

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7
STRUCT DAN ARRAY**



Disusun Oleh :

M. Faleno Albar Firjatulloh / 2311102297

S1-IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom.,M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

A. ARRAY

Array adalah tipe data dasar yang digunakan untuk menyimpan sekumpulan elemen dengan tipe data yang sama. Di bahasa Go, array memiliki ukuran tetap yang ditetapkan saat deklarasi.

Karakter Array:

1. Ukuran Standar:

Dibandingkan dengan tipe data lain seperti slice, ukuran array di Go ditentukan saat deklarasi dan tidak dapat diubah selama runtime.

2. Data homogen terdiri dari:

Semua elemen dalam array harus memiliki tipe data yang sama—misalnya, int, string, atau float64.

3. Index dimulai dengan zero:

Element awal array berada di indeks 0, elemen kedua berada di indeks 1, dan seterusnya.

4. Akses ke elemen:

Dengan menggunakan notasi array[index], Anda dapat mengakses elemen dalam array dengan menggunakan indeks.

Syarat Array:

Ukuran dan tipe data array dapat dideklarasikan. Dalam Go, ada beberapa cara untuk mendeklarasikan dan menginisialisasi array.

1. Deklarasi Array Kosong
2. Inisialisasi Array dengan Nilai
3. Deklarasi Array dengan Panjang Otomatis
4. Penggunaan Array
5. Array Multidimensi

B. STRUCT

Struktur sering digunakan untuk menunjukkan objek atau entitas dengan beberapa atribut. Ini adalah tipe data komposit yang memungkinkan kita mengelompokkan variabel dengan tipe yang berbeda dalam satu entitas.

Karakter Struktur:

1. Tipe Data Heterogen:

Dalam struct, field dapat memiliki berbagai tipe data, yang memungkinkan penggunaan tipe data yang lebih kompleks.

2. Nama Field:

Setiap field struct memiliki nama sendiri.

3. Notasi Titik:

Kita menggunakan notasi titik (.) untuk mengakses atau mengubah area.

4. Komposisi Struct:

Struct dapat menggabungkan struct lain karena mendukung komposisi.

Deklarasi Struktur:

Deklarasi struktur dimulai dengan kata kunci type, nama struktur, dan definisi field dalam kurung kurawal.

II. GUIDED

1. Guided 1

Soal Studi Case

Program sederhana untuk menghitung lama waktu parkir berdasarkan waktu kedatangan dan waktu pulang

Sourcecode

```
package main

import "fmt"

type waktu struct {
    jam, menit, detik int
}

func main() {
    var wParkir, wPulang, durasi waktu
    var dParkir, dPulang, lParkir int

    fmt.Scan(&wParkir.jam, &wParkir.menit,
&wParkir.detik)
    fmt.Scan(&wPulang.jam, &wPulang.menit,
&wPulang.detik)
    dParkir = wParkir.detik + wParkir.menit*60 +
wParkir.jam*3600 // Konversi ke detik
    dPulang = wPulang.detik + wPulang.menit*60 +
wPulang.jam*3600 // detik
    lParkir = dPulang -
dParkir //detik
    dari pulang-datang

    durasi.jam = lParkir / 3600
    durasi.menit = lParkir % 3600 / 60
    durasi.detik = lParkir % 3600 % 60 //17
    fmt.Printf("Lama Parkir : %d jam %d menit %d
detik", durasi.jam, durasi.menit, durasi.detik)
}
```

Screenshoot Output

```
PS D:\smt 3\Praktikum Alpro 2> go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\guided.go"
7 30 0
10 45 15
Lama Parkir : 3 jam 15 menit 15 detik
PS D:\smt 3\Praktikum Alpro 2> █
```

Deskripsi Program :

Program Go ini dirancang untuk menghitung durasi parkir kendaraan. Menggunakan struktur data `waktu` (jam, menit, detik), program menerima input waktu masuk dan keluar parkir. Program mengkonversi waktu ke detik untuk perhitungan, lalu menampilkan hasilnya dalam format "Lama Parkir: X jam X menit X detik". Sistem ini memungkinkan perhitungan durasi parkir yang akurat hingga level detik..

