

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7
STRUCK & ARRAY**



Disusun Oleh:

NAMA:IKRAM IRIANSYAH

NIM : 2311102184

KELAS : 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 dengan nama baru yang lebih ringkas dan familiar. Sebagai contoh “**integer**” dapat diubah 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>	type <nama alias> <tipe data>
3		
4	algoritma	func main(){
5
6		}
7		

Sebagai contoh :

```
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 }
```

Apabila diperhatikan subprogram **cetak()** di atas, terlihat pada baris ke-4 terdapat pemanggilan subprogram **cetak()** kembali. Misalnya apabila dieksekusi perintah **cetak()** maka akan menampilkan angka 5 6 7 8 9..dst.

B. Struct atau Record

Structure memungkinkan pemrogram untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi satu kesatuan. Masing-masing nilai tersimpan dalam fields dari structure tersebut.

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama struct> <	type <nama struct> struct {
3	<field 1> <tipe data>	<field 1> <tipe data>
4	<field 2> <tipe data>	<field 2> <tipe data>
5	<field 3> <tipe data>	<field 3> <tipe data>
6	>	}
7		

Beberapa dengan bahasa pemrograman lain, kesamaan tipe dari dua variabel berjenis structure bukan karena Namanya tetapi karena stukturanya. 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 structure tersebut didefinisikan sebagai sebuah tipe baru, sehingga structure 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 }

```

1. 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 **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]
```

2. Slice (Array dinamik)

Array dalam Go juga dapat mempunyai ukuran yang dinamik. (**Tidak digunakan di kelas Algoritma Pemograman**). 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 bild-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 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.

3. Map

Tipe array lain, sebuah array dinamik, indeksinya (di sini disebut **kunci**) dan 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"

```

```

18
19 // Menghapus entri dengan kunci "john"
20 delete(dct1, "john")

```

II. GUIDED

1. SOURCE CODE

```
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)  
    }  
}
```

OUTPUT

```
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7\GUIDED\guided1.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. 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\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7>
```

DESKRIPSI PROGRAM

Program diatas merupakan program yang digunakan untuk mengelola data mahasiswa dan menghitung peringkat mahasiswa berdasarkan rata-rata nilai dari tiga mata kuliah yaitu matematika, fisika, dan kimia. Program terdiri dari beberapa langkah yaitu mengurutkan data, menghitung rata-rata nilai, mengurutkan mahasiswa berdasarkan rata-rata tertinggi, dan menampilkan peringkat.

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)
}
}

```

OUTPUT

```

PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7> go run "c:\Users\Ikram\Documents\Folder Ikr
am\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7\GUIDED\guided2.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
-----
20231004     Dedi
20231001     Andi
20231002     Budi
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7>

```


DESKRIPSI PROGRAM

Program diatas merupakan program yang digunakan untuk mengelola data mahasiswa dengan menggunakan struktur data map, dengan menggunakan NIM sebgai kunci dan Nama sebagai nilai. Program memiliki beberapa fitur, yaitu tambah data, menampilkan data, menampilkan data dalam bentuk table .Mengakses data berdasarkan NIM, dan menghapus data tertentu.

III. UNGUIDED

1. SOURCE CODE

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    pusat Titik
    radius int
}

func jarak(p, q Titik) float64 {
    return math.Sqrt(float64((p.x - q.x)*(p.x - q.x) + (p.y - q.y)
    * (p.y - q.y)))
}

func inside(c Lingkaran, p Titik) bool {
    return jarak(c.pusat, p) <= float64(c.radius)
}

func main() {
    var l1, l2 Lingkaran
    var p Titik

    fmt.Scan(&l1.pusat.x, &l1.pusat.y, &l1.radius)
    fmt.Scan(&l2.pusat.x, &l2.pusat.y, &l2.radius)
    fmt.Scan(&p.x, &p.y)

    insideL1 := inside(l1, p)
    insideL2 := inside(l2, p)

    if insideL1 && insideL2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if insideL1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if insideL2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

OUTPUT

```
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7\UNGUIDED\unguided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7> |
```

DESKRIPSI PROGRAM

Program ini digunakan untuk menentukan posisi sebuah titik terhadap dua buah lingkaran dalam bidang koordinat. Program menggunakan struktur Titik untuk menyimpan koordinat x dan y, serta struktur Lingkaran yang terdiri dari pusat dan radius. Program meminta input koordinat pusat dan radius untuk dua lingkaran serta koordinat sebuah titik yang akan dicek posisinya. Menggunakan fungsi jarak yang menghitung jarak Euclidean antara dua titik, program menentukan apakah titik tersebut berada di dalam lingkaran 1, lingkaran 2, kedua lingkaran, atau di luar kedua lingkaran.

