

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 7  
ARRAY & STRUCT**



Dibuat Oleh:

M. AZKA HERMAWAN

2311102230

IF – 11- 02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM 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.

### B. 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 }
```

```
E:\DEV\GO>go build Demo.go
E:\DEV\GO> Demo.exe
1.8
```

### C. 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 yang 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 }

```

#### D. Array

Array mempunyai ukuran jumlah elemen) yang tetap (statls) 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]
```

### Slice (Array dinamik)

Array dalam Go juga dapat mempunyai ukuran yang dinamik. (Tidak digunakan di kelas Algoritma Pemrograman). Deklarasinya mirip dengan deklarasi array, tetapi jumlah elemennya dikosongkan.

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

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

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

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

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

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

### Map

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

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

## II. GUIDED

### 1. 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) / 3.0
}

func main() {
    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},
    }

    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 nilai
rata-rata: ")
    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

```

Peringkat mahasiswa berdasarkan nilai rata-rata:
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:\2311102230_M. Azka Hermawan_Modul 7>

```

### Deskripsi

Program ini dibuat untuk menghitung rata-rata nilai mahasiswa berdasarkan tiga mata pelajaran (matematika, fisika, dan kimia), kemudian mengurutkan mahasiswa berdasarkan nilai rata-rata dari yang tertinggi ke terendah, dan menampilkannya dengan urutan peringkat.

#### *Fungsi hitungRataRata*

Fungsi ini menerima pointer ke struktur Mahasiswa sebagai argumen. Fungsi ini kemudian menghitung rata-rata nilai dari ketiga mata kuliah dan menyimpan hasilnya ke dalam field rataRata dari struktur tersebut.

## 2. Guided 2

### Source Code

```
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

```
Daftar Mahasiswa:
NIM          Nama
-----
20231004     Dedi
20231001     Andi
20231002     Budi
20231003     Cici

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM          Nama
-----
20231004     Dedi
20231001     Andi
20231002     Budi
PS C:\2311102230_M. Azka Hermawan_Modul 7>
```

### Deskripsi

Program ini dibuat untuk mengelola data mahasiswa dalam bentuk map. Map adalah struktur data yang menyimpan pasangan kunci-nilai. Dalam kasus ini, kunci adalah NIM (Nomor Induk Mahasiswa) dan nilai adalah nama mahasiswa.

### III. UNGUIDED

#### 1. Unguided 1

##### Source Code

```
package main

import (
    "fmt"
    "math"
)

type lingkaran struct {
    titikX, titikY, cx, cy, r float64
}

func azka230(titikX, titikY float64, lingkaran1,
lingkaran2 lingkaran) string {
    jarakLingkaran1 := math.Sqrt(math.Pow(titikX-
lingkaran1.cx, 2) + math.Pow(titikY-lingkaran1.cy, 2))
    jarakLingkaran2 := math.Sqrt(math.Pow(titikX-
lingkaran2.cx, 2) + math.Pow(titikY-lingkaran2.cy, 2))

    if jarakLingkaran1 <= lingkaran1.r && jarakLingkaran2
<= lingkaran2.r {
        return "titik didalam lingkaran 1 dan 2"
    } else if jarakLingkaran1 <= lingkaran1.r {
        return "titik didalam lingkaran 1"
    } else if jarakLingkaran2 <= lingkaran2.r {
        return "titik didalam lingkaran 2"
    } else {
        return "titik diluar lingkaran 1 dan 2"
    }
}

func main() {
    var lingkaran1, lingkaran2 lingkaran

    fmt.Print("Masukkan pusat koordinat dan radius
lingkaran 1 (x y radius): ")
    fmt.Scan(&lingkaran1.cx, &lingkaran1.cy,
&lingkaran1.r)
```

```

        fmt.Print("Masukkan pusat koordinat dan radius
lingkaran 2 (x y radius): ")
        fmt.Scan(&lingkaran2.cx, &lingkaran2.cy,
&lingkaran2.r)

        fmt.Print("Masukkan titik koordinat (x y): ")
        fmt.Scan(&lingkaran1.titikX, &lingkaran1.titikY)

        hasil := azka230(lingkaran1.titikX, lingkaran1.titikY,
lingkaran1, lingkaran2)
        fmt.Println(hasil)
    }

