

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCT & ARRAY**



Oleh :

RAJA MUHAMMAD LUFHTI

103112400027

IF – 12 – 01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

DASAR TEORI – ARRAY KARAKTER, PEMBALIKAN URUTAN, DAN PALINDROM

1. Pengertian Array

Array adalah struktur data yang digunakan untuk menyimpan *kumpulan data sejenis* dalam satu variabel. Data dalam array disusun secara berurutan dan dapat diakses menggunakan *indeks*, yang dimulai dari angka 0.

Contoh sederhana array karakter:

[‘A’, ‘B’, ‘C’, ‘D’, ‘E’]

Array tersebut menyimpan 5 karakter, masing-masing memiliki posisi (indeks) dari 0 sampai 4.

2. Karakter dan Tipe Data Rune

Karakter adalah satuan huruf atau simbol yang bisa ditampilkan di layar. Dalam bahasa Go, karakter biasanya disimpan dalam tipe data rune, yang mampu menampung satu karakter Unicode.

Karakter bisa berupa huruf (contoh: A, Z), angka (contoh: 1, 9), atau simbol (contoh: \$, @).

3. Proses Input Karakter ke Array

Untuk mengisi array karakter, dilakukan proses pembacaan input dari pengguna satu per satu. Proses ini berjalan hingga ditemukan karakter *titik (.)* sebagai tanda berhenti, atau hingga jumlah maksimum elemen array tercapai.

Tujuannya adalah agar program dapat menyimpan barisan karakter yang dimasukkan oleh pengguna dalam urutan yang benar.

4. Mencetak Isi Array

Setelah array diisi, kita perlu menampilkan seluruh isi array ke layar. Proses ini dilakukan dengan cara membaca dan mencetak karakter satu per satu dari indeks pertama hingga terakhir yang valid (sesuai jumlah karakter yang dimasukkan).

Tujuan dari proses ini adalah untuk:

- Menampilkan hasil input pengguna
- Mempermudah verifikasi dan pengecekan data

5. Pembalikan Urutan Isi Array

Proses ini bertujuan untuk membalik urutan karakter dalam array. Karakter pertama akan ditukar dengan karakter terakhir, karakter kedua dengan karakter sebelum terakhir, dan seterusnya.

Contoh:

- Sebelum dibalik: ['H', 'A', 'L', 'O']
- Setelah dibalik: ['O', 'L', 'A', 'H']

Proses ini penting untuk digunakan sebagai bagian dari pengecekan *palindrom*, karena palindrom melibatkan perbandingan antara urutan asli dan urutan terbalik.

6. Pengertian Palindrom

Palindrom adalah kata, frasa, atau deretan karakter yang dibaca dari depan dan belakang tetap sama.

Contoh palindrom:

- "KATAK"
- "MALAM"
- "KASURRUSAK"

Contoh bukan palindrom:

- "SENANG"
- "BELAJAR"

Untuk menentukan apakah sebuah array karakter membentuk palindrom:

- Karakter pada posisi awal dibandingkan dengan posisi akhir
- Jika semua pasangan karakter tersebut sama, maka itu *palindrom*
- Jika ada satu saja yang berbeda, maka bukan palindrom

7. Fungsi dalam Program

Dalam sebuah program, kita biasanya membuat beberapa *fungsi (prosedur)* untuk memisahkan tugas-tugas berikut:

- *Mengisi array karakter* dari input pengguna
- *Mencetak isi array* ke layar
- *Membalik isi array* secara terbalik
- *Memeriksa apakah array membentuk palindrom*

Dengan membagi program menjadi beberapa fungsi, proses menjadi lebih teratur, mudah dibaca, dan efisien dalam pemeliharaan.

