

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK  
MODUL 7  
STRUCK & ARRAY**



Oleh:

MUHAMAD IHSAN

2311102077

IF - 11 – 02

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

## I. DASAR TEORI

### A. 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) 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 }
```

#### 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> <type data>	<field 1> <type data>
4	<field 2> <type data>	<field 2> <type data>
5	<field 3> <type data>	<field 3> <type 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 }

```

## B. 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]
```

## II. GUIDED

### Guided 1

```
package main

import (
    "fmt"
    "sort"
)

// Struktur untuk menampung data mahasiswa
type Mahasiswa struct {
    Nama      string
    Matematika int
    Fisika    int
    Kimia     int
    RataRata  float64
}

// Fungsi untuk menghitung rata-rata nilai tiap mahasiswa
func hitungRataRata(m *Mahasiswa) {
    total := m.Matematika + m.Fisika + m.Kimia
    m.RataRata = float64(total) / 3.0
}

// Fungsi utama untuk mengelola dan mengurutkan data mahasiswa
// berdasarkan nilai rata-rata
func main() {
    // Array untuk menampung data mahasiswa
    mahasiswa := []Mahasiswa{
        {"Ali", 85, 90, 80, 0},
        {"Budi", 70, 75, 80, 0},
        {"Cici", 90, 85, 95, 0},
        {"Doni", 60, 65, 70, 0},
        {"Eka", 100, 95, 90, 0},
    }

    // Menghitung rata-rata nilai tiap mahasiswa
    for i := range mahasiswa {
        hitungRataRata(&mahasiswa[i])
    }

    // Mengurutkan mahasiswa berdasarkan nilai rata-rata (descending)
    sort.Slice(mahasiswa, func(i, j int) bool {
        return mahasiswa[i].RataRata > mahasiswa[j].RataRata
    })
}
```

```

// Menampilkan hasil
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 output

```

PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Go1
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. Ali - 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\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>

```

## Deskripsi program

Program Go ini mengelola data mahasiswa dengan menyimpan nama dan nilai mata pelajaran Matematika, Fisika, dan Kimia, serta menghitung rata-rata nilainya. Struktur Mahasiswa memuat atribut untuk menyimpan data tersebut, dan fungsi hitungRataRata menghitung rata-rata nilai setiap mahasiswa. Program kemudian mengurutkan mahasiswa berdasarkan nilai rata-rata secara menurun menggunakan sort.Slice dan mencetak peringkat mereka ke layar dengan format yang mencantumkan nama, rata-rata nilai, serta nilai per mata pelajaran. Hasil akhirnya menampilkan daftar peringkat mahasiswa dari yang tertinggi ke terendah berdasarkan rata-rata nilai.

## Guided 2

```

package main

import "fmt"

func main() {
    // Membuat map dengan NIM sebagai kunci dan Nama sebagai nilai
    mahasiswa := map[string]string{
        "20231001": "Andi",
        "20231002": "Budi",
        "20231003": "Cici",
    }
}

```

```

}

// Menambahkan data baru ke map
mahasiswa["20231004"] = "Dedi"

// Menampilkan seluruh isi map dalam format kolom dan baris
fmt.Println("Daftar Mahasiswa:")
fmt.Println("NIM\t\tNama")
fmt.Println("-----")
for nim, nama := range mahasiswa {
    fmt.Printf("%s\t%s\n", nim, nama)
}

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

// Menghapus data berdasarkan NIM
delete(mahasiswa, "20231003")

// Menampilkan isi map setelah data dihapus dalam format kolom dan
baris
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)
}
}

```

## Screenshot output

```
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Gola
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\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
```

## Deskripsi program

Program Go ini menggunakan map untuk menyimpan data mahasiswa dengan NIM sebagai kunci dan nama sebagai nilai. Program menampilkan daftar mahasiswa, menambahkan data baru, mengakses nama berdasarkan NIM, dan menghapus data tertentu. Awalnya, map diisi beberapa data mahasiswa, lalu ditampilkan dalam format tabel. Data mahasiswa dengan NIM tertentu dapat diakses langsung, dan penghapusan dilakukan menggunakan fungsi delete. Setelah penghapusan, daftar mahasiswa diperbarui dan ditampilkan kembali. Program ini menunjukkan bagaimana cara menambahkan, mengakses, dan menghapus data dalam map di Go.



