

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII  
STRUCT DAN ARRAY**



**Disusun Oleh :**

**Ghilbran Alfaries Pryma**

**\2311102267 S1IF-11-05**

**Dosen Pengampu :**

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

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

### Alias

Alias dalam Go adalah cara mendefinisikan nama lain untuk tipe data yang sudah ada. Alias dapat digunakan untuk meningkatkan keterbacaan kode atau memberikan nama yang lebih bermakna. Dalam Go, alias didefinisikan menggunakan kata kunci `type`. Misalnya:

```
type MyInt int
```

Di atas, `MyInt` adalah alias untuk tipe `int`. Meskipun mendefinisikan alias, `MyInt` adalah tipe terpisah dan memerlukan konversi eksplisit untuk digunakan bersama tipe asli.

### Fungsi:

- Mempermudah pemahaman kode.
- Memberikan fleksibilitas untuk perubahan di masa depan.

### Struct

adalah tipe data yang digunakan dalam bahasa pemrograman Go (Golang) untuk mengelompokkan sejumlah atribut (field) ke dalam satu kesatuan data. Struct memungkinkan pengembang untuk mendefinisikan struktur data yang kompleks dengan cara yang sederhana, mirip dengan "class" dalam bahasa pemrograman berbasis objek, meskipun Go sendiri bukan bahasa yang sepenuhnya berorientasi objek.

Pengertian **Struct** adalah kumpulan definisi variabel (atau property) dan atau fungsi (atau method), yang dibungkus sebagai tipe data baru dengan nama tertentu. Property dalam struct, tipe datanya bisa bervariasi. Mirip seperti `map`, hanya saja key-nya sudah didefinisikan di awal, dan tipe data tiap itemnya bisa berbeda.

### Keunggulan Struct dalam Go

- Efisiensi: Struct memberikan representasi data yang ringan dan terorganisir.
- Fleksibilitas: Struct dapat diintegrasikan dengan interface untuk mendukung pola desain yang dinamis.
- Kemudahan Penanganan Data: Struct memudahkan pemrograman berbasis entitas dengan data yang kompleks.

### Metode pada Struct

Meskipun Go tidak mendukung konsep class, kita dapat menambahkan metode ke struct dengan mendefinisikan fungsi yang terkait dengan struct tertentu.

Contoh

```
func (p Person) Greet() string {  
    return "Hello, my name is " + p.Name  
}
```

### Array

Array adalah koleksi elemen yang memiliki tipe data yang sama, disusun secara berurutan dalam memori, dan memiliki ukuran tetap. Array di Golang dideklarasikan dengan menentukan tipe elemen dan panjang array.

Contoh :

```
var numbers [5]int // Array dengan panjang 5 elemen bertipe int
```

### Array di Golang memiliki sifat:

- Panjang array bersifat tetap (fixed size).
- Semua elemen harus memiliki tipe data yang sama.
- Elemen diakses melalui indeks, dimulai dari nol.

Contoh penggunaan array:

```
numbers := [5]int{1, 2, 3, 4, 5}  
fmt.Println(numbers[0]) // Output: 1
```

### Array Dinamik (Slice)

Slice adalah abstraksi yang lebih fleksibel dari array. Slice memungkinkan ukuran dinamis yang dapat bertambah atau berkurang selama runtime. Slice dibangun di atas array, tetapi hanya mengacu pada bagian dari array.

```
var nums []int
nums = append(nums, 1, 2, 3)
```

Slice menggunakan fungsi `append` untuk menambahkan elemen dan secara otomatis mengatur ulang ukuran.

### Kelebihan Array:

- Memberikan akses cepat ke elemen menggunakan indeks.
- Membantu pengorganisasian data secara berurutan.
- Namun, karena sifat ukuran tetap, array di Golang kurang fleksibel dibandingkan slice, yang merupakan ekstensi dari array.

### Map

Map adalah struktur data berbasis hash table yang digunakan untuk menyimpan pasangan key-value. Map sangat cepat untuk pencarian data berdasarkan kunci.

Contoh :

```
var scores map[string]int
scores = make(map[string]int)
scores["Alice"] = 90
scores["Bob"] = 85
```

Map mendukung operasi dasar seperti penambahan, pembaruan, dan penghapusan elemen. Operasi pengaksesan elemen dalam map bersifat konstan rata-rata ( $O(1)$ ).

### Manfaat:

- Sangat efisien untuk operasi pencarian.
- Memungkinkan pengelolaan data berdasarkan kunci unik.

## II. GUIDED I

### 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 = dParkir -
    dPulang //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

```
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Guided 1> go run
5 50 10
7 10 15
Lama Parkir : 1 jam 20 menit 5 detik
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Guided 1> █
```

## Deskripsi Program

Berikut adalah penjelasan program dalam bentuk teks tanpa format kode:

Program ini berfungsi untuk menghitung lama waktu parkir berdasarkan waktu kedatangan dan waktu pulang.

