

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII  
STRUCT & ARRAY**



**Disusun Oleh :**

**Muhammad Ihab Aufa Rafi / 2311102226**

**S1IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi, S.Kom., M.Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

### 1.1 Tipe Bentuk

Tipe bentuk memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipe bentuk ini dapat dibedakan atas dua jenis, yaitu Alias dan Struct.

#### 1.1.1. Alias (Type)

Bahasa pemrograman pada umumnya mengizinkan pemrograman untuk mengubah nama suatu tipe data dengan nama baru yang lebih ringkas dan familiar. Sebagai contoh "integer" dapat dirubah dengan nama alias "bilangan". Caranya dengan menggunakan kata kunci "type".

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama alias> <tipe data>	
3		type <nama alias> <tipe data>
4	algoritma	
5	...	func main(){
6		...
7		}

Sebagai contoh, perhatikan program Go berikut beserta hasil eksekusinya!

```
1 package main
2 import "fmt"
3 type bilangan int
4 type pecahan float64
5 func main(){
6     var a, b bilangan
7     var hasil pecahan
8     a = 9
9     b = 5
10    hasil = pecahan(a)/pecahan(b)
11    fmt.Println(hasil)
12 }
```

#### 1.1.2. Struct atau Record

Stucture memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari stucture tersebut.

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama struct> <	type <nama struct> struct {
3	<field 1> <tipe data>	<field 1> <tipe data>
4	<field 2> <tipe data>	<field 2> <tipe data>
5	<field 3> <tipe data>	<field 3> <tipe data>
6	>	}
7		

Berbeda dengan bahasa pemrograman lain. kesamaan tipe dari dua variabel berjenis stucture bukan karena namanya tetapi karena strukturnya. Dua variabel dengan nama-nama field dan tipe field yang sama (dan dalam urutan yang sama) dianggap mempunyai tipe yang sama. Tentunya akan lebih memudahkan jika stucture tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi stucture tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

```

1 package main
2 import "fmt"
3 type waktu struct {
4     jam, menit, detik int
5 }
6
7 func main(){
8     var wParkir, wPulang, durasi waktu
9     var dParkir, dPulang, lParkir int
10    fmt.Scan(&wParkir.jam, &wParkir.menit, &wParkir.detik)
11    fmt.Scan(&wPulang.jam, &wPulang.menit, &wPulang.detik)
12    dParkir = wParkir.detik + wParkir.menit*60 + wParkir.jam*3600
13    dPulang = wPulang.detik + wPulang.menit*60 + wPulang.jam*3600
14    lParkir = dPulang - dParkir
15    durasi.jam = lParkir / 3600
16    durasi.menit = lParkir % 3600 / 60
17    durasi.detik = lParkir % 3600 % 60
18    fmt.Printf("Lama parkir: %d jam %d menit %d detik",
19        durasi.jam, durasi.menit, durasi.detik)
20 }

```

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\dedededecade\dcd.go"
7 30 0
10 45 15
Lama parkir: 3 jam 15 menit 15 detik
PS D:\Alpro2Golang>

```

## 1.2 Array

Array mempunyai ukuran (jumlah elemen) yang tetap (statis) selama eksekusi program, sehingga jumlah elemen array menjadi bagian dari deklarasi variabel dengan tipe array.

	Notasi dalam bahasa Go
1	var (
2	// array arr mempunyai 73 elemen, masing-masing bertipe CircType2
3	arr [73]CircType
4	
5	// array buf dengan 5 elemen, dengan nilai awal 7, 3, 5, 2, dan 11.
6	buf = [5]byte{7, 3, 5, 2, 11}
7	
8	// mhs adalah array dengan 2000 elemen bertipe NewType
9	mhs [2000]NewType
10	
11	// rec adalah array dari array, yaitu matriks, atau array berdimensi-2
12	rec [20][40]float64
13	)

Jumlah elemen array dapat diminta dengan fungsi **len** yang tersedia. Sebagai contoh `len(arr)` akan menghasilkan 73 untuk contoh di atas.

Indeks array dimulai dari 0, sehingga indeks arr pada contoh adalah **0, 1.. len(arr)-1**.

Contoh:

