

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCT & ARRAY**



Disusun Oleh :

Hamzah Ziyad Ibadurrohman / 2311102254

IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

Tipe Bentukan

Tipe bentukan memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipe bentukan 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 }
```

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

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> <tiipe data>	<field 1> <tiipe data>
4	<field 2> <tiipe data>	<field 2> <tiipe data>
5	<field 3> <tiipe data>	<field 3> <tiipe 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 }

```

```

E:\DEV\GO>go build Demo.go
E:\DEV\GO> Demo.exe
7 30 0
10 45 15
Lama parkir: 3 jam 15 menit 15 detik

```

7.2 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]
```

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.

```

1 /* Append elemen baru, membuat slice baru, dan menyimpan kembali slice baru
   ke variabel semula. Boleh juga disimpan ke variabel lain, sehingga variabel
   semula masih menyimpan slice yang asli. */
2 sl01 = append(sl01, 17)
3 sl01 = append(sl01, 19, 23)

```

Sebuah slice baru juga dapat terbentuk dengan mengambil slice dari suatu array atau slice yang lain.

```

1 // Ambil 3 elemen pertama dari suatu slice atau array
2 sl04 = arr[:4]
3
4 // Ambil beberapa elemen terakhir, dimulai dari indeks 5
5 sl05 = sl01[5:]
6
7 // Salin semua dari slice/array aslinya
8 sl06 = sl05[:]
9
10 // Salin element dari indeks 3 sampai, tapi tidak termasuk, 5.
11 // Jadi dalam contoh hanya 2 elemen sl06[3] dan sl06[4] yang disalin
12 sl07 = sl06[3:5]

```

II. GUIDED

1. Soal Studi Case

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".

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// menyimpan titik dengan koordinat (x,y)
type titik struct {
    x int
    y int
}

// menyimpan lingkaran dengan pusat dan radius
type lingkaran struct {
    pusat titik
    radius int
}

// fungsi untuk menghitung jarak antara dua titik
func hitungjarak(a, b titik) float64 {
    return math.Sqrt(float64((a.x - b.x) + (a.y - b.y) * (a.y - b.y)))
}

// fungsi untuk memeriksa apakah titik berada di dalam lingkaran
func titikDiDalamLingkaran(t titik, l lingkaran) bool {
    jarak := hitungjarak(t, l.pusat)
    return jarak <= float64(l.radius)
}
```

```

func main() {
    // input untuk lingkaran 1
    var cx1, cy1, r1 int
    fmt.Print(" Masukkan koordinat pusat dan radius
lingkaran 1 (cx1, cy2, r1): ")
    fmt.Scanln(&cx1, &cy1, &r1)
    lingkaran1 := lingkaran{pusat: titik{x: cx1, y:
cy1}, radius: r1}

    //input untuk lingkaran 2
    var cx2, cy2, r2 int
    fmt.Print("Masukkan koordinat pusat dan radius
lingkaran 2 (cx2, cy2, r2): ")
    fmt.Scanln(&cx2, &cy2, &r2)
    lingkaran2 := lingkaran{pusat: titik{x: cx2, y:
cy2}, radius: r2}

    //input untuk titik sembarang
    var x, y int
    fmt.Print("Masukkan koordinat titik sembarang(x y):
")
    fmt.Scanln(&x, &y)
    titik := titik{x: x, y: y}

    // pengecekan posisi titik
    diDalam1 := titikDiDalamLingkaran(titik, lingkaran1)
    diDalam2 := titikDiDalamLingkaran(titik, lingkaran2)

    //menampilkan hasil sesuai kondisi
    if diDalam1 && diDalam2 {
        fmt.Println("Titik di dalam lingkaran 1 dan
2")
    } else if diDalam1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diDalam2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik berada di luar lingkaran")
    }
}

```

Screenshoot Output

```

PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\Pertemuan 6(modul 7)> go run guided1.go
Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy2, r1): 34 65 19
Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): 23 65 28
Masukkan koordinat titik sembarang(x y): 75 23
Titik berada di luar lingkaran
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\Pertemuan 6(modul 7)>

```

Deskripsi Program

Kode golang ini dibuat untuk menentukan posisi sebuah titik sembarang dalam dua lingkaran, Program ini menggunakan struct untuk menyimpan koordinat (x, y) dan lingkaran yang memiliki pusat berupa titik serta jari-jari (radius). Dalam program ini terdapat Fungsi hitungjarak yang digunakan untuk menghitung jarak Euclidean antara dua titik, sementara fungsi titikDiDalamLingkaran adalah fungsi yang menentukan apakah sebuah titik terletak di dalam lingkaran dengan menghitung jarak antara titik tersebut dan pusat lingkaran, lalu membandingkan jarak ini dengan jari-jari lingkaran.