2. Guided 2

Soal Studi Case

Program sederhana untuk validasi duplikasi nama pada daftar teman

Sourcecode

```
package main

import (
    "fmt"
)

func sudahAda(daftarTeman []string, nama string)
bool {
    for _, teman := range daftarTeman {
        if teman == nama {
            return true
        }
    }
    return false
}

func main() {
    daftarTeman := []string{"Andi", "Budi",
"Cici"}

    namaBaru := []string{"Dewi", "Budi", "Eka"}

    for _, nama := range namaBaru {
        if !sudahAda(daftarTeman, nama) {
```

```

                                daftarTeman = append(daftarTeman,
nama)
                                } else {
                                    fmt.Println("Nama", nama, "sudah ada
dalam daftar.")
                                }
                            }

                            fmt.Println("Daftar Teman:", daftarTeman)
                        }

```

Screenshoot Output

```

PS D:\smt 3\Praktikum Alpro 2> go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\guided2.go"
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\smt 3\Praktikum Alpro 2>

```

Deskripsi Program :

Program Go ini mengelola daftar nama teman dengan fitur pencegahan duplikasi. Menggunakan fungsi `sudahAda`, program memeriksa keberadaan nama dalam daftar sebelum menambahkan nama baru. Program dimulai dengan tiga nama awal dan hanya akan menambahkan nama baru jika belum terdaftar, sambil menampilkan pesan untuk nama yang sudah ada.

3. Guided 3

Soal Studi Case

Program sederhana untuk menampilkan daftar harga buah

Sourcecode

```

package main

import "fmt"

func main() {
    hargaBuah := map[string]int{
        "Apel": 5000,
        "Pisang": 3000,
        "Mangga": 7000,
    }

    fmt.Println("Harga Buah:")
    for buah, harga := range hargaBuah {
        fmt.Printf("%s: Rp%d\n", buah, harga)
    }
}

```

```
}  
  
        fmt.Print("Harga    buah    Mangga    =    ",  
hargaBuah["Mangga"])  
}
```

Screenshot Output

```
PS D:\smt 3\Praktikum Alpro 2> go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\guided3.go"  
Harga Buah:  
Pisang: Rp3000  
Mangga: Rp7000  
Apel: Rp5000  
Harga buah Mangga = 7000  
PS D:\smt 3\Praktikum Alpro 2>
```

Deskripsi Program :

Program Go ini menggunakan map untuk menyimpan daftar harga buah, dengan nama buah sebagai key dan harga sebagai value. Program menampilkan seluruh daftar harga menggunakan perulangan `for range`, serta dapat menampilkan harga spesifik suatu buah dengan mengakses langsung key-nya dalam map.

III. UNGUIDED

1. Unguided 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".

No	Masukan	Keluaran
1	1 1 5 8 8 4 2 2	Titik di dalam lingkaran 1
2	1 2 3 4 5 6 7 8	Titik di dalam lingkaran 2
3	5 10 15 -15 4 20 0 0	Titik di dalam lingkaran 1 dan 2
4	1 1 5 8 8 4 15 20	Titik di luar lingkaran 1 dan 2

Sourcecode

```
package main

import (
    "bufio"
    "fmt"
    "math"
    "os"
    "strconv"
    "strings"
)

// Struct untuk menyimpan koordinat titik
type Point struct {
    x, y float64
}

// Struct untuk menyimpan informasi lingkaran
type Circle struct {
    center Point
    radius float64
}

// Fungsi untuk menghitung jarak antara dua titik
func distance(p1, p2 Point) float64 {
    dx := p1.x - p2.x
    dy := p1.y - p2.y
    return math.Sqrt(dx*dx + dy*dy)
}

// Fungsi untuk mengecek apakah titik berada di
dalam lingkaran
```



```

func isPointInCircle(p Point, c Circle) bool {
    return distance(p, c.center) <= c.radius
}

// Fungsi untuk menentukan posisi titik terhadap
dua lingkaran
func determinePosition(p Point, c1, c2 Circle)
string {
    inCircle1 := isPointInCircle(p, c1)
    inCircle2 := isPointInCircle(p, c2)

    if inCircle1 && inCircle2 {
        return "Titik di dalam lingkaran 1 dan
2"
    } else if inCircle1 {
        return "Titik di dalam lingkaran 1"
    } else if inCircle2 {
        return "Titik di dalam lingkaran 2"
    }
    return "Titik di luar lingkaran 1 dan 2"
}

func main() {
    scanner := bufio.NewScanner(os.Stdin)

    // Membaca data lingkaran 1
    scanner.Scan()
    line1 := strings.Fields(scanner.Text())
    x1, _ := strconv.ParseFloat(line1[0], 64)
    y1, _ := strconv.ParseFloat(line1[1], 64)
    r1, _ := strconv.ParseFloat(line1[2], 64)
    circle1 := Circle{Point{x1, y1}, r1}

    // Membaca data lingkaran 2
    scanner.Scan()
    line2 := strings.Fields(scanner.Text())
    x2, _ := strconv.ParseFloat(line2[0], 64)
    y2, _ := strconv.ParseFloat(line2[1], 64)
    r2, _ := strconv.ParseFloat(line2[2], 64)
    circle2 := Circle{Point{x2, y2}, r2}

    // Membaca koordinat titik yang akan dicek
    scanner.Scan()
    line3 := strings.Fields(scanner.Text())
    x, _ := strconv.ParseFloat(line3[0], 64)
    y, _ := strconv.ParseFloat(line3[1], 64)
    point := Point{x, y}

