

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2  
MODUL 6  
STRUCT DAN ARRAY**



**DISUSUN OLEH:  
Keishin Naufa Alfaridzhi  
103112400061  
S1 IF-12-01**

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## I. DASAR TEORI

### A. Bahasa Yang Digunakan

Pada praktikum ini bahasa pemrograman yang digunakan adalah bahasa pemrograman Go, sesuai dengan modul yang menjadi acuan praktikum. Golang (atau) Go adalah bahasa pemrograman baru, yang mulai dilirik oleh para developer karena kelebihan-kelebihan yang dimilikinya. Sudah banyak Perusahaan besar yang menggunakan bahasa ini untuk produk-produk mereka hingga di level production.

### B. Komentar

Komentar biasa dimanfaatkan untuk menyisipkan catatan pada kode program, menulis penjelasan atau deskripsi mengenai suatu blok kode, atau bisa juga digunakan untuk me-remark kode (men-non-aktifkan kode yang tidak digunakan). Komentar akan diabaikan Ketika kompilasi maupun eksekusi program.

Ada 2 jenis komentar di Golang, yaitu inline dan multiline.

#### 1. Komentar Inline

Penulisan komentar jenis ini diawali dengan tanda *double slash* (*//*) lalu diikuti pesan komentarnya. Komentar inline hanya berlaku untuk satu baris pesan saja. Jika pesan komentar lebih dari satu baris, maka tanda *double slash* harus ditulis lagi di baris selanjutnya.

#### 2. Komentar Multiline

Komentar yang cukup panjang akan lebih rapi jika ditulis menggunakan teknik komentar multiline. Ciri dari komentar jenis ini adalah penulisannya diawali dengan tanda *(/\**) dan diakhiri *(\*/)*.

### C. Variabel

Golang mengadopsi 2 jenis penulisan variabel, yang dituliskan tipe data-nya dan yang tidak. Kedua cara tersebut intinya adalah sama, pembedanya hanyalah cara penulisannya saja. Untuk penulisan variabel dengan tipe data, keyword *var* digunakan untuk deklarasi variabel kemudian diakhiri dengan tipe data misalnya *string*. Kemudian untuk penulisan variabel tanpa tipe data, variabel dideklarasikan dengan menggunakan metode type inference. Penandanya tipe data tidak dituliskan pada saat deklarasi. Pada penggunaan metode ini, operand (*=*) harus diganti dengan (*:=*) dan keyword *var* dihilangkan.

Golang memiliki aturan unik yang tidak dimiliki bahasa lain, yaitu tidak boleh ada satupun variabel yang menganggur. Artinya, semua variabel yang dideklarasikan harus digunakan. Jika terdapat variabel yang tidak digunakan tapi dideklarasikan, program akan gagal dikompilasi. Untuk mengatasi itu, golang memiliki variabel yaitu underscore. Underscore ( \_ ) adalah predefined variabel yang bisa dimanfaatkan untuk menampung nilai yang tidak dipakai.

#### **D. Tipe Data**

Golang mengenal beberapa jenis tipe data, diantaranya adalah tipe data numerik (decimal dan non-desimal), string, dan boolean.

1. Tipe Data Numerik Non-Desimal (uint, int)
2. Tipe Data Numerik Desimal (float64, float32)
3. Tipe Data Bool (true, false)
4. Tipe Data String (string, “ “)

#### **E. Operator Aritmatika**

Operator aritmatika merupakan operator yang digunakan untuk operasi yang sifatnya perhitungan. Golang mendukung beberapa operator aritmatika standar, yaitu:

1. Penjumlahan (+)
2. Pengurangan (-)
3. Perkalian (\*)
4. Pembagian (/)
5. Modulus atau sisa hasil pembagian (%)

#### **F. Seleksi Kondisi**

Seleksi kondisi pada program berguna untuk mengontrol sebuah blok kode yang akan dieksekusi. Yang dijadikan acuan oleh seleksi kondisi adalah nilai bertipe bool, bisa berasal dari variabel, ataupun hasil operasi perbandingan. Nilai tersebut menentukan blok kode mana yang akan dieksekusi. Go memiliki 2 macam keyword untuk seleksi kondisi, yaitu if else dan switch.