Lalu, dalam fungsi main, Pengguna diminta untuk menginput koordinat pusat dan radius dari dua lingkaran (lingkaran1 dan lingkaran2) serta untuk titik sembarang. Kemudian program memeriksa apakah titik berada di dalam salah satu atau kedua lingkaran menggunakan fungsi titikDiDalamLingkaran. Jika titik berada di dalam kedua lingkaran, program menampilkan pesan bahwa titik tersebut berada di dalam kedua lingkaran. Jika hanya berada di dalam satu lingkaran, pesan yang relevan ditampilkan, dan jika tidak berada di dalam salah satu lingkaran, pesan lain yang menyatakan bahwa titik berada di luar ditampilkan.

III. UNGUIDED

1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

Soal Studi Case

Sebuah array digunakan untuk menampung sekumpulan bilangan bulat. Buatlah program golang yang digunakan untuk mengisi array tersebut sebanyak N elemen nilai. Asumsikan array memiliki kapasitas penyimpanan data sejumlah elemen tertentu. Program dapat menampilkan beberapa informasi berikut:

- a. Menampilkan keseluruhan isi dari array.
- b. Menampilkan elemen-elemen array dengan indeks ganjil saja.
- c. Menampilkan elemen-elemen array dengan indeks genap saja (asumsi indeks 0 adalah genap)
- d. Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x. x bisa diperoleh dari masukan pengguna.

- e. Menghapus elemen array pada indeks tertentu, asumsi indeks yang hapus selalu valid. Tampilkan keseluruhan isi dari arraynya, pastikan data yang dihapus tidak tampil.
- f. Menampilkan rata-rata dari bilangan yang ada di dalam array.
- g. Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut.
- h. Menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi tersebut.

Sourcecode

```
package main

package main

import (
    "fmt"
    "math"
)

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

    // Deklarasi array dengan kapasitas sesuai input N
    arr := make([]int, n)

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

    // a. Menampilkan keseluruhan isi dari array
    fmt.Println("Isi array:", arr)

    // b. Menampilkan elemen-elemen array dengan indeks ganjil saja
    fmt.Println("Elemen dengan indeks ganjil:")
    for i := 1; i < n; i += 2 {
        fmt.Printf("arr[%d] = %d\n", i, arr[i])
    }

    // c. Menampilkan elemen-elemen array dengan indeks genap
    fmt.Println("Elemen dengan indeks genap:")
    for i := 0; i < n; i += 2 {
        fmt.Printf("arr[%d] = %d\n", i, arr[i])
    }
}
```

```

        // d. Menampilkan elemen-elemen array dengan indeks
        kelipatan bilangan x
        var x int
        fmt.Print("Masukkan nilai x untuk menampilkan elemen
        dengan indeks kelipatan x: ")
        fmt.Scan(&x)
        fmt.Printf("Elemen dengan indeks kelipatan %d:\n",
        x)
        for i := 0; i < n; i++ {
            if i%x == 0 {
                fmt.Printf("arr[%d] = %d\n", i, arr[i])
            }
        }

        // e. Menghapus elemen array pada indeks tertentu
        var delIndx int
        fmt.Print("Masukkan indeks yang ingin dihapus: ")
        fmt.Scan(&delIndx)
        if delIndx >= 0 && delIndx < n {
            arr = append(arr[:delIndx],
        arr[delIndx+1:]...)
            fmt.Println("Isi array setelah penghapusan:",
        arr)
        } else {
            fmt.Println("Indeks tidak valid.")
        }

        // f. Menampilkan rata-rata dari bilangan yang ada
        di dalam array
        sum := 0
        for _, val := range arr {
            sum += val
        }
        ratarata := float64(sum) / float64(len(arr))
        fmt.Printf("Rata-rata dari elemen dalam array:
        %.2f\n", ratarata)

        // g. Menampilkan standar deviasi atau simpangan
        baku dari bilangan dalam array
        var varianceSum float64
        for _, val := range arr {
            varianceSum += math.Pow(float64(val)-ratarata,
        2)
        }
        stdDev := math.Sqrt(varianceSum / float64(len(arr)))
        fmt.Printf("Simpangan baku dari elemen dalam array:
        %.2f\n", stdDev)

        // h. Menampilkan frekuensi dari suatu bilangan
        tertentu di dalam array
        var target int