### III. UNGUIDED

#### Soal Latihan modul 7

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"
)

type Titik struct {
    x int
    y int
}

type lingkaran struct {
    center Titik
    radius int
}

func isTitikInlingkaran(p Titik, c lingkaran) bool {

    jarak := (p.x-c.center.x)*(p.x-c.center.x) + (p.y-c.center.y)*(p.y-
c.center.y)

    radius := c.radius * c.radius
    return jarak < radius
}
```

```

func main() {
    var lingkaran1 lingkaran
    var lingkaran2 lingkaran
    var titik Titik

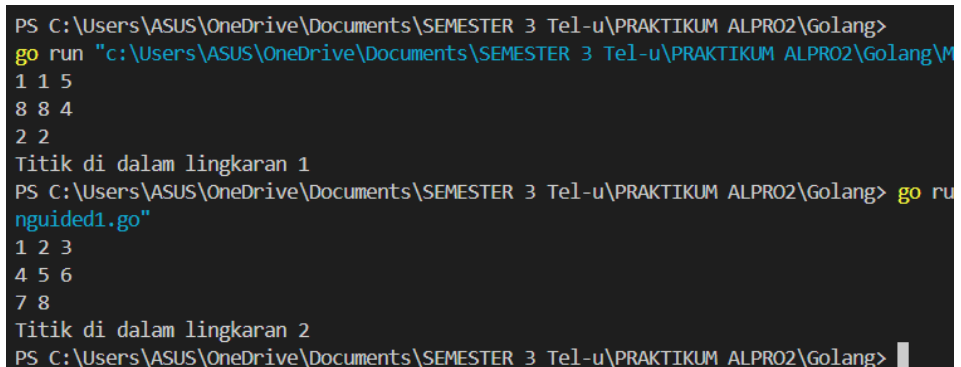
    fmt.Scan(&lingkaran1.center.x, &lingkaran1.center.y,
&lingkaran1.radius)
    fmt.Scan(&lingkaran2.center.x, &lingkaran2.center.y,
&lingkaran2.radius)
    fmt.Scan(&titik.x, &titik.y)

    inlingkaran1 := isTitikInlingkaran(titik, lingkaran1)
    inlingkaran2 := isTitikInlingkaran(titik, lingkaran2)

    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")
    }
}

```

## Screenshot output



```

PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang\Man
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang> go ru
nguided1.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>

```

## Deskripsi

Program di atas menentukan apakah sebuah titik berada di dalam satu atau dua lingkaran yang ditentukan. Struktur Titik digunakan untuk

menyimpan koordinat (x, y) titik, sedangkan struktur lingkaran menyimpan pusat dan jari-jari lingkaran. Fungsi `isTitikInlingkaran` menghitung jarak kuadrat antara titik dan pusat lingkaran, lalu membandingkannya dengan kuadrat jari-jari untuk mengecek apakah titik tersebut berada di dalam lingkaran. Program meminta input untuk koordinat dan jari-jari dua lingkaran serta satu titik, lalu mencetak hasil yang menunjukkan apakah titik tersebut berada di dalam lingkaran pertama, kedua, keduanya, atau di luar keduanya.

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

```
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.Printf("Elemen ke-%d: ", i)
        fmt.Scan(&array[i])
    }

    for {
        fmt.Println("\nMenu:")
        fmt.Println("1. Tampilkan seluruh isi array")
        fmt.Println("2. Tampilkan elemen dengan indeks ganjil")
        fmt.Println("3. Tampilkan elemen dengan indeks genap")
        fmt.Println("4. Tampilkan elemen dengan indeks kelipatan X")
        fmt.Println("5. Hapus elemen pada indeks tertentu")
        fmt.Println("6. Tampilkan rata-rata dari elemen array")
        fmt.Println("7. Tampilkan standar deviasi dari elemen array")
        fmt.Println("8. Keluar")
    }
}
```

```

fmt.Print("Pilih opsi: ")

var pilih int
fmt.Scan(&pilih)

switch pilih {
case 1:
    fmt.Println("Isi array:", array)