- **Import Package:** Program ini mengimpor package ``fmt``, yang digunakan untuk melakukan input dan output.
- **Definisi Tipe Data:** Program mendefinisikan tipe data baru bernama ``waktu``, yang terdiri dari tiga atribut: jam, menit, dan detik. Tipe data ini digunakan untuk menyimpan informasi waktu.
- **Fungsi ``main``:** Ini adalah titik awal eksekusi program. Di dalam fungsi ini, beberapa variabel dideklarasikan:
  - ``wParkir`` dan ``wPulang``: untuk menyimpan waktu kedatangan dan waktu pulang.
  - ``durasi``: untuk menyimpan lama parkir.
  - ``dParkir``, ``dPulang``, dan ``lParkir``: untuk menyimpan total detik dari waktu kedatangan, waktu pulang, dan lama parkir dalam detik.
- **Input Waktu Parkir dan Pulang:** Program meminta pengguna untuk memasukkan waktu kedatangan (``wParkir``) dan waktu pulang (``wPulang``) dalam format jam, menit, dan detik.
- **Konversi Waktu ke Detik:** Waktu kedatangan dan pulang dikonversi ke dalam satuan detik untuk memudahkan perhitungan. Total detik dari waktu kedatangan disimpan dalam variabel ``dParkir``, sedangkan total detik dari waktu pulang disimpan dalam variabel ``dPulang``.
- **Menghitung Lama Parkir:** Lama parkir dihitung dengan mengurangi total detik waktu kedatangan dari total detik waktu pulang. Hasilnya disimpan dalam variabel ``lParkir``.
- **Mengonversi Detik Kembali ke Jam, Menit, dan Detik:** Setelah mendapatkan lama parkir dalam detik, program mengonversinya kembali ke format jam, menit, dan detik. Jam dihitung dengan membagi total detik dengan 3600, menit dihitung dengan sisa pembagian total detik setelah dibagi jam (menggunakan modulus) dibagi 60, dan detik dihitung dengan mengambil sisa pembagian setelah menghitung menit.

- Output Hasil: Program mencetak lama parkir dalam format yang mudah dibaca oleh pengguna.

Secara keseluruhan, program ini adalah contoh sederhana namun efektif untuk menghitung lama parkir berdasarkan input waktu kedatangan dan pulang. Pengguna hanya perlu memasukkan waktu dalam format yang ditentukan untuk mendapatkan lama parkir secara otomatis.

## GUIDED II

### 1. Slice

#### Source Code

```
package main

import (
    "fmt"
)

// Fungsi untuk mengecek apakah nama sudah ada di dalam slice
func sudahTersedia(listNama []string, namaBaru string) bool {
    for _, nama := range listNama {
        if nama == namaBaru {
            return true
        }
    }
    return false
}

func main() {
    // Slice awal untuk daftar kontak dengan beberapa data
    kontakAwal := []string{"Ari", "Bayu", "Citra"}

    // Nama-nama baru yang ingin dimasukkan
    kontakTambahan := []string{"Dian", "Bayu", "Elok"}

    // Menambahkan nama baru hanya jika belum ada di daftar
    for _, kontak := range kontakTambahan {
        if !sudahTersedia(kontakAwal, kontak) {
            kontakAwal = append(kontakAwal, kontak)
        } else {
            fmt.Println("Nama", kontak, "sudah ada di daftar kontak.")
        }
    }
}
```

```

    }

    // Menampilkan daftar kontak akhir
    fmt.Println("Daftar Kontak:", kontakAwal)
}

```

## Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Guided 2 Slice> go run
Nama Bayu sudah ada di daftar kontak.
Daftar Kontak: [Ari Bayu Citra Dian Elok]
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Guided 2 Slice>

```

## Deskripsi Program

Program berfungsi untuk mengelola daftar kontak dengan menambahkan nama baru hanya jika nama tersebut belum ada dalam daftar. Berikut adalah penjelasan rinci tentang cara kerja program ini:

- **Import Package:** Program ini mengimpor package `fmt`, yang digunakan untuk melakukan input dan output, termasuk mencetak ke layar.
- **Fungsi `sudahTersedia`:**
  - Fungsi ini menerima dua parameter: `listNama`, yaitu slice (daftar) yang berisi nama-nama, dan `namaBaru`, yaitu nama yang ingin dicek keberadaannya.
  - Fungsi ini melakukan iterasi (perulangan) pada setiap nama dalam `listNama`. Jika `namaBaru` ditemukan dalam daftar, fungsi akan mengembalikan nilai `true`, menandakan bahwa nama tersebut sudah ada.
  - Jika setelah memeriksa semua nama tidak ditemukan yang cocok, fungsi akan mengembalikan nilai `false`.
- **Fungsi `main`:**
  - Di dalam fungsi `main`, terdapat slice bernama `kontakAwal` yang berisi beberapa nama kontak awal: "Ari", "Bayu", dan "Citra".
  - Ada juga slice bernama `kontakTambahan` yang berisi nama-nama baru yang ingin ditambahkan ke dalam daftar kontak: "Dian", "Bayu", dan "Elok".
- **Menambahkan Nama Baru:**
  - Program melakukan iterasi pada setiap nama dalam `kontakTambahan`.
  - Untuk setiap nama, program memanggil fungsi `sudahTersedia` untuk mengecek apakah nama tersebut sudah ada di dalam `kontakAwal`.



- Jika nama belum ada (fungsi mengembalikan nilai `false`), maka nama tersebut ditambahkan ke dalam `kontakAwal` menggunakan fungsi `append`.
- Jika nama sudah ada (fungsi mengembalikan nilai `true`), program mencetak pesan bahwa nama tersebut sudah ada di daftar kontak.
- Menampilkan Daftar Kontak Akhir:
  - Setelah proses penambahan selesai, program mencetak daftar kontak akhir menggunakan perintah `fmt.Println`.

## GUIDED II

### 2.Map

#### Source Code

