

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCK & ARRAY**



Disusun Oleh :

Raihan Ramadhan/2311102040

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

Array adalah kumpulan data yang memiliki tipe yang sama dan disimpan dalam satu variabel. Ukuran array ditentukan saat array dibuat, sehingga jumlah elemen yang dapat disimpan tidak boleh melebihi kapasitas yang telah dialokasikan.

Setiap elemen dalam array memiliki nilai awal berdasarkan tipe datanya. Misalnya, elemen bertipe int akan bernilai 0 secara default, sedangkan elemen bertipe bool akan bernilai false, dan seterusnya. Masing-masing elemen array diidentifikasi dengan indeks berupa angka yang menunjukkan posisi elemen tersebut, di mana indeks dimulai dari 0.

```
var names [4]string
names[0] = "trafalgar"
names[1] = "d"
names[2] = "water"
names[3] = "law"

fmt.Println(names[0], names[1], names[2], names[3])
```

Variabel names dideklarasikan sebagai array string dengan alokasi kapasitas elemen adalah 4 slot. Cara mengisi slot elemen array bisa dilihat di kode di atas, yaitu dengan langsung mengakses elemen menggunakan indeks, lalu mengisinya.

Slice adalah kumpulan elemen yang mirip dengan array, tetapi memiliki ukuran yang dinamis. Ini berarti bahwa ukuran slice dapat berubah selama runtime, memungkinkan elemen untuk ditambahkan atau dihapus secara fleksibel

Membuat Slice

slice := []type{elem1, elem2, ..., elemN}

Di mana type adalah tipe data elemen dalam slice (seperti int, string, dll.), dan elem1 hingga elemN adalah nilai-nilai elemen yang ingin ditambahkan. Contoh: mySlice := []int{1, 2, 3, 4, 5}

Map di Go adalah tipe data asosiatif yang memungkinkan penyimpanan nilai dengan menggunakan kunci yang unik. Kunci dapat berupa tipe data apa pun yang dapat dibandingkan, sedangkan nilai dapat berupa tipe data apa pun. Map mirip dengan slice, tetapi menggunakan kunci sebagai pengakses nilai alih-alih indeks numerik.

Deklarasi dan Inisialisasi

```
var myMap map[string]int
```

II. Guided

1. Guided 1 Source Code

```
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)
}
```

Screenshot Output

```
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\PRAKTIKUM\MODUL VII>
1 20 50
2 40 50
Lama Parkir : 1 jam 20 menit 0 detik
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\PRAKTIKUM\MODUL VII>
```

Deskripsi Program

Program ini berfungsi untuk menghitung lama waktu parkir berdasarkan waktu kedatangan dan kepulangan dalam satuan jam, menit, dan detik. Struktur `waktu`

dibuat dengan tiga atribut: jam, menit, dan detik, yang digunakan untuk merepresentasikan data waktu. Pada fungsi main, variabel wParkir dan wPulang disiapkan untuk menyimpan waktu kedatangan dan waktu pulang yang dimasukkan oleh pengguna. Selanjutnya, kedua waktu tersebut dikonversi ke dalam satuan detik dan disimpan pada variabel dParkir dan dPulang, agar selisih waktu (dihitung dalam variabel lParkir) bisa dihitung dengan mudah. Akhirnya, hasil durasi dalam bentuk jam, menit, dan detik ditampilkan sebagai output dengan format "Lama Parkir : [jam] jam [menit] menit [detik] detik".