```

```

    fmt.Print("Masukkan nilai untuk menghitung
frekuensinya dalam array: ")
    fmt.Scan(&target)
    frekuensi := 0
    for _, val := range arr {
        if val == target {
            frekuensi++
        }
    }
    fmt.Printf("Frekuensi %d di dalam array: %d\n",
target, frekuensi)
}

```

Screenshoot Output

```

PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> go run unguided1.go
masukkan nilai suku nya: 8
Hasilnya adalah: 21
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4>

```

Deskripsi Program

Kode golang ini dibuat untuk mengisi array tersebut sebanyak N elemen nilai. Asumsikan array memiliki kapasitas penyimpanan data sejumlah elemen tertentu. Pertama, program meminta input dari pengguna untuk menentukan jumlah elemen array, yang kemudian digunakan untuk membuat array arr dengan kapasitas yang sesuai. Setelah itu, elemen-elemen array diisi berdasarkan input pengguna dalam loop for. Program selanjutnya menampilkan semua elemen array dan memberikan fitur untuk menampilkan elemen-elemen dengan indeks ganjil dan genap melalui loop yang terpisah.

Selain itu, program juga dapat menampilkan elemen-elemen array berdasarkan indeks kelipatan bilangan x yang dimasukkan pengguna. Untuk manipulasi array lebih lanjut, pengguna dapat menghapus elemen pada indeks tertentu, dan array akan diperbarui jika indeks tersebut valid. Dari segi analisis data, program menghitung rata-rata dari semua elemen dalam array dengan membagi jumlah elemen dengan panjang array. Selanjutnya, simpangan baku atau standar deviasi dihitung untuk menunjukkan seberapa tersebar nilai-nilai di dalam array dari nilai rata-rata. Terakhir, program memiliki fitur untuk menghitung dan menampilkan frekuensi kemunculan suatu bilangan tertentu dalam array, berdasarkan input dari pengguna. Program ini menggunakan operasi dasar seperti pengisian, penampilan, penghapusan elemen, serta perhitungan statistik untuk menunjukkan fleksibilitas dalam pengolahan data array di Go.

2. Soal Studi Case

Sebuah program digunakan untuk menyimpan dan menampilkan nama-nama klub yang memenangkan pertandingan bola pada suatu grup pertandingan. Buatlah program golang yang digunakan untuk merekap skor pertandingan bola 2 buah klub bola yang berlaga.

Pertama-tama program meminta masukan nama-nama klub yang bertanding, kemudian program meminta masukan skor hasil pertandingan kedua klub tersebut. Yang disimpan dalam array adalah nama-nama klub yang menang saja. Proses input skor berhenti ketika skor salah satu atau kedua klub tidak valid (negatif). Di akhir program, tampilkan daftar klub yang memenangkan pertandingan.

Perhatikan sesi interaksi pada contoh berikut ini (teks bergaris bawah adalah input/read).

Sourcecode

```
package main

import "fmt"

// struct untuk menyimpan informasi klub
type Klub struct {
    nama string
    skor int
}

func main() {
    var klubA, klubB Klub
    var pemenang []string

    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&klubA.nama)
    fmt.Print("Masukkan nama Klub B: ")
```

```

        fmt.Scanln(&klubB.nama)

        for {

            fmt.Printf("\nMasukkan skor %s: ", klubA.nama)

            fmt.Scan(&klubA.skor)

            fmt.Printf("Masukkan skor %s: ", klubB.nama)

            fmt.Scan(&klubB.skor)

            // Hentikan input jika salah satu atau kedua skor
            tidak valid (negatif)

            if klubA.skor < 0 || klubB.skor < 0 {

                fmt.Println("\nSkor tidak valid. Proses input
                dihentikan.")

                break

            }

            // Tentukan pemenang dan simpan nama klub yang
            menang

            if klubA.skor > klubB.skor {

                fmt.Printf("%s memenangkan pertandingan\n",
                klubA.nama)

                pemenang = append(pemenang, klubA.nama)

            } else if klubB.skor > klubA.skor {

                fmt.Printf("%s memenangkan pertandingan\n",
                klubB.nama)

                pemenang = append(pemenang, klubB.nama)

            } else {

                fmt.Println("Pertandingan berakhir seri")

            }

        }

    }

