

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7  
STRUCT DAN ARRAY**



**Disusun Oleh :**

**Agnes Refilina Fiska/ 2311102126**

**S1-IF-11-05**

**Dosen Pengampu :**

**Arif Amrulloh, S.Kom.,M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

### A. ARRAY

Array adalah tipe data dasar yang digunakan untuk menyimpan sekumpulan elemen dengan tipe data yang sama. Di bahasa Go, array memiliki ukuran tetap yang ditetapkan saat deklarasi.

Karakter Array:

#### 1. Ukuran Standar:

Dibandingkan dengan tipe data lain seperti slice, ukuran array di Go ditentukan saat deklarasi dan tidak dapat diubah selama runtime.

Contoh :

```
var angka [5]int // Array dengan ukuran tetap 5
```

#### 2. Data homogen terdiri dari:

Semua elemen dalam array harus memiliki tipe data yang sama—misalnya, int, string, atau float64.

Contoh :

```
var nama [3]string = [3]string{"Ayu", "Budi", "Citra"}
```

#### 3. Index dimulai dengan zero:

Element awal array berada di indeks 0, elemen kedua berada di indeks 1, dan seterusnya.

Contoh :

```
fmt.Println(nama[0]) // Mengakses elemen pertama
```

#### 4. Akses ke elemen:

Dengan menggunakan notasi array[index], Anda dapat mengakses elemen dalam array dengan menggunakan indeks.

Contoh :

```
angka[0] = 10 // Mengubah elemen pertama menjadi 10
```

Syarat Array:

Ukuran dan tipe data array dapat dideklarasikan. Dalam Go, ada beberapa cara untuk mendeklarasikan dan menginisialisasi array.

1. Deklarasi Array Kosong
2. Inisialisasi Array dengan Nilai
3. Deklarasi Array dengan Panjang Otomatis
4. Penggunaan Array
5. Array Multidimensi

### B. STRUCT

Struktur sering digunakan untuk menunjukkan objek atau entitas dengan beberapa atribut. Ini adalah tipe data komposit yang memungkinkan kita mengelompokkan variabel dengan tipe yang berbeda dalam satu entitas.

Karakter Struktur:

#### 1. Tipe Data Heterogen:

Dalam struct, field dapat memiliki berbagai tipe data, yang memungkinkan penggunaan tipe data yang lebih kompleks.

## 2. Nama Field:

Setiap field struct memiliki nama sendiri.

## 3. Notasi Titik:

Kita menggunakan notasi titik (.) untuk mengakses atau mengubah area.

Contoh :

```
mahasiswa.Umur = 21
```

## 4. Komposisi Struct:

Struct dapat menggabungkan struct lain karena mendukung komposisi.

## Deklarasi Struktur:

Deklarasi struktur dimulai dengan kata kunci type, nama struktur, dan definisi field dalam kurung kurawal.

## II. GUIDED

### 1. Guided 1

#### Soal Studi Case

Program sederhana untuk menghitung lama waktu parkir berdasarkan waktu kedatangan dan waktu pulang

#### Sourcecode

```
package main

import "fmt"

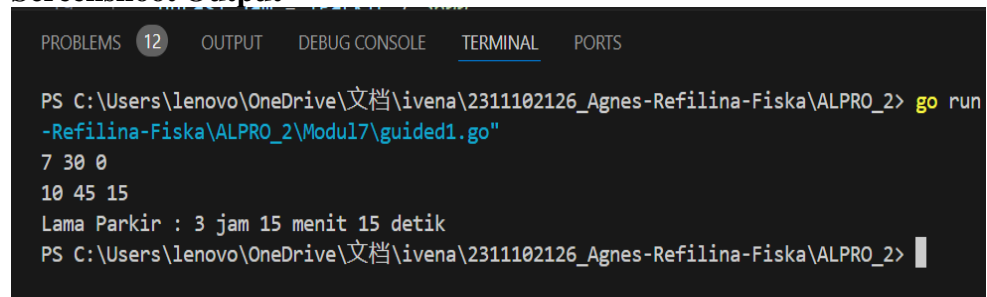
type waktu struct {
    jam, menit, detik int
}

func main() {
    var wParkir, wPulang, durasi waktu
    var dParkir, dPulang, lParkir int

    fmt.Scan(&wParkir.jam, &wParkir.menit,
&wParkir.detik)
    fmt.Scan(&wPulang.jam, &wPulang.menit,
&wPulang.detik)
    dParkir = wParkir.detik + wParkir.menit*60 +
wParkir.jam*3600 // Konversi ke detik
    dPulang = wPulang.detik + wPulang.menit*60 +
wPulang.jam*3600 // detik
    lParkir = dPulang -
dParkir //detik
    dari pulang-datang

    durasi.jam = lParkir / 3600
    durasi.menit = lParkir % 3600 / 60
    durasi.detik = lParkir % 3600 % 60 //17
    fmt.Printf("Lama Parkir : %d jam %d menit %d
detik", durasi.jam, durasi.menit, durasi.detik)
}
```