##### **1. If Expression**

If adalah salah satu kata kunci yang digunakan dalam percabangan. Percabangan artinya kita bisa mengeksekusi kode program tertentu ketika suatu kondisi terpenuhi. Hampir semua bahasa pemrograman mendukung if expression.

##### **2. Else if expression**

Terkadang kita butuh membuat beberapa kondisi. Kasus seperti ini dapat menggunakan else if expression. If mendukung short statement sebelum kondisi.

Hal ini sangat cocok untuk membuat statement yang sederhana sebelum melakukan pengecekan terhadap kondisi.

### 3. Switch-Case

Switch merupakan seleksi kondisi yang sifatnya fokus pada satu variabel, lalu kemudian di-cek nilainya. Contoh sederhananya seperti penentuan apakah nilai variabel x adalah: 1, 2, 3, atau lainnya. Perlu diketahui, switch pada pemrograman Go memiliki perbedaan dibanding bahasa lain. Di Go, ketika sebuah case terpenuhi, tidak akan dilanjutkan ke pengecekan case selanjutnya, meskipun tidak ada keyword “break” di situ. Konsep ini berkebalikan dengan switch pada umumnya pemrograman lain (yang ketika sebuah case terpenuhi, maka akan tetap dilanjutkan mengecek case selanjutnya kecuali ada keyword “break”).

## G. Perulangan

Perulangan merupakan proses mengulang dan mengeksekusi blok kode tanpa henti sesuai dengan kondisi yang dijadikan acuan. Biasanya disiapkan variabel untuk iterasi atau penanda kapan perulangan akan dihentikan.

### a. For Loop

For loop merupakan statement perulangan dasar dan cukup sering ditemui. Format for loop yaitu sebagai berikut.

- *Init Statement*: bagian ini akan dieksekusi sebelum perulangan dimulai. Biasanya diisi dengan mendeklarasi variabel iterasi.
- *Condition Expression*: bagian ini akan dicek dan dieksekusi setiap perulangan yang dilakukan, jika true maka perulangan akan terus berjalan hingga kondisi bernilai false.
- *Post Statement*: statement ini akan dieksekusi pada akhir iterasi. Jika terdapat range, maka perulangan akan dieksekusi untuk setiap item pada range.

### b. While Loop

While loop merupakan perulangan yang akan terus berjalan hingga suatu kondisi terpenuhi. Penulisan while loop adalah dengan menuliskan kondisi setelah keyword for (hanya kondisi). Deklarasi dan iterasi variabel counter tidak dituliskan setelah keyword, hanya kondisi perulangan saja. Konsepnya mirip seperti while milik bahasa pemrograman lain.

### c. Repeat Until

Untuk Repeat Until ini mirip seperti for loop biasa namun hanya menggunakan inisiasi dan kondisi saja.

## H. Fungsi

Dalam konteks pemrograman, fungsi adalah sekumpulan blok kode yang dibungkus dengan nama tertentu. Penerapan fungsi yang tepat akan menjadikan kode lebih modular dan juga *dry* (singkatan dari *don't repeat yourself*) yang artinya kita tidak perlu menuliskan banyak kode untuk kegunaan yang sama berulang kali. Cukup deklarasikan sekali saja blok kode sebagai suatu fungsi, lalu panggil sesuai kebutuhan.

### 1. Penerapan Fungsi

Sebenarnya kita sudah mengimplementasikan fungsi pada banyak praktek sebelumnya, yaitu fungsi `main()`. Fungsi `main()` sendiri merupakan fungsi utama pada program Go, yang akan dieksekusi ketika program dijalankan.

Selain fungsi `main()`, kita juga bisa membuat fungsi lainnya. Dan caranya cukup mudah, yaitu dengan menuliskan keyword `func` kemudian diikuti nama fungsi, lalu kurung `()` (yang bisa diisi parameter), dan diakhiri dengan kurung kurawal untuk membungkus blok kode.

Parameter merupakan variabel yang menempel di fungsi yang nilainya ditentukan saat pemanggilan fungsi tersebut. Parameter sifatnya opsional, suatu fungsi bisa tidak memiliki parameter, atau bisa saja memiliki satu atau banyak parameter (tergantung kebutuhan).

### 2. Fungsi dengan Nilai Balik / Return Value

Selain parameter, fungsi bisa memiliki attribute **return value** atau nilai balik. Fungsi yang memiliki return value, saat deklarasinya harus ditentukan terlebih dahulu tipe data dari nilai baliknya.