```

```

        // Menentukan dan mencetak hasil
        result := determinePosition(point, circle1,
circle2)
        fmt.Println(result)
    }

```

Screenshoot Output

```

PS D:\smt 3\Praktikum Alpro 2> go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\unguided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\smt 3\Praktikum Alpro 2> go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\unguided1.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS D:\smt 3\Praktikum Alpro 2> go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\unguided1.go"
5 10 15
-15 4 20
0 0go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\unguided1.go"
Titik di dalam lingkaran 1 dan 2
PS D:\smt 3\Praktikum Alpro 2>

```

Deskripsi Program :

Program Go ini menganalisis posisi suatu titik terhadap dua lingkaran menggunakan struktur data `Point` dan `Circle`. Program memiliki fungsi untuk menghitung jarak antar titik dan mengecek posisi titik terhadap lingkaran. Berdasarkan input koordinat dan jari-jari dari pengguna, program menentukan apakah titik berada di dalam salah satu lingkaran, di dalam keduanya, atau di luar keduanya.

2. Unguided 2

Soal Studi Case

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. Program dapat menampilkan beberapa informasi berikut:

- Menampilkan keseluruhan isi dari array.
- Menampilkan elemen-elemen array dengan indeks ganjil saja.
- Menampilkan elemen-elemen array dengan indeks genap saja (asumsi indek ke-0 adalah genap).
- 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

import (
    "fmt"
    "math"
)

type ArrayOperations struct {
    data []float64
    size int
}

// Membuat array baru
func NewArrayOperations(capacity int) *ArrayOperations {
    return &ArrayOperations{
        data: make([]float64, 0, capacity),
        size: 0,
    }
}

// Menambah elemen ke array
func (a *ArrayOperations) Add(value float64) {
    if a.size < cap(a.data) {
        a.data = append(a.data, value)
        a.size++
    }
}

// a. Menampilkan seluruh isi array
func (a *ArrayOperations) DisplayAll() {
    fmt.Println("Seluruh isi array:")
    for i, v := range a.data {
        fmt.Printf("Index %d: %.2f\n", i, v)
    }
}
```

```

// b. Menampilkan elemen dengan indeks ganjil
func (a *ArrayOperations) DisplayOddIndices() {
    fmt.Println("Elemen dengan indeks ganjil:")
    for i, v := range a.data {
        if i%2 != 0 {
            fmt.Printf("Index %d: %.2f\n", i, v)
        }
    }
}

// c. Menampilkan elemen dengan indeks genap
func (a *ArrayOperations) DisplayEvenIndices() {
    fmt.Println("Elemen dengan indeks genap:")
    for i, v := range a.data {
        if i%2 == 0 {
            fmt.Printf("Index %d: %.2f\n", i, v)
        }
    }
}

// d. Menampilkan elemen dengan indeks kelipatan
x
func (a *ArrayOperations)
DisplayMultipleIndices(x int) {
    fmt.Printf("Elemen dengan indeks kelipatan
%d:\n", x)
    for i, v := range a.data {
        if i%x == 0 {
            fmt.Printf("Index %d: %.2f\n", i, v)
        }
    }
}