2. Guided 2

Source Code

```
package main

import (
    "fmt"
)

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

func main() {
    // Slice awal untuk daftar teman dengan beberapa data
    daftarTeman := []string{"Andi", "Budi", "Cici"}

    // Nama-nama baru yang ingin ditambahkan
    namaBaru := []string{"Dewi", "Budi", "Eka"}

    // Menambahkan nama baru hanya jika belum ada di daftar
    for _, nama := range namaBaru {
        if !sudahAda(daftarTeman, nama) {
            daftarTeman = append(daftarTeman, nama)
        } else {
            fmt.Println("Nama", nama, "sudah ada dalam daftar.")
        }
    }
}
```

```
// Menampilkan daftar teman akhir
fmt.Println("Daftar Teman:", daftarTeman)
}
```

Screenshoot Output

```
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\
```

Deskripsi Program

Program di atas dibuat menggunakan bahasa Go untuk menambahkan nama-nama baru ke dalam daftar teman, tapi hanya jika nama tersebut belum ada di dalam daftar. Awalnya, program memiliki daftar teman berisi Andi, Budi, dan Cici, serta daftar nama baru seperti Dewi, Budi, dan Eka. Ada fungsi bernama sudahAda yang bertugas mengecek apakah sebuah nama sudah ada di daftar teman dengan membandingkan setiap nama satu per satu. Jika nama baru belum ada, maka nama tersebut ditambahkan ke daftar menggunakan fungsi append, sedangkan jika nama sudah ada, program akan memberi tahu bahwa nama itu sudah ada. Setelah semua nama diproses, program akan menampilkan daftar teman yang sudah diperbarui.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    // Membuat map dengan nama buah sebagai kunci dan harga sebagai
    nilai
    hargaBuah := map[string]int{
        "Apel": 5000,
        "Pisang": 3000,
        "Mangga": 7000,
    }

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

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

```
}
```

Screenshot Output

```
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\PRAKTIKUM\MODUL VII>
Harga Buah:
Mangga: Rp7000
Apel: Rp5000
Pisang: Rp3000
Harga buah Mangga = 7000
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\PRAKTIKUM\MODUL VII>
```

Deskripsi Program

Program ini adalah contoh sederhana dalam bahasa Go yang memanfaatkan map untuk menyimpan data harga beberapa jenis buah. Di dalam map hargaBuah, nama buah dijadikan sebagai kunci (bertipe string), dan harga buah sebagai nilai (bertipe int), dengan data harga Apel, Pisang, dan Mangga. Pada langkah berikutnya, program menampilkan seluruh harga buah melalui perulangan for dengan range, yang berfungsi untuk mencetak setiap nama buah beserta harganya. Selain itu, program juga secara khusus menampilkan harga buah Mangga dengan mengakses nilai dalam map `hargaBuah` menggunakan kunci "Mangga". Hasil output menunjukkan daftar harga masing-masing buah serta harga khusus untuk buah Mangga. (n-1) dan mengalikan hasilnya dengan (2). Pada fungsi main(), program meminta input dari pengguna untuk nilai (n), kemudian memanggil fungsi pangkat() dan menampilkan hasilnya di layar.

III. Unguided

1. Unguided 1

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

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

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

Source Code

```
package main

import (
    "fmt"
```

```

    "math"
)

// Definisikan struktur untuk titik dengan koordinat x dan y
type Titik struct {
    x, y float64
}

// Definisikan struktur untuk lingkaran dengan titik pusat dan radius
type Lingkaran struct {
    pusat Titik
    radius float64
}

// Fungsi untuk menghitung jarak antara dua titik
func jarak(t1, t2 Titik) float64 {
    return math.Sqrt((t1.x-t2.x)*(t1.x-t2.x) + (t1.y-t2.y)*(t1.y-t2.y))
}

// Fungsi untuk mengecek apakah titik berada di dalam lingkaran
func diDalamLingkaran(t Titik, l Lingkaran) bool {
    return jarak(t, l.pusat) <= l.radius
}

// Fungsi untuk menentukan posisi titik terhadap dua lingkaran
func cekPosisiTitik(t Titik, l1, l2 Lingkaran) string {
    diLingkaran1 := diDalamLingkaran(t, l1)
    diLingkaran2 := diDalamLingkaran(t, l2)

    switch {
    case diLingkaran1 && diLingkaran2:
        return "Titik di dalam lingkaran 1 dan 2"
    case diLingkaran1:
        return "Titik di dalam lingkaran 1"
    case diLingkaran2:
        return "Titik di dalam lingkaran 2"
    default:
        return "Titik di luar lingkaran 1 dan 2"
    }
}

func main() {
    var lingkaran1, lingkaran2 Lingkaran