2. SOURCE CODE

```
package main

import (
    "fmt"
    "math"
)

type Array []int

func show_array(arr []int) {
    fmt.Println("ARRAY :", arr)
}

func index_ganjil(arr []int) {
    fmt.Print("Indeks Ganjil : ")
    for i := 1; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}

func index_genap(arr []int) {
    fmt.Print("Indeks Genap : ")
    for i := 0; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
}
```

```

        fmt.Println()
    }

    func kelipatan_index(arr []int, x int) {
        fmt.Printf("Indeks Kelipatan %d : ", x)
        for i := 0; i < len(arr); i += x {
            fmt.Print(arr[i], " ")
        }
        fmt.Println()
    }

    func hapus_index(arr []int, index int) []int {
        if index < 0 || index >= len(arr) {
            fmt.Println("Indeks Tidak Valid")
            return arr
        }
        fmt.Printf("Elemen di Indeks %d (nilai : %d) dihapus!\n",
            index, arr[index])
        return append(arr[:index], arr[index+1:]...)
    }

    func rata_rata(arr []int) float64 {
        sum := 0
        for _, v := range arr {
            sum += v
        }
        return float64(sum) / float64(len(arr))
    }

    func deviasi(arr []int) float64 {
        mean := rata_rata(arr)
        var sum float64
        for _, v := range arr {
            sum += math.Pow(float64(v)-mean, 2)
        }
        return math.Sqrt(sum / float64(len(arr)))
    }

    func frekuensi(arr []int, nilai int) int {
        count := 0
        for _, v := range arr {
            if v == nilai {
                count++
            }
        }
        return count
    }

    func main() {
        var n, x, index, nilai int

        fmt.Print("Masukkan jumlah elemen array : ")
    }

```

```

fmt.Scan(&n)

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

show_array(arr)
index_ganjil(arr)
index_genap(arr)

fmt.Print("\nMasukkan nilai x untuk kelipatan indeks : ")
fmt.Scan(&x)
kelipatan_index(arr, x)

    fmt.Print("\nMasukkan indeks untuk menghapus elemen : ")
    fmt.Scan(&index)
    arr = hapus_index(arr, index)
    show_array(arr)

fmt.Printf("\nRATA RATA : %.2f\n", rata_rata(arr))

fmt.Printf("DEVIASI : %.2f\n", deviasi(arr))

    fmt.Print("\nMasukkan nilai untuk menghitung frekuensi : ")
    fmt.Scan(&nilai)
    fmt.Printf("FREKUENSI %d : %d\n", nilai, frekuensi(arr,
nilai))
}

```

OUTPUT

```

PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7> go run "c:\Users\Ikram\Documents\Folder Ikr
am\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7\UNGUIDED\unguided2.go"
Masukkan jumlah elemen array : 5
Masukkan elemen array : 1 2 3 4 5
ARRAY : [1 2 3 4 5]
Indeks Ganjil : 2 4
Indeks Genap : 1 3 5

Masukkan nilai x untuk kelipatan indeks : 3
Indeks Kelipatan 3 : 1 4

Masukkan indeks untuk menghapus elemen : 3
Elemen di Indeks 3 (nilai : 4) dihapus!
ARRAY : [1 2 3 5]

RATA RATA : 2.75
DEVIASI : 1.48

Masukkan nilai untuk menghitung frekuensi : 3
FREKUENSI 3 : 1
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7> █

```

DESKRIPSI PROGRAM

Program ini digunakan untuk melakukan berbagai operasi analisis dan manipulasi pada array bilangan bulat. Program meminta user memasukkan jumlah elemen array dan nilai-nilainya, kemudian menyediakan berbagai fungsi untuk menganalisis array tersebut. Fungsi-fungsi yang tersedia meliputi menampilkan elemen pada indeks ganjil/genap, menampilkan elemen pada indeks kelipatan tertentu, menghapus elemen pada indeks tertentu, menghitung rata-rata array, menghitung deviasi standar, dan menghitung frekuensi kemunculan nilai tertentu dalam array.

