LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 6

MATERI

Oleh:

DAFFA TSAQIFNA FAUZTSANY

103112400032

S1 IF-12-01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I. DASAR TEORI

Struct & Array

1. Tipe Bentukan

Tipe bentukan (user-defined type) digunakan untuk membuat tipe data baru. Terdiri dari:

a. Alias (Type)
Mengganti nama tipe data agar lebih mudah digunakan.

```
type bilangan int
type pecahan float64
```

b. Struct
Mengelompokkan beberapa data dalam satu kesatuan.

```
type waktu struct {
   jam, menit, detik int
}
```

2. Array

Array menyimpan banyak nilai dalam satu variabel, dengan indeks dari 0 dan ukuran tetap.

```
var angka [5]int = [5]int{1, 2, 3, 4, 5}
fmt.Println(angka[0]) // akses elemen pertama
```

3. Slice

Versi dinamis dari array (ukuran fleksibel).

```
var sl = []int{1, 2, 3}
sl = append(sl, 4) // menambahkan elemen
```

4. Map

Struktur data mirip array, tetapi menggunakan kunci (key) bukan indeks angka.

```
var nilai map[string]int = map[string]int{"Ali": 90, "Budi": 80}
fmt.Println(nilai["Ali"])
```

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

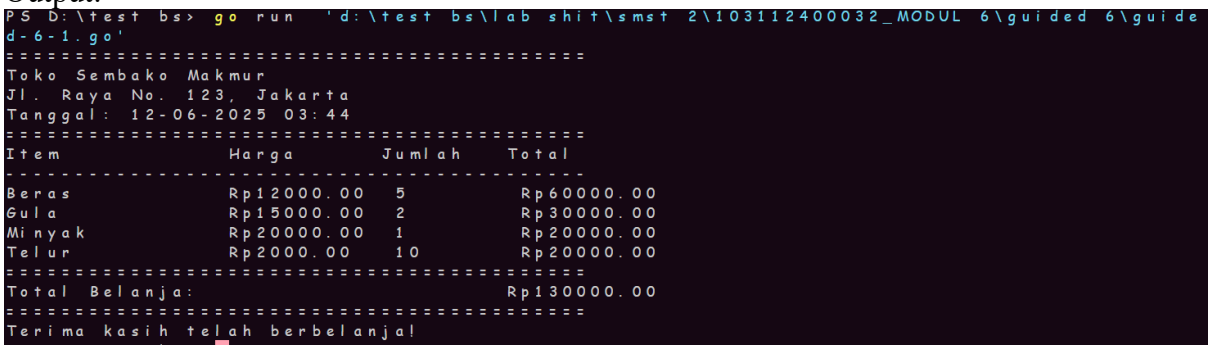
1. **GUIDED 1**

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:



Deskripsi Program:

Program ini membuat dan mencetak struk belanja lengkap menggunakan fitur struct, slice, dan method di bahasa Go. Program mengatur informasi toko, tanggal transaksi, dan daftar item belanja (nama, harga, jumlah), lalu mencetaknya dalam format tabel yang rapi.

Penjelasan Detail:

1. Struct Item

Digunakan untuk mewakili satu barang dalam daftar belanja:

- Name: nama barang.
- Price: harga per satuan (float).
- Quantity: jumlah barang.

2. Struct Receipt

Mewakili keseluruhan struk belanja, terdiri dari:

- StoreInfo: informasi toko.
- Date: tanggal dan waktu pembelian (menggunakan time.Time).
- Items: slice dari Item, berisi daftar belanja.
- TotalAmount: total biaya belanja.

3. Method CalculateTotal()

Mengiterasi seluruh Items dan menghitung total harga:

```
total += item.Price * float64(item.Quantity)
```

Hasilnya disimpan dalam TotalAmount.

