LAPORAN

PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 6

STRUCT & ARRAY

Oleh:

NAMA: NUFAIL ALAUDDI TSAQIF

NIM: 103112400084

KELAS: IF-12-01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I. DASAR TEORI

1. Tipe Bentukan (User-Defined Types)

Tipe bentukan adalah fitur dalam bahasa pemrograman yang memungkinkan programmer mendefinisikan tipe data baru untuk memodelkan entitas yang lebih kompleks dan bermakna. Dalam bahasa Go, tipe bentukan dibedakan menjadi dua kategori utama: alias dan struct.

- **Alias(TypeAlias)**
  Alias digunakan untuk memberikan nama baru pada tipe data yang sudah ada guna meningkatkan keterbacaan dan kemudahan pemrograman. Misalnya, tipe int dapat diberi nama baru seperti bilangan menggunakan kata kunci type, sehingga pemrogram dapat menyesuaikan konteks semantik tipe data dengan kebutuhan aplikasi.

- **Struct(StrukturDataKomposit)**
  Struct merupakan kumpulan beberapa elemen data yang bisa memiliki tipe berbeda, tetapi digabungkan dalam satu entitas logis. Contoh umum adalah representasi waktu sebagai struct yang memuat atribut jam, menit, dan detik. Dengan struct, pengelolaan data yang saling terkait menjadi lebih terorganisasi dan efisien.

2. Array

Array adalah struktur data statis yang menyimpan sejumlah elemen bertipe sama dan diakses menggunakan indeks numerik yang dimulai dari nol. Di Go, ukuran array harus ditentukan saat deklarasi dan tidak dapat diubah selama runtime, sehingga cocok untuk kasus di mana ukuran data telah diketahui sejak awal.

3. Slice dan Map: Struktur Data Dinamis

Untuk kebutuhan pengelolaan data yang lebih fleksibel, Go menyediakan slice dan map sebagai alternatif struktur dinamis. Slice merupakan abstraksi dari array yang memungkinkan ukuran dinamis dan dapat diperbesar menggunakan fungsi append. Slice mereferensikan array dasar, sehingga perubahan pada slice dapat memengaruhi array tersebut. Sementara itu, map adalah struktur data asosiatif yang menyimpan pasangan kunci-nilai, di mana kunci dapat berupa berbagai tipe data seperti string atau integer. Map sangat berguna ketika data perlu diakses secara efisien berdasarkan identifikasi unik (key), bukan sekadar indeks numerik.