8. Tujuan Pembelajaran

Materi ini mengajarkan dasar penting dalam pemrograman:

- Cara menyimpan dan memanipulasi kumpulan data (karakter)
- Pemahaman struktur data array
- Penggunaan logika dan algoritma dasar
- Implementasi prosedur dan fungsi dalam bahasa pemrograman
- Penerapan konsep matematika/logika seperti pencocokan (untuk palindrom)

II. GUIDED

1. Guided 1

Source Code :

```
package main

import (
    "fmt"
    "time"
)

// Struct untuk barang dalam struk belanja
type Item struct {
    Name    string
    Price   float64
    Quantity int
}

// Struct untuk struk belanja
type Receipt struct {
    StoreInfo string
    Date      time.Time
    Items     []Item
    TotalAmount float64
}

// Method untuk menghitung total harga semua item
func (r *Receipt) CalculateTotal() {
    var total float64
    for _, item := range r.Items {
        total += item.Price * float64(item.Quantity)
    }
    r.TotalAmount = total
}

// Method untuk mencetak struk belanja
func (r Receipt) PrintReceipt() {
```

```

    fmt.Println("=====
")
    fmt.Println(r.StoreInfo)
    fmt.Println("Tanggal:", r.Date.Format("02-01-2006 15:04"))
    fmt.Println("=====
")
    fmt.Printf("%-15s %-10s %-8s %-10s\n", "Item", "Harga", "Jumlah",
"Total")
    fmt.Println("-----")

    for _, item := range r.Items {
        itemTotal := item.Price * float64(item.Quantity)
        fmt.Printf("%-15s Rp%-9.2f %-8d Rp%-9.2f\n", item.Name,
item.Price, item.Quantity, itemTotal)
    }

    fmt.Println("=====
")
    fmt.Printf("%-35s Rp%-9.2f\n", "Total Belanja:", r.TotalAmount)
    fmt.Println("=====
")
    fmt.Println("Terima kasih telah berbelanja!")
}

func main() {
    receipt := Receipt{
        StoreInfo: "Toko Sembako Makmur\nJl. Raya No. 123, Jakarta",
        Date:      time.Now(),
        Items: []Item{
            {Name: "Beras", Price: 12000, Quantity: 5},
            {Name: "Gula", Price: 15000, Quantity: 2},
            {Name: "Minyak", Price: 20000, Quantity: 1},
            {Name: "Telur", Price: 2000, Quantity: 10},
        },
    }

    receipt.CalculateTotal()
    receipt.PrintReceipt()
}

```

Output :

```
PS D:\ALPRO 2\103112400027_MODUL 6> go run "d:\ALPRO 2\103112400027_MODUL 6\1\1.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 14:42
=====
Item          Harga      Jumlah    Total
-----
Beras         Rp12000.00    5      Rp60000.00
Gula          Rp15000.00    2      Rp30000.00
Minyak        Rp20000.00    1      Rp20000.00
Telur         Rp2000.00    10     Rp20000.00
=====
Total Belanja:                Rp130000.00
=====
Terima kasih telah berbelanja!
PS D:\ALPRO 2\103112400027_MODUL 6> |
```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go dan bertujuan untuk mencetak struk belanja. Terdapat dua struktur utama dalam aplikasi ini: yang pertama adalah Item, yang merepresentasikan barang dengan atribut nama, harga, dan jumlah. Yang kedua adalah Receipt, yang menyimpan informasi mengenai toko, tanggal transaksi, daftar item, serta total belanja.

Dalam program ini, terdapat dua metode utama. Metode pertama, CalculateTotal, berfungsi untuk menghitung total harga semua item yang dibeli. Metode kedua, PrintReceipt, digunakan untuk mencetak struk belanja dengan format yang rapi.

Pada fungsi main, program ini membuat objek Receipt yang berisi informasi tentang toko serta daftar item yang dibeli. Setelah itu, total belanja dihitung dan struk tersebut dicetak.

