

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII  
STRUCK DAN ARRAY**



**Disusun Oleh :**

Maulisa Elvita Sari / 2311102259

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**

### **Pengertian Array**

Array adalah struktur data yang menyimpan sekumpulan elemen dengan tipe data yang sama dalam urutan tetap. Ukuran array ditentukan saat deklarasi dan tidak dapat diubah setelahnya. Contohnya, untuk mendeklarasikan array integer dengan lima elemen, maka ada beberapa sintaks yang dapat digunakan, diantaranya:

- Deklarasi dan Inisialisasi: Array dapat dideklarasikan dan diinisialisasi secara bersamaan.
- Akses dan modifikasi Elemen: Elemen dalam array diakses menggunakan indeks dimulai dari 0.
- Iterasi melalui array: iterasi dapat dilakukan menggunakan loop 'for' atau 'for-range'.
- Array multidimensi
- Kelebihan dan Keterbatasan Array: kelebihan utama dari array adalah efisiensi dalam penyimpanan dan akses data Ketika ukuran data sudah diketahui. Namun, keterbatasan utama adalah ukuran yang tetap, sehingga jika diperlukan struktur data yang lebih fleksibel, slice dapat menjadi alternatif yang lebih baik.

### **Pengertian Struct**

Struct adalah Kumpulan variabel (atau property) dengan tipe data yang berbeda-beda yang dibungkus dalam satu tipe data baru. Struct memungkinkan pengelompokan data yang relevan menjadi satu kesatuan logis. Dalam Go, struct digunakan untuk membuat tipe data kompleks yang mirip dengan kelas dalam pemrograman berorientasi objek.

## II. GUIDED

### Soal Studi Case

#### Sourcecode

```
package main

import "fmt"

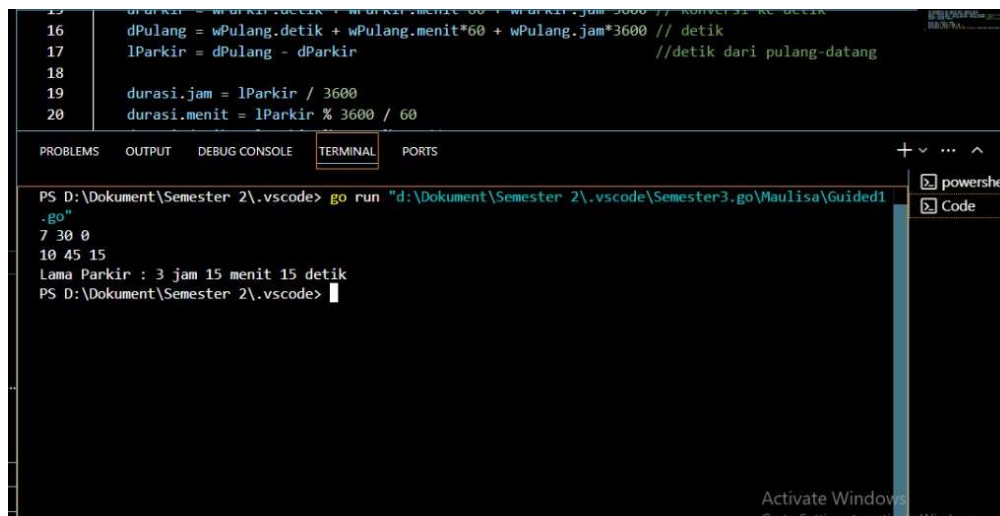
type waktu struct {
    jam, menit, detik int
}

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



```
16 dPulang = wPulang.detik + wPulang.menit*60 + wPulang.jam*3600 // detik
17 lParkir = dPulang - dParkir                                //detik dari pulang-datang
18
19 durasi.jam = lParkir / 3600
20 durasi.menit = lParkir % 3600 / 60

PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Maulisa\Guided1
.go"
7 30 0
10 45 15
Lama Parkir : 3 jam 15 menit 15 detik
PS D:\Dokument\Semester 2\.vscode>
```

## **Deskripsi Program**

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

Program tersebut merupakan program dalam bahasa GO dimana program di atas meminta pengguna untuk menghitung durasi parkir berdasarkan waktu kedatangan dan waktu pulang. Program menerima input berupa waktu kedatangan dan waktu pulang dalam format jam, menit, dan detik, kemudian menghitung selisih waktu tersebut untuk menampilkan lama parkir dalam format jam, menit, dan detik.

## Sourcecode guided 2/slice