```
1 // Mengganti isi elemen ke-0 dengan nilai dari elemen ke-7
2 arr[0] = arr[7]
3
4 // Mengambil data field x dari elemen ke-i
5 currX = arr[i].center.x
6
7 // Mengambil elemen terakhir
8 n := len(arr)
9 buf := arr[n-1]
```

### 1.2.1 Slice (Array Dinamik)

Array dalam Go juga dapat mempunyai ukuran yang dinamik. Deklarasinya mirip dengan deklarasi array, tetapi jumlah elemennya dikosongkan.

```
1 // declaring chop as an empty slice of float64
2 var chop []float64
3
4 // declaring sl01 as a slice
5 var sl01 = []int{ 11, 2, 3, 5, 7, 13 }
```

Sebuah slice dapat diprealokasi menggunakan fungsi built-in **make**.

```

1 // Prealokasi 10 elemen untuk sl02 dan sejumlah tempat tambahan
2 var sl02 []int = make([]int, 10, 20)
3
4 // Prealokasi 7 elemen untuk sl03 tanpa tempat tambahan
5 var sl03 []circType = make([]circType, 7)

```

Fungsi built-in **len** dapat digunakan untuk mengetahui ukuran slice. Fungsi lain, **cap**, dapat digunakan untuk mengetahui total tempat yang disediakan untuk slice tersebut.

```

1 // Cetak jumlah elemen dan tempat yang tersedia untuk sl02
2 fmt.Println( len(sl02), cap(sl02) )

```

Fungsi built-in **append** dapat digunakan untuk menambahkan elemen ke suatu slice, dan bila perlu mempersebar tempat untuk slice tersebut.

```

1 /* Append elemen baru, membuat slice baru, dan menyimpan kembali slice baru
   ke variabel semula. Boleh juga disimpan ke variabel lain, sehingga variabel
   semula masih menyimpan slice yang asli. */
2 sl01 = append(sl01, 17)
3 sl01 = append(sl01, 19, 23)

```

Sebuah slice baru juga dapat terbentuk dengan mengambil slice dari suatu array atau slice yang lain.

## 1.2.2 Map

Tipe array lain, sebuah array dinamik. Indeksnya (di sini disebut **kunci**) tidak harus berbentuk integer. Indeks dapat berasal dari tipe apa saja. Struktur ini disebut **map**.

```

1 // Deklarasi variabel dct sebagai map bilangan bulat dengan kunci string
2 var dct map[string]int
3
4 // Deklarasi map lain dct1 dari elemen string dengan kunci juga string
5 // Mempunyai nilai awal dct1["john"] = "hi", dct1["anne"] = "darling"
6 var dct1 = map[string]string{ "john":"hi", "anne":"darling" }
7
8 // Deklarasi dan prealokasi tempat untuk map dct2
9 var dct2 map[float64]int = make(map[float64]int, 10)
10
11 // Mengambil nilai yang tersimpan dengan kunci "john"
12 fmt.Println( dct1["john"] )
13
14 // Mengganti nilai yang tersimpan pada kunci "anne", dan
15 // Membuat entri baru dengan kunci "boy"
16 dct1["anne"] = "lovely"
17 dct1["boy"] = "runaround"
18
19 // Menghapus entri dengan kunci "john"
20 delete(dct1, "john")

```

## II. GUIDED

### 1. Soal Studi Case

Membuat tipe data baru menggunakan TYPE.

#### Sourcecode

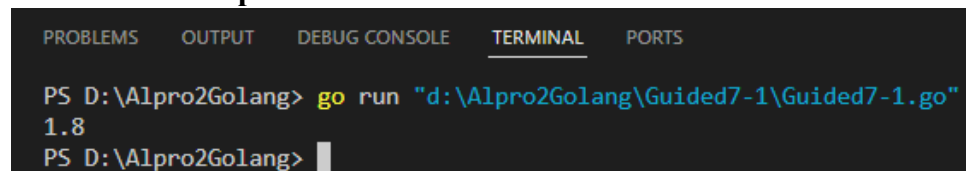
```
package main