2. Guided 2

Source Code :

```
package main

import (
    "fmt"
)

func main() {
    // Deklarasi dan inisialisasi array nilai mahasiswa
```

```

nilaiMahasiswa := [5]int{85, 90, 78, 88, 95}

fmt.Println("Data Nilai Mahasiswa:")
fmt.Println("=====")

// Menampilkan nilai per mahasiswa
for i, nilai := range nilaiMahasiswa {
    fmt.Printf("Mahasiswa %d: %d\n", i+1, nilai)
}

// Menghitung rata-rata nilai
var total int
for _, nilai := range nilaiMahasiswa {
    total += nilai
}
rataRata := float64(total) / float64(len(nilaiMahasiswa))

fmt.Println("=====")
fmt.Printf("Rata-rata nilai: %.2f\n", rataRata)

// Mencari nilai tertinggi dan terendah
tertinggi := nilaiMahasiswa[0]
terendah := nilaiMahasiswa[0]

for _, nilai := range nilaiMahasiswa {
    if nilai > tertinggi {
        tertinggi = nilai
    }
    if nilai < terendah {
        terendah = nilai
    }
}

fmt.Printf("Nilai tertinggi: %d\n", tertinggi)
fmt.Printf("Nilai terendah: %d\n", terendah)

// Contoh array 2 dimensi
fmt.Println("\nContoh Array 2 Dimensi:")
fmt.Println("=====")

```



```
// Nilai ujian mahasiswa dalam 2 mata kuliah (Matematika, Bahasa)
nilaiUjian := [3][2]int{
    {80, 85},
    {90, 75},
    {70, 95},
}

// Menampilkan nilai ujian per mahasiswa
fmt.Println("Nilai Ujian Mahasiswa (Matematika, Bahasa):")
for i, nilai := range nilaiUjian {
    fmt.Printf("Mahasiswa %d: Matematika = %d, Bahasa = %d\n", i+1,
nilai[0], nilai[1])
}
}
```

Output :

```
PS D:\ALPRO 2\103112400027_MODUL 6> go run "d:\ALPRO 2\103112400027_MODUL 6\2\2.go"
Data Nilai Mahasiswa:
=====
Mahasiswa 1: 85
Mahasiswa 2: 90
Mahasiswa 3: 78
Mahasiswa 4: 88
Mahasiswa 5: 95
=====
Rata-rata nilai: 87.20
Nilai tertinggi: 95
Nilai terendah: 78

Contoh Array 2 Dimensi:
=====
Nilai Ujian Mahasiswa (Matematika, Bahasa):
Mahasiswa 1: Matematika = 80, Bahasa = 85
Mahasiswa 2: Matematika = 90, Bahasa = 75
Mahasiswa 3: Matematika = 70, Bahasa = 95
PS D:\ALPRO 2\103112400027_MODUL 6> |
```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go yang bertujuan untuk mengelola dan menganalisis nilai mahasiswa. Di dalam aplikasi ini, terdapat sebuah array satu dimensi yang menyimpan nilai dari lima mahasiswa, yang kemudian ditampilkan secara berurutan. Program ini juga dapat menghitung rata-rata nilai serta menentukan nilai tertinggi dan terendah di antara nilai-nilai yang ada.

Selain itu, terdapat pula contoh penggunaan array dua dimensi untuk menyimpan nilai ujian mahasiswa di dua mata kuliah, yaitu Matematika dan Bahasa. Nilai ujian ditampilkan dengan format yang jelas, memudahkan pengguna untuk melihat performa akademik setiap mahasiswa.

III. UNGUIDED

1. UnGuided 1

Source Code :

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    pusat Titik
    radius int
}

func hitungJarak(a, b Titik) float64 {
    return math.Sqrt(math.Pow(float64(a.x-b.x), 2) +
        math.Pow(float64(a.y-b.y), 2))
}

func diDalamLingkaran(t Titik, l Lingkaran) bool {
    return hitungJarak(t, l.pusat) <= float64(l.radius)
}