```
package main

import (
    "fmt"
)

func main() {
    // Membuat map dengan nama makanan sebagai kunci dan harga
    // sebagai nilai
    hargaMakanan := map[string]int{
        "Nasi Goreng": 20000,
        "Rendang":     35000,
        "Sate":        25000,
    }

    // Menampilkan harga dari setiap makanan
    fmt.Println("Harga Makanan:")
    for makanan, harga := range hargaMakanan {
        fmt.Printf("%s: Rp%d\n", makanan, harga)
    }

    fmt.Print("Harga makanan Sate = ", hargaMakanan["Sate"])
}
```

#### Screenshoot Output

```
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Guided 2 Map> go run "
Harga Makanan:
Nasi Goreng: Rp20000
Rendang: Rp35000
Sate: Rp25000
Harga makanan Sate = 25000
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Guided 2 Map>
```

## Deskripsi Program

Program yang Anda berikan ditulis dalam bahasa pemrograman Go (Golang) dan berfungsi untuk menyimpan dan menampilkan harga makanan menggunakan struktur data `map`. Berikut adalah penjelasan rinci tentang cara kerja program ini:

- **Import Package:**
  - Program ini mengimpor package `fmt`, yang digunakan untuk melakukan input dan output, termasuk mencetak informasi ke layar.
- **Fungsi `main`:**
  - Fungsi `main` adalah titik awal eksekusi program.
- **Membuat Map:**
  - Di dalam fungsi `main`, program membuat sebuah map bernama `hargaMakanan`. Map ini menggunakan tipe data string sebagai kunci (key) dan tipe data int sebagai nilai (value).
  - Map ini diisi dengan beberapa entri yang mengaitkan nama makanan dengan harganya:
    - "Nasi Goreng" dengan harga 20.000
    - "Rendang" dengan harga 35.000
    - "Sate" dengan harga 25.000
- **Menampilkan Harga Makanan:**
  - Program mencetak judul "Harga Makanan:".
  - Menggunakan perulangan `for` dengan `range`, program iterasi melalui setiap pasangan kunci-nilai dalam map `hargaMakanan`.
  - Untuk setiap makanan dan harga, program mencetak nama makanan dan harganya dalam format yang terstruktur, yaitu "Nama Makanan: RpHarga".
- **Menampilkan Harga Makanan Tertentu:**
  - Setelah menampilkan semua harga makanan, program juga mencetak harga khusus untuk makanan "Sate" dengan menggunakan akses langsung ke map.
  - Dengan menulis `hargaMakanan["Sate"]`, program mengambil nilai yang terkait dengan kunci "Sate" dan mencetaknya.

Program ini memungkinkan pengguna untuk melihat daftar harga makanan dan mendapatkan harga spesifik untuk salah satu item dalam daftar tersebut.

## III. UNGUIDED

### UNGUIDED I

#### 1. Soal

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 radlusnya.

**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".

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

Fungsi untuk menghitung jarak titik (a,b) dan (c,d) dimana rumus jarak adalah:

$$jarak = \sqrt{(a - c)^2 + (b - d)^2}$$

Dan juga fungsi untuk menentukan posisi sebuah titik sembarang berada di dalam suatu lingkaran atau tidak.

```
function jarak(p, q titik) -> real
{Mengembalikan jarak antara titik p(x,y) dan titik q(x,y)}

function didalam(c:lingkaran, p:titik) -> boolean
{Mengembalikan true apabila titik p(x,y) berada di dalam lingkaran c yang
memiliki titik pusat (cx,cy) dan radius r}
```

**Catatan:** Lihat paket **math** dalam lampiran untuk menggunakan fungsi **math.Sqrt()** untuk menghitung akar kuadrat.

**Source Code**

```
package main

import (
    "fmt"
```

```

    "math"
)

// Tipe bentukan Titik untuk menyimpan koordinat x dan y
type Titik struct {
    x, y int
}

// Tipe bentukan Lingkaran untuk menyimpan titik pusat (cx, cy) dan radius r
type Lingkaran struct {
    cx, cy, r int
}

// Fungsi untuk menghitung jarak antara dua titik (x1, y1) dan (x2, y2)
func jarak(x1, y1, x2, y2 int) float64 {
    return math.Sqrt(float64((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1)))
}