### 3. Fungsi tanpa Nilai Balik / Return Value

Fungsi juga dapat tidak memiliki nilai balik yang dapat disebut juga sebagai Prosedur. Prosedur dapat dianggap sebagai potongan beberapa instruksi program menjadi suatu instruksi baru yang dibuat untuk mengurangi kerumitan dari kode program yang kompleks pada suatu program yang besar.

## **I. Array**

Array adalah kumpulan data bertipe sama, yang disimpan dalam sebuah variabel. Array memiliki kapasitas yang nilainya ditentukan pada saat pembuatan, menjadikan elemen/data yang disimpan di array tersebut jumlahnya tidak boleh melebihi yang sudah dialokasikan.

Default nilai tiap elemen array pada awalnya tergantung dari tipe datanya. Jika `int` maka tiap element zero value-nya adalah `0`, jika `bool` maka `false`, dan seterusnya. Setiap elemen array memiliki indeks berupa angka yang merepresentasikan posisi urutan elemen tersebut. Indeks array dimulai dari `0`.

## II. GUIDED

### 1. Source Code:

```
package main
import (
    "fmt"
    "time"
)

// Struct untuk barang dalam struk belanja
type Item struct {
    Name string
    Price float64
    Quantity int
}

// Struct untuk struk belanja
type Receipt struct {
    StoreInfo string
    Date time.Time
    Items []Item
    TotalAmount float64
}

// Method untuk menghitung total harga semua item
func (r *Receipt) CalculateTotal() {
    var total float64
    for _, item := range r.Items {
        total += item.Price * float64(item.Quantity)
    }
    r.TotalAmount = total
}

func (r *Receipt) PrintReceipt() {
    fmt.Println("=====")
    fmt.Println(r.StoreInfo)
    fmt.Println("Tanggal:", r.Date.Format("02-01-2006 15:04"))
    fmt.Println("=====")
    fmt.Printf("%-15s %-10s %-8s %-10s\n", "Item", "Harga", "Jumlah", "Total")
    fmt.Println("-----")

    for _, item := range r.Items {
        itemTotal := item.Price * float64(item.Quantity)
        fmt.Printf("%-15s Rp%-9.2f %-8d Rp%-9.2f\n", item.Name, item.Price,
            item.Quantity, itemTotal)
    }

    fmt.Println("=====")
    fmt.Printf("%-35s Rp%-9.2f\n", "Total Belanja:", r.TotalAmount)
```

```

    fmt.Println("=====")
    fmt.Println("Terima kasih telah berbelanja!")
}

func main() {
    receipt := Receipt{
        StoreInfo: "Toko Sembako Makmur\nJl. Raya No. 123, Jakarta",
        Date:      time.Now(),
        Items: []Item{
            {Name: "Beras", Price: 12000, Quantity: 5},
            {Name: "Gula", Price: 15000, Quantity: 2},
            {Name: "Minyak", Price: 20000, Quantity: 1},
            {Name: "Telur", Price: 2000, Quantity: 10},
        },
    }

    receipt.CalculateTotal()
    receipt.PrintReceipt()
}

```

Output:

```

D:\Tugas\SEM2\Alpro\Praktikum\modul6\MODUL6_103112400061>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul6\MODUL6_103112400061
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 26-04-2025 00:07
=====
Item      Harga      Jumlah  Total
-----
Beras     Rp12000.00    5      Rp60000.00
Gula      Rp15000.00    2      Rp30000.00
Minyak    Rp20000.00    1      Rp20000.00
Telur     Rp2000.00    10     Rp20000.00
=====
Total Belanja:                      Rp130000.00
=====
Terima kasih telah berbelanja!

```

Penjelasan:

Program kasir untuk memilih produk dari penyimpanan dan mencetak detail belanja ke dalam struk belanja.