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

    // Input dari user
    fmt.Println("Masukkan data lingkaran 1 (x y radius):")
    fmt.Scan(&l1.pusat.x, &l1.pusat.y, &l1.radius)

    fmt.Println("Masukkan data lingkaran 2 (x y radius):")
```

```

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

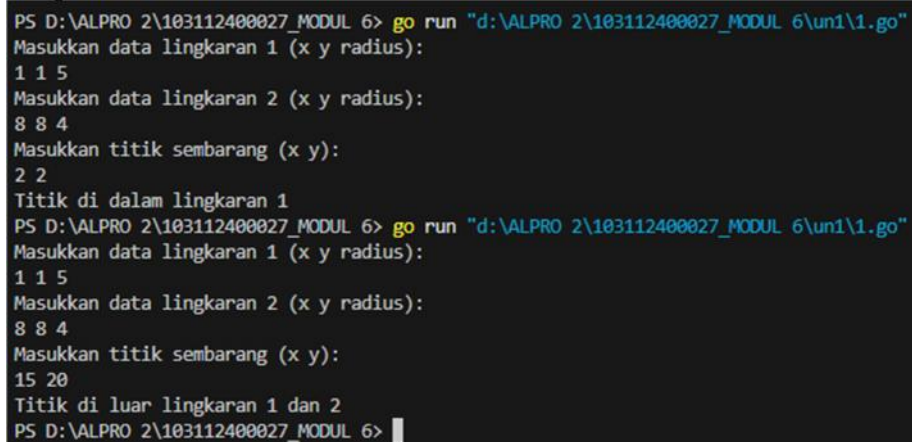
fmt.Println("Masukkan titik sembarang (x y):")
fmt.Scan(&titik.x, &titik.y)

// Logika posisi titik terhadap dua lingkaran
dalam1 := diDalamLingkaran(titik, l1)
dalam2 := diDalamLingkaran(titik, l2)

if dalam1 && dalam2 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if dalam1 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if dalam2 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

Output :



```

PS D:\ALPRO 2\103112400027_MODUL 6> go run "d:\ALPRO 2\103112400027_MODUL 6\un1\1.go"
Masukkan data lingkaran 1 (x y radius):
1 1 5
Masukkan data lingkaran 2 (x y radius):
8 8 4
Masukkan titik sembarang (x y):
2 2
Titik di dalam lingkaran 1
PS D:\ALPRO 2\103112400027_MODUL 6> go run "d:\ALPRO 2\103112400027_MODUL 6\un1\1.go"
Masukkan data lingkaran 1 (x y radius):
1 1 5
Masukkan data lingkaran 2 (x y radius):
8 8 4
Masukkan titik sembarang (x y):
15 20
Titik di luar lingkaran 1 dan 2
PS D:\ALPRO 2\103112400027_MODUL 6>

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go yang bertujuan untuk menentukan posisi sebuah titik relatif terhadap dua lingkaran yang telah ditentukan. Dengan memanfaatkan fungsi `hitungJarak`, program ini menghitung jarak antara pusat lingkaran dan titik yang ingin diperiksa. Selanjutnya, fungsi `titikDalamLingkaran` akan memverifikasi apakah titik tersebut berada di dalam lingkaran berdasarkan jarak yang telah dihitung dan radius masing-masing lingkaran. Di dalam fungsi utama, pengguna akan diminta untuk memasukkan koordinat pusat serta radius dari kedua lingkaran, disertai dengan koordinat titik yang ingin dianalisis. Setelah melakukan pengecekan, program ini akan menampilkan hasil yang menunjukkan apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, atau justru berada di luar keduanya.

2. UnGuided 2

Source Code :

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n, x, hapusIdx, cari int
    fmt.Print("Jumlah elemen: ")
    fmt.Scan(&n)
    data := make([]int, n)
    fmt.Println("Masukkan elemen:")
    for i := 0; i < n; i++ {
        fmt.Printf("indeks ke-%d: ", i)
        fmt.Scan(&data[i])
    }

    fmt.Print("a. Isi array: ")
    for _, v := range data {
        fmt.Print(v, " ")
    }
    fmt.Println()

    fmt.Print("b. Indeks ganjil: ")
    for i := 1; i < len(data); i += 2 {
        fmt.Print(data[i], " ")
    }
    fmt.Println()

    fmt.Print("c. Indeks genap: ")
    for i := 0; i < len(data); i += 2 {
        fmt.Print(data[i], " ")
    }
    fmt.Println()

    fmt.Print("Masukkan Indeks kelipatan x: ")
    fmt.Scan(&x)
    fmt.Printf("d. Indeks kelipatan %d: ", x)
```

```

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

fmt.Print("Masukan Indeks yang ingin dihapus: ")
fmt.Scan(&hapusIdx)
data = append(data[:hapusIdx], data[hapusIdx+1:]...)

fmt.Print("e. Array Setelah dihapus: ")
for _, v := range data {
    fmt.Print(v, " ")
}
fmt.Println()

var total float64
for _, v := range data {
    total += float64(v)
}
rata := total / float64(len(data))
fmt.Printf("f. Rata-rata: %.2f\n", rata)

var jumlahKuadrat float64
for _, v := range data {
    selisih := float64(v) - rata
    jumlahKuadrat += selisih * selisih
}
sd := math.Sqrt(jumlahKuadrat / float64(len(data)))
fmt.Printf("g. Simpangan baku: %.2f\n", sd)

fmt.Print("Masukan Nilai yang ingin dicari frekuensinya: ")
fmt.Scan(&cari)
freq := 0
for _, v := range data {
    if v == cari {
        freq++
    }
}