## Screenshoot Output



```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2> go run
-Refilina-Fiska\ALPRO_2\Modul7\guided1.go
7 30 0
10 45 15
Lama Parkir : 3 jam 15 menit 15 detik
PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2> |
```

## Deskripsi Program :

Program di atas merupakan program Go yang menghitung durasi parkir kendaraan berdasarkan waktu masuk dan waktu keluar. Program menggunakan struktur data `waktu` yang memiliki tiga komponen: jam, menit, dan detik. Program menerima input waktu parkir (masuk) dan waktu pulang (keluar) dari pengguna, kemudian mengkonversi kedua waktu tersebut ke dalam satuan detik untuk memudahkan perhitungan selisih waktu. Setelah mendapatkan selisih waktu dalam detik, program mengkonversi kembali ke format jam, menit, dan detik yang lebih mudah dibaca, lalu menampilkan hasil durasi parkir dalam format "Lama Parkir: X jam X menit X detik". Program ini berguna untuk menghitung berapa lama sebuah kendaraan parkir di suatu tempat dengan tepat hingga satuan detik.

## 2. Guided 2

### Soal Studi Case

Program sederhana untuk validasi duplikasi nama pada daftar teman

### Sourcecode

```
package main

import (
    "fmt"
)

func sudahAda(daftarTeman []string, nama string)
bool {
    for _, teman := range daftarTeman {
        if teman == nama {
            return true
        }
    }
    return false
}
```

```

func main() {
    daftarTeman := []string{"Andi", "Budi",
"Cici"}

    namaBaru := []string{"Dewi", "Budi", "Eka"}

    for _, nama := range namaBaru {
        if !sudahAda(daftarTeman, nama) {
            daftarTeman = append(daftarTeman,
nama)
        } else {
            fmt.Println("Nama", nama, "sudah ada
dalam daftar.")
        }
    }

    fmt.Println("Daftar Teman:", daftarTeman)
}

```

### Screenshoot Output

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2> go run
-Refilina-Fiska\ALPRO_2\Modul7\guided2.go
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2>

```

### Deskripsi Program :

Program di atas adalah program Go yang mengelola daftar nama teman dengan mencegah duplikasi nama. Program memiliki fungsi `sudahAda` yang bertugas memeriksa apakah sebuah nama sudah terdapat dalam daftar teman yang ada. Program dimulai dengan daftar awal yang berisi tiga nama ("Andi", "Budi", "Cici"), kemudian mencoba menambahkan tiga nama baru ("Dewi", "Budi", "Eka"). Saat menambahkan nama baru, program memeriksa setiap nama menggunakan fungsi `sudahAda` - jika nama belum ada, maka nama tersebut akan ditambahkan ke daftar, namun jika nama sudah ada (seperti "Budi"), program akan menampilkan pesan bahwa nama tersebut sudah ada dalam daftar. Pada akhirnya, program menampilkan daftar teman yang sudah diperbarui, yang hanya akan menambahkan nama-nama yang belum ada sebelumnya.

### 3. Guided 3

#### Soal Studi Case

Program sederhana untuk menampilkan daftar harga buah

#### Sourcecode

```
package main

import "fmt"

func main() {
    hargaBuah := map[string]int{
        "Apel": 5000,
        "Pisang": 3000,
        "Mangga": 7000,
    }

    fmt.Println("Harga Buah:")
    for buah, harga := range hargaBuah {
        fmt.Printf("%s: Rp%d\n", buah, harga)
    }

    fmt.Print("Harga buah Mangga = ",
        hargaBuah["Mangga"])
}
```

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The command executed is 'go run -Refilina-Fiska\ALPRO\_2\Modul7\guided3.go'. The output shows the price of three fruits: 'Apel: Rp5000', 'Pisang: Rp3000', and 'Mangga: Rp7000'. Below this, it prints 'Harga buah Mangga = 7000'. The prompt 'PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126\_Agnes-Refilina-Fiska\ALPRO\_2>' is visible at the bottom.

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2> go run
-Refilina-Fiska\ALPRO_2\Modul7\guided3.go
Harga Buah:
Apel: Rp5000
Pisang: Rp3000
Mangga: Rp7000
Harga buah Mangga = 7000
PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2>
```