// e. Menghapus elemen pada indeks tertentu
func (a *ArrayOperations) DeleteAtIndex(index
int) {
    if index >= 0 && index < a.size {
        a.data = append(a.data[:index],
a.data[index+1:]...)
        a.size--
    }
}

// f. Menghitung rata-rata
func (a *ArrayOperations) CalculateAverage()
float64 {

```

```

        if a.size == 0 {
            return 0
        }
        sum := 0.0
        for _, v := range a.data {
            sum += v
        }
        return sum / float64(a.size)
    }

// g. Menghitung standar deviasi
func (a *ArrayOperations)
CalculateStandardDeviation() float64 {
    if a.size < 2 {
        return 0
    }
    mean := a.CalculateAverage()
    sumSquaredDiff := 0.0
    for _, v := range a.data {
        diff := v - mean
        sumSquaredDiff += diff * diff
    }
    variance := sumSquaredDiff / float64(a.size-
1)
    return math.Sqrt(variance)
}

// h. Menghitung frekuensi suatu bilangan
func (a *ArrayOperations)
CalculateFrequency(number float64) int {
    frequency := 0
    for _, v := range a.data {
        if v == number {
            frequency++
        }
    }
    return frequency
}

func main() {
    // Membuat array dengan kapasitas 10
    arr := NewArrayOperations(10)

    // Menambahkan beberapa nilai
    arr.Add(1.0)
    arr.Add(2.0)
    arr.Add(3.0)

```

```
arr.Add(4.0)
arr.Add(5.0)
arr.Add(2.0)
arr.Add(3.0)

// Demonstrasi semua operasi
fmt.Println("\n=== Demonstrasi Operasi Array
===")

// a. Menampilkan semua elemen
arr.DisplayAll()

fmt.Println()
// b. Menampilkan indeks ganjil
arr.DisplayOddIndices()

fmt.Println()
// c. Menampilkan indeks genap
arr.DisplayEvenIndices()

fmt.Println()
// d. Menampilkan kelipatan 2
arr.DisplayMultipleIndices(2)

fmt.Println()
// e. Menghapus elemen index 2
fmt.Println("Menghapus elemen index 2:")
arr.DeleteAtIndex(2)
arr.DisplayAll()

fmt.Println()
// f. Menampilkan rata-rata
        fmt.Printf("Rata-rata:      %.2f\n",
arr.CalculateAverage())

// g. Menampilkan standar deviasi
        fmt.Printf("Standar deviasi:  %.2f\n",
arr.CalculateStandardDeviation())

// h. Menampilkan frekuensi angka 2.0
        fmt.Printf("Frekuensi angka 2.0:  %d\n",
arr.CalculateFrequency(2.0))
}
```

Screenshot Output

```

PS D:\smt 3\Praktikum Alpro 2> go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\unguided2.go"

=== Demonstrasi Operasi Array ===
Seluruh isi array:
Index 0: 1.00
Index 1: 2.00
Index 2: 3.00
Index 3: 4.00
Index 4: 5.00
Index 5: 2.00
Index 6: 3.00

Elemen dengan indeks ganjil:
Index 1: 2.00
Index 3: 4.00
Index 5: 2.00

Elemen dengan indeks genap:
Index 0: 1.00
Index 2: 3.00
Index 4: 5.00
Index 6: 3.00

Elemen dengan indeks kelipatan 2:
Index 0: 1.00
Index 2: 3.00
Index 4: 5.00
Index 6: 3.00

Menghapus elemen index 2:
Seluruh isi array:
Index 0: 1.00
Index 1: 2.00
Index 2: 4.00
Index 3: 5.00
Index 4: 2.00
Index 5: 3.00

Rata-rata: 2.83
Standar deviasi: 1.47
Frekuensi angka 2.0: 2
PS D:\smt 3\Praktikum Alpro 2>

```

Deskripsi Program :

Program Go ini mengimplementasikan sistem pengelolaan array menggunakan struktur data `ArrayOperations`. Program menyediakan berbagai fungsi untuk manipulasi dan analisis array floating-point, termasuk penambahan, penghapusan, penampilan data berdasarkan indeks, serta perhitungan statistik seperti rata-rata, standar deviasi, dan frekuensi kemunculan nilai.

3. Unguided 3

Soal Studi Case

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. 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)