```

```
fmt.Printf("h. Frekuensi %d: %d\n", cari, freq)
}
```

Output :

```
PS D:\ALPRO 2\103112400027_MODUL 6> go run "d:\ALPRO 2\103112400027_MODUL 6\un2\2.go"
Jumlah elemen: 5
Masukkan elemen:
indeks ke-0: 1
indeks ke-1: 2
indeks ke-2: 3
indeks ke-3: 4
indeks ke-4: 1
a. Isi array: 1 2 3 4 1
b. Indeks ganjil: 2 4
c. Indeks genap: 1 3 1
Masukkan Indeks kelipatan x: 2
d. Indeks kelipatan 2: 1 3 1
Masukan Indeks yang ingin dihapus: 2
e. Array Setelah dihapus: 1 2 4 1
f. Rata-rata: 2.00
g. Simpangan baku: 1.22
Masukan Nilai yang ingin dicari frekuensinya: 1
h. Frekuensi 1: 2
```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go yang bertujuan untuk mengelola dan menganalisis sekumpulan bilangan bulat yang disimpan dalam sebuah array. Pengguna diminta untuk memasukkan jumlah elemen yang ingin dimasukkan ke dalam array, diikuti dengan nilai-nilai dari elemen-elemen tersebut. Setelah itu, program menawarkan berbagai fungsi analisis, seperti menampilkan semua elemen, elemen

dengan indeks ganjil dan genap, serta elemen dengan indeks kelipatan tertentu yang ditentukan oleh pengguna.

Selain itu, program ini memberi kesempatan kepada pengguna untuk menghapus elemen pada indeks tertentu dan menampilkan isi array setelah penghapusan dilakukan. Program juga menghitung dan menampilkan rata-rata serta simpangan baku (standar deviasi) dari elemen-elemen dalam array. Di akhir program, pengguna dapat mencari frekuensi kemunculan bilangan tertentu di dalam array.