#### Screenshot Output

#### Deskripsi Program :

Program di atas adalah program Go yang menggunakan struktur data map untuk menyimpan daftar harga buah-buahan. Program membuat sebuah map bernama `hargaBuah` yang memetakan nama buah (string) dengan harganya (integer), di mana terdapat tiga jenis buah yaitu Apel seharga Rp5000,

Pisang seharga Rp3000, dan Mangga seharga Rp7000. Program kemudian menampilkan seluruh daftar harga buah menggunakan perulangan `for range` yang mengiterasi setiap pasangan key-value dalam map, dan mencetak nama buah beserta harganya. Terakhir, program menampilkan harga spesifik untuk buah Mangga dengan mengakses langsung nilai dalam map menggunakan key "Mangga".

### III. UNGUIDED

#### 1. Unguided 1

##### Soal Studi Case

Suatu lingkaran didefinisikan dengan koordinat titik pusat  $(cx, cy)$  dengan radius  $r$ . Apabila diberikan dua buah lingkaran, maka tentukan posisi sebuah titik sembarang  $(x, y)$  berdasarkan dua lingkaran tersebut. Gunakan tipe bentukan titik untuk menyimpan koordinat, dan tipe bentukan lingkaran untuk menyimpan titik pusat lingkaran dan radiusnya.

**Masukan** terdiri dari beberapa tiga baris. Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingkaran 2, sedangkan baris ketiga adalah koordinat titik sembarang. Asumsi sumbu x dan y dari semua titik dan juga radius direpresentasikan dengan bilangan bulat.

**Keluaran** berupa string yang menyatakan posisi titik "Titik di dalam lingkaran 1 dan 2", "Titik di dalam lingkaran 1", "Titik di dalam lingkaran 2", atau "Titik di luar lingkaran 1 dan 2".

No	Masukan	Keluaran
1	1 1 5 8 8 4 2 2	Titik di dalam lingkaran 1
2	1 2 3 4 5 6 7 8	Titik di dalam lingkaran 2
3	5 10 15 -15 4 20 0 0	Titik di dalam lingkaran 1 dan 2
4	1 1 5 8 8 4 15 20	Titik di luar lingkaran 1 dan 2

##### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// Point represents a 2D point with x,y
coordinates
```



```

type Point struct {
    x, y float64
}

// Circle represents a circle with center point
and radius
type Circle struct {
    center Point
    radius float64
}

// distance calculates Euclidean distance between
two points
func distance(p1, p2 Point) float64 {
    dx := p1.x - p2.x
    dy := p1.y - p2.y
    return math.Sqrt(dx*dx + dy*dy)
}

// isPointInside checks if a point is inside a
circle
func (c Circle) isPointInside(p Point) bool {
    return distance(c.center, p) <= c.radius
}

// checkPointPosition determines the position of a
point relative to two circles
func checkPointPosition(c1, c2 Circle, p Point)
string {
    in1 := c1.isPointInside(p)
    in2 := c2.isPointInside(p)

    if in1 && in2 {
        return "Titik di dalam lingkaran 1 dan 2"
    } else if in1 {
        return "Titik di dalam lingkaran 1"
    } else if in2 {
        return "Titik di dalam lingkaran 2"
    }
    return "Titik di luar lingkaran 1 dan 2"
}

func main() {
    var T int
    fmt.Println("Masukkan jumlah test case:")
    fmt.Scan(&T)

    for t := 1; t <= T; t++ {
        // Read first circle coordinates and
radius
        var x1, y1, r1 float64
        fmt.Scan(&x1, &y1, &r1)
        circle1 := Circle{center: Point{x1, y1},

```

```

radius: r1}

        // Read second circle coordinates and
radius
        var x2, y2, r2 float64
        fmt.Scan(&x2, &y2, &r2)
        circle2 := Circle{center: Point{x2, y2},
radius: r2}

        // Read test point coordinates
        var px, py float64
        fmt.Scan(&px, &py)
        testPoint := Point{px, py}