import "fmt"

type bilangan int
type pecahan float64

func main(){
    var a, b bilangan
    var hasil pecahan
    a = 9
    b = 5
    hasil = pecahan(a)/pecahan(b)
    fmt.Println(hasil)
}
```

#### Screenshoot Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Guided7-1\Guided7-1.go"
1.8
PS D:\Alpro2Golang> 
```

#### Deskripsi Program

Program di atas merupakan contoh sederhana dalam bahasa Go yang mendemonstrasikan penggunaan tipe data khusus (alias) untuk operasi pembagian antara dua bilangan. Dalam program ini, dua tipe data baru didefinisikan menggunakan kata kunci type.

### 2. Soal Studi Case

Menghitung durasi parkir menggunakan **type struct**.

#### Sourcecode

```
package main

import "fmt"

type waktu struct {
    jam, menit, detik int
}
```

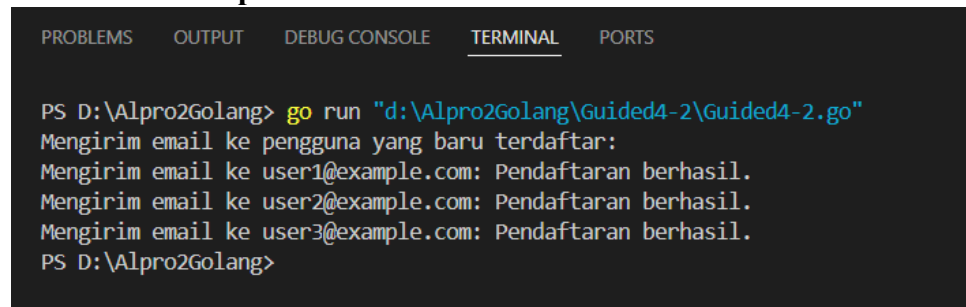
```

}

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
    dPulang = wPulang.detik + wPulang.menit*60 +
wPulang.jam*3600
    lParkir = dPulang - dParkir
    durasi.jam = lParkir / 3600
    durasi.menit = lParkir % 3600 / 60
    durasi.detik = lParkir % 3600 % 60
    fmt.Printf("Lama parkir: %d jam %d menit %d detik",
durasi.jam, durasi.menit, durasi.detik)
}