## 2. Source Code:

```

package main
import "fmt"

func main() {
    nilaiMahasiswa := [5]int{85, 90, 78, 88, 95}

    fmt.Println("Data Nilai Mahasiswa")
    fmt.Println("=====")

    for i, nilai := range nilaiMahasiswa {
        fmt.Printf("Mahasiswa %d: %d\n", i+1, nilai)
    }
}

```



```

var total int
for _, nilai := range nilaiMahasiswa {
    total += nilai
}
rataRata := float64(total) / float64(len(nilaiMahasiswa))

fmt.Println("=====")
fmt.Printf("Rata-rata nilai: %.2f\n", rataRata)

// Mencari nilai tertinggi dan terendah
tertinggi := nilaiMahasiswa[0]
terendah := nilaiMahasiswa[0]

for _, nilai := range nilaiMahasiswa {
    if nilai > tertinggi {
        tertinggi = nilai
    }
    if nilai < terendah {
        terendah = nilai
    }
}

fmt.Printf("Nilai tertinggi: %d\n", tertinggi)
fmt.Printf("Nilai terendah: %d\n", terendah)

// Contoh array 2 dimensi
fmt.Println("\nContoh Array 2 Dimensi:")
fmt.Println("=====")

// Nilai ujian mahasiswa dalam 2 mata kuliah (Matematika, Bahasa)
nilaiUjian := [3][2]int{
    {80, 85},
    {90, 75},
    {70, 95},
}

// Menampilkan nilai ujian per mahasiswa
fmt.Println("Nilai Ujian Mahasiswa (Matematika, Bahasa):")
for i, nilai := range nilaiUjian {
    fmt.Printf("Mahasiswa %d: Matematika = %d, Bahasa = %d\n", i+1, nilai[0],
nilai[1])
}
}

```

Output:

```
D:\Tugas\SEM2\Alpro\Praktikum\modul6\103112400061_MODUL6>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul6\103112400061_
Data Nilai Mahasiswa
=====
Mahasiswa 1: 85
Mahasiswa 2: 90
Mahasiswa 3: 78
Mahasiswa 4: 88
Mahasiswa 5: 95
=====
Rata-rata nilai: 87.20
Nilai tertinggi: 95
Nilai terendah: 78

Contoh Array 2 Dimensi:
=====
Nilai Ujian Mahasiswa (Matematika, Bahasa):
Mahasiswa 1: Matematika = 80, Bahasa = 85
Mahasiswa 2: Matematika = 90, Bahasa = 75
Mahasiswa 3: Matematika = 70, Bahasa = 95
```

Penjelasan:

Program mencetak nilai siswa, mencari rata-rata nilai siswa, mencari nilai tertinggi dan terendah, dan juga mencetak nilai ujian mahasiswa menggunakan array 2 dimensi.

### III. UNGUIDED

#### 1. Latihan no. 1

Source Code:

```
// KEISHIN NAUFA ALFARIDZHI
// 103112400061
package main
import (
    "fmt"
    "math"
)

type titik struct {
    x, y int
}

type lingkaran struct {
    pusatLingkaran titik
    radius int
}

func main() {
    var (
        lingkaran1, lingkaran2 lingkaran
        p titik
        dalamLingkaran1, dalamLingkaran2 bool
    )
    fmt.Scan(&lingkaran1.pusatLingkaran.x, &lingkaran1.pusatLingkaran.y,
        &lingkaran1.radius)
```

```

    fmt.Scan(&lingkaran2.pusatLingkaran.x, &lingkaran2.pusatLingkaran.y,
&lingkaran2.radius)
    fmt.Scan(&p.x, &p.y)

    dalamLingkaran1 = didalam(lingkaran1, p)
    dalamLingkaran2 = didalam(lingkaran2, p)

    if dalamLingkaran1 == true && dalamLingkaran2 == true {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dalamLingkaran1 == true {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if dalamLingkaran2 == true {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

func jarak(p, q titik) float64 {
    cx := p.x - q.x
    cy := p.y - q.y

    return math.Sqrt(float64(cx*cx + cy*cy))
}

func didalam(c lingkaran, p titik) bool {
    jarakMenujuPusat := jarak(c.pusatLingkaran, p)
    return jarakMenujuPusat <= float64(c.radius)
}

```

Output:

```

D:\Tugas\SEM2\Alpro\Praktikum\modul6\MODUL6_103112400061>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul6\MODUL6_103112400061\
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1

D:\Tugas\SEM2\Alpro\Praktikum\modul6\MODUL6_103112400061>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul6\MODUL6_103112400061\
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2

D:\Tugas\SEM2\Alpro\Praktikum\modul6\MODUL6_103112400061>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul6\MODUL6_103112400061\
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2

D:\Tugas\SEM2\Alpro\Praktikum\modul6\MODUL6_103112400061>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul6\MODUL6_103112400061\
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2

```

Deskripsi Program:

Menentukan suatu titik koordinat apakah titik tersebut berada dalam lingkaran 1 atau lingkaran 2 atau keduanya. Memiliki tempat untuk menyimpan tipe data yaitu struct.