3. SOURCE CODE

```
package main

import (
    "fmt"
)

type Klub struct {
    klubA, klubB string
}

func input_klub() Klub {
    var klub Klub
    fmt.Print("Klub A : ")
    fmt.Scan(&klub.klubA)
    fmt.Print("Klub B : ")
    fmt.Scan(&klub.klubB)
    return klub
}

func input_skor(pertandingan int) (int, int) {
    var skorA, skorB int
    fmt.Printf("Pertandingan  %d  (Skor  A,  Skor  B)  :  ",
    pertandingan)
    fmt.Scan(&skorA, &skorB)
    return skorA, skorB
}

func hasil_pemenang(skorA, skorB int, klub Klub) string {
    if skorA > skorB {
        return klub.klubA
    } else if skorB > skorA {
        return klub.klubB
    }
    return "Draw"
}

func hasil_pertandingan(pemenang []string) {
    fmt.Println("\nHasil pertandingan :")
    for i, winner := range pemenang {
```

```

        fmt.Printf("Hasil %d : %s\n", i+1, winner)
    }
    fmt.Println("Pertandingan selesai.")
}

func main() {
    // Input nama klub
    klub := input_klub()

    var pemenang []string
    pertandingan := 1

    for {
        skorA, skorB := input_skor(pertandingan)

        if skorA < 0 || skorB < 0 {
            break
        }

        winner := hasil_pemenang(skorA, skorB, klub)
        pemenang = append(pemenang, winner)

        pertandingan++
    }
    hasil_pertandingan(pemenang)
}

```

OUTPUT

```

PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7> go run "c:\Users\Ikram\Documents\Folder Ikr
am\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7\UNGUIDED\unguided3.go"
Klub A : MU
Klub B : Inter
Pertandingan 1 (Skor A, Skor B) : 2 0
Pertandingan 2 (Skor A, Skor B) : 1 2
Pertandingan 3 (Skor A, Skor B) : 2 2
Pertandingan 4 (Skor A, Skor B) : 0 1
Pertandingan 5 (Skor A, Skor B) : 3 2
Pertandingan 6 (Skor A, Skor B) : 1 0
Pertandingan 7 (Skor A, Skor B) : 5 2
Pertandingan 8 (Skor A, Skor B) : 2 3
Pertandingan 9 (Skor A, Skor B) : -1 2

Hasil pertandingan :
Hasil 1 : MU
Hasil 2 : Inter
Hasil 3 : Draw
Hasil 4 : Inter
Hasil 5 : MU
Hasil 6 : MU
Hasil 7 : MU
Hasil 8 : Inter
Pertandingan selesai.
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7>

```

DESKRIPSI PROGRAM

Program ini digunakan untuk mencatat dan menampilkan hasil pertandingan antara dua klub olahraga secara berurutan. Program menggunakan struktur Klub untuk menyimpan nama kedua klub yang bertanding. Program akan meminta input nama kedua klub dan kemudian meminta skor pertandingan secara berulang hingga dimasukkan skor negatif sebagai kondisi berhenti. Setiap pertandingan akan ditentukan pemenangnya berdasarkan skor, dengan hasil "Draw" jika skornya sama. Di akhir program, akan ditampilkan rekap hasil seluruh pertandingan yang telah dilakukan secara berurutan.

4. SOURCE CODE

```
package main

import (
    "fmt"
)

const NMAX int = 127
type tabel [NMAX]rune

func array() (t tabel, n int) {
    fmt.Println("Masukkan karakter (akhiri dengan titik :)")
    for n < NMAX {
        var char rune
        fmt.Printf("Karakter ke-%d : ", n+1)
        fmt.Scanf("%c\n", &char)
        if char == '.' {
            break
        }
        t[n] = char
        n++
    }
    return
}

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

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



```

func palindrom(t tabel, n int) bool {
    reversed := balik_array(t, n)
    for i := 0; i < n; i++ {
        if t[i] != reversed[i] {
            return false
        }
    }
    return true
}

func main() {
    tab, m := array()
    cetak_array(tab, m, "Original")

    reversed := balik_array(tab, m)
    cetak_array(reversed, m, "Reversed")

    if palindrom(tab, m) {
        fmt.Println("Palindrom True")
    } else {
        fmt.Println("Palindrom : False")
    }
}

```

OUTPUT

```

PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7> go run "c:\Users\Ikram\Documents\Folder Ikr
am\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7\UNGUIDED\unguided4.go"
Masukkan karakter (akhiri dengan titik) :
Karakter ke-1 : S
Karakter ke-2 : E
Karakter ke-3 : N
Karakter ke-4 : A
Karakter ke-5 : N
Karakter ke-6 : G
Karakter ke-7 : .
Original: SENANG
Reversed: GNANES
Palindrom : False
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 7>

```

DESKRIPSI PROGRAM

Program ini digunakan untuk melakukan operasi pada array karakter dan mengecek apakah array tersebut membentuk palindrom (kata yang dibaca sama dari depan maupun belakang). Program akan meminta input karakter satu per satu dari user hingga dimasukkan karakter titik sebagai penanda akhir input. Program kemudian akan menampilkan array karakter asli, menampilkan array yang sudah dibalik urutannya menggunakan fungsi `balik_array`, dan mengecek apakah susunan karakternya membentuk palindrom dengan membandingkan array asli dengan array yang sudah dibalik menggunakan fungsi `palindrom`.