```

## Screenshoot Output



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Guided4-2\Guided4-2.go"
Mengirim email ke pengguna yang baru terdaftar:
Mengirim email ke user1@example.com: Pendaftaran berhasil.
Mengirim email ke user2@example.com: Pendaftaran berhasil.
Mengirim email ke user3@example.com: Pendaftaran berhasil.
PS D:\Alpro2Golang>

```

## Deskripsi Program

Program di atas merupakan aplikasi sederhana dalam bahasa Go untuk menghitung durasi parkir berdasarkan waktu kedatangan dan waktu pulang. Program ini menggunakan struktur data struct untuk merepresentasikan waktu dalam bentuk jam, menit, dan detik.

### III. 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 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".

#### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

//struktur koordinat
type Titik struct {
    x int
    y int
}

//struktur lingkaran
type Lingkaran226 struct {
    pusat Titik
    radius int
}

//fungsi euclidean
func hitungJarak(a,b Titik) float64 {
    return math.Sqrt(float64((a.x-b.x)*(a.x-b.x) + (a.y-
b.y)*(a.y-b.y)))
}

func titikDalamLingkaran(t Titik, l Lingkaran226) bool {
```



```

        jarak := hitungJarak(t, l.pusat)
        return jarak <= float64(l.radius)
    }

func main() {
    //input untuk lingkaran 1
    var cx1, cy1, r1 int
    fmt.Print("Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): ")
    fmt.Scanln(&cx1, &cy1, &r1)
    lingkaran1 := Lingkaran226{pusat: Titik{x: cx1, y:cy1}, radius: r1}

    //input untuk lingkaran 2
    var cx2, cy2, r2 int
    fmt.Print("Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): ")
    fmt.Scanln(&cx2, &cy2, &r2)
    lingkaran2 := Lingkaran226{pusat: Titik{x: cx2, y:cy2}, radius: r2}

    //input titik sembarang
    var x, y int
    fmt.Print("Masukkan koordinat titik sembarang (x y): ")
    fmt.Scanln(&x, &y)
    titik := Titik{x: x, y: y}

    //pengecekan posisi titik
    diDalamL1 := titikDalamLingkaran(titik, lingkaran1)
    diDalamL2 := titikDalamLingkaran(titik, lingkaran2)

    //menampilkan hasil sesuai kondisi
    if diDalamL1 && diDalamL2{
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    }else if diDalamL1{
        fmt.Println("Titik di dalam lingkaran 1")
    }else if diDalamL2{
        fmt.Println("Titik di dalam lingkaran 2")
    }else {
        fmt.Println("Titik berada di luar kedua lingkaran")
    }
}

```

## Screenshoot Output

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided7-1\Unguided7-1.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): 1 1 5
Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): 4 1 1
Masukkan koordinat titik sembarang (x y): 0 0
Titik di dalam lingkaran 1
PS D:\Alpro2Golang>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided7-1\Unguided7-1.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): 1 1 2
Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): 2 2 2
Masukkan koordinat titik sembarang (x y): 3 3
Titik di dalam lingkaran 2
PS D:\Alpro2Golang>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided7-1\Unguided7-1.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): 3 3 3
Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): 2 2 3
Masukkan koordinat titik sembarang (x y): 3 3
Titik di dalam lingkaran 1 dan 2
PS D:\Alpro2Golang>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided7-1\Unguided7-1.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): 2 3 4
Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): 5 3 2
Masukkan koordinat titik sembarang (x y): 5 2
Titik berada di luar kedua lingkaran
PS D:\Alpro2Golang>
```

## Deskripsi Program

Program ini merupakan implementasi untuk mengecek apakah sebuah titik berada di dalam satu atau dua lingkaran pada bidang dua dimensi. Program menggunakan dua struct: **Titik** untuk menyimpan koordinat (x, y) dan **Lingkaran226** untuk menyimpan pusat lingkaran (**Titik**) dan radius (**int**). Fungsi **hitungJarak** menghitung jarak antara dua titik menggunakan rumus Euclidean, sedangkan **titikDalamLingkaran** mengecek apakah jarak titik ke pusat lingkaran kurang atau sama dengan radius.

Di dalam fungsi main, pengguna memasukkan koordinat dan radius dua lingkaran serta koordinat titik sembarang. Program kemudian memeriksa apakah titik tersebut berada di dalam lingkaran 1, lingkaran 2, atau keduanya. Hasil pengecekan ditampilkan sesuai kondisi yang

terpenuhi. Misalnya, jika titik berada di dalam kedua lingkaran, program akan menampilkan "Titik di dalam lingkaran 1 dan 2".

## 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 indeks ke-0 adalah genap).
- Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x. nilai x bisa diperoleh dari masukan pengguna.
- Menghapus elemen array pada indeks tertentu, asumsi indeks yang dihapus 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 tampilArray(arr []int, n int) {
    fmt.Print("Isi array: ")
    for i := 0; i < n; i++ {
        fmt.Printf("%d ", arr[i])
    }
    fmt.Println()
}
```

```

func tampilIndeksGanjil(arr []int, n int) {
    fmt.Print("Elemen dengan indeks ganjil: ")
    for i := 1; i < n; i += 2 {
        fmt.Printf("%d ", arr[i])
    }
    fmt.Println()
}

func tampilIndeksGenap(arr []int, n int) {
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 0; i < n; i += 2 {
        fmt.Printf("%d ", arr[i])
    }
    fmt.Println()
}

func tampilKelipatanX(arr []int, n int, x int) {
    fmt.Printf("Elemen dengan indeks kelipatan %d: ", x)
    for i := 0; i < n; i++ {
        if i%x == 0 {
            fmt.Printf("%d ", arr[i])
        }
    }
    fmt.Println()
}

func hapusElemen(arr []int, n *int, indeks int) {
    for i := indeks; i < *n-1; i++ {
        arr[i] = arr[i+1]
    }
    *n--
}

func hitungRataRata(arr []int, n int) float64 {
    sum := 0
    for i := 0; i < n; i++ {
        sum += arr[i]
    }
    return float64(sum) / float64(n)
}

func hitungStandarDeviasi(arr []int, n int) float64 {
    mean := hitungRataRata(arr, n)
    sumSqDiff := 0.0