Terdapat 2 struct, bernama titik dan lingkaran, dengan titik diisi dengan koordinat x dan y, lalu lingkaran berisi titik pusat lingkaran dan radius dari lingkaran. Menggunakan 2 function yaitu jarak dan didalam. Function jarak digunakan untuk mencari jarak antara kedua lingkaran dengan cara mengurangi pusat lingkaran dengan titik yang diinputkan oleh user. Kemudian function didalam digunakan untuk cek apakah titik yang diinputkan oleh user terdapat dalam lingkaran 1 atau lingkaran 2 atau keduanya.

## 2. Latihan no. 2

Source Code:

```
// KEISHIN NAUFA ALFARIDZHI
// 103112400061
package main

import (
    "fmt"
    "math"
)

func main() {
    var (
        storage [10]int
        temporaryStorage []int
        n, x, indexToDelete int
    )

    fmt.Print("Masukkan data ke dalam storage: ")
    n = len(storage)
    for i := 0; i < n; i++ {
        fmt.Scan(&storage[i])
    }

    // a. Menampilkan keseluruhan isi dari array
    fmt.Print("Berikut merupakan isi dari keseluruhan storage (array): ")
    fmt.Println(storage)

    // b. Menampilkan elemen-elemen array dengan indeks ganjil saja
    fmt.Print("Berikut merupakan isi dari storage (array) dengan indeks ganjil: ")
    for i := 0; i < n; i++ {
        if i % 2 == 1 {
            temporaryStorage = append(temporaryStorage, storage[i])
        }
    }
    fmt.Println(temporaryStorage)
    temporaryStorage = nil
}
```

**// c. Menampilkan elemen-elemen array dengan indeks genap saja (asumsi indek ke-0 adalah genap)**

```
fmt.Print("Berikut merupakan isi dari storage (array) dengan indeks genap: ")
for i := 0; i < n; i++ {
    if i % 2 == 0 {
        temporaryStorage = append(temporaryStorage, storage[i])
    }
}
fmt.Println(temporaryStorage)
temporaryStorage = nil
```

**// d. Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x. x bisa diperoleh dari masukan pengguna**

```
fmt.Print("Berikut merupakan isi dari storage (array) dengan indeks kelipatan x: ")
fmt.Scan(&x)
for i, num := range storage {
    if i % x == 0 && i != 0 {
        temporaryStorage = append(temporaryStorage, storage[num])
    }
}
fmt.Println(temporaryStorage)
```

**// e. Menghapus elemen array pada indeks tertentu, asumsi indeks yang hapus selalu valid. Tampilkan keseluruhan isi dari arraynya, pastikan data yang dihapus tidak tampil**

```
temporaryStorage = storage[:]
fmt.Print("Masukkan index yang akan dihapus: ")
fmt.Scan(&indexToDelete)
temporaryStorage = append(temporaryStorage[:indexToDelete],
temporaryStorage[indexToDelete+1:]...)
fmt.Println(temporaryStorage)
temporaryStorage = nil
```

**// f. Menampilkan rata-rata dari bilangan yang ada di dalam array**

```
var rataRata, totalNum float64
for _, num := range storage {
    totalNum += float64(storage[num])
}
rataRata = totalNum / float64(n)
fmt.Print("Rata-rata dari bilangan dalam storage: ")
fmt.Println(rataRata)
```

**// g. Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut**

```
var mean, sum, diff, squaredDiff, variance float64
sum = 0.0
for _, num := range storage {
    sum += float64(num)
}
mean = sum / float64(len(storage))
```

```

squaredDiff = 0.0
for _, num := range storage {
    diff = float64(num) - mean
    squaredDiff += diff * diff
}
variance = math.Sqrt(squaredDiff / float64(len(storage)-1))
fmt.Println("Standar deviasi dari array storage:", variance)

// h. Menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang
telah diisi tersebut
fmt.Println("\n===== Frekuensi =====")
frequency := make(map[int]int)
for _, num := range storage {
    frequency[num]++
}
for num, freq := range frequency {
    fmt.Printf("Frekuensi dari bilangan %d: %d\n", num, freq)
}
fmt.Println("===== ")
}