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

case 3:
    fmt.Println("Elemen dengan indeks genap:")
    for i := 0; i < len(array); i += 2 {
        fmt.Printf("Index %d: %d\n", i, array[i])
    }

case 4:
    var x int
    fmt.Print("Masukkan nilai X: ")
    fmt.Scan(&x)
    fmt.Printf("Elemen dengan indeks kelipatan %d:\n", x)
    for i := 0; i < len(array); i++ {
        if i%x == 0 {
            fmt.Printf("Index %d: %d\n", i, array[i])
        }
    }

case 5:
    var index int
    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&index)
    if index >= 0 && index < len(array) {
        array = append(array[:index], array[index+1:]...)
        fmt.Println("Array setelah penghapusan:", array)
    } else {
        fmt.Println("Indeks tidak valid.")
    }

case 6:
    sum := 0
    for _, val := range array {

```

```

        sum += val
    }
    rerata := float64(sum) / float64(len(array))
    fmt.Printf("Rata-rata elemen array: %.2f\n", rerata)

case 7:
    sum := 0
    for _, val := range array {
        sum += val
    }
    mean := float64(sum) / float64(len(array))

    variasiTambah := 0.0
    for _, val := range array {
        diff := float64(val) - mean
        variasiTambah += diff * diff
    }
    variasi := variasiTambah / float64(len(array))
    deviasi := math.Sqrt(variasi)
    fmt.Printf("Standar deviasi elemen array: %.2f\n", deviasi)

case 8:
    fmt.Println("Keluar dari program.")
    return

default:
    fmt.Println("Pilihan tidak valid.")
}
}
}

```

**Screenshot output**

```
e:\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang\Modul7
Masukkan jumlah elemen dalam array: 10
Masukkan elemen-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
Elemen ke-6: 7
Elemen ke-7: 8
Elemen ke-8: 9
Elemen ke-9: 0

Menu:
1. Tampilkan seluruh isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan X
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata dari elemen array
7. Tampilkan standar deviasi dari elemen array
8. Keluar
Pilih opsi: 2
Elemen dengan indeks ganjil:
Index 1: 2
Index 3: 4
Index 5: 6
Index 7: 8
Index 9: 0

Menu:
1. Tampilkan seluruh isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan X
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata dari elemen array
7. Tampilkan standar deviasi dari elemen array
8. Keluar
Pilih opsi: 3
Elemen dengan indeks genap:
Index 0: 1
Index 2: 3
Index 4: 5
Index 6: 7
Index 8: 9
```

```
Pilih opsi: 4
Masukkan nilai X: 6
Elemen dengan indeks kelipatan 6:
Index 0: 1
Index 6: 7

Menu:
1. Tampilkan seluruh isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan X
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata dari elemen array
7. Tampilkan standar deviasi dari elemen array
8. Keluar
Pilih opsi: 5
Masukkan indeks yang ingin dihapus: 7
Array setelah penghapusan: [1 2 3 4 5 6 7 9 0]

Menu:
1. Tampilkan seluruh isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan X
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata dari elemen array
7. Tampilkan standar deviasi dari elemen array
8. Keluar
Pilih opsi: 6
Rata-rata elemen array: 4.11

Menu:
1. Tampilkan seluruh isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan X
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata dari elemen array
7. Tampilkan standar deviasi dari elemen array
8. Keluar
Pilih opsi: 7
Standar deviasi elemen array: 2.77
```

## Deskripsi program

Program di atas memungkinkan pengguna untuk mengelola array dengan berbagai opsi. Setelah memasukkan jumlah dan elemen array, pengguna dapat memilih untuk menampilkan elemen array (seluruhnya, dengan indeks ganjil, genap, atau kelipatan tertentu), menghapus elemen pada indeks tertentu, serta menghitung rata-rata dan standar deviasi elemen array. Program menggunakan menu interaktif dengan perulangan for dan switch untuk menjalankan fungsi-fungsi tersebut hingga pengguna memilih untuk keluar.

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.

Source code

```
package main

import "fmt"

func skorBola(arr []string) {
    var skor1, skor2 int
    var i int = 0
    fmt.Print("masukan klub 1: ")
    fmt.Scan(&arr[0])
    fmt.Print("masukan klub 2: ")
    fmt.Scan(&arr[1])

    fmt.Println()

    for {
        i++
        fmt.Print("pertandingan ", i, ": ")
        fmt.Scan(&skor1, &skor2)

        if skor1 < 0 || skor2 < 0 {
            break
        }

        if skor1 > skor2 {
            fmt.Println("hasil ", i, ": ", arr[0])
        } else if skor2 > skor1 {
            fmt.Println("hasil ", i, ": ", arr[1])
        } else if skor1 == skor2 || skor2 == skor1 {
            fmt.Println("hasil ", i, ": Draw")
        }
    }
}

func main() {
    arr := make([]string, 2)
    skorBola(arr)
}
```

## Screenshot output