```

```

var titik Titik

// Input baris pertama untuk lingkaran 1
fmt.Println("Masukkan koordinat pusat dan radius lingkaran 1 (x
y radius):")
fmt.Scan(&lingkaran1.pusat.x, &lingkaran1.pusat.y,
&lingkaran1.radius)

// Input baris kedua untuk lingkaran 2
fmt.Println("Masukkan koordinat pusat dan radius lingkaran 2 (x
y radius):")
fmt.Scan(&lingkaran2.pusat.x, &lingkaran2.pusat.y,
&lingkaran2.radius)

// Input baris ketiga untuk titik yang akan dicek
fmt.Println("Masukkan koordinat titik yang akan dicek (x y):")
fmt.Scan(&titik.x, &titik.y)

// Menentukan posisi titik
hasil := cekPosisiTitik(titik, lingkaran1, lingkaran2)
fmt.Println(hasil)
}

```

Screenshot Output

```

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\PRAKTIKUM\MODUL VII>
Masukkan koordinat pusat dan radius lingkaran 1 (x y radius):
1 1 5
Masukkan koordinat pusat dan radius lingkaran 2 (x y radius):
8 8 4
Masukkan koordinat titik yang akan dicek (x y):
2 2
Titik di dalam lingkaran 1
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\PRAKTIKUM\MODUL VII>

```

Deskripsi Program

Program ini dibuat untuk mengecek apakah sebuah titik ada di dalam lingkaran 1, lingkaran 2, kedua lingkaran, atau di luar keduanya. Pengguna memasukkan tiga baris data: baris pertama berisi koordinat pusat dan radius lingkaran 1, baris kedua berisi koordinat pusat dan radius lingkaran 2, dan baris ketiga berisi koordinat titik yang ingin dicek. Program ini menghitung jarak antara titik dan pusat lingkaran menggunakan rumus jarak, lalu membandingkan jarak tersebut dengan radius lingkaran. Berdasarkan hasilnya, program akan memberi tahu apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, atau di luar keduanya. Dengan begitu, pengguna bisa mengetahui dengan jelas posisi titik tersebut terhadap lingkaran-lingkaran yang dimasukkan.

2. Unguided 2

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. 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.

Source Code

```
package main

import (
    "fmt"
    "math"
)

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

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

    for {
        var choice int
        fmt.Println("\nMenu:")
        fmt.Println("1. Tampilkan semua elemen")
        fmt.Println("2. Tampilkan elemen pada indeks ganjil")
        fmt.Println("3. Tampilkan elemen pada indeks genap")
        fmt.Println("4. Tampilkan elemen pada indeks kelipatan x")
        fmt.Println("5. Hapus elemen pada indeks tertentu")
        fmt.Println("6. Tampilkan rata-rata elemen")
        fmt.Println("7. Tampilkan deviasi standar elemen")
```

```

        fmt.Println("8. Tampilkan frekuensi angka tertentu")
        fmt.Println("9. Keluar")
        fmt.Print("Pilih opsi: ")
        fmt.Scan(&choice)

        switch choice {
        case 1:
            tampilkanSemua(array)
        case 2:
            tampilkanIndeksGanjil(array)
        case 3:
            tampilkanIndeksGenap(array)
        case 4:
            var x int
            fmt.Print("Masukkan nilai x: ")
            fmt.Scan(&x)
            tampilkanKelipatanX(array, x)
        case 5:
            var index int
            fmt.Print("Masukkan indeks untuk dihapus: ")
            fmt.Scan(&index)
            array = hapusPadaIndeks(array, index)
            tampilkanSemua(array)
        case 6:
            fmt.Printf("Rata-rata: %.2f\n", rataRata(array))
        case 7:
            fmt.Printf("Deviasi Standar: %.2f\n",
deviasiStandar(array))
        case 8:
            var number int
            fmt.Print("Masukkan angka untuk mencari frekuensinya: ")
            fmt.Scan(&number)
            fmt.Printf("Frekuensi angka %d: %d\n", number,
frekuensi(array, number))
        case 9:
            fmt.Println("Keluar...")
            return
        default:
            fmt.Println("Pilihan tidak valid. Silakan coba lagi.")
        }
    }
}

func tampilkanSemua(array []int) {
    fmt.Println("Elemen array:", array)
}