3. UnGuided 3

Source Code :

```
package main

import (
    "fmt"
)

func main() {
    var klubA, klubB string
    var skorA, skorB int
    var pemenang []string
    pertandingan := 1

    fmt.Print("Klub A : ")
    fmt.Scan(&klubA)
    fmt.Print("Klub B : ")
    fmt.Scan(&klubB)

    for {
        fmt.Printf("Pertandingan %d : ", pertandingan)
        _, err := fmt.Scan(&skorA, &skorB)

        if err != nil || skorA < 0 || skorB < 0 {
            break
        }

        if skorA > skorB {
            fmt.Printf("// %s = %d sedangkan %s = %d\n", klubA, skorA,
klubB, skorB)
            fmt.Printf("Hasil %d : %s\n", pertandingan, klubA)
            pemenang = append(pemenang, klubA)
        } else if skorB > skorA {
```



```

fmt.Printf("// %s = %d sedangkan %s = %d\n", klubA, skorA, klubB,
skorB)
    fmt.Printf("Hasil %d : %s\n", pertandingan, klubB)
    pemenang = append(pemenang, klubB)
} else {
    fmt.Printf("// %s = %d sedangkan %s = %d\n", klubA, skorA,
klubB, skorB)
    fmt.Printf("Hasil %d : Draw\n", pertandingan)
}

    pertandingan++
}

fmt.Println("Pertandingan selesai")
fmt.Println("Daftar klub pemenang pertandingan:")

for i, klub := range pemenang {
    fmt.Printf("Pertandingan %d dimenangkan oleh: %s\n", i+1, klub)
}
}

```

Output :

```

PS D:\ALPRO 2\103112400027_MODUL 6> go run "d:\ALPRO 2\103112400027_MODUL 6\un3\3.go"
Klub A : MU
Klub B : INTER
Pertandingan 1 : 2 0
// MU = 2 sedangkan INTER = 0
Hasil 1 : MU
Pertandingan 2 : 1 2
// MU = 1 sedangkan INTER = 2
Hasil 2 : INTER
Pertandingan 3 : 0 1
// MU = 0 sedangkan INTER = 1
Hasil 3 : INTER
Pertandingan 4 : 2 2
// MU = 2 sedangkan INTER = 2
Hasil 4 : Draw
Pertandingan 5 : 3 2
// MU = 3 sedangkan INTER = 2
Hasil 5 : MU
Pertandingan 6 : 1 0
// MU = 1 sedangkan INTER = 0
Hasil 6 : MU
Pertandingan 7 : 5 2
// MU = 5 sedangkan INTER = 2
Hasil 7 : MU
Pertandingan 8 : 2 3
// MU = 2 sedangkan INTER = 3
Hasil 8 : INTER
Pertandingan 9 : -1 2
Pertandingan selesai
Daftar klub pemenang pertandingan:
Pertandingan 1 dimenangkan oleh: MU
Pertandingan 2 dimenangkan oleh: INTER
Pertandingan 3 dimenangkan oleh: INTER
Pertandingan 4 dimenangkan oleh: MU
Pertandingan 5 dimenangkan oleh: MU
Pertandingan 6 dimenangkan oleh: MU

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go yang bertujuan untuk merekam dan menampilkan hasil pertandingan antara dua klub sepak bola. Pengguna diminta untuk memasukkan nama kedua klub, diikuti dengan skor pertandingan yang berlangsung. Program akan terus meminta input skor hingga pengguna memasukkan nilai negatif, yang menandakan akhir dari proses input. Setelah setiap skor dimasukkan, program akan menentukan hasil pertandingan, apakah Klub A menang, Klub B menang,

atau jika hasilnya imbang (draw) dan menyimpan informasi tersebut dalam sebuah daftar. Setelah semua skor tercatat, program akan menampilkan hasil dari setiap pertandingan yang telah dicatat.

4. UnGuided

SOURCE GUIDE

```

package main

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

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    reader := bufio.NewReader(os.Stdin)
    fmt.Print("Masukkan teks (akhiri dengan titik): ")
    i := 0
    for i < NMAX {
        ch, _, err := reader.ReadRune()
        if err != nil {
            fmt.Println("Terjadi kesalahan saat membaca input:", err)
            break
        }
        if ch == '.' {
            break
        }
        (*t)[i] = ch
        i++
    }
    *n = 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-1-i] = t[n-1-i], t[i]
}
}