```

```

    for i := 0; i < n; i++ {
        diff := float64(arr[i]) - mean
        sumSqDiff += diff * diff
    }
    return math.Sqrt(sumSqDiff / float64(n))
}

func hitungFrekuensi(arr []int, n int, bilangan int) int {
    frekuensi := 0
    for i := 0; i < n; i++ {
        if arr[i] == bilangan {
            frekuensi++
        }
    }
    return frekuensi
}

func main() {
    var n, x, indeksHapus, cariBilangan int
    var arr [100]int

    fmt.Print("Masukkan jumlah elemen array (N): ")
    fmt.Scan(&n)

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

    tampilArray(arr[:], n)
    tampilIndeksGanjil(arr[:], n)
    tampilIndeksGenap(arr[:], n)

    fmt.Print("Masukkan nilai X untuk menampilkan kelipatan: ")
    fmt.Scan(&x)
    tampilKelipatanX(arr[:], n, x)

    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&indeksHapus)
    hapusElemen(arr[:], &n, indeksHapus)

    fmt.Println("Array setelah penghapusan:")
    tampilArray(arr[:], n)

```

```

    fmt.Printf("Rata-rata: %.2f\n", hitungRataRata(arr[:],
n))

    fmt.Printf("Standar deviasi: %.2f\n",
hitungStandarDeviasi(arr[:], n))

    fmt.Print("Masukkan bilangan yang ingin dihitung
frekuensinya: ")
    fmt.Scan(&cariBilangan)
    frek := hitungFrekuensi(arr[:], n, cariBilangan)
    fmt.Printf("Frekuensi bilangan %d: %d\n", cariBilangan,
frek)
}