```
//Guided 2 - Slice
package main

import (
    "fmt"
)

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

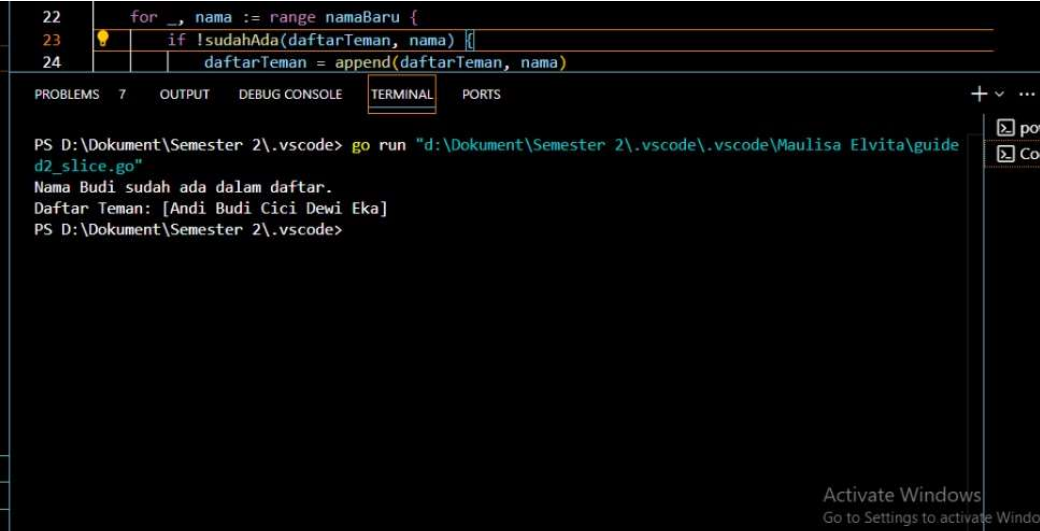
func main() {
    daftarTeman := []string{"Andi", "Budi", "Cici"}

    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)
}
```

## Screenshot Output



The screenshot shows a VS Code editor window with a Go program. The code is as follows:

```
22     for _, nama := range namaBaru {
23         if !sudahAda(daftarTeman, nama) {
24             daftarTeman = append(daftarTeman, nama)

```

The terminal output shows the execution of the program:

```
PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\vscode\Maulisa Elvita\guide
d2_slice.go"
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\Dokument\Semester 2\vscode>
```

The terminal output matches the expected behavior of the code: it checks for duplicates and only adds new names to the list.

## **Deskripsi Program**

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

Program tersebut merupakan program dalam bahasa GO dimana program di atas meminta pengguna untuk menambahkan nama-nama baru ke dalam daftar teman, dengan pengecekan untuk menghindari duplikasi nama. Jika nama yang ingin ditambahkan sudah ada dalam daftar, program akan memberikan pemberitahuan. Daftar teman ahir akan ditampilkan setelah proses selesai. Pertama fungsi 'sudahAda' akan memeriksa apakah sebuah nama sudah ada dalam daftar teman 'daftarTeman', jika ditemukan nama yang sama, maka fungsi akan mengembalikan 'true'. Dan jika tidak ditemukan nama yang sama, maka fungsi akan mengembalikan 'false'. Selanjutnya, daftar teman awal diinisialisasi sebagai slice dengan nilai, dan nama-nama baru akan ditambahkan lalu disimpan dalam slice 'namaBaru'. Lalu, program akan melakukan iterasi melalui semua elemen dalam 'namaBaru', jika nama sudah ada maka program akan mencetak ke layar bahwa nama tersebut sudah ada dalam daftar.

## Sourcecode guided 2/Map