        // Check and print result
        result := checkPointPosition(circle1,
circle2, testPoint)
        fmt.Printf("Case #%d: %s\n", t, result)
    }
}

```

## Screenshoot Output

```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2> go run
-Refilina-Fiska\ALPRO_2\Modul7\unguided1.go
Masukkan jumlah test case:
3
1 1 5
8 8 4
2 2
Case #1: Titik di dalam lingkaran 1
3
1 2 3
4 5 6
7 8
Case #2: Titik di dalam lingkaran 2
3
5 10 15
-15 4 20
Case #3: Titik di luar lingkaran 1 dan 2
PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2>

```

## Deskripsi Program :

Program di atas adalah program Go yang dirancang untuk memeriksa posisi suatu titik relatif terhadap dua lingkaran dalam bidang 2D. Program menerima beberapa test case, di mana untuk setiap test case, dua lingkaran dan satu titik diuji untuk menentukan apakah titik tersebut berada di dalam salah satu atau kedua lingkaran.

Struktur `Point` digunakan untuk merepresentasikan titik dengan koordinat x dan y, sementara struktur `Circle` merepresentasikan lingkaran dengan pusat dan radius. Program ini menghitung jarak Euclidean antara titik dan pusat lingkaran untuk menentukan apakah titik berada di dalam lingkaran.

Fungsi `checkPointPosition` digunakan untuk memeriksa posisi titik relatif terhadap kedua lingkaran dan memberikan hasil yang sesuai.

Pengguna diminta untuk memasukkan jumlah test case, kemudian untuk setiap test case, koordinat dan radius kedua lingkaran, serta koordinat titik yang akan diuji. Program kemudian menampilkan hasil apakah titik berada di dalam kedua lingkaran, hanya satu lingkaran, atau berada di luar kedua lingkaran. Dengan menggunakan konsep dasar geometri dan penghitungan jarak, program ini dapat memberikan informasi yang akurat tentang posisi titik tersebut.

## 2. Unguided 2

### Soal Studi Case

Sebuah array digunakan untuk menampung sekumpulan bilangan bulat. Buatlah program yang digunakan untuk mengisi array tersebut sebanyak N elemen nilai. Asumsikan array memiliki kapasitas penyimpanan data sejumlah elemen tertentu. Program dapat menampilkan beberapa informasi berikut:

- a. Menampilkan keseluruhan isi dari array.
- b. Menampilkan elemen-elemen array dengan indeks ganjil saja.
- c. Menampilkan elemen-elemen array dengan indeks genap saja (asumsi indek ke-0 adalah genap).
- d. Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x. x bisa diperoleh dari masukan pengguna.
- e. Menghapus elemen array pada indeks tertentu, asumsi indeks yang hapus selalu valid. Tampilkan keseluruhan isi dari arraynya, pastikan data yang dihapus tidak tampil
- f. Menampilkan rata-rata dari bilangan yang ada di dalam array.
- g. Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut.
- h. Menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi tersebut.

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// ArrayOps struct to hold the array and its
operations
type ArrayOps struct {
    arr    []int
    length int
}

// NewArrayOps creates a new ArrayOps instance
func NewArrayOps(capacity int) *ArrayOps {
    return &ArrayOps{
        arr:    make([]int, 0, capacity),
        length: 0,
    }
}
```

```

// inputArray fills the array with user input
func (a *ArrayOps) inputArray() {
    fmt.Print("Masukkan jumlah elemen (N): ")
    var n int
    fmt.Scan(&n)
    a.length = n

    fmt.Println("Masukkan", n, "bilangan:")
    for i := 0; i < n; i++ {
        var num int
        fmt.Scan(&num)
        a.arr = append(a.arr, num)
    }
}

// displayAll shows all elements
func (a *ArrayOps) displayAll() {
    fmt.Print("Seluruh elemen array: ")
    for _, v := range a.arr {
        fmt.Print(v, " ")
    }
    fmt.Println()
}

// displayOddIndices shows elements at odd indices
func (a *ArrayOps) displayOddIndices() {
    fmt.Print("Elemen dengan indeks ganjil: ")
    for i := 1; i < len(a.arr); i += 2 {
        fmt.Print(a.arr[i], " ")
    }
    fmt.Println()
}

// displayEvenIndices shows elements at even
indices
func (a *ArrayOps) displayEvenIndices() {
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 0; i < len(a.arr); i += 2 {
        fmt.Print(a.arr[i], " ")
    }
    fmt.Println()
}

// displayMultipleIndices shows elements at
indices multiple of x
func (a *ArrayOps) displayMultipleIndices() {
    fmt.Print("Masukkan nilai x untuk menampilkan
kelipatan: ")
    var x int
    fmt.Scan(&x)

