

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 7
STRUCT & ARRAY**



Oleh:

GALIH TRISNA

2311102050

IF-11-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

Struct adalah tipe data di Go yang digunakan untuk mengelompokkan beberapa data yang berbeda tipe dalam satu kesatuan. Struct mirip dengan objek di bahasa pemrograman lain, di mana kita dapat membuat tipe data yang lebih kompleks dengan menggabungkan berbagai tipe data sederhana. Struct sangat berguna untuk merepresentasikan data yang memiliki berbagai atribut atau properti, seperti data mahasiswa dengan atribut nama, NIM, dan jurusan.

Array adalah kumpulan elemen dengan tipe data yang sama dan ukuran tetap. Array digunakan untuk menyimpan data dalam jumlah banyak yang memiliki tipe data sejenis dalam satu variabel. Dalam bahasa Go, ukuran array bersifat tetap setelah ditentukan, sehingga kita tidak bisa menambah atau mengurangi elemen array setelah deklarasi. Array biasanya digunakan dalam situasi di mana kita mengetahui jumlah elemen yang akan disimpan.

Struct

```
type [Nama_Struct] struct {  
    [Nama_Field] [Tipe_Data]  
    [Nama_Field2] [Tipe_Data2]  
}
```

Array

```
var [nama_array] = [ukuran_array][tipe_data]
```

II. GUIDED

1. Source Code

```
package main

import (
    "fmt"
    "sort"
)

type Mahasiswa struct {
    Nama      string
    Matematika int
    Fisika     int
    Kimia      int
    RataRata   float64
}

func hitungRataRata(m *Mahasiswa) {
    total := m.Matematika + m.Fisika + m.Kimia
    m.RataRata = float64(total) / 3
}

func main() {
    mahasiswa := []Mahasiswa{
        {"Aldi", 85, 90, 80, 0},
        {"Budi", 70, 75, 80, 0},
        {"Cici", 90, 85, 95, 0},
        {"Doni", 60, 65, 70, 0},
        {"Eka", 100, 95, 90, 0},
    }

    for i := range mahasiswa {
        hitungRataRata(&mahasiswa[i])
    }

    sort.Slice(mahasiswa, func(i, j int) bool {
        return mahasiswa[i].RataRata > mahasiswa[j].RataRata
    })

    fmt.Println("Peringkat mahasiswa berdasarkan rata - rata nilai: ")
    for i, m := range mahasiswa {
        fmt.Printf("%d. %s - Rata-rata: %.2f (Matematika: %d, Fisika: %d, Kimia: %d)\n", i+1, m.Nama, m.RataRata, m.Matematika, m.Fisika, m.Kimia)
    }
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> go run
\guided_1.go"
Peringkat mahasiswa berdasarkan rata - rata nilai:
1. Eka - Rata-rata: 95.00 (Matematika: 100, Fisika: 95, Kimia: 90)
2. Cici - Rata-rata: 90.00 (Matematika: 90, Fisika: 85, Kimia: 95)
3. Aldi - Rata-rata: 85.00 (Matematika: 85, Fisika: 90, Kimia: 80)
4. Budi - Rata-rata: 75.00 (Matematika: 70, Fisika: 75, Kimia: 80)
5. Doni - Rata-rata: 65.00 (Matematika: 60, Fisika: 65, Kimia: 70)
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> █
```

Penjelasan

Program ini menghitung rata-rata nilai Matematika, Fisika, dan Kimia setiap mahasiswa, lalu menampilkan peringkat berdasarkan nilai rata-rata tertinggi. Program dimulai dengan mendefinisikan struct Mahasiswa, yang memiliki atribut nama dan nilai setiap mata pelajaran dengan atribut RataRata untuk menyimpan rata-rata nilai. Fungsi hitungRataRata menerima pointer ke struct Mahasiswa, menghitung rata-rata dari total nilai Matematika, Fisika, dan Kimia, kemudian menyimpannya di RataRata. Di dalam main, data mahasiswa dimasukkan dalam slice mahasiswa. Setiap mahasiswa dihitung rata-ratanya, kemudian diurutkan berdasarkan RataRata secara desc menggunakan sort.Slice. Setelah itu, program mencetak peringkat mahasiswa dan rata-rata dan detail nilai tiap pelajaran.