```
//Guided 2 - Map

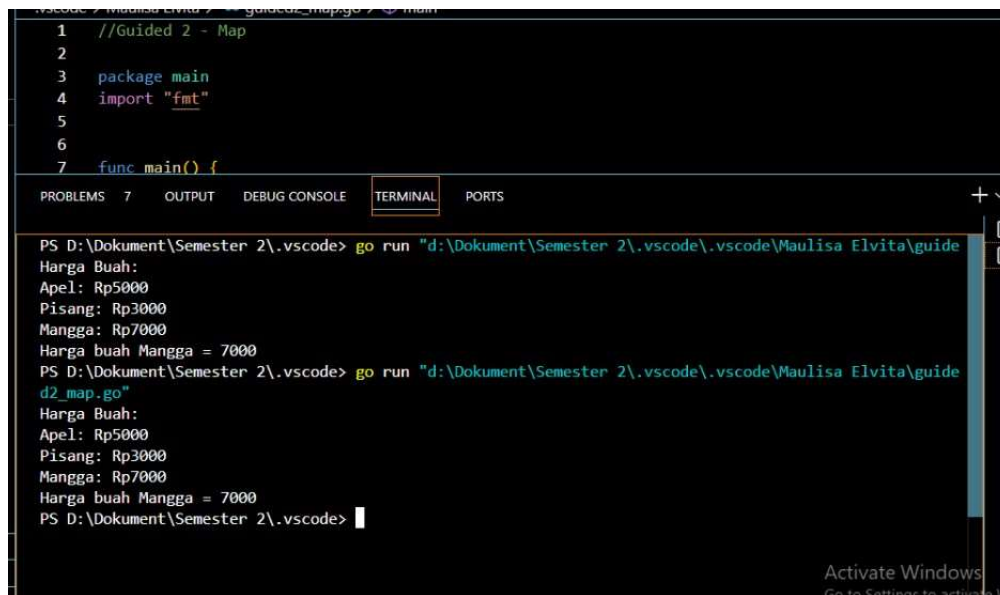
import (
    "fmt"
)

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"])
}
```

## Screenshoot Output



```
1 //Guided 2 - Map
2
3 package main
4 import "fmt"
5
6
7 func main() {
8     hargaBuah := map[string]int{
9         "Apel": 5000,
10        "Pisang": 3000,
11        "Mangga": 7000,
12    }
13
14    fmt.Println("Harga Buah:")
15    for buah, harga := range hargaBuah {
16        fmt.Printf("%s: Rp%d\n", buah, harga)
17    }
18
19    fmt.Print("Harga buah Mangga = ", hargaBuah["Mangga"])
20 }
```

PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\vscode\Maulisa Elvita\guide  
Harga Buah:  
Apel: Rp5000  
Pisang: Rp3000  
Mangga: Rp7000  
Harga buah Mangga = 7000  
PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\vscode\Maulisa Elvita\guide  
d2\_map.go"  
Harga Buah:  
Apel: Rp5000  
Pisang: Rp3000  
Mangga: Rp7000  
Harga buah Mangga = 7000  
PS D:\Dokument\Semester 2\vscode>

## Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

Program tersebut merupakan program dalam bahasa GO dimana program di atas meminta pengguna untuk menyimpan dan menampilkan harga buah menggunakan struktur data map.

Map adalah koleksi data yang berisi pasangan key-value, dimana key adalah nama buah, dan value adalah harga buah. Program akan mencetak daftar harga semua buah dalam map dan menampilkan harga yang spesifik untuk buah tertentu (Mangga).



## UNGUIDED

### 1. Soal Studi Case

Suatu lingkaran didefinisikan dengan koordinat titik pusat ( $c_x$ ,  $c_y$ ) 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 sernua 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".

#### Contoh

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"
)

type Titik struct {
    x, y float64
}

type Lingkaran struct {
    pusat Titik
    radius float64
}

func jarak(p1, p2 Titik) float64 {
    return math.Sqrt(math.Pow(p1.x-p2.x, 2) + math.Pow(p1.y-p2.y, 2))
}

func diDalamLingkaran(l Lingkaran, t Titik) bool {
    return jarak(l.pusat, t) <= l.radius
}

func main() {
    var cx1, cy1, r1 float64
    var cx2, cy2, r2 float64

    fmt.Println("Masukkan data lingkaran 1 (pusat x, pusat y, radius):")
    fmt.Scan(&cx1, &cy1, &r1)

    fmt.Println("Masukkan data lingkaran 2 (pusat x, pusat y, radius):")
    fmt.Scan(&cx2, &cy2, &r2)

    lingkaran1 := Lingkaran{pusat: Titik{x: cx1, y: cy1}, radius: r1}
    lingkaran2 := Lingkaran{pusat: Titik{x: cx2, y: cy2}, radius: r2}

    var tx, ty float64
    fmt.Println("Masukkan koordinat titik (x, y):")
    fmt.Scan(&tx, &ty)
    titik := Titik{x: tx, y: ty}

    diLingkaran1 := diDalamLingkaran(lingkaran1, titik)
    diLingkaran2 := diDalamLingkaran(lingkaran2, titik)

    if diLingkaran1 && diLingkaran2 {
```

```

        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if diLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

## Screenshoot Output