```

```

        fmt.Printf("Elemen dengan indeks kelipatan %d:
", x)

        for i := 0; i < len(a.arr); i++ {
            if i%x == 0 {
                fmt.Print(a.arr[i], " ")
            }
        }
        fmt.Println()
    }

// deleteElement removes element at specified
index
func (a *ArrayOps) deleteElement() {
    fmt.Print("Masukkan indeks yang akan dihapus:
")

    var idx int
    fmt.Scan(&idx)

    a.arr = append(a.arr[:idx], a.arr[idx+1:]...)
    a.length--

    fmt.Print("Array setelah penghapusan: ")
    a.displayAll()
}

// calculateAverage computes mean of array
elements
func (a *ArrayOps) calculateAverage() float64 {
    sum := 0
    for _, v := range a.arr {
        sum += v
    }
    avg := float64(sum) / float64(len(a.arr))
    fmt.Printf("Rata-rata: %.2f\n", avg)
    return avg
}

// calculateStdDev computes standard deviation
func (a *ArrayOps) calculateStdDev() {
    avg := a.calculateAverage()
    sumSquaredDiff := 0.0

    for _, v := range a.arr {
        diff := float64(v) - avg
        sumSquaredDiff += diff * diff
    }

    stdDev := math.Sqrt(sumSquaredDiff /
float64(len(a.arr)))

```

```

        fmt.Printf("Standar deviasi: %.2f\n", stdDev)
    }

    // calculateFrequency counts occurrences of a
number
    func (a *ArrayOps) calculateFrequency() {
        fmt.Print("Masukkan bilangan yang ingin
dihitung frekuensinya: ")
        var num int
        fmt.Scan(&num)

        freq := 0
        for _, v := range a.arr {
            if v == num {
                freq++
            }
        }

        fmt.Printf("Frekuensi bilangan %d: %d\n", num,
freq)
    }

    func main() {
        // Initialize array with capacity 100
        arrOps := NewArrayOps(100)

        // Input array elements
        arrOps.inputArray()

        for {
            fmt.Println("\nMenu:")
            fmt.Println("1. Tampilkan semua elemen")
            ganjil")
            fmt.Println("2. Tampilkan elemen indeks
genap")
            fmt.Println("3. Tampilkan elemen indeks
kelipatan x")
            fmt.Println("4. Tampilkan elemen indeks")
            fmt.Println("5. Hapus elemen")
            fmt.Println("6. Hitung rata-rata")
            fmt.Println("7. Hitung standar deviasi")
            fmt.Println("8. Hitung frekuensi")
            fmt.Println("0. Keluar")

            fmt.Print("Pilih menu (0-8): ")
            var choice int
            fmt.Scan(&choice)

            switch choice {
            case 0:

```

```

        return
    case 1:
        arrOps.displayAll()
    case 2:
        arrOps.displayOddIndices()
    case 3:
        arrOps.displayEvenIndices()
    case 4:
        arrOps.displayMultipleIndices()
    case 5:
        arrOps.deleteElement()
    case 6:
        arrOps.calculateAverage()
    case 7:
        arrOps.calculateStdDev()
    case 8:
        arrOps.calculateFrequency()
    default:
        fmt.Println("Pilihan tidak valid!")
    }
}
}

```

## Screenshot Output

```

PROBLEMS 27 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2> go run '-Refilina-Fiska\ALPRO_2\Modul17\unguided2.go'
Masukkan jumlah elemen (N): 7
Masukkan 7 bilangan:
0 1 2 3 4 5 6

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen indeks kelipatan x
5. Hapus elemen
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi
0. Keluar
Pilih menu (0-8): 2
Elemen dengan indeks ganjil: 1 3 5

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen indeks kelipatan x
5. Hapus elemen
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi
0. Keluar
Pilih menu (0-8): 3
Elemen dengan indeks genap: 0 2 4 6

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen indeks kelipatan x
5. Hapus elemen

```



Masukkan nilai x untuk menampilkan kelipatan: 2  
Elemen dengan indeks kelipatan 2: 0 2 4 6