```
Klub A : MU
Klub B : Inter
Pertandingan 1 : 2 0 // MU = 2 sedangkan Inter = 0
Pertandingan 2 : 1 2
Pertandingan 3 : 2 2
Pertandingan 4 : 0 1
Pertandingan 5 : 3 2
Pertandingan 6 : 1 0
Pertandingan 7 : 5 2
Pertandingan 8 : 2 3
Pertandingan 9 : -1 2
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
```

Sourcecode

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strings"
)
```



```

func main() {
    scanner := bufio.NewScanner(os.Stdin)
    var winners []string

    // Input nama klub
    fmt.Print("Klub A : ")
    scanner.Scan()
    clubA := scanner.Text()

    fmt.Print("Klub B : ")
    scanner.Scan()
    clubB := scanner.Text()

    // Proses input skor pertandingan
    matchNumber := 1
    for {
        fmt.Printf("Pertandingan  %d  :  ",
matchNumber)
        scanner.Scan()
        input := scanner.Text()

        // Split input menjadi dua skor
        scores := strings.Split(input, " ")
        if len(scores) != 2 {
            continue
        }

        // Convert string ke integer
        var scoreA, scoreB int
        _, err1 := fmt.Sscanf(scores[0], "%d",
&scoreA)
        _, err2 := fmt.Sscanf(scores[1], "%d",
&scoreB)

        // Cek apakah skor valid
        if err1 != nil || err2 != nil || scoreA
< 0 || scoreB < 0 {
            fmt.Printf("\nHasil pertandingan:\n")
            for i, winner := range winners {
                fmt.Printf("Hasil %d : %s\n",
i+1, winner)
            }
            fmt.Println("Pertandingan selesai")
            break
        }
    }
}

```

```

        // Tentukan pemenang
        var winner string
        if scoreA > scoreB {
            winner = clubA
        } else if scoreB > scoreA {
            winner = clubB
        } else {
            winner = "Draw"
        }

        // Tambahkan pemenang ke array
        winners = append(winners, winner)
        matchNumber++
    }
}

```

Screenshot Output

```

PS D:\smt 3\Praktikum Alpro 2> go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\un
guided3.go"
Klub A : MU
Klub B : INTER
Pertandingan 1 : 2 0
Pertandingan 2 : 1 2
Pertandingan 3 : 2 2
Pertandingan 4 : 0 1
Pertandingan 5 : 3 2
Pertandingan 6 : 1 0
Pertandingan 7 : 5 2
Pertandingan 8 : 2 3
Pertandingan 9 : -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 D:\smt 3\Praktikum Alpro 2>

```

Deskripsi Program :

Program Go ini dibuat untuk mencatat hasil pertandingan antar dua klub olahraga. Program meminta input nama klub dan skor setiap pertandingan, menentukan pemenang berdasarkan skor, dan menyimpan hasilnya. Program berhenti saat menerima input tidak valid, lalu menampilkan rekap seluruh hasil pertandingan.

1. Unguided 4

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.

Lengkapi potongan algoritma berikut ini!

```
package main
import "fmt"
const NMAX int = 127
type tabel [NMAX]rune
    tab : tabel
    m : integer

func isiArray(t *tabel, n *int)
/*I.S. Data tersedia dalam piranti masukan
   F.S. Array t berisi sejumlah n karakter yang dimasukkan user,
   Proses input selama karakter bukanlah TITIK dan n <= NMAX */

func cetakArray(t tabel, n int)
/*I.S. Terdefinisi array t yang berisi sejumlah n karakter
   F.S. n karakter dalam array muncul di layar */

func balikanArray(t *tabel, n int)
/*I.S. Terdefinisi array t yang berisi sejumlah n karakter
   F.S. Urutan isi array t terbalik */

func main(){
    var tab tabel
    var m int
    // si array tab dengan memanggil prosedur isiArray

    // Balikian isi array tab dengan memanggil balikanArray

    // Cetak is array tab
}
```

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

Teks	: S E N A N G .
Reverse teks	: G N A N E S
Teks	: K A T A K .
Reverse teks	: K A T A K

Modifikasi program tersebut dengan menambahkan fungsi palindrom. Tambahkan instruksi untuk memanggil fungsi tersebut dan menampilkan hasilnya pada program utama.