// Fungsi untuk menentukan apakah titik berada di dalam lingkaran
func titikDiDalamLingkaran(t Titik, l Lingkaran) bool {
    // Menghitung jarak antara titik dan pusat lingkaran
    jarakTitikKePusat := jarak(t.x, t.y, l.cx, l.cy)
    // Jika jarak titik ke pusat lingkaran lebih kecil atau sama dengan radius, maka titik berada di dalam lingkaran
    return jarakTitikKePusat <= float64(l.r)
}

func main() {
    var l1, l2 Lingkaran
    var t Titik

    // Membaca input untuk lingkaran 1 (cx, cy, r)
    fmt.Scan(&l1.cx, &l1.cy, &l1.r)

    // Membaca input untuk lingkaran 2 (cx, cy, r)
    fmt.Scan(&l2.cx, &l2.cy, &l2.r)

    // Membaca input untuk titik sembarang (x, y)
    fmt.Scan(&t.x, &t.y)

    // Mengecek apakah titik di dalam lingkaran 1 dan 2
    inLingkaran1 := titikDiDalamLingkaran(t, l1)
    inLingkaran2 := titikDiDalamLingkaran(t, l2)

    // Menentukan hasil berdasarkan posisi titik terhadap lingkaran
    if inLingkaran1 && inLingkaran2 {

```

```

        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if inLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if inLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

### Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Unguided 1> go run
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Unguided 1>

```

### Deskripsi Program

Program berfungsi untuk menentukan posisi suatu titik relatif terhadap dua lingkaran. Berikut adalah penjelasan rinci tentang cara kerja program ini:

- Definisi Tipe Bentuk:
  - Titik: Tipe bentuk ``Titik`` didefinisikan untuk menyimpan koordinat x dan y dari suatu titik.
  - Lingkaran: Tipe bentuk ``Lingkaran`` didefinisikan untuk menyimpan koordinat pusat lingkaran (cx, cy) dan radius r.
- Fungsi ``jarak``:
  - Fungsi ini digunakan untuk menghitung jarak antara dua titik (x1, y1) dan (x2, y2) menggunakan rumus jarak Euclidean. Hasilnya adalah nilai float64 yang menunjukkan jarak antara kedua titik.
- Fungsi ``titikDiDalamLingkaran``:
  - Fungsi ini menerima dua parameter: sebuah titik (``t``) dan sebuah lingkaran (``l``).
  - Di dalam fungsi, jarak antara titik dan pusat lingkaran dihitung dengan memanggil fungsi ``jarak``.
  - Fungsi ini mengembalikan nilai boolean yang menunjukkan apakah jarak titik ke pusat lingkaran lebih kecil atau sama dengan radius lingkaran. Jika ya, berarti titik berada di dalam lingkaran.
- Fungsi ``main``:
  - Di dalam fungsi ``main``, dua variabel bertipe ``Lingkaran`` (``l1`` dan ``l2``) serta satu variabel bertipe ``Titik`` (``t``) dideklarasikan.
  - Program meminta pengguna untuk memasukkan data untuk lingkaran pertama (cx, cy, r) dan lingkaran kedua (cx, cy, r).
  - Kemudian, program meminta pengguna untuk memasukkan koordinat titik sembarang (x, y).

- Pemeriksaan Posisi Titik:
  - Program memanggil fungsi `titikDiDalamLingkaran` untuk mengecek apakah titik berada di dalam lingkaran pertama (`inLingkaran1`) dan lingkaran kedua (`inLingkaran2`).
  - Berdasarkan hasil pemeriksaan tersebut, program menentukan posisi titik relatif terhadap kedua lingkaran:
  - Jika titik berada di dalam kedua lingkaran, program mencetak "Titik di dalam lingkaran 1 dan 2".
  - Jika hanya berada di dalam lingkaran pertama, program mencetak "Titik di dalam lingkaran 1".
  - Jika hanya berada di dalam lingkaran kedua, program mencetak "Titik di dalam lingkaran 2".
  - Jika titik tidak berada di dalam keduanya, program mencetak "Titik di luar lingkaran 1 dan 2".

## UNGUIDED II

### 2. Soal

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 Indeks 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.

## Source Code

```
package main

import (
    "fmt"
    "math"
)

// Menampilkan seluruh isi array
func tampilkanArray(arr []int) {
    fmt.Println("Isi array:", arr)
}

// Menampilkan elemen dengan indeks genap
func tampilkanIndeksGenap(arr []int) {
    fmt.Println("Elemen dengan indeks genap:")
    for i := 0; i < len(arr); i++ {
        if i%2 == 0 { // Indeks genap
            fmt.Printf("Indeks %d: %d\n", i, arr[i])
        }
    }
    fmt.Println()
}

// Menampilkan elemen dengan indeks ganjil
func tampilkanIndeksGanjil(arr []int) {
    fmt.Println("Elemen dengan indeks ganjil:")
    for i := 0; i < len(arr); i++ {
        if i%2 != 0 { // Indeks ganjil
            fmt.Printf("Indeks %d: %d\n", i, arr[i])
        }
    }
    fmt.Println()
}

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

// Menghapus elemen array pada indeks tertentu
func hapusElemen(arr []int, indeks int) []int {
```

```

        return append(arr[:indeks], arr[indeks+1:]...)
    }

    // Menghitung rata-rata
    func hitungRataRata(arr []int) float64 {
        var sum int
        for _, v := range arr {
            sum += v
        }
        return float64(sum) / float64(len(arr))
    }