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:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL6> go run "d:\ALGORITMA PROGRAM
========================================
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 16:25
========================================
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:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL6>
```

DEKSRIPSI:

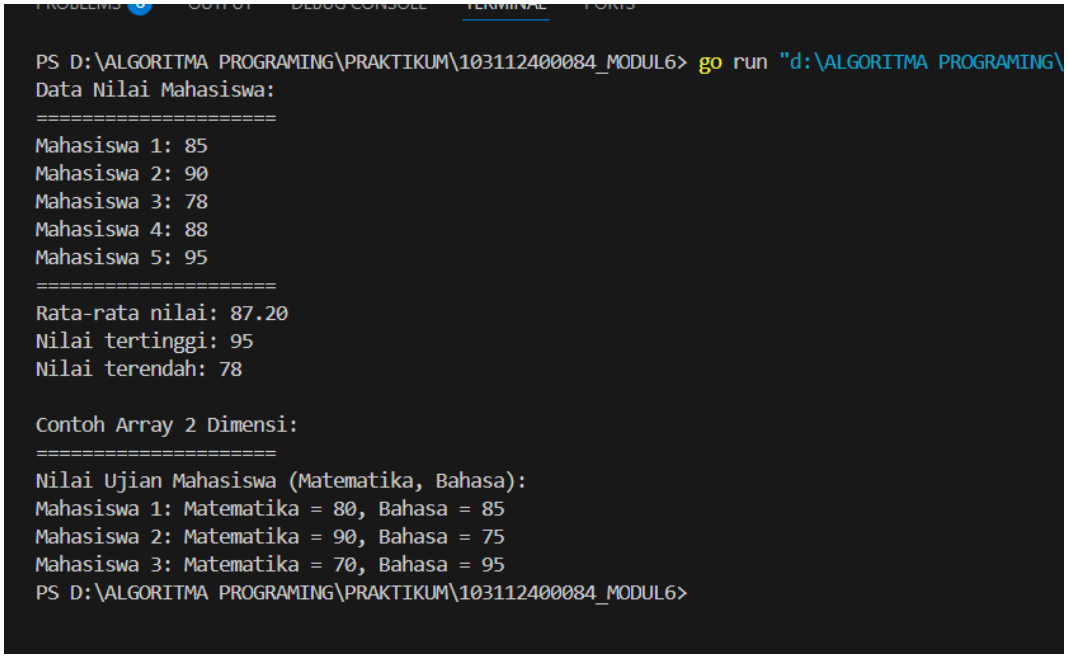
Program ini adalah simulasi pencetakan struk belanja, yang memanfaatkan konsep struct untuk merepresentasikan data barang dan struk secara terstruktur. Struct Item menyimpan informasi nama, harga, dan jumlah barang, sementara struct Receipt menyimpan detail toko, tanggal transaksi, daftar belanja, serta total pembayaran. Program menghitung total harga belanjaan melalui method CalculateTotal() dan mencetak struk lengkap secara terformat dengan PrintReceipt(), menampilkan informasi toko, tanggal, daftar barang lengkap dengan subtotal, dan total belanja. Dengan pendekatan modular dan berbasis objek, program ini mencerminkan praktik nyata dalam membangun sistem kasir sederhana yang rapi, informatif, dan siap dikembangkan lebih lanjut.

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:



DEKSRIPSI:

Program yang mendemonstrasikan penggunaan array satu dimensi dan dua dimensi dalam pengolahan data nilai mahasiswa. Pada bagian pertama, program menyimpan dan menampilkan nilai lima mahasiswa, lalu menghitung rata-rata, nilai tertinggi, dan nilai terendah dari array satu dimensi. Selanjutnya, program memperkenalkan array dua dimensi untuk merepresentasikan nilai ujian tiga mahasiswa dalam dua mata kuliah—Matematika dan Bahasa—dan menampilkannya secara terstruktur. Melalui pendekatan ini, program memperkuat pemahaman dasar tentang manipulasi array, perulangan, dan logika analitis yang sering diterapkan dalam pemrosesan data akademik.

I. UNGUIDED

UNGUIDED 1

SOURCE CODE

```
package main
//Nufail Alauddin Tsaqif
//013112400084
import (
    "fmt"
    "math"
)

type Titik struct {
    X, Y float64
}

type Lingkaran struct {
    Pusat  Titik
    Radius float64
}

func hitungJarak(a, b Titik) float64 {
    return math.Hypot(a.X-b.X, a.Y-b.Y)
}

func titikDalamLingkaran(l Lingkaran, t Titik) bool {
    return hitungJarak(l.Pusat, t) <= l.Radius
}

func inputLingkaran(nama string) Lingkaran {
    var x, y, r float64
    fmt.Printf("Masukkan %s (x y r): ", nama)
    fmt.Scan(&x, &y, &r)
    return Lingkaran{Pusat: Titik{X: x, Y: y}, Radius: r}
}

func inputTitik() Titik {
    var x, y float64
    fmt.Print("Masukkan koordinat titik (x y): ")
    fmt.Scan(&x, &y)
    return Titik{X: x, Y: y}
}
```

```
func main() {
    lingkaran1 := inputLingkaran("lingkaran 1")
    lingkaran2 := inputLingkaran("lingkaran 2")
    titik := inputTitik()

    dalamL1 := titikDalamLingkaran(lingkaran1, titik)
    dalamL2 := titikDalamLingkaran(lingkaran2, titik)

    if dalamL1 && dalamL2 {
        fmt.Println("Titik berada di dalam lingkaran 1 dan 2.")
    } else if dalamL1 {
        fmt.Println("Titik berada di dalam lingkaran 1 saja.")
    } else if dalamL2 {
        fmt.Println("Titik berada di dalam lingkaran 2 saja.")
    } else {
        fmt.Println("Titik berada di luar kedua lingkaran.")
    }
}
```

OUTPUT