Menu:

1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen indeks kelipatan x
5. Hapus elemen
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi
0. Keluar

Pilih menu (0-8): 6

Rata-rata: 3.00

Menu:

1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen indeks kelipatan x
5. Hapus elemen
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi
0. Keluar

Pilih menu (0-8): 7

Rata-rata: 3.00

Standar deviasi: 2.00

```
Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen indeks kelipatan x
5. Hapus elemen
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi
0. Keluar
Pilih menu (0-8): 8
Masukkan bilangan yang ingin dihitung frekuensinya: 6
Frekuensi bilangan 6: 1
```

### **Deskripsi Program :**

Program di atas adalah implementasi Go yang memungkinkan pengguna untuk melakukan berbagai operasi pada array, seperti menampilkan elemen, menghitung rata-rata, menghapus elemen, dan menghitung frekuensi elemen tertentu. Pengguna dapat berinteraksi dengan array melalui menu yang menawarkan berbagai pilihan. Program menggunakan struktur `ArrayOps` yang menyimpan array dan menyediakan metode untuk mengelola array. Metode yang tersedia termasuk memasukkan elemen ke dalam array, menampilkan elemen pada indeks ganjil/genap, menampilkan elemen pada indeks kelipatan tertentu, menghapus elemen berdasarkan indeks, serta menghitung rata-rata, standar deviasi, dan frekuensi elemen tertentu. Setelah array diisi, pengguna dapat memilih operasi yang diinginkan melalui menu dan dapat terus memilih hingga memilih opsi "Keluar". Program memberikan validasi input yang sederhana dan memungkinkan pengguna untuk kembali ke menu utama setelah setiap operasi. Dengan fitur-fitur ini, program memudahkan pengelolaan array dan cocok untuk belajar pengoperasian array di Go.

### **3. Unguided 3**

#### **Soal Studi Case**

Sebuah program digunakan untuk menyimpan dan menampilkan nama-nama klub yang memenangkan pertandingan bola pada suatu grup pertandingan. Buatlah program yang digunakan untuk merekap skor pertandingan bola 2 buah klub bola yang berlaga. Pertama-tama program meminta masukan nama-nama klub yang bertanding, kemudian program meminta masukan skor hasil pertandingan kedua klub tersebut. Yang disimpan dalam array adalah nama-nama klub yang menang saja. Proses

input skor berhenti ketika skor salah satu atau kedua klub tidak valid (negatif). Di akhir program, tampilkan daftar klub yang memenangkan pertandingan.

Perhatikan sesi interaksi pada contoh berikut ini (teks bergaris bawah adalah input/read)

```
Klub A : MU
Klub B : Inter
Pertandingan 1 : 2 0 // MU = 2 sedangkan Inter = 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
```

## Sourcecode

```
package main

import (
    "fmt"
)

type Match struct {
    clubAScore int
    clubBScore int
}

func determineWinner(clubA, clubB string, match Match) string {
    if match.clubAScore < 0 || match.clubBScore < 0 {
        return "Invalid"
    }
    if match.clubAScore > match.clubBScore {
        return clubA
    } else if match.clubBScore > match.clubAScore {
        return clubB
    }
    return "Draw"
}

func main() {
    var clubA, clubB string
```

```

// Input klub names
fmt.Print("Klub A : ")
fmt.Scan(&clubA)
fmt.Print("Klub B : ")
fmt.Scan(&clubB)

matches := make([]Match, 9) // Array untuk 9
pertandingan
winners := make([]string, 9) // Array untuk
menyimpan pemenang

// Input skor pertandingan
for i := 0; i < 9; i++ {
    fmt.Printf("Pertandingan %d : ", i+1)
    var scoreA, scoreB int
    fmt.Scan(&scoreA, &scoreB)

    // Jika skor negatif, hentikan input
    if scoreA < 0 || scoreB < 0 {
        fmt.Println("Skor tidak valid
(negatif). Program berhenti.")
        break
    }

    matches[i] = Match{clubAScore: scoreA,
clubBScore: scoreB}
    winners[i] = determineWinner(clubA, clubB,
matches[i])

    // Tampilkan komentar untuk pertandingan
pertama
    if i == 0 {
        fmt.Printf("// %s = %d sedangkan %s =
%d\n",
            clubA, scoreA, clubB, scoreB)
    }
}

// Tampilkan hasil pertandingan
fmt.Println("\nHasil pertandingan:")
for i := 0; i < len(winners); i++ {
    if winners[i] == "" {
        continue
    }
    fmt.Printf("Hasil %d : %s\n", i+1,
winners[i])
}

fmt.Println("Pertandingan selesai")
}