```

```

        // Tampilkan daftar klub pemenang

        fmt.Println("\nDaftar klub yang memenangkan
pertandingan:")

        for i, klub := range pemenang {

            fmt.Printf("Pemenang %d: %s\n", i+1, klub)

        }

    }
}

```

Screenshot output

```

PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\Pertemuan 6(modul 7)> go run unguided2.go
Masukkan nama Klub A: MU
Masukkan nama Klub B: chelsea

Masukkan skor MU: 4
Masukkan skor chelsea: 2
MU memenangkan pertandingan

Masukkan skor MU: 5
Masukkan skor chelsea: 6
chelsea memenangkan pertandingan

Masukkan skor MU: 0
Masukkan skor chelsea: -1

Skor tidak valid. Proses input dihentikan.

Daftar klub yang memenangkan pertandingan:
Pemenang 1: MU
Pemenang 2: chelsea
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\Pertemuan 6(modul 7)>

```

Deskripsi Output

Ini adalah Program golang yang dibuat untuk menyimpan dan menampilkan dan menampilkan nama-nama klub yang memenangkan pertandingan bola pada suatu grup pertandingan. Koding ini memakai array untuk menyimpan nama klub dan skor pertandingan. Sekaligus memakai struct agar mengelompokkannya. Pertama program meminta untuk diinput nama 2 klub untuk disimpan di variabel klub a dan klub b. Kemudian program menggunakan loop tidak terbatas untuk meminta pengguna skor klub tersebut. Jika ada skor yang negatif maka loop berhenti. Dalam iterasi tersebut setelah skor diinput program akan membandingkan skor mana yang lebih tinggi. Jika sudah maka program akan menampilkan pemenang dari pertandingan tersebut. Jika skor itu sama, maka pertandingannya seri. Setelah loop berhenti, program akan menampilkan tiap pemenang di setiap pertandingan ke layar.

3. Soal Studi Case

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

modifikasi program tersebut dengan menambahkan fungsi palindrom, tambahkan intruksi untuk memanggil fungsi tersebut dan menampilkan hasilnya pada program utama.

Sourcecode

```
package main

import (
    "fmt"
)

const NMAX int = 127
type tabel [NMAX]rune

// Fungsi untuk mengisi array dengan karakter
func isiArray(t *tabel, n *int) {
    fmt.Println("Masukkan karakter (akhiri dengan titik '.'): ")
    var input rune
    *n = 0
    for *n < NMAX {
        fmt.Scanf("%c", &input)
        if input == '.' {
            break
        }
        *n++
    }
}
```

```

        }

        (*t)[*n] = input

        *n++

    }

}

// Fungsi untuk mencetak isi array
func cetakArray(t tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Printf("%c", t[i])
    }

    fmt.Println()
}

// Fungsi untuk membalikkan isi array
func balikanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        t[i], t[n-i-1] = t[n-i-1], t[i]
    }
}

// Fungsi untuk memeriksa apakah array membentuk
palindrom
func palindrom(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 tab tabel

    var m int

    isiArray(&tab, &m)

    fmt.Print("Teks: ")

    cetakArray(tab, m)

    balikanArray(&tab, m)

    fmt.Print("Reverse teks: ")

    cetakArray(tab, m)

    if palindrom(tab, m) {

        fmt.Println("Palindrom? benar")

    } else {

        fmt.Println("Palindrom? salah")

    }

}

```

Screenshot Output

```

PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\Pertemuan 6(modul 7)> go run unguided3.go
Masukkan karakter (akhiri dengan titik '.'):
katak.
Teks: katak
Reverse teks: katak
Palindrom? true
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\Pertemuan 6(modul 7)>

```

Deskripsi Program

Ini adalah program yang dibuat untuk mengecek kata dalam array bahwa apakah kata ini merupakan palindrom atau tidak. Cara kerjanya ialah array nya nanti akan dibalik dan diperiksa apakah itu palindrom atau tidak. Program ini memakai konstanta nmax untuk ukuran maksimum array dan tipe tabel yang merepresentasikan array karakter (rune). Pengguna memasukkan karakter hingga karakter titik (.) ditemukan atau hingga batas maksimum tercapai. Program menyediakan fungsi untuk mengisi array, mencetak isinya, membalikkan susunan elemen array, dan memeriksa apakah array tersebut membentuk palindrom. Dan ketika pemeriksaan terjadi prosesnya adalah elemen dari tiap ujung array akan dibandingkan hingga tengah array. Jika ini adalah palindrom, program akan menampilkan bahwa palindrom? Benar. Dan juga sebaliknya.