```
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL6> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\10311
Masukkan lingkaran 1 (x y r): 1 1 5
Masukkan lingkaran 2 (x y r): 8 8 4
Masukkan koordinat titik (x y): 2 2
Titik berada di dalam lingkaran 1 saja.
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL6> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\10311
Masukkan lingkaran 1 (x y r): 1 2 3
Masukkan lingkaran 2 (x y r): 4 5 6
Masukkan koordinat titik (x y): 7 8
Titik berada di dalam lingkaran 2 saja.
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL6> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\10311
Masukkan lingkaran 1 (x y r): 5 10 15
Masukkan lingkaran 2 (x y r): - 15 4 20
Masukkan koordinat titik (x y): Titik berada di dalam lingkaran 1 saja.
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL6> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\10311
Masukkan lingkaran 1 (x y r): 1 1 5
Masukkan lingkaran 2 (x y r): 8 8 4
Masukkan koordinat titik (x y): 15 20
Titik berada di luar kedua lingkaran.
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL6>
```

DEKSRIPSI

Program yang digunakan untuk menentukan posisi relatif suatu titik terhadap dua buah lingkaran pada bidang kartesian. Pengguna diminta untuk memasukkan koordinat pusat dan jari-jari dari masing-masing lingkaran, serta koordinat dari sebuah titik. Program kemudian menghitung jarak Euclidean antara titik tersebut dan pusat setiap lingkaran menggunakan fungsi math.Hypot, lalu membandingkan jarak itu dengan jari-jari masing-masing lingkaran untuk menentukan apakah titik berada di dalam salah satu, kedua, atau di luar lingkaran tersebut. Output program berupa keterangan posisi titik yang diinformasikan secara eksplisit berdasarkan hasil perhitungan tersebut. Program ini disusun secara modular dan hanya memanfaatkan paket fmt untuk interaksi input/output serta math untuk perhitungan matematis.

UNGUIDED 2

SOURCE CODE

```
package main
// Nufail Alauddin Tsaqif
// 103112400084
import "fmt"
func hitungRataRata(data []int) float64 {
    total := 0
    for _, nilai := range data {
        total += nilai
    }
    return float64(total) / float64(len(data))
}

func hitungSimpanganBaku(data []int) float64 {
    rata := hitungRataRata(data)
    var jumlahKuadrat float64
    for _, nilai := range data {
        selisih := float64(nilai) - rata
        jumlahKuadrat += selisih * selisih
    }
    return jumlahKuadrat / float64(len(data))
}

func hitungFrekuensi(data []int, target int) int {
    hitung := 0
    for _, nilai := range data {
        if nilai == target {
            hitung++
        }
    }
    return hitung
}

func tampilkanBerdasarkanIndeks(data []int, ganjil bool) {
    for i := 0; i < len(data); i++ {
        if ganjil && i%2 != 0 {
            fmt.Print(data[i], " ")
        }
        if !ganjil && i%2 == 0 {
            fmt.Print(data[i], " ")
        }
    }
    fmt.Println()
```

```
}

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

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

    fmt.Println("\nIsi array:", angka)

    fmt.Print("Elemen pada indeks ganjil: ")
    tampilkanBerdasarkanIndeks(angka, true)

    fmt.Print("Elemen pada indeks genap: ")
    tampilkanBerdasarkanIndeks(angka, false)

    fmt.Printf("Rata-rata: %.2f\n", hitungRataRata(angka))
        fmt.Printf("Simpangan baku (tanpa akar): %.2f\n",
hitungSimpanganBaku(angka))

    var cari int
    fmt.Print("Masukkan angka yang ingin dicari frekuensinya: ")
    fmt.Scan(&cari)
    fmt.Printf("Frekuensi angka %d: %d\n", cari, hitungFrekuensi(angka, cari))
}
```

OUTPUT

```
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL6> go run "d:\ALGORIT
Masukkan jumlah elemen: 3
Masukkan elemen ke-0: 1
Masukkan elemen ke-1: 2
Masukkan elemen ke-2: 3