2. Source code

```
package main

import "fmt"

func main() {
    mahasiswa := map[string]string{
        "20231001": "Andi",
        "20231002": "Budi",
        "20231003": "Cici",
    }

    mahasiswa["20231004"] = "Dedi"

    fmt.Println("Daftar Mahasiswa:")
    fmt.Println("NIM\t\tNama")
    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }

    nim := "20231002"
    fmt.Println("\nNama Mahasiswa dengan NIM", nim, "adalah",
mahasiswa[nim])

    delete(mahasiswa, "20231003")

    fmt.Println("\nDaftar Mahasiswa setelah dihapus:")
    fmt.Println("NIM\t\tNama")
    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }
}
```

Screenshoot

```

PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> go
\guided_2.go"
Daftar Mahasiswa:
NIM          Nama
-----
20231001     Andi
20231002     Budi
20231003     Cici
20231004     Dedi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM          Nama
-----
20231001     Andi
20231002     Budi
20231004     Dedi
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> 

```

Penjelasan :

Program diatas adalah program yang menggunakan map untuk menyimpan daftar mahasiswa dengan NIM sebagai key dan nama sebagai value. Program dimulai dengan membuat map mahasiswa, yang berisi beberapa entri yaitu NIM dan nama mahasiswa. pada awal program akan membuat entri baru, yaitu 20231004: "Dedi", ditambahkan ke map. Program kemudian mencetak daftar mahasiswa awal dengan header dan format NIM serta nama menggunakan for range untuk mengiterasi setiap elemen dalam map. Setelah itu, program mencari dan menampilkan nama mahasiswa berdasarkan NIM tertentu, yaitu 20231002. Kemudian, entri dengan NIM 20231003 dihapus menggunakan fungsi delete. Terakhir, program menampilkan kembali daftar mahasiswa setelah penghapusan, dengan format yang sama.

III. UNGUIDED

1. Source Code

```
package main

import (
    "fmt"
    "math"
)

type titik struct {
    x int
    y int
}

type lingkaran struct {
    pusat titik
    radius int
}

func jarak(p, q titik) float64 {
    return math.Pow(float64(p.x-q.x), 2) + math.Pow(float64(p.y-q.y),
2)
}

func didalam(c lingkaran, p titik) bool {
    var jarak_titik = math.Sqrt(jarak(c.pusat, p))
    if jarak_titik < float64(c.radius) {
        return true
    }
    return false
}
```

```

func main() {
    var lingkaran1, lingkaran2 lingkaran
    var titikSembarang titik
    var hasil1, hasil2 bool
    fmt.Print("Titik pusat (x,y) dan jejari lingkaran 1 : ")
    fmt.Scanln(&lingkaran1.pusat.x, &lingkaran1.pusat.y,
&lingkaran1.radius)
    fmt.Print("Titik pusat (x,y) dan jejari lingkaran 2 : ")
    fmt.Scanln(&lingkaran2.pusat.x, &lingkaran2.pusat.y,
&lingkaran2.radius)
    fmt.Print("Titik sembarang (x,y) : ")
    fmt.Scanln(&titikSembarang.x, titikSembarang.y)
    hasil1 = didalam(lingkaran1, titikSembarang)
    hasil2 = didalam(lingkaran2, titikSembarang)
    if hasil1 && hasil2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if hasil1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if hasil2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

Screenshot

```

PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> go
\1\unguided_1.go"
Titik pusat (x,y) dan jejari lingkaran 1 : 1 1 5
Titik pusat (x,y) dan jejari lingkaran 2 : 8 8 4
Titik sembarang (x,y) : 2 2
Titik di dalam lingkaran 1
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> go
\1\unguided_1.go"
Titik pusat (x,y) dan jejari lingkaran 1 : 1 2 3
Titik pusat (x,y) dan jejari lingkaran 2 : 4 5 6
Titik sembarang (x,y) : 7 8
Titik di dalam lingkaran 2
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> go
\1\unguided_1.go"
Titik pusat (x,y) dan jejari lingkaran 1 : 5 10 15
Titik pusat (x,y) dan jejari lingkaran 2 : -15 4 20
Titik sembarang (x,y) : 0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>

```


Penjelasan

Program diatas merupakan program untuk menentukan apakah suatu titik berada di dalam salah satu dari dua lingkaran atau di luar keduanya. Struct titik menyimpan koordinat (x, y), dan struct lingkaran menyimpan titik pusat dan radius lingkaran. Fungsi jarak menghitung jarak kuadrat antara dua titik, ketika fungsi didalam memeriksa apakah sebuah titik berada di dalam lingkaran dengan membandingkan jarak dari titik ke pusat lingkaran dengan radius lingkaran. Di dalam main, program meminta pengguna memasukkan koordinat dan radius untuk dua lingkaran dan koordinat titik yang akan diperiksa. Fungsi didalam digunakan untuk memeriksa keberadaan titik tersebut dalam masing-masing lingkaran. Berdasarkan hasilnya, program mencetak apakah titik berada di dalam lingkaran 1, lingkaran 2, di dalam keduanya, atau di luar keduanya.

2. Source Code