***Palindrom adalah teks yang dibaca dari awal atau akhir adalah sama, contoh: KATAK, APA, KASUR_RUSAK.**

func palindrom(t tabel, n int) bool	
/* Mengembalikan true apabila susunan karakter di dalam t membentuk palindrom, dan false apabila sebaliknya. Petunjuk: Manfaatkan prosedur balikanArray */	

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

Teks	: K A T A K
Palindrom	? true
Teks	: S E N A N G
Palindrom	? false

Sourcecode

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

const NMAX int = 127

type tabel [NMAX]rune

// Fungsi untuk mengisi array dengan karakter
func isiArray(t *tabel, n *int) {
    scanner := bufio.NewScanner(os.Stdin)
    fmt.Print("Teks : ")
    scanner.Scan()
    text := scanner.Text()

    // Menghilangkan spasi dan tanda baca
    text = strings.ReplaceAll(text, " ", "")
    text = strings.ReplaceAll(text, ".", "")
    text = strings.ReplaceAll(text, ",", "")

    *n = 0
    // Mengisi array dengan karakter dari input
```

```

        for _, char := range text {
            if *n < NMAX {
                t[*n] = char
                *n++
            }
        }
    }

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

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

    // Fungsi untuk mengecek apakah array membentuk
    palindrom
    func palindrom(t tabel, n int) bool {
        // Membuat salinan array untuk dibalik
        var temp tabel
        copy(temp[:], t[:n])

        // Balik array salinan
        for i := 0; i < n/2; i++ {
            temp[i], temp[n-1-i] = temp[n-1-i],
temp[i]
        }

        // Bandingkan array asli dengan array yang
        sudah dibalik
        for i := 0; i < n; i++ {
            if t[i] != temp[i] {
                return false
            }
        }
        return true
    }

```

```

func main() {
    var tab tabel
    var m int

    // Input dan proses array
    isiArray(&tab, &m)

    // Tampilkan array yang dibalik
    var reversed tabel
    copy(reversed[:], tab[:m])
    balikanArray(&reversed, m)
    cetakArray(reversed, m)

    // Cek palindrom
    fmt.Print("Palindrom ? ")
    if palindrom(tab, m) {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }
}

```

Screenshot Output

```

PS D:\smt 3\Praktikum Alpro 2> go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\unguided4.go"
Teks : SENANG
Reverse teks : GNANES
Palindrom ? false
PS D:\smt 3\Praktikum Alpro 2> go run "d:\smt 3\Praktikum Alpro 2\Pertemuan 7\unguided4.go"
Teks : KATAK
Reverse teks : KATAK
Palindrom ? true
PS D:\smt 3\Praktikum Alpro 2>

```

Deskripsi Program :

Program Go ini menganalisis palindrom dari input teks. Menggunakan array rune, program memiliki fungsi untuk membersihkan teks dari spasi dan tanda baca, membalik urutan karakter, dan membandingkan teks asli dengan teks yang dibalik untuk menentukan apakah input merupakan palindrom atau bukan.

IV. KESIMPULAN

Pembelajaran tentang struct dan array merupakan aspek penting dalam pemrograman yang membantu dalam pengelolaan data secara efisien. Struct, atau struktur, memungkinkan pengembang untuk mengelompokkan berbagai tipe data yang berbeda menjadi satu kesatuan yang terorganisir, sehingga memudahkan representasi objek yang kompleks. Sementara itu, array menyediakan cara untuk menyimpan sekumpulan data dengan tipe yang sama dalam urutan tertentu, memungkinkan akses dan manipulasi data secara cepat. Keduanya saling melengkapi; struct memberikan fleksibilitas dalam mendefinisikan tipe data yang lebih kompleks, sedangkan array menawarkan cara untuk mengelola koleksi data secara sistematis. Dengan memahami dan menguasai kedua konsep ini, programmer dapat meningkatkan kemampuan dalam merancang program yang lebih terstruktur dan efisien, serta memecahkan masalah dengan lebih baik.

V. REFERENSI

- [1] Modul 7 Praktikum Algoritma 2
- [2] A. A. A. Donovan and B. W. Kernighan, *The Go Programming Language*. Boston, MA: Addison-Wesley, 2015.