    // Menghitung standar deviasi
    func hitungStandarDeviasi(arr []int) float64 {
        if len(arr) <= 1 {
            return 0
        }

        mean := hitungRataRata(arr)
        var variance float64
        for _, v := range arr {
            variance += math.Pow(float64(v)-mean, 2)
        }
        variance /= float64(len(arr))
        return math.Sqrt(variance)
    }

    // Menghitung frekuensi dari bilangan tertentu
    func hitungFrekuensi(arr []int, bilangan int) int {
        count := 0
        for _, v := range arr {
            if v == bilangan {
                count++
            }
        }
        return count
    }

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

        var arr []int
        for i := 0; i < n; i++ {
            var elemen int
            fmt.Printf("Masukkan elemen ke-%d: ", i+1)
            fmt.Scan(&elemen)

```



```

        arr = append(arr, elemen)
    }

    // a. Menampilkan seluruh isi array
    tampilkanArray(arr)

    // b. Menampilkan elemen dengan indeks genap
    tampilkanIndeksGenap(arr)

    // c. Menampilkan elemen dengan indeks ganjil
    tampilkanIndeksGanjil(arr)

    // d. Menampilkan elemen dengan indeks kelipatan x
    var x int
    fmt.Print("Masukkan nilai x untuk indeks kelipatan: ")
    fmt.Scan(&x)
    tampilkanIndeksKelipatanX(arr, x)

    // e. Menghapus elemen pada indeks tertentu
    var indeksHapus int
    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&indeksHapus)
    arr = hapusElemen(arr, indeksHapus)
    fmt.Println("Array setelah penghapusan elemen:", arr)

    // f. Menampilkan rata-rata
    rataRata := hitungRataRata(arr)
    fmt.Printf("Rata-rata elemen dalam array: %.2f\n", rataRata)

    // g. Menampilkan standar deviasi
    standarDeviasi := hitungStandarDeviasi(arr)
    fmt.Printf("Standar deviasi elemen dalam array: %.2f\n",
    standarDeviasi)

    // h. Menampilkan frekuensi suatu bilangan tertentu
    var bilangan int
    fmt.Print("Masukkan bilangan untuk menghitung frekuensinya:
    ")
    fmt.Scan(&bilangan)
    frekuensi := hitungFrekuensi(arr, bilangan)
    fmt.Printf("Frekuensi bilangan %d: %d\n", bilangan,
    frekuensi)
}

```

## Screenshoot Program

```
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Unguided 2> go run
Masukkan jumlah elemen array: 5
Masukkan elemen ke-1: 1
Masukkan elemen ke-2: 2
Masukkan elemen ke-3: 3
Masukkan elemen ke-4: 4
Masukkan elemen ke-5: 5
Isi array: [1 2 3 4 5]
Elemen dengan indeks genap:
Indeks 0: 1
Elemen dengan indeks genap:
Elemen dengan indeks genap:
Indeks 0: 1
Indeks 2: 3
Indeks 4: 5

Elemen dengan indeks ganjil:
Indeks 1: 2
Indeks 3: 4

Masukkan nilai x untuk indeks kelipatan: 5
Elemen dengan indeks kelipatan 5:
Indeks 0: 1