```
package main

import (
    "fmt"
    "math"
)

func genap(arr *[]int) {
    fmt.Print("[")
    for i := 0; i < len(*arr); i++ {
        if i%2 == 0 {
            fmt.Printf("%v, ", (*arr)[i])
        }
    }
    fmt.Print("]\n")
}

func ganjil(arr *[]int) {
    fmt.Print("[")
    for i := 0; i < len(*arr); i++ {
        if i%2 != 0 {
            fmt.Printf("%v, ", (*arr)[i])
        }
    }
    fmt.Print("]\n")
}

func kelipatan(arr *[]int) {
    var x int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scanln(&x)
    fmt.Print("[")
    for i := 0; i < len(*arr); i++ {
        if (*arr)[i]%x == 0 {
            fmt.Printf("%v, ", (*arr)[i])
        }
    }
    fmt.Print("]\n")
}

func hapus(arr *[]int) {
    var indeks int
    fmt.Print("Masukkan indeks: ")
    fmt.Scanln(&indeks)
    *arr = append((*arr)[:indeks], (*arr)[indeks+1:]...)
    fmt.Printf("%v\n", arr)
}

func rata2(arr *[]int) {
    var total int
    for i := 0; i < len(*arr); i++ {
        total += (*arr)[i]
    }
    fmt.Printf("Rata - rata: %v\n", total/len(*arr))
}
```

```

func simpangan_baku(arr *[]int) {
    var total, rerata, simpangan_baku float64
    for i := 0; i < len(*arr); i++ {
        total += float64((*arr)[i])
    }
    rerata = total / float64(len(*arr))
    for i := 0; i < len(*arr); i++ {
        simpangan_baku += math.Pow(float64((*arr)[i]) - rerata, 2)
    }
    simpangan_baku = math.Sqrt(simpangan_baku / float64(len(*arr)))
    fmt.Printf("Simpangan baku dari array adalah %v\n", simpangan_baku)
}

func frekuensi(arr *[]int) {
    var x, total int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scanln(&x)
    for i := 0; i < len(*arr); i++ {
        if (*arr)[i] == x {
            total++
        }
    }
    fmt.Printf("Frekuensi %v dalam array adalah %v\n", x, total)
}

func main() {
    var arrSize, menu int
    fmt.Print("Masukkan ukuran array: ")
    fmt.Scanln(&arrSize)
    arr := make([]int, arrSize)
    for i := 0; i < len(arr); i++ {
        fmt.Printf("Masukkan nilai elemen ke - %v: ", i+1)
        fmt.Scanln(&arr[i])
    }
    for true {
        fmt.Printf("\n[PROGRAM MENU]\n")
        fmt.Printf("1. Tampilkan semua elemen\n")
        fmt.Printf("2. Tampilkan elemen indeks ganjil\n")
        fmt.Printf("3. Tampilkan elemen indeks genap\n")
        fmt.Printf("4. Tampilkan elemen kelipatan x\n")
        fmt.Printf("5. Hapus elemen\n")
        fmt.Printf("6. Tampilkan rata - rata semua elemen\n")
        fmt.Printf("7. Tampilkan simpangan baku\n")
        fmt.Printf("8. Tampilkan frekuensi bilangan x\n")
        fmt.Print("Pilih menu: ")
        fmt.Scanln(&menu)
        switch menu {
            case 1:
                fmt.Printf("%v\n", arr)
                break
            case 2:
                ganjil(&arr)
                break
            case 3:
                genap(&arr)
                break
            case 4:
                kelipatan(&arr)
                break
            case 5:
                hapus(&arr)
                break
            case 6:
                rata2(&arr)
                break
            case 7:
                simpangan_baku(&arr)
                break
            case 8:
                frekuensi(&arr)
                break
            default:
                fmt.Println("Masukkan nomor yang sesuai dengan menu!")
                break
        }
    }
}

```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
ramman 2\7-121124\unguided\2\unguided_2.go"
Masukkan ukuran array: 5
Masukkan nilai elemen ke - 1: 10
Masukkan nilai elemen ke - 2: 12
Masukkan nilai elemen ke - 3: 14
Masukkan nilai elemen ke - 4: 17
Masukkan nilai elemen ke - 5: 18

[PROGRAM MENU]
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen kelipatan x
5. Hapus elemen
6. Tampilkan rata - rata semua elemen
7. Tampilkan simpangan baku
8. Tampilkan frekuensi bilangan x
Pilih menu: 1
[10 12 14 17 18]
```

```
[PROGRAM MENU]
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen kelipatan x
5. Hapus elemen
6. Tampilkan rata - rata semua elemen
7. Tampilkan simpangan baku
8. Tampilkan frekuensi bilangan x
Pilih menu: 2
[12, 17, ]
```

```
[PROGRAM MENU]
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen kelipatan x
5. Hapus elemen
6. Tampilkan rata - rata semua elemen
7. Tampilkan simpangan baku
8. Tampilkan frekuensi bilangan x
Pilih menu: 3
[10, 14, 18, ]
```

```
[PROGRAM MENU]
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen kelipatan x
5. Hapus elemen
6. Tampilkan rata - rata semua elemen
7. Tampilkan simpangan baku
8. Tampilkan frekuensi bilangan x
Pilih menu: 4
Masukkan nilai x: 2
[10, 12, 14, 18, ]
```

```
[PROGRAM MENU]
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen kelipatan x
5. Hapus elemen
6. Tampilkan rata - rata semua elemen
7. Tampilkan simpangan baku
8. Tampilkan frekuensi bilangan x
Pilih menu: 5
Masukkan indeks: 4
&[10 12 14 17]
```

```

[PROGRAM MENU]
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen kelipatan x
5. Hapus elemen
6. Tampilkan rata - rata semua elemen
7. Tampilkan simpangan baku
8. Tampilkan frekuensi bilangan x
Pilih menu: 6
Rata - rata: 13

[PROGRAM MENU]
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen kelipatan x
5. Hapus elemen
6. Tampilkan rata - rata semua elemen
7. Tampilkan simpangan baku
8. Tampilkan frekuensi bilangan x
Pilih menu: 7
Simpangan baku dari array adalah 2.5860201081971503