```

pertandingan 1: 3 0
hasil 1 : persib
pertandingan 2: 4 0
hasil 2 : persib
pertandingan 3: 2 0
hasil 3 : persib
pertandingan 4: 0 4
hasil 4 : MU
pertandingan 5: 3 1
hasil 5 : persib
pertandingan 6: 4 4
hasil 6 : Draw
pertandingan 7: 3 1
hasil 7 : persib
pertandingan 8: 3 3
hasil 8 : Draw
pertandingan 9: 2 0
hasil 9 : persib
pertandingan 10: 1- 3
hasil 10 : persib
pertandingan 11: -1 7
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\GoLa

```

## Deskripsi program

Program Go ini mencatat hasil pertandingan antara dua klub. Pengguna memasukkan nama kedua klub, lalu memasukkan skor pertandingan secara berulang. Program menampilkan pemenang untuk setiap pertandingan atau "Draw" jika hasilnya seri. Proses berlanjut hingga pengguna memasukkan skor negatif, yang menghentikan program

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"
)

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Scanf("%c", &t[i])
    }
}

```



```

}

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

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]
    }
}

func isPalindrome(t *tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t[i] != t[n-i-1] {
            return false
        }
    }
    return true
}

func main() {
    var t tabel
    var n int

    fmt.Print("Masukkan jumlah karakter (maksimal 127): ")
    fmt.Scan(&n)

    fmt.Println("Masukkan karakter:")
    var dummy byte
    fmt.Scanf("%c", &dummy)

    isiArray(&t, n)

    fmt.Print("Array yang dimasukkan: ")
    cetakArray(&t, n)

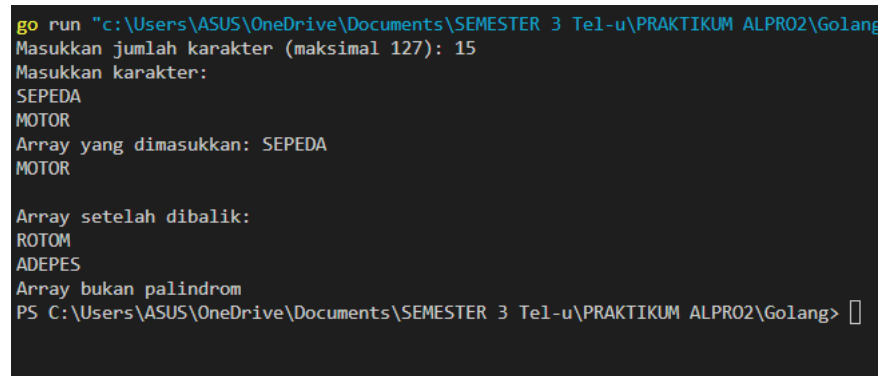
    balikanArray(&t, n)
    fmt.Print("Array setelah dibalik: ")
    cetakArray(&t, n)

    if isPalindrome(&t, n) {
        fmt.Println("Array adalah palindrom")
    }
}

```

```
    } else {  
        fmt.Println("Array bukan palindrom")  
    }  
}
```

### Screenshot output



```
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang  
Masukkan jumlah karakter (maksimal 127): 15  
Masukkan karakter:  
SEPEDA  
MOTOR  
Array yang dimasukkan: SEPEDA  
MOTOR  
  
Array setelah dibalik:  
ROTOM  
ADEPES  
Array bukan palindrom  
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang> █
```

### Deskripsi program

Program di atas memproses array karakter (maksimal 127 elemen) dengan beberapa fungsi. Pengguna diminta untuk memasukkan jumlah karakter dan menginput karakter-karakter tersebut. Fungsi `isiArray` mengisi array dengan input karakter, sedangkan `cetakArray` menampilkan array. Fungsi `balikanArray` membalik urutan elemen dalam array, dan `isPalindrome` memeriksa apakah array tersebut adalah palindrom (membaca sama dari depan dan belakang). Program menampilkan array asli, array yang telah dibalik, dan memberi tahu apakah array tersebut adalah palindrom.