```

### Screenshot

```

Masukkan pusat koordinat dan radius lingkaran 1 (x y radius): 1 1 5
Masukkan pusat koordinat dan radius lingkaran 2 (x y radius): 8 8 4
Masukkan titik koordinat (x y): 2 2
titik didalam lingkaran 1

```

### Deskripsi

Program ini berfungsi untuk menentukan posisi suatu titik terhadap dua buah lingkaran yang telah didefinisikan. Program ini menggunakan konsep jarak antara titik dengan pusat lingkaran untuk menentukan apakah titik tersebut berada di dalam, di luar, atau tepat pada lingkaran. *Func azka230*, berfungsi untuk menerima koordinat titik dan dua objek lingkaran, kemudian mengembalikan string yang mengindikasikan posisi titik terhadap kedua lingkaran.

## 2. Unguided 2

### Source Code

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

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

    for {
        fmt.Println("\nPilihan:")
        fmt.Println("a. Tampilkan seluruh elemen array")
        fmt.Println("b. Tampilkan elemen dengan indeks ganjil")
        fmt.Println("c. Tampilkan elemen dengan indeks genap")
        fmt.Println("d. Tampilkan elemen dengan indeks kelipatan bilangan x")
        fmt.Println("e. Hapus elemen pada indeks tertentu")
        fmt.Println("f. Tampilkan rata-rata bilangan dalam array")
        fmt.Println("g. Tampilkan simpangan baku bilangan dalam array")
        fmt.Println("h. Tampilkan frekuensi dari bilangan tertentu")
        fmt.Println("i. Keluar")

        var pilihan string
        fmt.Print("Pilih menu: ")
        fmt.Scan(&pilihan)

        switch pilihan {
            case "a":
                fmt.Println("Seluruh elemen array:", arr)

            case "b":
                fmt.Print("Elemen dengan indeks ganjil: ")
                for i := 1; i < N; i += 2 {
```

```

        fmt.Print(arr[i], " ")
    }
    fmt.Println()

case "c":
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 0; i < N; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

case "d":
    var x int
    fmt.Print("Masukkan bilangan x: ")
    fmt.Scan(&x)
    fmt.Printf("Elemen dengan indeks kelipatan %d:", x)

    for i := 0; i < N; i++ {
        if i%x == 0 {
            fmt.Print(arr[i], " ")
        }
    }
    fmt.Println()

case "e":
    var idx int
    fmt.Print("Masukkan indeks yang ingin dihapus:")

    fmt.Scan(&idx)
    if idx >= 0 && idx < N {
        arr = append(arr[:idx], arr[idx+1:]...)
        N--
        fmt.Println("Array setelah penghapusan:", arr)
    } else {
        fmt.Println("Indeks tidak valid!")
    }

case "f":
    total := 0
    for _, val := range arr {
        total += val
    }
    rataRata := float64(total) / float64(N)

```

```

        fmt.Println("Rata-rata nilai array:",
rataRata)

    case "g":
        total := 0
        for _, val := range arr {
            total += val
        }
        rataRata := float64(total) / float64(N)

        var sumSquare float64
        for _, val := range arr {
            sumSquare += (float64(val) - rataRata) *
(float64(val) - rataRata)
        }
        simpanganBaku := sqrt(sumSquare / float64(N))
        fmt.Println("Simpangan baku dari array:",
simpanganBaku)

    case "h":
        var target int
        fmt.Print("Masukkan bilangan yang ingin
dihitung frekuensinya: ")
        fmt.Scan(&target)
        frekuensi := 0
        for _, val := range arr {
            if val == target {
                frekuensi++
            }
        }
        fmt.Printf("Frekuensi bilangan %d dalam array:
%d\n", target, frekuensi)