```

Output:

```

D:\Tugas\SEM2\Alpro\Praktikum>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul6\103112400061
Masukkan data ke dalam storage: 1 1 1 1 2 3 4 4 5 6
Berikut merupakan isi dari keseluruhan storage (array): [1 1 1 1 2 3 4 4 5 6]
Berikut merupakan isi dari storage (array) dengan indeks ganjil: [1 1 3 4 6]
Berikut merupakan isi dari storage (array) dengan indeks genap: [1 1 2 4 5]
Berikut merupakan isi dari storage (array) dengan indeks kelipatan x: 3
[1 2 4]
Masukkan index yang akan dihapus: 2
[1 1 1 2 3 4 4 5 6]
Rata-rata dari bilangan dalam storage: 2.4
Standar deviasi dari array storage: 2.0027758514399734

===== Frekuensi =====
Frekuensi dari bilangan 4: 2
Frekuensi dari bilangan 5: 1
Frekuensi dari bilangan 6: 2
Frekuensi dari bilangan 1: 3
Frekuensi dari bilangan 2: 1
Frekuensi dari bilangan 3: 1
=====

```

Deskripsi Program:

Program untuk menampilkan output dari perintah poin a hingga h pada soal.

Berikut deskripsi tiap poin:

- Menampilkan array dengan `fmt.Print()`
- Cek index ganjil pada array **storage** kemudian memasukkan nilai index ganjil tersebut ke dalam temporary array **temporaryStorage**.

- c. Cek index genap pada array **storage** kemudian memasukkan nilai index genap tersebut ke dalam temporary array **temporaryStorage**.
- d. Cek index kelipatan-x pada array **storage** kemudian memasukkan nilai index kelipatan-x tersebut ke dalam temporary array **temporaryStorage**.
- e. Memindahkan isi array **storage** ke **temporaryStorage** untuk kemudian dihapus indexnya menggunakan slice.
- f. Mencari nilai index tidak null yang kemudian akan dihitung rata-rata data nya.
- g. Menampilkan standar deviasi dengan mengambil data dalam array.
- h. Mencari frekuensi tiap data dalam array menggunakan map.

### 3. Latihan no. 3

Source Code:

```
// KEISHIN NAUFA ALFARIDZHI
// 103112400061
package main
import "fmt"

func main() {
    var (
        match [50]string
        klubA, klubB string
        skorKlubA, skorKlubB, jumlahMatch int
    )

    fmt.Print("Klub A: ")
    fmt.Scan(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scan(&klubB)

    for i := 0; i < len(match); i++ {
        fmt.Printf("Pertandingan %d : ", i+1)
        fmt.Scan(&skorKlubA, &skorKlubB)
        if skorKlubA < 0 || skorKlubB < 0 {
            break
        }
        if skorKlubA > skorKlubB {
            match[i] = match[i] + klubA
        } else if skorKlubA < skorKlubB {
            match[i] = match[i] + klubB
        } else {
            match[i] = match[i] + "Draw"
        }
        jumlahMatch += 1
    }
}
```

```

for i := 0; i < jumlahMatch; i++ {
    fmt.Printf("Hasil %d : %s\n", i+1, match[i])
}
fmt.Println("Pertandingan selesai")
}

```

Output:

```

D:\Tugas\SEM2\Alpro\Praktikum>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul6\103112400061
Klub A: MU
Klub B: Inter
Pertandingan 1 : 2 0
Pertandingan 2 : 1 2
Pertandingan 3 : 2 2
Pertandingan 4 : 0 1
Pertandingan 5 : 3 2
Pertandingan 6 : 1 0
Pertandingan 7 : 5 2
Pertandingan 8 : 2 3
Pertandingan 9 : -1 2
Hasil 1 : MU
Hasil 2 : Inter
Hasil 3 : Draw
Hasil 4 : Inter
Hasil 5 : MU
Hasil 6 : MU
Hasil 7 : MU
Hasil 8 : Inter
Pertandingan selesai

```

Deskripsi Program:

Program untuk mencatat pertandingan antara dua klub sepak bola. Menggunakan for loop untuk perulangan dengan kondisi perulangan akan tetap berjalan selama tidak melewati batas array yang telah ditentukan yakni **[50]string**. Kemudian tinggal menentukan apakah klubA atau klubB yang menang atau Draw tiap indeks array nya.