```

PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\Maulisa Elvita\ungul
1 1 5
Masukkan data lingkaran 2 (pusat x, pusat y, radius):
8 8 4
Masukkan koordinat titik (x, y):
2 2
Titik di dalam lingkaran 1
PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\Maulisa Elvita\ungul
1 2 3
Masukkan data lingkaran 2 (pusat x, pusat y, radius):
4 5 6
Masukkan koordinat titik (x, y):
7 8
Titik di dalam lingkaran 2
PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\Maulisa Elvita\ungul
5 10 15
Masukkan data lingkaran 2 (pusat x, pusat y, radius):
-15 4 20
Masukkan koordinat titik (x, y):
0 0
Titik di dalam lingkaran 1 dan 2
PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\Maulisa Elvita\ungul
1 1 5
Masukkan data lingkaran 2 (pusat x, pusat y, radius):
8 8 4
Masukkan koordinat titik (x, y):
15 20
Titik di luar lingkaran 1 dan 2
PS D:\Dokument\Semester 2\vscode>

```

## Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

Program di atas merupakan program menggunakan bahasa Go yang dimana digunakan untuk mengevaluasi apakah suatu titik berada di dalam, di luar, atau tepat di perbatasan dua lingkaran yang diberikan oleh pengguna. Program ini menggunakan pendekatan matematis dengan memanfaatkan konsep jarak Eulidean antara dua titik dan sifat-sifat lingkaran. Pertama program meminta input untuk data dua lingkaran untuk lingkaran pertama dan lingkaran kedua, selanjutnya data akan disimpan dalam dua objek lingkaran. Program meminta input koordinat titik yang kemudian disimpan dalam objek titik. Lalu, fungsi 'diDalamLingkaran' dijalankan untuk mengevaluasi apakah titik berada di dalam lingkaran 1, lingkaran 2, atau keduanya. Maka, hasil dari evaluasi akan dicetak sebagai output berupa posisi relatif titik terhadap kedua lingkaran.

## 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:

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

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n int
    fmt.Print("Masukkan jumlah elemen array (N): ")
    fmt.Scan(&n)

    array := make([]int, n)
    fmt.Println("Masukkan elemen array:")
    for i := 0; i < n; i++ {
        fmt.Printf("Elemen ke-%d: ", i)
        fmt.Scan(&array[i])
    }
}
```

```

        fmt.Println("\na. Isi array:")
        fmt.Println(array)

        fmt.Println("\nb. Elemen dengan indeks ganjil:")
        for i := 1; i < len(array); i += 2 {
            fmt.Printf("array[%d] = %d\n", i, array[i])
        }

        fmt.Println("\nc. Elemen dengan indeks genap:")
        for i := 0; i < len(array); i += 2 {
            fmt.Printf("array[%d] = %d\n", i, array[i])
        }

        var x int
        fmt.Print("\nd. Masukkan nilai x untuk kelipatan indeks:
")

        fmt.Scan(&x)
        fmt.Println("Elemen dengan indeks kelipatan", x, ":")
        for i := 0; i < len(array); i++ {
            if i%x == 0 {
                fmt.Printf("array[%d] = %d\n", i, array[i])
            }
        }

        var idx int
        fmt.Print("\ne. Masukkan indeks elemen yang akan dihapus:
")

        fmt.Scan(&idx)
        if idx >= 0 && idx < len(array) {
            array = append(array[:idx], array[idx+1:]...)
            fmt.Println("Isi array setelah elemen dihapus:")
            fmt.Println(array)
        } else {
            fmt.Println("Indeks tidak valid.")
        }

        sum := 0
        for _, v := range array {
            sum += v
        }
        rataRata := float64(sum) / float64(len(array))
        fmt.Printf("\nf. Rata-rata elemen array: %.2f\n",
rataRata)

        var deviasiSum float64
        for _, v := range array {
            deviasiSum += math.Pow(float64(v)-rataRata, 2)
        }
        standarDeviasi := math.Sqrt(deviasiSum /
float64(len(array)))
        fmt.Printf("g. Standar deviasi elemen array: %.2f\n",
standarDeviasi)

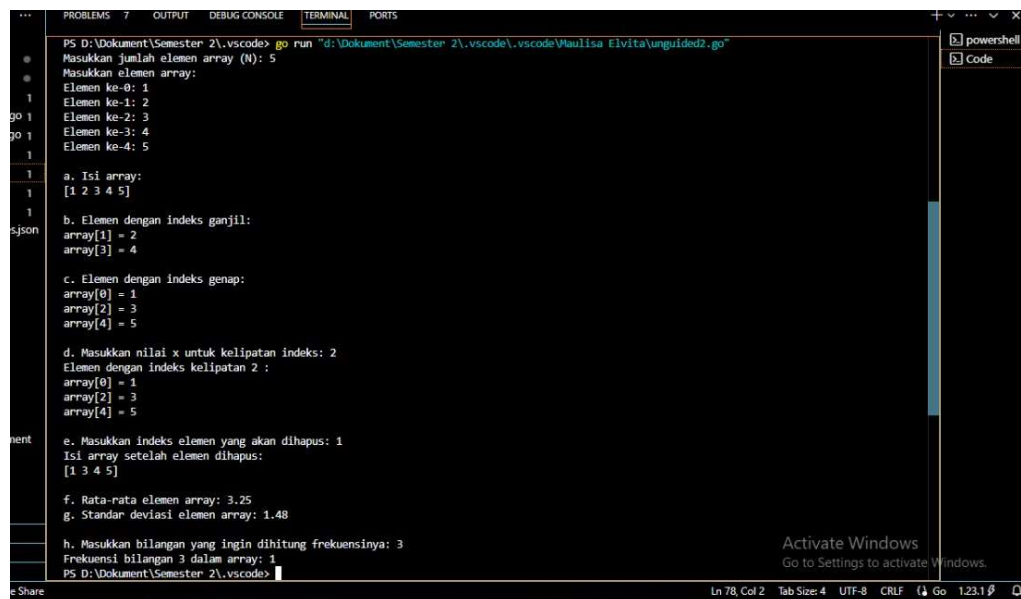
```

```

        var target int
        fmt.Print("\nh. Masukkan bilangan yang ingin dihitung
frekuensinya: ")
        fmt.Scan(&target)
        frekuensi := 0
        for _, v := range array {
            if v == target {
                frekuensi++
            }
        }
        fmt.Printf("Frekuensi bilangan %d dalam array: %d\n",
target, frekuensi)
    }