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

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

```

## Screenshot

```
Masukkan jumlah elemen dalam array: 2
Masukkan elemen ke-0: 1
Masukkan elemen ke-1: 2

Pilih list yang ingin anda cari:
1. Tampilkan seluruh elemen array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata bilangan dalam array
7. Tampilkan simpangan baku bilangan dalam array
8. Tampilkan frekuensi dari bilangan tertentu
9. Keluar
Pilih menu: 1
Seluruh elemen array: [1 2]

Pilih list yang ingin anda cari:
1. Tampilkan seluruh elemen array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata bilangan dalam array
7. Tampilkan simpangan baku bilangan dalam array
8. Tampilkan frekuensi dari bilangan tertentu
9. Keluar
Pilih menu: 7
Simpangan baku dari array: 0.5
```

## Deskripsi

Program ini berfungsi untuk membantu user untuk melakukan berbagai operasi pada sebuah array (daftar) angka.

Dalam program tersebut saya menggunakan switch case untuk membandingkan nilai dari variabel pil230 (yang menyimpan pilihan pengguna) dengan berbagai kasus yang telah ditentukan. Jika ada kecocokan, maka blok kode di dalam kasus tersebut akan dijalankan.

### 3. Unguided 3

#### Source Code

```
package main

import "fmt"

func inputNamaKlub() (string, string) {
    var klubA, klubB string
    fmt.Print("Klub A: ")
    fmt.Scanln(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scanln(&klubB)
    return klubA, klubB
}

func inputSkor(pertandingan int, klubA, klubB string)
(int, int, bool) {
    var skorA, skorB int
    fmt.Printf("Pertandingan %d: ", pertandingan)
    fmt.Scanln(&skorA, &skorB)

    if skorA < 0 || skorB < 0 {
        return skorA, skorB, false
    }
    fmt.Printf("// %s %d sedangkan %s %d\n", klubA, skorA,
klubB, skorB)
    return skorA, skorB, true
}

func tentukanPemenang(skorA, skorB int, klubA, klubB
string) string {
    if skorA > skorB {
        return klubA
    } else if skorB > skorA {
        return klubB
    }
    return "Seri"
}

func tampilkanHasil(pemenang []string) {
    fmt.Println("Hasil pertandingan:")
    for i, hasil := range pemenang {
        fmt.Printf("Hasil %d: %s\n", i+1, hasil)
    }
}
```



```
    }  
    fmt.Println("Pertandingan selesai")  
}  
  
func main() {  
    klubA, klubB := inputNamaKlub()  
  
    var pemenang []string  
    var pertandingan int = 1  
  
    for {  
        skorA, skorB, valid := inputSkor(pertandingan,  
klubA, klubB)  
  
        if !valid {  
            fmt.Println("Skor tidak valid")  
            break  
        }  
  
        hasil := tentukanPemenang(skorA, skorB, klubA,  
klubB)  
        pemenang = append(pemenang, hasil)  
        pertandingan++  
  