Isi array: [1 2 3]
Elemen pada indeks ganjil: 2
Elemen pada indeks genap: 1 3
Rata-rata: 2.00
Simpangan baku (tanpa akar): 0.67
Masukkan angka yang ingin dicari frekuensinya:
```

DEKSRIPSI

Program yang memungkinkan pengguna menginput sejumlah bilangan bulat untuk dianalisis. Program menghitung dan menampilkan rata-rata, simpangan baku tanpa akar, serta membedakan elemen berdasarkan indeks ganjil dan genap. Selain itu, pengguna dapat memasukkan sebuah angka untuk mengetahui frekuensi kemunculannya dalam array. Seluruh perhitungan dilakukan secara manual tanpa menggunakan pustaka eksternal, dengan pendekatan modular yang memudahkan pemeliharaan dan pengembangan kode.

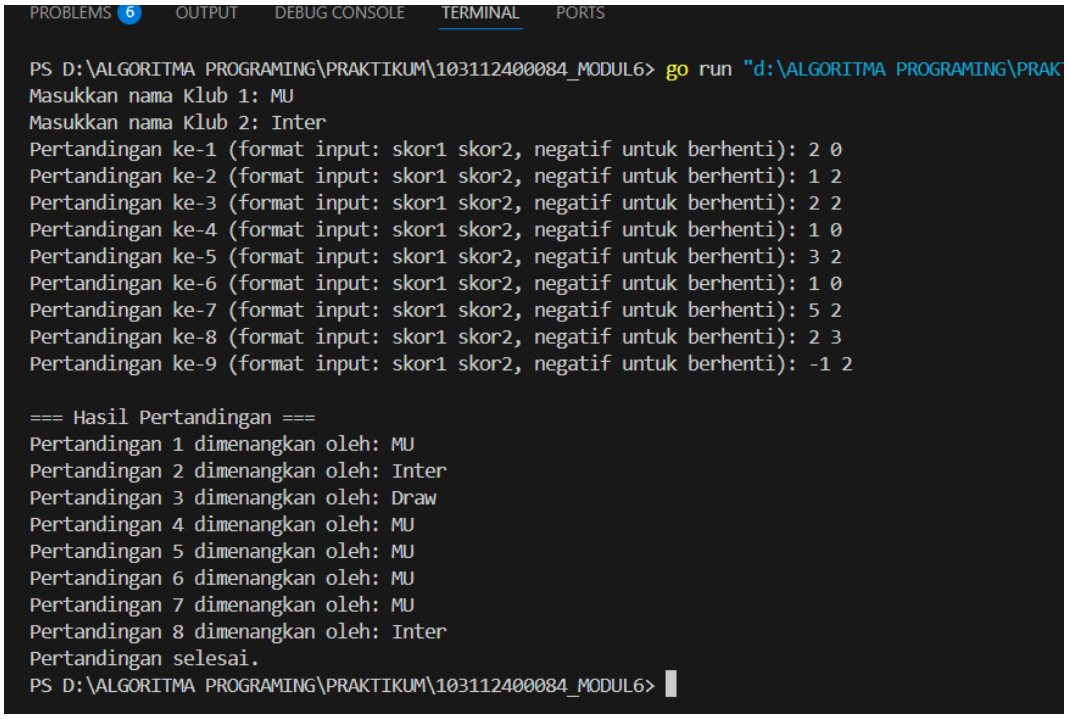
UNGUIDED 3

SOURCE CODE

```
package main
// Nufail Alauddin Tsaqif
// 103112400084
import "fmt"
func tampilkanHasil(hasil []string) {
    fmt.Println("\n=== Hasil Pertandingan ===")
    for i, pemenang := range hasil {
        fmt.Printf("Pertandingan %d dimenangkan oleh: %s\n", i+1, pemenang)
    }
    fmt.Println("Pertandingan selesai.")
}

func main() {
    var klub1, klub2 string
    var skor1, skor2 int
    var hasil []string
    nomor := 1

    fmt.Print("Masukkan nama Klub 1: ")
    fmt.Scan(&klub1)

    fmt.Print("Masukkan nama Klub 2: ")
    fmt.Scan(&klub2)

    for {
        fmt.Printf("Pertandingan ke-%d (format input: skor1 skor2, negatif untuk berhenti): ", nomor)
        fmt.Scan(&skor1, &skor2)

        if skor1 < 0 || skor2 < 0 {
            break
        }

        switch {
        case skor1 > skor2:
            hasil = append(hasil, klub1)
        case skor2 > skor1:
            hasil = append(hasil, klub2)
        default:
            hasil = append(hasil, "Draw")
        }
```