```

### Screenshoot Output

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided7-2\Unguided7-2.go"
Masukkan jumlah elemen array (N): 6
Masukkan elemen array:
Elemen ke-0: 1
Elemen ke-1: 2
Elemen ke-2: 3
Elemen ke-3: 4
Elemen ke-4: 5
Elemen ke-5: 6
Isi array: 1 2 3 4 5 6
Elemen dengan indeks ganjil: 2 4 6
Elemen dengan indeks genap: 1 3 5
Masukkan nilai X untuk menampilkan kelipatan: 3
Elemen dengan indeks kelipatan 3: 1 4
Masukkan indeks yang ingin dihapus: 0
Array setelah penghapusan:
Isi array: 2 3 4 5 6
Rata-rata: 4.00
Standar deviasi: 1.41
Masukkan bilangan yang ingin dihitung frekuensinya: 2
Frekuensi bilangan 2: 1
PS D:\Alpro2Golang>

```

### Deskripsi Program

Program di atas merupakan implementasi dalam bahasa Go yang mengelola array integer dan melakukan berbagai operasi dasar, seperti menampilkan elemen, menghitung statistik, serta menghapus elemen. Pada awal program, terdapat beberapa fungsi yang masing-masing memiliki tujuan tertentu.

Pertama, fungsi **tampilArray** digunakan untuk menampilkan semua elemen dari array. Dalam fungsi ini, elemen-elemen array diiterasi satu per satu dari indeks 0 hingga  $n-1$  dan dicetak ke layar. Selanjutnya, fungsi **tampilIndeksGanjil** menampilkan elemen yang berada pada indeks ganjil, seperti indeks 1, 3, dan seterusnya. Fungsi ini mengiterasi array dengan langkah 2, dimulai dari indeks 1. Sementara itu, fungsi **tampilIndeksGenap** melakukan hal serupa tetapi hanya untuk indeks genap, seperti 0, 2, 4, dengan mengiterasi array mulai dari indeks 0.

Kemudian, ada fungsi **tampilKelipatanX**. Dalam fungsi ini, kondisi  $i \% x == 0$  digunakan untuk mengecek apakah suatu indeks adalah kelipatan  $x$ . Misalnya, jika  $x$  adalah 2, maka hanya elemen pada indeks 0, 2, 4, dst., yang akan ditampilkan.

Selanjutnya, fungsi **hapusElemen** bertanggung jawab untuk menghapus elemen array pada indeks yang ditentukan pengguna. Fungsi lain yang terdapat dalam program adalah **hitungRataRata**, yang menghitung rata-rata dari semua elemen array. Fungsi ini menjumlahkan semua elemen dan kemudian membaginya dengan jumlah elemen  $n$ . Fungsi **hitungStandarDeviasi** menghitung standar deviasi dari array, yang merupakan ukuran seberapa tersebar nilai-nilai dalam array terhadap rata-ratanya. Selanjutnya, terdapat fungsi **hitungFrekuensi** yang menghitung berapa kali suatu bilangan muncul dalam array. Dalam fungsi ini, array diiterasi dari indeks 0 hingga  $n-1$ , dan setiap kali elemen yang sesuai ditemukan, nilai frekuensi akan bertambah satu.

### 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 ber laga.

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"

func main() {
    var klubA, klubB string
    var skorA, skorB int
    var hasil_226 []string

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

    for i := 1; ; i++ {
        fmt.Printf("Pertandingan %d : ", i)
        fmt.Scan(&skorA, &skorB)
```



```

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

        if skorA > skorB {
            hasil_226 = append(hasil_226, klubA)
        } else if skorB > skorA {
            hasil_226 = append(hasil_226, klubB)
        } else {
            hasil_226 = append(hasil_226, "Draw")
        }
    }

    fmt.Println()
    for i, h := range hasil_226 {
        fmt.Printf("Hasil %d : %s\n", i+1, h)
    }
    fmt.Println("Pertandingan selesai")
}

```

### Screenshot Output



```

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided7-3\Unguided7-3.go"
Klub A : Indonesia
Klub B : Jepang
Pertandingan 1 : 0 4
Pertandingan 2 : 0 4
Pertandingan 3 : 0 4
Pertandingan 4 : -1 8

Hasil 1 : Jepang
Hasil 2 : Jepang
Hasil 3 : Jepang
Pertandingan selesai
PS D:\Alpro2Golang>

```

### Deskripsi Program

Program ini merupakan sistem pencatatan hasil pertandingan antara dua klub olahraga. Program meminta input nama dua klub dan skor pertandingan secara berulang hingga pengguna memasukkan skor negatif. Untuk setiap pertandingan, program menentukan pemenang berdasarkan skor - jika **skorA** lebih besar dari **skorB** maka **klubA** menang, jika **skorB** lebih besar dari **skorA** maka **klubB** menang, dan jika skor sama maka hasilnya "Draw". Hasil setiap pertandingan disimpan dalam array **hasil\_226**. Setelah input selesai (ditandai dengan skor negatif), program

menampilkan hasil seluruh pertandingan secara berurutan dan pesan "Pertandingan selesai". Sebagai contoh, jika diinput klub "Persib" vs "Persija" dengan skor 2-1, 1-1, 0-2, program akan menampilkan hasil: Persib, Draw, Persija.