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

func main() {
    var tab tabel
    var m int

    isiArray(&tab, &m)

    fmt.Print("Teks Asli      : ")
    cetakArray(tab, m)

    if palindrome(tab, m) {
        fmt.Println("Palindrome?   : true")
    } else {
        fmt.Println("Palindrome?   : false")
    }

    balikanArray(&tab, m)
    fmt.Print("Teks Terbalik  : ")
    cetakArray(tab, m)
}

```

Output :

```

PS D:\ALPRO 2\103112400027_MODUL 6> go run "d:\ALPRO 2\103112400027_MODUL 6\un4\4.go"
Masukkan teks (akhiri dengan titik): K A T A K.
Teks Asli      : K A T A K
Palindrome?    : true
Teks Terbalik  : K A T A K
PS D:\ALPRO 2\103112400027_MODUL 6> go run "d:\ALPRO 2\103112400027_MODUL 6\un4\4.go"
Masukkan teks (akhiri dengan titik): S E N A N G.
Teks Asli      : S E N A N G
Palindrome?    : false
Teks Terbalik  : G N A N E S
PS D:\ALPRO 2\103112400027_MODUL 6>

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go yang bertujuan untuk mengelola dan menganalisis sekumpulan karakter yang dimasukkan oleh pengguna. Pengguna akan diminta untuk menginput teks karakter satu per

satu hingga mencapai tanda titik (.) atau hingga batas maksimum karakter yang telah ditentukan. Karakter yang telah dimasukkan akan disimpan dalam sebuah array. Setelah semua karakter terinput, program akan menampilkan isi array tersebut. Selanjutnya, program akan memeriksa apakah urutan karakter dalam array tersebut membentuk palindrom, yaitu kata atau frasa yang dapat dibaca sama baik dari depan maupun dari belakang. Hasil pemeriksaan palindrom ini kemudian akan disampaikan kepada pengguna.

IV. KESIMPULAN

Modul ini membahas tipe bentukan dalam bahasa pemrograman Go, khususnya mengenai **alias**, **struct**, **array**, **slice**, dan **map**, serta penerapannya dalam pemecahan masalah praktikum.

1. Tipe Bentukan:

- **Alias** digunakan untuk memberi nama baru pada tipe data yang sudah ada agar lebih mudah digunakan.
- **Struct (struktur)** memungkinkan pengelompokan beberapa variabel dengan tipe berbeda menjadi satu kesatuan logis.

2. Array:

- Merupakan struktur data dengan ukuran tetap selama program berjalan.
- Digunakan untuk menyimpan sekumpulan data sejenis.
- Operasi dasar yang umum digunakan: akses indeks, iterasi, manipulasi nilai.

3. Slice:

- Versi dinamis dari array.
- Dapat berubah ukuran selama eksekusi program.
- Didukung oleh fungsi-fungsi seperti `len()`, `cap()`, dan `append()`.

4. Map:

- Struktur data asosiatif yang menyimpan pasangan kunci-nilai.
- Kunci dapat bertipe apa pun, tidak terbatas pada integer.
- Digunakan untuk pencarian data yang efisien berdasarkan kunci.

5. Latihan Praktikum:

- Mengaplikasikan struct dan array dalam kasus nyata, seperti:
 - Menentukan posisi titik terhadap lingkaran.
 - Memanipulasi array bilangan bulat (tampilkan, hapus, cari frekuensi, dsb).
 - Merekam hasil pertandingan bola.
 - Mengecek apakah sebuah string membentuk palindrom.

V. REFERENSI

Modul 7 - Praktikum Alpro 2

- | | | | | |
|-------|---------|---------|-----------|---|
| A.24. | Struct. | (n.d.). | Retrieved | from |
| | | | | https://dasarpemrogramangolang.novalagung.com/A-struct.html |
| A.15. | Array. | (n.d.). | Retrieved | from |
| | | | | https://dasarpemrogramangolang.novalagung.com/A-array.html |
| A.16. | Slice. | (n.d.). | Retrieved | from |
| | | | | https://dasarpemrogramangolang.novalagung.com/A-slice.html |