```
        nomor++
    }

    tampilkanHasil(hasil)
}
```

OUTPUT



DEKSRIPSI

Program yang digunakan untuk merekap hasil pertandingan antara dua klub sepak bola berdasarkan skor yang dimasukkan pengguna. Pengguna diminta memasukkan nama kedua klub, kemudian secara berulang memasukkan skor masing-masing klub untuk setiap pertandingan. Program menentukan pemenang dari setiap pertandingan—baik klub pertama, klub kedua, atau hasil seri—dan menyimpannya dalam daftar. Proses input berakhir jika skor negatif dimasukkan, setelah itu program menampilkan seluruh hasil pertandingan secara terformat. Pendekatan modular dengan fungsi khusus untuk menampilkan hasil membuat program ini lebih terstruktur dan mudah dikembangkan.

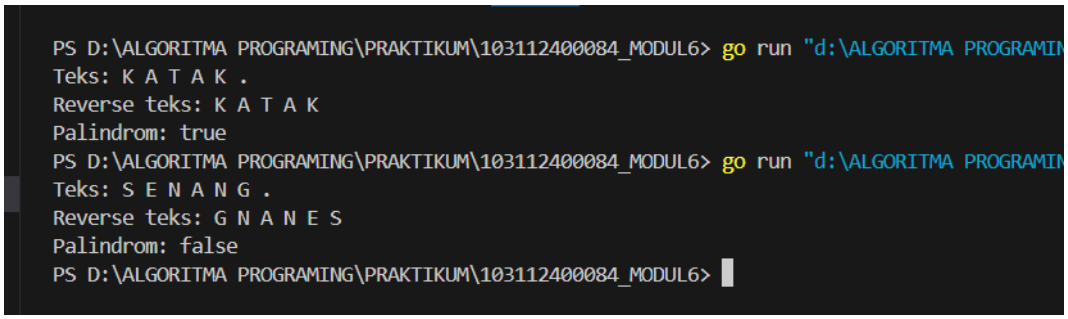
UNGUIDED 4

SOURCE CODE

```
package main
// Nufail Alauddin Tsaqif
//103112400084
import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
   var Teks rune
   *n = 0
   fmt.Print("Teks: ")
   for {
      fmt.Scanf("%c", &Teks)
      if Teks == '.' || *n > NMAX {
         break
      }
      if Teks != ' ' && Teks != '\n' {
         (*t)[*n] = Teks
         *n++
      }
   }
}

func cetakArray(t tabel, n int) {
   fmt.Print("Reverse teks: ")
   for i := 0; i < n; i++ {
      fmt.Printf("%c ", 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 palindrom(t tabel, n int) bool {
   var tabClone tabel
   copy(tabClone[:], t[:])
```

```
    balikanArray(&tabClone, n)
    for i := 0; i < n; i++ {
        if t[i] != tabClone[i] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var m int
    isiArray(&tab, &m)
    balikanArray(&tab, m)
    cetakArray(tab, m)
    fmt.Print("Palindrom: ", palindrom(tab, m))
}
```

OUTPUT



DEKSRIPSI

Program yang memproses teks karakter demi karakter hingga tanda titik (.) sebagai penanda akhir input. Teks yang dimasukkan akan disimpan dalam array bertipe rune, mengabaikan spasi dan karakter newline. Setelah itu, program membalik urutan karakter dalam array dan mencetaknya dalam bentuk terbalik. Selain itu, program juga melakukan pengecekan apakah urutan karakter tersebut merupakan palindrom—yakni susunan karakter yang sama jika dibaca dari depan maupun belakang—dengan membandingkan teks asli dan versi terbaliknya. Pendekatan ini menggunakan manipulasi array secara

langsung dan menghindari penggunaan pustaka eksternal, sehingga cocok sebagai latihan dasar manipulasi string dan struktur data di Go.

II. KESIMPULAN

Praktikum memperlihatkan penerapan nyata konsep-konsep fundamental dalam pemrograman, seperti array, slice, dan tipe bentukan (struct), dalam menyelesaikan berbagai permasalahan komputasional yang relevan dengan kehidupan sehari-hari. Dari menentukan letak titik terhadap lingkaran, menganalisis data numerik seperti rata-rata dan simpangan baku, hingga mengidentifikasi apakah sebuah kata merupakan palindrom, setiap studi kasus dirancang untuk memperkuat logika algoritmik serta keterampilan manipulasi struktur data secara sistematis. Praktikum ini tidak hanya mendorong mahasiswa menulis kode yang efisien dan modular, tetapi juga menekankan pentingnya representasi data yang tepat guna menghasilkan solusi yang akurat, terstruktur, dan mudah dioptimalkan dalam konteks pemrograman yang lebih kompleks.

REFERENSI

MODUL 6 PROSEDUR ALGORITMA PEMOGRAMAN 2