```

### Screenshot Output

PROBLEMS 10

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2> go run -Refilina-Fiska\ALPRO_2\Modul7\unguided3.go"
```

Klub A : MU

Klub B : Inter

Pertandingan 1 : 2 0

// MU = 2 sedangkan Inter = 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

Skor tidak valid (negatif). Program berhenti.

Hasil pertandingan:

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

```
PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2>
```

### Deskripsi Program :

Program di atas adalah program Go untuk mencatat hasil pertandingan antara dua klub olahraga. Pengguna diminta memasukkan nama kedua klub serta skor dari setiap pertandingan. Program dapat mencatat hingga 9 pertandingan, namun akan berhenti jika ditemukan skor yang tidak valid (negatif). Setiap skor yang dimasukkan akan diproses untuk menentukan pemenang dengan aturan sederhana: jika skor Klub A lebih tinggi, maka Klub A menang; sebaliknya, jika skor Klub B lebih tinggi, Klub B menang. Jika skor sama, hasilnya adalah seri ("Draw"). Program juga memberikan komentar khusus pada skor pertandingan pertama dan mencetak hasil pertandingan satu per satu. Jika skor negatif dimasukkan, program langsung berhenti dan menampilkan pesan kesalahan. Setelah semua skor valid diproses, program akan mencetak hasil akhir dan menyelesaikan eksekusi. Contohnya, jika pengguna memasukkan skor 2-1 untuk pertandingan pertama, hasilnya akan menunjukkan Klub A sebagai pemenang. Program ini memprioritaskan validasi data dan kemudahan pengguna dalam mencatat hasil pertandingan.

## 1. Unguided 4

### Soal Studi Case

Sebuah array digunakan untuk menampung sekumpulan karakter, Anda diminta untuk membuat sebuah subprogram untuk melakukan membalikkan urutan isi array dan memeriksa apakah membentuk palindrom.

Lengkapi potongan algoritma berikut ini!

```
package main
import "fmt"
const NMAX int = 127
type tabel [NMAX]rune
    tab : tabel
    m : integer

func isiArray(t *tabel, n *int)
/*I.S. Data tersedia dalam piranti masukan
  F.S. Array t berisi sejumlah n karakter yang dimasukkan user,
  Proses input selama karakter bukanlah TITIK dan n <= NMAX */

func cetakArray(t tabel, n int)
/*I.S. Terdefinisi array t yang berisi sejumlah n karakter
  F.S. n karakter dalam array muncul di layar */

func balikanArray(t *tabel, n int)
/*I.S. Terdefinisi array t yang berisi sejumlah n karakter
  F.S. Urutan isi array t terbalik */

func main(){
    var tab tabel
    var m int
    // si array tab dengan memanggil prosedur isiArray

    // Balikian isi array tab dengan memanggil balikanArray

    // Cetak is array tab
}
```

Perhatikan sesi interaksi pada contoh berikut ini (**teks bergaris bawah adalah Input/read**)

Teks	: <b>S E N A N G .</b>
Reverse teks	: G N A N E S
Teks	: <b>K A T A K .</b>
Reverse teks	: K A T A K

Modifikasi program tersebut dengan menambahkan fungsi palindrom. Tambahkan instruksi untuk memanggil fungsi tersebut dan menampilkan hasilnya pada program utama.

**\*Palindrom adalah teks yang dibaca dari awal atau akhir adalah sama, contoh: KATAK, APA, KASUR\_RUSAK.**

func palindrom(t tabel, n int) bool	
/* Mengembalikan true apabila susunan karakter di dalam t membentuk palindrom, dan false apabila sebaliknya. Petunjuk: Manfaatkan prosedur balikanArray */	

Perhatikan sesi interaksi pada contoh berikut ini (**teks bergaris bawah adalah Input/read**)

Teks	: <b>K A T A K</b>
Palindrom	? true
Teks	: <b>S E N A N G</b>
Palindrom	? false

### Sourcecode

```
package main

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

const NMAX int = 127

type tabel [NMAX]rune

// Fungsi untuk mengisi array dengan karakter
func isiArray(t *tabel, n *int) {
    scanner := bufio.NewScanner(os.Stdin)
    fmt.Print("Teks : ")
    scanner.Scan()
    text := scanner.Text()