```

```

[PROGRAM MENU]
1. Tampilkan semua elemen
2. Tampilkan elemen indeks ganjil
3. Tampilkan elemen indeks genap
4. Tampilkan elemen kelipatan x
5. Hapus elemen
6. Tampilkan rata - rata semua elemen
7. Tampilkan simpangan baku
8. Tampilkan frekuensi bilangan x
Pilih menu: 8
Masukkan nilai x: 2
Frekuensi 2 dalam array adalah 0

```

Penjelasan

Program diatas merupakan program untuk mengelola dan menganalisis elemen-elemen dalam array berdasarkan input pengguna. Di awal, pengguna menentukan ukuran array dan memasukkan nilai setiap elemen. Program kemudian menampilkan menu dengan delapan opsi yaitu menampilkan elemen array, menampilkan elemen pada indeks ganjil/genap, menampilkan elemen yang merupakan kelipatan dari angka tertentu, menghapus elemen berdasarkan indeks, menghitung rata-rata semua elemen, menghitung simpangan baku, dan menghitung frekuensi kemunculan nilai tertentu. Setiap operasi memiliki fungsi khusus yang menerima array sebagai parameter pointer dan melakukan tugasnya sesuai menu yang dipilih. Program berulang menampilkan menu hingga pengguna menghentikannya, dengan setiap pilihan akan mengolah array.

3. Source Code

```
package main

import "fmt"

type pertandingan struct {
    klub1 string
    klub2 string
    hasil []string
}

func main() {
    var pertandingan pertandingan
    var skor_k1, skor_k2 int
    fmt.Print("Klub A: ")
    fmt.Scanln(&pertandingan.klub1)
    fmt.Print("Klub B: ")
    fmt.Scanln(&pertandingan.klub2)
    for i := 1; i > 0; i++ {
        fmt.Printf("Pertandingan %v: ", i)
        fmt.Scanln(&skor_k1, &skor_k2)
        if skor_k1 < 0 || skor_k2 < 0 {
            break
        } else if skor_k1 > skor_k2 {
            pertandingan.hasil = append(pertandingan.hasil,
pertandingan.klub1)
        } else if skor_k1 < skor_k2 {
            pertandingan.hasil = append(pertandingan.hasil,
pertandingan.klub2)
        } else {
            pertandingan.hasil = append(pertandingan.hasil, "Draw")
        }
    }
    for i := 0; i < len(pertandingan.hasil); i++ {
        fmt.Printf("Hasil %v: %v\n", i+1, pertandingan.hasil[i])
    }
    fmt.Println("Pertandingan selesai")
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> go run ramman 2\7-121124\unguided\3\unguided_3.go"
Klub A: PSG
Klub B: CITY
Pertandingan 1: 1 3
Pertandingan 2: 5 3
Pertandingan 3: 2 1
Pertandingan 4: 4 2
Pertandingan 5: 4 2
Pertandingan 6: 1 2
Pertandingan 7: 3 2
Pertandingan 8: 4 1
Pertandingan 9: 2 1
Pertandingan 10: 0 0
Pertandingan 11: -1 1
Hasil 1: CITY
Hasil 2: PSG
Hasil 3: PSG
Hasil 4: PSG
Hasil 5: PSG
Hasil 6: CITY
Hasil 7: PSG
Hasil 8: PSG
Hasil 9: PSG
Hasil 10: Draw
Pertandingan selesai
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas merupakan program untuk mencatat hasil pertandingan antara dua klub yang dimasukkan oleh pengguna. Struct pertandingan menyimpan nama dua klub (klub1 dan klub2) dan slice hasil untuk mencatat pemenang setiap pertandingan. Dalam fungsi main, pengguna pertama-tama menginput nama kedua klub. Program kemudian memasuki loop di mana pengguna menginput skor untuk setiap pertandingan. Jika skor salah satu klub negatif, loop berhenti, menandakan akhir dari rangkaian pertandingan. Berdasarkan skor, program menambahkan nama klub pemenang ke dalam slice hasil. namun jika skornya sama, "Draw" yang akan ditambahkan. Setelah loop selesai, program mencetak hasil dari setiap pertandingan yang tersimpan dalam slice hasil.

4. Source Code