Masukkan indeks yang ingin dihapus: 2
Array setelah penghapusan elemen: [1 2 4 5]
Rata-rata elemen dalam array: 3.00
Standar deviasi elemen dalam array: 1.58
Masukkan bilangan untuk menghitung frekuensinya: 2
Frekuensi bilangan 2: 1
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Unguided 2> |
```

## Deskripsi Program

Program berfungsi untuk mengelola dan menganalisis array integer. Program ini mencakup berbagai fungsi untuk menampilkan elemen array, menghitung statistik, dan memanipulasi data dalam array. Berikut adalah penjelasan rinci tentang cara kerja program ini:

- Program ini mengimpor package `fmt` untuk input dan output, serta package `math` untuk melakukan perhitungan matematika seperti akar kuadrat.
- Fungsi-Fungsi Utama
- Program ini memiliki beberapa fungsi yang masing-masing memiliki tujuan tertentu:
  - tampilkanArray: Menampilkan seluruh isi array.
  - tampilkanIndeksGenap: Menampilkan elemen dengan indeks genap (0, 2, 4, dst.).
  - tampilkanIndeksGanjil: Menampilkan elemen dengan indeks ganjil (1, 3, 5, dst.).
  - tampilkanIndeksKelipatanX: Menampilkan elemen dengan indeks yang merupakan kelipatan dari nilai x yang diberikan.
  - hapusElemen: Menghapus elemen pada indeks tertentu dari array dan mengembalikan array yang baru.
  - hitungRataRata: Menghitung rata-rata dari elemen-elemen dalam array.

- `hitungStandarDeviasi`: Menghitung standar deviasi dari elemen-elemen dalam array.
- `hitungFrekuensi`: Menghitung frekuensi kemunculan suatu bilangan tertentu dalam array.
- Fungsi ``main``  
 Fungsi ``main`` adalah titik awal eksekusi program. Berikut adalah langkah-langkah yang dilakukan di dalamnya:
  - Input Jumlah Elemen Array:
    - Program meminta pengguna untuk memasukkan jumlah elemen yang ingin dimasukkan ke dalam array.
  - Input Elemen Array:
    - Program melakukan iterasi sebanyak jumlah elemen yang diminta untuk menerima input dari pengguna dan menambahkan setiap elemen ke dalam slice ``arr``.
  - Menampilkan Seluruh Isi Array:
    - Memanggil fungsi ``tampilkanArray`` untuk menampilkan semua elemen dalam array.
  - Menampilkan Indeks Genap dan Ganjil:
    - Memanggil fungsi ``tampilkanIndeksGenap`` untuk menampilkan elemen dengan indeks genap.
    - Memanggil fungsi ``tampilkanIndeksGanjil`` untuk menampilkan elemen dengan indeks ganjil.
  - Menampilkan Indeks Kelipatan X:
    - Program meminta pengguna untuk memasukkan nilai x dan kemudian memanggil fungsi ``tampilkanIndeksKelipatanX`` untuk menampilkan elemen pada indeks kelipatan x.
  - Menghapus Elemen pada Indeks Tertentu:
    - Program meminta pengguna untuk memasukkan indeks yang ingin dihapus, kemudian memanggil fungsi ``hapusElemen`` untuk menghapus elemen tersebut dari array dan menampilkan hasilnya.
  - Menghitung Rata-Rata dan Standar Deviasi:
    - Memanggil fungsi ``hitungRataRata`` untuk menghitung dan menampilkan rata-rata dari elemen-elemen yang tersisa di dalam array.
    - Memanggil fungsi ``hitungStandarDeviasi`` untuk menghitung dan menampilkan standar deviasi dari elemen-elemen tersebut.

- Menghitung Frekuensi Bilangan Tertentu:
  - Program meminta pengguna untuk memasukkan bilangan yang ingin dihitung frekuensinya, kemudian memanggil fungsi `hitungFrekuensi` untuk menghitung dan menampilkan frekuensi bilangan tersebut dalam array.

### UNGUIDED III

#### 3.Soal

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                                // MU = 2 sedangkan Inter = 0
Pertandingan 1:2
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: M
Hasil 7: MU
Hasil 8 Inter
Pertandingan selesai
```

## Source code

```
package main

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

// Struct untuk merepresentasikan pertandingan
type Pertandingan struct {
    KlubA    string
    KlubB    string
    SkorA    int
    SkorB    int
    Pemenang string
}

func main() {
    // Membuat scanner untuk input
    scanner := bufio.NewScanner(os.Stdin)

    // Input nama klub
    fmt.Print("Klub A: ")
    scanner.Scan()
    klubA := scanner.Text()

    fmt.Print("Klub B: ")
    scanner.Scan()
    klubB := scanner.Text()

    // Daftar pemenang dan semua pertandingan
    var pemenang []string
    var pertandinganList []Pertandingan

    // Mulai loop untuk input skor pertandingan
    pertandinganKe := 1
    for {
        fmt.Printf("Pertandingan %d: ", pertandinganKe)
        scanner.Scan()
        input := scanner.Text()
        skor := strings.Split(input, " ")

        // Validasi input
        if len(skor) != 2 {
```

```

        fmt.Println("Input tidak valid. Masukkan dua angka
dipisahkan dengan spasi.")
        continue
    }

    // Parsing skor
    skorA, errA := strconv.Atoi(skor[0])
    skorB, errB := strconv.Atoi(skor[1])
    if errA != nil || errB != nil {
        fmt.Println("Input tidak valid. Masukkan angka.")
        continue
    }

    // Periksa validitas skor
    if skorA < 0 || skorB < 0 {
        fmt.Println("Pertandingan selesai")
        break
    }

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

    // Simpan data pertandingan ke struct
    pertandingan := Pertandingan{
        KlubA: klubA,
        KlubB: klubB,
        SkorA: skorA,
        SkorB: skorB,
        Pemenang: pemenangPertandingan,
    }
    pertandinganList = append(pertandinganList, pertandingan)

    // Tampilkan hasil pertandingan
    fmt.Printf("Hasil %d: %s\n", pertandinganKe,
pemenangPertandingan)

    pertandinganKe++
}

```

```

// Tampilkan daftar semua pertandingan
fmt.Println("\nDetail pertandingan:")
for i, pertandingan := range pertandinganList {
    fmt.Printf("Pertandingan %d: %s %d-%d %s, Pemenang: %s\n",
        i+1, pertandingan.KlubA, pertandingan.SkorA,
        pertandingan.SkorB, pertandingan.KlubB, pertandingan.Pemenang)
}

// Tampilkan daftar klub yang menang
if len(pemenang) > 0 {
    fmt.Println("\nDaftar klub pemenang:")
    for _, klub := range pemenang {
        fmt.Println(klub)
    }
} else {
    fmt.Println("\nTidak ada klub yang menang.")
}
}