    // Menghilangkan spasi dan tanda baca
    text = strings.ReplaceAll(text, " ", "")
    text = strings.ReplaceAll(text, ".", "")
    text = strings.ReplaceAll(text, ",", "")

    *n = 0
    // Mengisi array dengan karakter dari input
```

```

        for _, char := range text {
            if *n < NMAX {
                t[*n] = char
                *n++
            }
        }
    }

    // Fungsi untuk mencetak isi array
    func cetakArray(t tabel, n int) {
        fmt.Print("Reverse teks : ")
        for i := 0; i < n; i++ {
            fmt.Printf("%c", t[i])
        }
        fmt.Println()
    }

    // Fungsi untuk membalikkan array
    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]
        }
    }

    // Fungsi untuk mengecek apakah array membentuk
    palindrom
    func palindrom(t tabel, n int) bool {
        // Membuat salinan array untuk dibalik
        var temp tabel
        copy(temp[:], t[:n])

        // Balik array salinan
        for i := 0; i < n/2; i++ {
            temp[i], temp[n-1-i] = temp[n-1-i],
temp[i]
        }

        // Bandingkan array asli dengan array yang
        sudah dibalik
        for i := 0; i < n; i++ {
            if t[i] != temp[i] {
                return false
            }
        }
        return true
    }

```



```

func main() {
    var tab tabel
    var m int

    // Input dan proses array
    isiArray(&tab, &m)

    // Tampilkan array yang dibalik
    var reversed tabel
    copy(reversed[:], tab[:m])
    balikanArray(&reversed, m)
    cetakArray(reversed, m)

    // Cek palindrom
    fmt.Print("Palindrom ? ")
    if palindrom(tab, m) {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }
}

```



```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2> go run
-Refilina-Fiska\ALPRO_2\Modul7\unguided4.go"
Teks : SENANG
Reverse teks : GNANES
Palindrom ? false
PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2> go run
-Refilina-Fiska\ALPRO_2\Modul7\unguided4.go"
Teks : KATAK
Reverse teks : KATAK
Palindrom ? true
PS C:\Users\lenovo\OneDrive\文档\ivena\2311102126_Agnes-Refilina-Fiska\ALPRO_2>

```

## Screenshot Output

### Deskripsi Program :

Program di atas adalah program Go yang dirancang untuk memproses teks dan menganalisis apakah teks tersebut merupakan palindrom (kata atau kalimat yang dibaca sama baik dari depan maupun belakang). Program menggunakan array bertipe `rune` dengan kapasitas maksimal 127 karakter

dan memiliki beberapa fungsi utama: ``isiArray`` untuk menerima input teks dan menyimpannya dalam array (sambil menghilangkan spasi dan tanda baca), ``cetakArray`` untuk menampilkan isi array, ``balikanArray`` untuk membalikkan urutan karakter dalam array, dan ``palindrom`` untuk mengecek apakah teks tersebut merupakan palindrom dengan cara membandingkan array asli dengan array yang sudah dibalik. Dalam fungsi ``main``, program menerima input teks dari pengguna, menampilkan teks tersebut dalam bentuk terbalik, dan memberikan output berupa ``true`` jika teks tersebut adalah palindrom atau ``false`` jika bukan.

#### **IV. KESIMPULAN**

Pemahaman tentang struct dan array adalah kunci dalam pemrograman untuk mengelola data secara lebih efektif. Struct memungkinkan pengelompokan berbagai tipe data berbeda menjadi satu kesatuan yang terorganisir, memudahkan representasi objek yang kompleks. Sebaliknya, array menyediakan mekanisme untuk menyimpan kumpulan data bertipe sama dalam urutan tertentu, memungkinkan akses dan manipulasi data yang cepat. Keduanya saling melengkapi; struct memberikan kemampuan untuk membangun tipe data yang lebih kompleks, sementara array mendukung pengelolaan koleksi data secara terstruktur. Penguasaan kedua konsep ini membantu programmer merancang program yang lebih efisien, terorganisir, dan mampu menyelesaikan masalah dengan lebih baik.

#### **V. REFERENSI**

- [1] Modul 7 Praktikum Algoritma 2
- [2] A. A. A. Donovan and B. W. Kernighan, *The Go Programming Language*. Boston, MA: Addison-Wesley, 2015.