#### 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 isi array tab
}
```

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

```
Teks      : S E N A N G .
Reverse teks : G N A N E S

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

## Sourcecode

```
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]int

func isiArray(t *tabel, n *int) {
    var kar int
    *n = 0
    fmt.Print("Masukkan karakter (. untuk berhenti): ")
    fmt.Scanf("%c", &kar)
    for kar != '.' && *n <= NMAX {
        t[*n] = kar
        *n++
        fmt.Scanf("%c", &kar)
    }
}

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

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

```

}

func main() {
    var tab tabel
    var m int

    isiArray(&tab, &m)
    fmt.Print("Array awal: ")
    cetakArray(tab, m)
    balikkanArray(&tab, m)
    fmt.Print("Array setelah dibalik: ")
    cetakArray(tab, m)
}

```

### Program setelah modifikasi

```

package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]int

func isiArray(t *tabel, n *int) {
    var kar int
    *n = 0
    fmt.Print("Teks      : ")
    fmt.Scanf("%c", &kar)
    for kar != '.' && *n < NMAX {
        if kar != ' ' {
            t[*n] = kar
            *n++
        }
        fmt.Scanf("%c", &kar)
    }
}

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

```

```

func balikkanArray(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 {
    for i := 0; i < n/2; i++ {
        if t[i] != t[n-1-i] {
            return false
        }
    }
    return true
}

func main() {
    var tab_226 tabel
    var m_226 int

    isiArray(&tab_226, &m_226)

    var tabReverse tabel = tab_226
    balikkanArray(&tabReverse, m_226)

    fmt.Printf("Palindrom    ? %t\n", palindrom(tab_226,
m_226))
}

```

## Screenshoot Output

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided7-4\Unguided7-4.go"
Masukkan karakter (. untuk berhenti): k a s u r t i d u r .
Array awal: k a s u r t i d u r
Array setelah dibalik: r u d i t r u s a k
PS D:\Alpro2Golang>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided7-4\Unguided7-4.go"
Teks          : k a s u r r u s a k .
Palindrom    ? true
PS D:\Alpro2Golang>

```

## Deskripsi Program

Program ini merupakan implementasi untuk membalik urutan karakter dalam array. Program menggunakan tipe data tabel dengan ukuran maksimal 127 karakter. Program memiliki tiga prosedur utama: **isiArray** untuk menerima input karakter hingga ditemui titik (.), **cetakArray** untuk menampilkan isi array dalam bentuk karakter, dan **balikkanArray** untuk membalik urutan elemen array dengan menukar posisi elemen dari ujung ke tengah. Di fungsi main, program memanggil ketiga prosedur tersebut secara berurutan - pertama mengisi array, menampilkan array awal, membalik array, dan menampilkan hasil array yang sudah dibalik. Sebagai contoh, jika diinput "s e n a n g.", program akan menampilkan "Array awal: s e n a n g" dan "Array setelah dibalik: g n a n e s".

Program ini merupakan pengecekan palindrom dari sebuah teks. Program memiliki beberapa prosedur utama: **isiArray** untuk menerima input karakter (mengabaikan spasi) hingga ditemui titik (.), **cetakArray** untuk menampilkan array dengan label tertentu, **balikkanArray** untuk membalik urutan elemen array, dan fungsi palindrom untuk mengecek apakah teks tersebut palindrom dengan membandingkan karakter dari ujung ke tengah. Di fungsi main, program memanggil **isiArray** untuk input teks, membuat salinan array yang dibalik, dan menampilkan hasil pengecekan palindrom (true/false). Sebagai contoh, jika diinput "K A S U R R U S A K.", program akan menghapus spasi dan menampilkan "Palindrom ? true" karena "K A S U R R U S A K" adalah palindrom.