```
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    fmt.Print("Jumlah karakter [k < 127]: ")
    fmt.Scanln(n)
    if *n > NMAX {
        fmt.Println("Jumlah karakter terlalu banyak!")
        isiArray(t, n)
    } else {
        fmt.Print("Teks: ")
        for i := 0; i < *n; i++ {
            fmt.Scanf("%c", &t[i])
        }
    }
}

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

func balikanArray(t *tabel, n int) {
    var temp tabel
    for i := 0; i < n; i++ {
        temp[i] = (*t)[n-1-i]
    }
    *t = temp
}

func main() {
    var tab tabel
    var m int
    isiArray(&tab, &m)
    balikanArray(&tab, m)
    fmt.Print("Reverse teks: ")
    cetakArray(tab, m)
}
```


Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
ramman 2\7-121124\unguided\4\unguided_4.go"
Jumlah karakter [k < 127]: 6
Teks: ASTAGA
Reverse teks: A G A T S A
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas merupakan program yang dapat membaca sekumpulan karakter dari pengguna, menyimpannya dalam array, kemudian mencetaknya dalam urutan terbalik. Array tabel dengan kapasitas maksimum NMAX digunakan untuk menyimpan karakter-karakter input. Fungsi isiArray meminta pengguna menginput jumlah karakter, memastikan jumlah tersebut tidak melebihi batas yang ditetapkan. Jika jumlah karakter valid, fungsi kemudian membaca karakter satu per satu dan menyimpannya dalam array t. Fungsi balikanArray menyalin karakter dari t ke array sementara temp dalam urutan terbalik, lalu menyalin hasilnya kembali ke t. Akhirnya, cetakArray menampilkan array terbalik ke layar sebagai output.

5. Source Code (UNGUIDED 4 MOD)

```
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    fmt.Print("Jumlah karakter [k < 127]: ")
    fmt.Scanln(n)
    if *n > NMAX {
        fmt.Println("Jumlah karakter terlalu banyak!")
    } else {
        fmt.Print("Teks: ")
        for i := 0; i < *n; i++ {
            fmt.Scanf("%c", &t[i])
        }
    }
}

func balikanArray(t *tabel, n int) {
    var temp tabel
    for i := 0; i < n; i++ {
        temp[i] = (*t)[n-1-i]
    }
    *t = temp
}

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 tabel
    var m int
    isiArray(&tab, &m)
    balikanArray(&tab, m)

    if palindrom(tab, m) {
        fmt.Println("Palindrom: true")
    } else {
        fmt.Println("Palindrom: false")
    }
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
ramman 2\7-121124\unguided\4-mod\unguided_4-mod.go"
Jumlah karakter [k < 127]: 5
Teks: GALIH
Palindrom: false
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
ramman 2\7-121124\unguided\4-mod\unguided_4-mod.go"
Jumlah karakter [k < 127]: 3
Teks: SOS
Palindrom: true
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas merupakan program yang sama dengan program sebelumnya. namun menambahkan fungsi palindrom yang memeriksa apakah teks sama ketika dibaca dari depan ke belakang dan sebaliknya. Jika karakter pertama sama dengan karakter terakhir, kedua sama dengan karakter kedua terakhir, dan seterusnya hingga tengah teks, maka teks dianggap palindrom dan program akan mengeluarkan hasil true. jika tidak, hasilnya adalah false.