#### 4. Latihan no. 4

Source Code:

```

// KEISHIN NAUFA ALFARIDZHI
// 103112400061
package main
import "fmt"

const NMAX int = 127
type tabel [NMAX]rune

func main() {
    var (
        tabel tabel

```



```

        n int
        isPalindrom bool
    )

    isiArray(&tabel, &n)

    isPalindrom = palindrom(tabel, n)

    balikanArray(&tabel, n)

    fmt.Printf("%-10s : ", "Reverse teks")
    cetakArray(tabel, n)

    fmt.Printf("%-10s %t\n", "Palindrom ?", isPalindrom)
}

func isiArray(t *tabel, n *int) {
    var char rune

    fmt.Printf("%-12s : ", "Teks")
    for {
        fmt.Scanf("%c", &char)

        if char == ' ' {
            continue
        }

        if char == '.' {
            break
        }

        if *n < NMAX {
            (*t)[*n] = char
            *n++
        } else {
            break
        }
    }
}

func cetakArray(t tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Printf("%c ", t[i])
    }
    fmt.Println()
}

func balikanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        (*t)[i], (*t)[n-1-i] = (*t)[n-1-i], (*t)[i]
    }
}

```

```

    }
}

func palindrom(t tabel, n int) bool {
    var tempTabel tabel

    for i := 0; i < n; i++ {
        tempTabel[i] = t[i]
    }
    balikanArray(&tempTabel, n)
    for i := 0; i < n; i++ {
        if t[i] != tempTabel[i] {
            return false
        }
    }
    return true
}

```

Output:

```

D:\Tugas\SEM2\Alpro\Praktikum>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul6\103112400061_
Teks          : K A T A K .
Reverse teks  : K A T A K
Palindrom ? true

D:\Tugas\SEM2\Alpro\Praktikum>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul6\103112400061_
Teks          : S E N A N G .
Reverse teks  : G N A N E S
Palindrom ? false

```

Deskripsi Program:

Program untuk reverse teks dari sebuah array dan menentukan apakah teks yang diinputkan merupakan palindrom. Memiliki array bernama tabel sebesar 127 yang bertipe data rune. Menggunakan beberapa fungsi:

1. isiArray()

Untuk input suatu kata ke dalam array. Menggunakan `fmt.Scanf("%c")` untuk mengubah input string menjadi bentuk character yang dapat dibaca oleh tipe data rune. Tiap karakter akan dimasukkan kepada indeks menggunakan pointer (\*n).

2. cetakArray()

Digunakan untuk mencetak karakter yang ada dalam array.

3. balikanArray()

Digunakan untuk membalik isi data array dari bagian depan-ke-belakang dan belakang-ke-depan.

#### 4. palindrom()

Untuk cek apakah kata yang diinputkan merupakan palindrom dengan cara membandingkan array **tabel** dengan **tempTabel**. **tempTabel** akan menggunakan fungsi `balikanArray()` untuk membalik kata, setelah itu dibandingkan apakah **tabel** dengan **tempTabel** memiliki isi data yang urut dan sama.

## IV. KESIMPULAN

Pada praktikum ini telah dibahas perihal cara menerapkan dan menggunakan array dalam bahasa Go. Array dapat menyimpan banyak data sekaligus tanpa harus mendeklarasikan variabel satu per satu untuk data nya. Array sangat cocok digunakan pada program dengan data yang banyak, dapat digunakan seperti database.

## V. DAFTAR PUSTAKA

Noval Agung Prayogo. *Dasar Pemrograman Golang*. Diakses pada 01 Oktober 2024.  
<https://dasarpemrogramangolang.novalagung.com>

Annisa Nur Isnaeni. *Golang — Seleksi Kondisi*. Diakses pada 01 Oktober 2024.  
<https://medium.com/@annisaisna/golang-seleksi-kondisi-f988ead004b4>

Parvez Alam, *Golang for loop example* | *Golang Loops Tutorial – Phpflow.com*  
<https://medium.com/@parvez1487/golang-for-loop-example-golang-loops-tutorial-phpflow-com-f4b2b0e57944>