```

#### Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Unguided 3> go run "d:\TUGAS
Klub A: MU
Klub B: Inter
Pertandingan 1: 2 0
Hasil 1: MU
Pertandingan 2: 1 2
Hasil 2: Inter
Pertandingan 3: 2 2
Hasil 3: Draw
Pertandingan 4: 0 1
Hasil 4: Inter
Pertandingan 5: 3 2
Hasil 5: MU
Pertandingan 6: 1 0
Hasil 6: MU
Pertandingan 7: 5 2
Hasil 7: MU
Pertandingan 8: 2 3
Hasil 8: Inter
Pertandingan 9: -1 2
Pertandingan selesai

Detail pertandingan:
Pertandingan 1: MU 2-0 Inter, Pemenang: MU
Pertandingan 2: MU 1-2 Inter, Pemenang: Inter
Pertandingan 3: MU 2-2 Inter, Pemenang: Draw
Pertandingan 4: MU 0-1 Inter, Pemenang: Inter
Pertandingan 5: MU 3-2 Inter, Pemenang: MU
Pertandingan 6: MU 1-0 Inter, Pemenang: MU
Pertandingan 7: MU 5-2 Inter, Pemenang: MU
Pertandingan 8: MU 2-3 Inter, Pemenang: Inter

Daftar klub pemenang:
MU
Inter
Inter
MU
MU
MU
Inter

```

## Deskripsi Program

**Program yang Anda berikan ditulis dalam bahasa pemrograman Go** (Golang) dan berfungsi untuk mencatat hasil pertandingan antara dua klub sepak bola. Program ini memungkinkan pengguna untuk memasukkan nama klub, mencatat skor pertandingan, dan menentukan pemenang. Berikut adalah penjelasan rinci tentang cara kerja program ini:

- Definisi Struct
  - Program mendefinisikan sebuah struct bernama ``Pertandingan`` yang memiliki atribut:
    - ``KlubA``: Nama klub A.
    - ``KlubB``: Nama klub B.
    - ``SkorA``: Skor yang dicetak oleh klub A.
    - ``SkorB``: Skor yang dicetak oleh klub B.
    - ``Pemenang``: Nama klub yang menang atau "Draw" jika hasilnya imbang.
- Fungsi ``main``
  - Membuat Scanner untuk Input:
    - Program membuat scanner menggunakan ``bufio.NewScanner(os.Stdin)`` untuk membaca input dari konsol.
  - Input Nama Klub:
    - Program meminta pengguna untuk memasukkan nama Klub A dan Klub B dengan menggunakan ``scanner.Scan()`` dan menyimpan hasilnya.
  - Inisialisasi Daftar Pemenang dan Pertandingan:
    - Dua slice dideklarasikan: satu untuk menyimpan nama pemenang (``pemenang``) dan satu lagi untuk menyimpan daftar pertandingan (``pertandinganList``).
  - Loop untuk Input Skor Pertandingan:
    - Program memasuki loop di mana pengguna diminta untuk memasukkan skor pertandingan satu per satu.
    - Input diharapkan dalam format "skorA skorB", yang kemudian dipisahkan menggunakan ``strings.Split``.
  - Validasi Input:
    - Jika input tidak valid (tidak ada dua angka), program memberikan pesan kesalahan dan meminta input ulang.
    - Jika skor yang dimasukkan adalah angka negatif, program menganggap pertandingan selesai dan keluar dari loop.
  - Menentukan Pemenang:
    - Program membandingkan skor untuk menentukan pemenang pertandingan:
      - Jika skorA lebih besar dari skorB, Klub A menang.
      - Jika skorB lebih besar dari skorA, Klub B menang.
      - Jika kedua skor sama, hasilnya dianggap "Draw".



- Pemenang disimpan dalam slice `pemenang` dan juga dalam struct `Pertandingan`.
- Menyimpan Data Pertandingan:
  - Data pertandingan disimpan dalam struct `Pertandingan` dan ditambahkan ke slice `pertandinganList`.
- Menampilkan Hasil Pertandingan:
  - Setelah setiap input, hasil pertandingan ditampilkan di konsol.
- Menampilkan Detail Semua Pertandingan:
  - Setelah loop berakhir, program menampilkan detail semua pertandingan yang telah dicatat.
- Menampilkan Daftar Klub yang Menang:
  - Program menampilkan daftar klub yang menang berdasarkan data yang telah dikumpulkan selama pertandingan.

## UNGUIDED IV

### 4.Soal

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

:integer

func ls1Array(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 <= 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 mint

// 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 : SENANG.
Reverse teks : GNANES

Teks : KATAK.
Reverse teks : KATAK

```

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)

**Source Code**

```

package main

import (
    "fmt"
)

const NMAX int = 127

type tabel [NMAX]rune

// Fungsi untuk mengisi array
func isiArray(t *tabel, n *int) {

```

```

    var ch rune
    *n = 0
    fmt.Println("Masukkan karakter (akhiri dengan '.' atau maksimal
127 karakter):")
    for *n < NMAX {
        fmt.Scanf("%c", &ch)
        if ch == '.' {
            break
        }
        (*t)[*n] = ch
        *n++
    }
}

// Fungsi untuk mencetak array
func cetakArray(t tabel, n int) {
    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-i-1] = t[n-i-1], t[i]
    }
}

// Fungsi untuk memeriksa apakah array adalah palindrom
func palindrom(t tabel, n int) bool {
    var reversed tabel
    copy(reversed[:n], t[:n])
    balikanArray(&reversed, n)
    for i := 0; i < n; i++ {
        if t[i] != reversed[i] {
            return false
        }
    }
    return true
}

// Fungsi utama
func main() {
    var tab tabel
    var n int

    // Mengisi array