```

```

}

func tampilkanIndeksGanjil(array []int) {
    fmt.Print("Elemen pada indeks ganjil: ")
    for i := 1; i < len(array); i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}

func tampilkanIndeksGenap(array []int) {
    fmt.Print("Elemen pada indeks genap: ")
    for i := 0; i < len(array); i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}

func tampilkanKelipatanX(array []int, x int) {
    fmt.Printf("Elemen pada indeks yang merupakan kelipatan dari %d: ", x)
    for i := x; i < len(array); i += x {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}

func hapusPadaIndeks(array []int, index int) []int {
    if index < 0 || index >= len(array) {
        fmt.Println("Indeks tidak valid.")
        return array
    }
    return append(array[:index], array[index+1:]...)
}

func rataRata(array []int) float64 {
    sum := 0
    for _, value := range array {
        sum += value
    }
    return float64(sum) / float64(len(array))
}

func deviasiStandar(array []int) float64 {
    avg := rataRata(array)

```

```

var varianceSum float64
for _, value := range array {
    varianceSum += math.Pow(float64(value)-avg, 2)
}
return math.Sqrt(varianceSum / float64(len(array)))
}

func frekuensi(array []int, number int) int {
    count := 0
    for _, value := range array {
        if value == number {
            count++
        }
    }
    return count
}

```

Screenshot Output

```

PS D:\Wuliah\Wartul\SEMESTER 3\ALPRO PRAKTEK\PRAKTIK
Masukkan jumlah elemen dalam array: 5
Masukkan elemen-elemen array:
1 3 5 7 9

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen pada indeks ganjil
3. Tampilkan elemen pada indeks genap
4. Tampilkan elemen pada indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan deviasi standar elemen
8. Tampilkan frekuensi angka tertentu
9. Keluar
Pilih opsi: 1
Elemen array: [1 3 5 7 9]

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen pada indeks ganjil
3. Tampilkan elemen pada indeks genap
4. Tampilkan elemen pada indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan deviasi standar elemen
8. Tampilkan frekuensi angka tertentu
9. Keluar
Pilih opsi: 2
Elemen pada indeks ganjil: 3 7

```

```

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen pada indeks ganjil
3. Tampilkan elemen pada indeks genap
4. Tampilkan elemen pada indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan deviasi standar elemen
8. Tampilkan frekuensi angka tertentu
9. Keluar
Pilih opsi: 3
Elemen pada indeks genap: 1 5 9

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen pada indeks ganjil
3. Tampilkan elemen pada indeks genap
4. Tampilkan elemen pada indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan deviasi standar elemen
8. Tampilkan frekuensi angka tertentu
9. Keluar
Pilih opsi: 4
Masukkan nilai x: 3
Elemen pada indeks yang merupakan kelipatan dari 3: 7

```

```

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen pada indeks ganjil
3. Tampilkan elemen pada indeks genap
4. Tampilkan elemen pada indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan deviasi standar elemen
8. Tampilkan frekuensi angka tertentu
9. Keluar
Pilih opsi: 5
Masukkan indeks untuk dihapus: 2
Elemen array: [1 3 7 9]

```

```

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen pada indeks ganjil
3. Tampilkan elemen pada indeks genap
4. Tampilkan elemen pada indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan deviasi standar elemen
8. Tampilkan frekuensi angka tertentu
9. Keluar
Pilih opsi: 6
Rata-rata: 5.00

```

```

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen pada indeks ganjil
3. Tampilkan elemen pada indeks genap
4. Tampilkan elemen pada indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan deviasi standar elemen
8. Tampilkan frekuensi angka tertentu
9. Keluar
Pilih opsi: 7
Deviasi Standar: 3.16