```

## Screenshoot Output



```

PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\Haulisa Elvita\unguided2.go"
Masukkan jumlah elemen array (N): 5
Masukkan elemen array:
Elemen ke-0: 1
Elemen ke-1: 2
Elemen ke-2: 3
Elemen ke-3: 4
Elemen ke-4: 5

a. Isi array:
[1 2 3 4 5]

b. Elemen dengan indeks ganjil:
array[1] = 2
array[3] = 4

c. Elemen dengan indeks genap:
array[0] = 1
array[2] = 3
array[4] = 5

d. Masukkan nilai x untuk kelipatan indeks: 2
Elemen dengan indeks kelipatan 2 :
array[0] = 1
array[2] = 3
array[4] = 5

e. Masukkan indeks elemen yang akan dihapus: 1
Isi array setelah elemen dihapus:
[1 3 4 5]

f. Rata-rata elemen array: 3.25
g. Standar deviasi elemen array: 1.48

h. Masukkan bilangan yang ingin dihitung frekuensinya: 3
Frekuensi bilangan 3 dalam array: 1
PS D:\Dokument\Semester 2\vscode>

```

## Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

Program di atas merupakan program menggunakan bahasa Go yang dimana digunakan untuk manipulasi array dan analisis data sederhana. Pengguna dapat melakukan berbagai operasi seperti mencetak elemen array, menghapus elemen, menghitung rata-rata, standar deviasi, dan frekuensi suatu nilai dalam array.

