut, dan menampilkan hasilnya.

## DAFTAR PUSTAKA

Laporan Dasar Pemrograman Tipe Data. (2015-05-17)

*Retrieved from <http://Academia.edu>: <http://Academia.edu>*

Laporan Hasil Praktikum Modul II Tipe Data. (2014-11-05).

*Retrieved from <http://Slideshare.net>: <http://Slideshare.net>*

Laporan Praktikum 3 Tipe Data dan Operasi I/O. (2020-01-01).

*Retrieved from <http://Academia.edu>: <http://Academia.edu>*

Polinema Programming. (2020-09-21). Retrieved from Jobsheet 03.

Praktikum 4 & 5 – Tipe Data, Variabel, Konstanta dan Operator. (2016-05-26).

*Retrieved from <http://Matheusrumetna.com>: <http://Matheusrumetna>.*