```

```

Menu:
1. Tampilkan semua elemen
2. Tampilkan elemen pada indeks ganjil
3. Tampilkan elemen pada indeks genap
4. Tampilkan elemen pada indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan deviasi standar elemen
8. Tampilkan frekuensi angka tertentu
9. Keluar
Pilih opsi: 8
Masukkan angka untuk mencari frekuensinya: 2
Frekuensi angka 2: 0

```

Deskripsi Program

Program ini memungkinkan kita untuk mengelola array dengan berbagai cara, seperti menampilkan, menghapus, atau menganalisis data yang ada. Pertama-tama, kita diminta untuk memasukkan jumlah elemen dan nilai-nilai yang ingin disimpan

dalam array. Setelah itu, program memberikan pilihan menu yang bisa digunakan untuk melihat elemen pada posisi genap atau ganjil, menghitung rata-rata, deviasi standar, atau frekuensi angka tertentu. Program juga memungkinkan kita untuk menghapus elemen pada posisi tertentu dan menampilkan elemen yang ada pada posisi kelipatan angka yang dimasukkan. Program akan terus berjalan sampai kita memilih untuk keluar dari menu.

3. Unguided 3

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.

Source Code

```
package main

import (
    "fmt"
)

func main() {
    // Input nama klub
    var clubA, clubB string
    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&clubA)

    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scanln(&clubB)

    var winners []string // Array untuk menyimpan pemenang

    // Input skor pertandingan
    for i := 1; ; i++ {
        var scoreA, scoreB int
        fmt.Printf("Masukkan skor pertandingan %d (%s vs %s):\n", i,
clubA, clubB)

        fmt.Printf("Skor %s: ", clubA)
        fmt.Scanln(&scoreA)

        fmt.Printf("Skor %s: ", clubB)
```

```

        fmt.Scanln(&scoreB)

        // Cek jika skor tidak valid (negatif)
        if scoreA < 0 || scoreB < 0 {
            fmt.Println("Skor tidak valid! Program selesai.")
            break
        }

        // Tentukan pemenang
        if scoreA > scoreB {
            winners = append(winners, clubA)
            fmt.Printf("Hasil pertandingan %d: %s menang\n", i,
clubA)
        } else if scoreA < scoreB {
            winners = append(winners, clubB)
            fmt.Printf("Hasil pertandingan %d: %s menang\n", i,
clubB)
        } else {
            winners = append(winners, "Draw")
            fmt.Printf("Hasil pertandingan %d: Seri (Draw)\n", i)
        }
    }

    // Tampilkan daftar pemenang
    fmt.Println("\nDaftar pemenang pertandingan:")
    for _, winner := range winners {
        if winner != "Draw" {
            fmt.Println(winner)
        }
    }
}

```

Screenshot Output

```

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\
Masukkan nama Klub A: ars
Masukkan nama Klub B: man
Masukkan skor pertandingan 1 (ars vs man):
Skor ars: 1
Skor man: 2
Hasil pertandingan 1: man menang
Masukkan skor pertandingan 2 (ars vs man):
Skor ars: 2
Skor man: 2
Hasil pertandingan 2: Seri (Draw)
Masukkan skor pertandingan 3 (ars vs man):
Skor ars: 2
Skor man: 1
Hasil pertandingan 3: ars menang
Skor ars: 2
Skor man: 1
Hasil pertandingan 3: ars menang

```

```

Masukkan skor pertandingan 4 (ars vs man):
Skor ars: 3
Skor man: 1
Skor man: 1
Hasil pertandingan 4: ars menang
Hasil pertandingan 4: ars menang
Masukkan skor pertandingan 5 (ars vs man):
Skor ars: 1
Skor man: 3
Hasil pertandingan 5: man menang
Masukkan skor pertandingan 6 (ars vs man):
Skor ars: 4
Skor man: 4
Hasil pertandingan 6: Seri (Draw)
Masukkan skor pertandingan 7 (ars vs man):
Skor ars: -1
Skor man: 3
Skor tidak valid! Program selesai.