        if pertandingan > 12 {  
            break  
        }  
    }  
  
    tampilkanHasil(pemenang)  
}
```

## Screenshot

```
Klub A: Realmadrid
Klub B: Barcelona
Pertandingan 1: 1 4
// Realmadrid 1 sedangkan Barcelona 4
Pertandingan 2: 4 1
// Realmadrid 4 sedangkan Barcelona 1
Pertandingan 3: 2 3
// Realmadrid 2 sedangkan Barcelona 3
Pertandingan 4: 3 2
// Realmadrid 3 sedangkan Barcelona 2
Pertandingan 5: 5 4
// Realmadrid 5 sedangkan Barcelona 4
Pertandingan 6: 4 5
// Realmadrid 4 sedangkan Barcelona 5
Pertandingan 7: 1 2
// Realmadrid 1 sedangkan Barcelona 2
Pertandingan 8: 2 1
// Realmadrid 2 sedangkan Barcelona 1
Pertandingan 9: 1 3
// Realmadrid 1 sedangkan Barcelona 3
Pertandingan 10: 3 1
// Realmadrid 3 sedangkan Barcelona 1
Pertandingan 11: 2 4
// Realmadrid 2 sedangkan Barcelona 4
Pertandingan 12: 4 2
// Realmadrid 4 sedangkan Barcelona 2
Hasil pertandingan:
Hasil 1: Barcelona
Hasil 2: Realmadrid
Hasil 3: Barcelona
Hasil 4: Realmadrid
Hasil 5: Realmadrid
Hasil 6: Barcelona
Hasil 7: Barcelona
Hasil 8: Realmadrid
Hasil 9: Barcelona
Hasil 10: Realmadrid
Hasil 11: Barcelona
Hasil 12: Realmadrid
Pertandingan selesai
PS C:\2311102230_M. Azka Hermawan_Modul 7> |
```

## Deskripsi

Program ini dibuat untuk mencatat hasil pertandingan antara dua klub (klubA dan klubB). Program akan meminta input nama kedua klub, kemudian secara berulang meminta input skor untuk setiap pertandingan. Setelah semua data pertandingan dimasukkan, program akan menampilkan hasil akhir dari setiap pertandingan.

*Func inputNamaKlub*, berfungsi untuk meminta input nama dua klub sepak bola, yaitu Klub A dan Klub B.

*Func inputSkor*, berfungsi untuk meminta input skor pertandingan untuk setiap pertandingan. Skor untuk Klub A dan Klub B diminta, dan jika ada skor yang tidak valid (misalnya, nilai negatif), fungsi akan

mengembalikan false untuk menunjukkan kesalahan. Jika skor valid, maka skor dan status valid (true) dikembalikan.

*Func tentukanPemenang*, berfungsi untuk membandingkan skor antara Klub A dan Klub B. Jika skor Klub A lebih tinggi, Klub A dianggap menang. Jika skor Klub B lebih tinggi, Klub B yang menang. Jika skor kedua klub sama, hasilnya "Seri".

*Func tampilkanHasil*, berfungsi untuk menerima daftar hasil pertandingan (pemenang) dan menampilkan hasil setiap pertandingan satu per satu, diikuti dengan informasi bahwa pertandingan telah selesai.

#### 4. Unguided 4

##### Source Code

```
package main

import "fmt"

func main() {
    var tab tabel
    var m int

    isiArray(&tab, &m)
    cetakArray(tab, m)
}

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var char rune
    *n = 0
    fmt.Print("Masukkan karakter (gunakan .) untuk berhenti): ")

    for *n < NMAX {
        fmt.Scanf("%c", &char)
        if char == '.' {
            break
        }

        if char != '\n' {
            t[*n] = char
            *n++
        }
    }
}

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

```

}

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 cetakArray(t tabel, n int) {
    var balik tabel = balikanArray(t, n)
    isPalindrome := palindrom(t, n)

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

    fmt.Print("Reverse teks\t:")
    for i := 0; i < n; i++ {
        fmt.Printf(" %c", balik[i])
    }
    fmt.Println()

    fmt.Print("Palindrom\t? ")
    if isPalindrome {
        fmt.Print("true")
    } else {
        fmt.Print("false")
    }
}

```

### Screenshot

```

Masukkan karakter (gunakan .) untuk berhenti): AZKA.
Teks           : A Z K A
Reverse teks   : A K Z A
Palindrom      ? false

```

## **Deskripsi**

Program ini dibuat untuk memeriksa apakah sebuah kata atau kalimat yang dimasukkan oleh pengguna merupakan palindrom atau bukan. Palindrom adalah kata atau kalimat yang jika dibaca dari depan maupun belakang tetap sama, contohnya "katak", "kasur rusak".

*func isiArray*, digunakan untuk membaca karakter demi karakter dari input pengguna hingga pengguna menekan titik('.') atau batas maksimum karakter tercapai.

*Fungsi balikanArray*, membalik urutan karakter dalam array tabel dan menyimpannya dalam array baru.

*Fungsi palindrom*, membandingkan karakter-karakter pada indeks yang sama dari awal dan akhir array. Jika semua karakter sama, maka kata tersebut adalah palindrom.