```

```

isiArray(&tab, &n)

// Menampilkan teks asli
fmt.Print("Teks : ")
cetakArray(tab, n)

// Membalikkan isi array
balikanArray(&tab, n)
fmt.Print("Reverse teks : ")
cetakArray(tab, n)

// Memeriksa apakah palindrom
if palindrom(tab, n) {
    fmt.Println("Teks tersebut adalah palindrom.")
} else {
    fmt.Println("Teks tersebut bukan palindrom.")
}
}

```

### Screenshoot Program

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Unguided 4> go run
Masukkan karakter (akhiri dengan '.' atau maksimal 127 karakter):
GHILBRAN.
Teks : GHILBRAN
Reverse teks : NARBLIHG
Teks tersebut bukan palindrom.
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 7\Unguided 4> 

```

### Deskripsi Program

Program yang Anda berikan adalah program dalam bahasa pemrograman Go (Golang) yang berfungsi untuk mengisi, mencetak, membalikkan, dan memeriksa apakah suatu string adalah palindrom. Berikut adalah penjelasan rinci tentang cara kerja program ini:

- Konstanta dan Tipe Data
  - NMAX: Didefinisikan sebagai 127, yang merupakan batas maksimum jumlah karakter yang dapat dimasukkan ke dalam array.
  - tabel: Didefinisikan sebagai tipe data array dari `rune` dengan ukuran `NMAX`. Tipe `rune` digunakan untuk menyimpan karakter Unicode.
- Fungsi-Fungsi Utama
- Program ini memiliki tujuan:
  - isiArray:
  - Fungsi ini digunakan untuk mengisi array dengan karakter yang dimasukkan oleh pengguna.

- Pengguna diminta untuk memasukkan karakter satu per satu hingga mencapai batas maksimum (127 karakter) atau hingga pengguna memasukkan titik (`. `) untuk menandakan akhir input.
- Karakter yang dimasukkan disimpan dalam array dan jumlah karakter yang dimasukkan disimpan dalam variabel `n`.
- cetakArray:  
Fungsi ini mencetak karakter-karakter dalam array hingga jumlah `n`.
- balikanArray:  
Fungsi ini membalikkan urutan elemen dalam array.  
Menggunakan teknik swapping untuk menukar elemen dari depan dan belakang hingga mencapai tengah array.
- palindrom:  
Fungsi ini memeriksa apakah array adalah palindrom.  
Pertama, fungsi membuat salinan dari array asli dan membalikkan salinan tersebut.  
Kemudian, fungsi membandingkan elemen-elemen dari array asli dengan salinan yang dibalik. Jika semua elemen sama, maka string tersebut adalah palindrom.
- 4. Fungsi `main`  
Fungsi `main` adalah titik awal eksekusi program. Berikut adalah langkah-langkah yang dilakukan di dalamnya:
  - Deklarasi Variabel:
    - Variabel `tab` bertipe `tabel` untuk menyimpan karakter.
    - Variabel `n` untuk menyimpan jumlah karakter yang dimasukkan.
  - Mengisi Array:
    - Memanggil fungsi `isiArray` untuk meminta pengguna memasukkan karakter ke dalam array.
  - Menampilkan Teks Asli:
    - Memanggil fungsi `cetakArray` untuk menampilkan teks asli yang telah dimasukkan.
  - Membalikkan Isi Array:
    - Memanggil fungsi `balikanArray` untuk membalikkan urutan karakter dalam array.
    - Menampilkan teks yang telah dibalik menggunakan fungsi `cetakArray`.

- Memeriksa Apakah Palindrom:
  - Memanggil fungsi `palindrom` untuk memeriksa apakah teks asli adalah palindrom.
  - Menampilkan hasil pemeriksaan ke layar.

#### **IV.DAFTAR PUSTAKA**

- [1] "Alias dalam Go," Deskripsi tentang cara mendefinisikan alias tipe data di bahasa Go, termasuk manfaat dan contohnya. [Online]. Available: <https://golang.org/doc/>. [Accessed: Nov. 17, 2024].
- [2] "Struct di Golang," Pengertian dan keunggulan struct dalam bahasa pemrograman Go, serta contoh penggunaannya. [Online]. Available: [https://golang.org/doc/effective\\_go.html](https://golang.org/doc/effective_go.html). [Accessed: Nov. 17, 2024].
- [3] "Array dalam Go," Penjelasan konsep array, array dinamik (slice), dan kelebihan dalam pengorganisasian data di Golang. [Online]. Available: <https://golang.org/pkg/>. [Accessed: Nov. 17, 2024].
- [4] "Map dalam Go," Definisi, manfaat, dan contoh penggunaan map untuk pengelolaan data berbasis key-value. [Online]. Available: <https://golang.org/ref/spec>. [Accessed: Nov. 17, 2024].