```

```

Daftar pemenang pertandingan:
man
ars
ars
man

```

Deskripsi Program

Program ini dirancang untuk mencatat hasil pertandingan antara dua klub dengan meminta pengguna memasukkan nama klub dan skor setiap pertandingan. Pengguna terlebih dahulu menginput nama kedua klub, yang kemudian disimpan dalam variabel clubA dan clubB. Program menjalankan loop untuk mencatat skor dari setiap pertandingan, memvalidasi input agar hanya menerima skor positif; jika skor negatif dimasukkan, program akan berhenti. Pemenang pertandingan ditentukan berdasarkan skor yang lebih tinggi, dengan nama klub pemenang disimpan dalam array winners, sementara hasil seri dicatat sebagai "Draw". Setelah semua pertandingan dicatat, program menampilkan daftar klub yang menang, tanpa menyertakan pertandingan yang berakhir seri, dan memberikan ringkasan hasil pertandingan kepada pengguna.

4. Unguided 4

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

Source Code

```
package main

import (
    "fmt"
    "strings"
)

const NMAX int = 127

type tabel [NMAX]rune

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 */
    var input string
    fmt.Println("Masukkan kata atau teks (akhiri dengan titik):")
    fmt.Scanln(&input)

    // Hapus titik jika ada
    input = strings.TrimSuffix(input, ".")
    *n = len(input)

    for i := 0; i < *n; i++ {
```

```

        t[i] = rune(input[i]) // Mengisi array dengan karakter-
        karakter input
    }
}

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

func balikanArray(t *tabel, n int) {
    /* I.S. Terdefinisi array t yang berisi sejumlah n karakter
       F.S. Urutan isi array t terbalik */
    for i, j := 0, n-1; i < j; i, j = i+1, j-1 {
        t[i], t[j] = t[j], t[i]
    }
}

func isPalindrome(t tabel, n int) bool {
    /* Mengembalikan true apabila susunan karakter di dalam t
    membentuk palindrom,
    dan false apabila sebaliknya. */
    for i := 0; i < n/2; i++ {
        if t[i] != t[n-i-1] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var m int

    // Isi array tab dengan memanggil prosedur isiArray
    isiArray(&tab, &m)

    // Cetak isi array tab
    fmt.Print("Teks: ")
    cetakArray(tab, m)

    // Balikkan isi array tab dengan memanggil balikanArray

```



```

    fmt.Print("Reverse teks: ")
    balikanArray(&tab, m)
    cetakArray(tab, m)

    // Periksa apakah array adalah palindrom
    fmt.Print("Palindrome: ")
    if isPalindrome(tab, m) {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }
}

```

Screenshot Output

```

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\PRAKTIKUM\MODUL VII>
Masukkan kata atau teks (akhiri dengan titik):
senang.
Teks: senang
Reverse teks: gnanes
Palindrome: false
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\PRAKTIKUM\MODUL VII>

```

Deskripsi Program

Program ini digunakan untuk memproses sebuah teks yang dimasukkan oleh pengguna. Program akan membalikkan urutan teks tersebut dan mengecek apakah teks itu sama jika dibaca dari depan maupun belakang disebut palindrom. Fungsi `isiArray` bertugas untuk membaca teks dari pengguna hingga menemukan tanda titik (`.`), lalu menyimpannya ke dalam sebuah array sambil menghilangkan titik tersebut. Fungsi `cetakArray` digunakan untuk menampilkan isi array, dan fungsi `balikanArray` akan membalikkan urutan karakter dalam array dengan cara menukar elemen di awal dan akhir hingga mencapai tengah. Fungsi `isPalindrome` mengecek apakah teks di array sama saat dibaca dari depan dan belakang, lalu memberikan hasil `true` jika teks adalah palindrom. Di bagian utama program, semua fungsi ini dijalankan secara berurutan untuk membaca, membalik, mencetak teks, dan memeriksa apakah teks adalah palindrom.