### 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.

```
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"
)

func main() {
    var klubA, klubB string
    var skorA, skorB int
    var pemenang []string

    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&klubA)
    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scanln(&klubB)

    fmt.Println("\nMasukkan skor pertandingan (masukkan skor
negatif untuk menghentikan):")
    for i := 1; ; i++ {
        fmt.Printf("Pertandingan %d - Skor %s: ", i, klubA)
        fmt.Scan(&skorA)
        fmt.Printf("Pertandingan %d - Skor %s: ", i, klubB)
        fmt.Scan(&skorB)

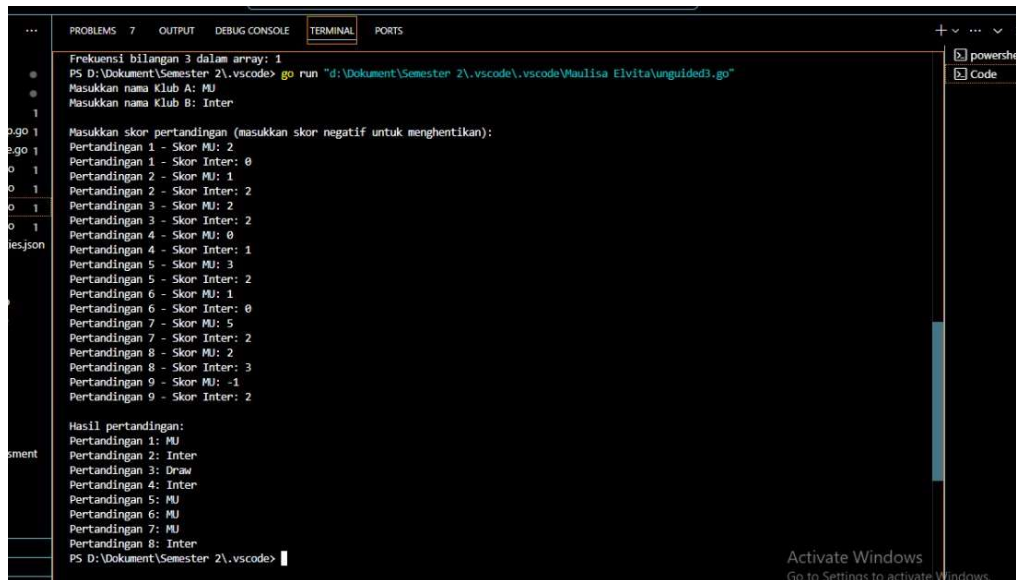
        if skorA < 0 || skorB < 0 {
            break
        }

        if skorA > skorB {
            pemenang = append(pemenang, klubA)
        } else if skorB > skorA {
            pemenang = append(pemenang, klubB)
        } else {
            pemenang = append(pemenang, "Draw")
        }
    }

    fmt.Println("\nHasil pertandingan:")
    for i, hasil := range pemenang {
        fmt.Printf("Pertandingan %d: %s\n", i+1, hasil)
    }
}
```

## Screenshot Output





```
Frekuensi bilangan 3 dalam array: 1
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Maulisa Elvita\unguided3.go"
Masukkan nama Klub A: MU
Masukkan nama Klub B: Inter

Masukkan skor pertandingan (masukkan skor negatif untuk menghentikan):
Pertandingan 1 - Skor MU: 2
Pertandingan 1 - Skor Inter: 0
Pertandingan 2 - Skor MU: 1
Pertandingan 2 - Skor Inter: 2
Pertandingan 3 - Skor MU: 2
Pertandingan 3 - Skor Inter: 2
Pertandingan 4 - Skor MU: 0
Pertandingan 4 - Skor Inter: 1
Pertandingan 5 - Skor MU: 3
Pertandingan 5 - Skor Inter: 2
Pertandingan 6 - Skor MU: 1
Pertandingan 6 - Skor Inter: 0
Pertandingan 7 - Skor MU: 5
Pertandingan 7 - Skor Inter: 2
Pertandingan 8 - Skor MU: 2
Pertandingan 8 - Skor Inter: 3
Pertandingan 9 - Skor MU: -1
Pertandingan 9 - Skor Inter: 2

Hasil pertandingan:
Pertandingan 1: MU
Pertandingan 2: Inter
Pertandingan 3: Draw
Pertandingan 4: Inter
Pertandingan 5: MU
Pertandingan 6: MU
Pertandingan 7: MU
Pertandingan 8: Inter
PS D:\Dokument\Semester 2\.vscode>
```

## Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

Program di atas merupakan program menggunakan bahasa Go yang dimana digunakan untuk mencatat hasil pertandingan antara dua klub sepak bola. Program ini menerima input nama kedua klub dan skor dari beberapa pertandingan. Berdasarkan skor yang dimasukkan, program menentukan pemenang untuk setiap pertandingan atau mencatat hasil imbang (draw). Proses berlanjut hingga pengguna memasukkan skor negatif untuk menghentikan.

#### 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.

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

Teks	: <u>S E N A N G .</u>
Reverse teks	: G N A N E S
Teks	: <u>K A I A K .</u>
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.**

<pre>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 */</pre>
---

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

Teks	: <u>K A I A K</u>
Palindrom	? true
Teks	: <u>S E N A N G</u>
Palindrom	? false

#### Sourcecode

```
package main

import (
    "fmt"
)

const NMAX = 127

type Tabel [NMAX]rune

func isArray(t *Tabel, n *int) {
    var input string
    fmt.Print("Masukkan teks (akhiri dengan TITIK '.'): ")
    fmt.Scanln(&input)

    *n = 0
    for i := 0; i < len(input) && input[i] != '.'; i++ {
        (*t)[*n] = rune(input[i])
```

```

        *n++
    }
}

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, j := 0, n-1; i < j; i, j = i+1, j-1 {
        t[i], t[j] = t[j], t[i]
    }
}

func palindrom(t Tabel, n int) bool {
    for i, j := 0, n-1; i < j; i, j = i+1, j-1 {
        if t[i] != t[j] {
            return false
        }
    }
    return true
}

func main() {
    var tab Tabel
    var n int

    isArray(&tab, &n)

    fmt.Print("Teks: ")
    cetakArray(tab, n)

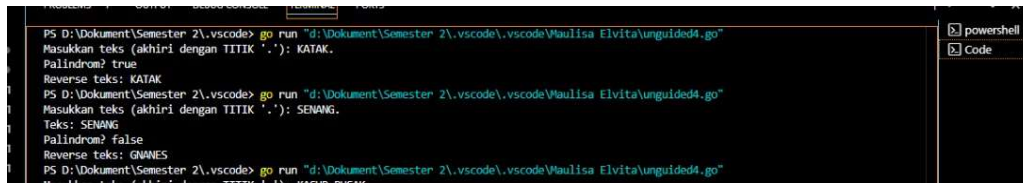
    if palindrom(tab, n) {
        fmt.Println("Palindrom? true")
    } else {
        fmt.Println("Palindrom? false")
    }

    balikanArray(&tab, n)

    fmt.Print("Reverse teks: ")
    cetakArray(tab, n)
}

```

## Screenshoot Output



```
PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\vscode\Maulisa Elvita\unguided4.go"
Masukkan teks (akhiri dengan TITIK '.'): KATAK.
Palindrom? true
Reverse teks: KATAK
PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\vscode\Maulisa Elvita\unguided4.go"
Masukkan teks (akhiri dengan TITIK '.'): SENANG.
Teks: SENANG
Palindrom? false
Reverse teks: GNANES
PS D:\Dokument\Semester 2\vscode> go run "d:\Dokument\Semester 2\vscode\vscode\Maulisa Elvita\unguided4.go"
```

## Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

Program di atas merupakan program menggunakan bahasa Go yang dimana digunakan untuk memproses string (teks) dengan beberapa fitur, yaitu mencetak teks asli, memeriksa apakah teks tersebut merupakan polindrom, dan mencetak teks dalam bentuk terbalik. Program ini menggunakan array bertipe 'rune' untuk menyimpan teks dan mengelola operasi terhadap teks tersebut.

## **DAFTAR PUSTAKA**

<https://gepcode.com/post/golang-tipe-data-array>

<https://www.petanikode.com/c-struct/>