4. Method PrintReceipt()

Menampilkan isi struk ke layar dengan format:

- Informasi toko dan tanggal (diformat: dd-mm-yyyy hh:mm)
- Daftar item: nama, harga satuan, jumlah, total per item
- Total keseluruhan (TotalAmount)
- Pesan penutup

5. Fungsi main()

- Membuat sebuah objek Receipt dengan data toko dan empat barang.
- Memanggil CalculateTotal() untuk menghitung total harga.
- Mencetak hasilnya dengan PrintReceipt().

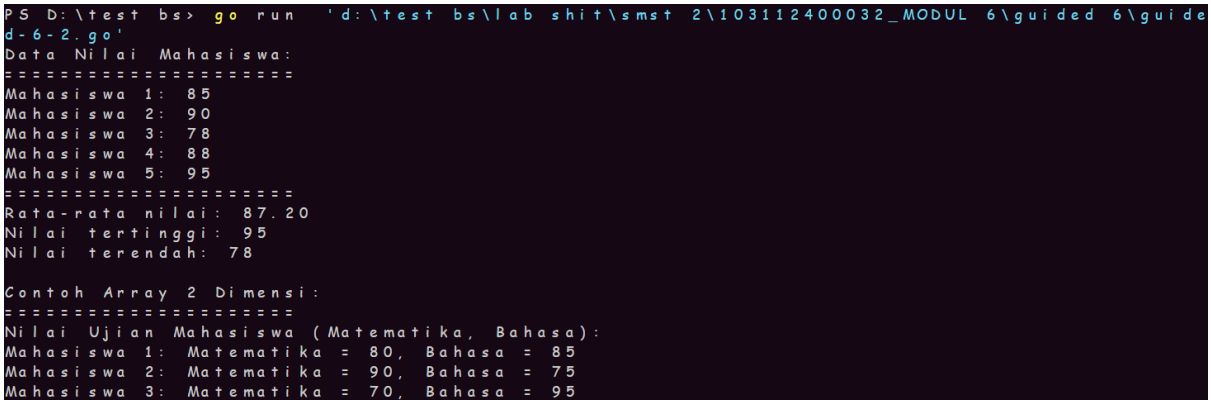
2. GUIDED 2

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:



Deskripsi Program:

Program ini digunakan untuk mengelola dan menampilkan data nilai mahasiswa, baik dalam bentuk array 1 dimensi (nilai akhir) maupun 2 dimensi (nilai beberapa mata kuliah). Program juga menghitung rata-rata, nilai tertinggi, dan terendah, serta menampilkan nilai masing-masing mahasiswa dengan format yang jelas.

Penjelasan Detail:

1. Array 1 Dimensi: Nilai Akhir Mahasiswa

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

   Menyimpan nilai akhir dari 5 mahasiswa.
2. Menampilkan Nilai Mahasiswa
   Menggunakan perulangan for range untuk mencetak nilai setiap mahasiswa:

```
Mahasiswa 1: 85
Mahasiswa 2: 90
...
```

3. Menghitung Rata-rata Nilai

```
Mahasiswa 1: 85
Mahasiswa 2: 90
...
```

   Total semua nilai dibagi jumlah mahasiswa.
   Tipe data dikonversi ke float64 agar hasil tidak dibulatkan.
4. Mencari Nilai Tertinggi dan Terendah
   Diperoleh dengan membandingkan setiap nilai terhadap nilai awal:

```
if nilai > tertinggi { ... }
if nilai < terendah { ... }
```

5. Array 2 Dimensi: Nilai per Mata Kuliah

```
nilaiUjian := [3][2]int{
        {80, 85},
        {90, 75},
        {70, 95},
}
```

   Menyimpan nilai dari 3 mahasiswa, masing-masing untuk dua mata kuliah: Matematika dan Bahasa.
6. Menampilkan Nilai 2D
   Menampilkan nilai berdasarkan indeks:

```
Mahasiswa 1: Matematika = 80, Bahasa = 85
```

Mahasiswa 2: Matematika = 90, Bahasa = 75
...

III. UNGUIDED

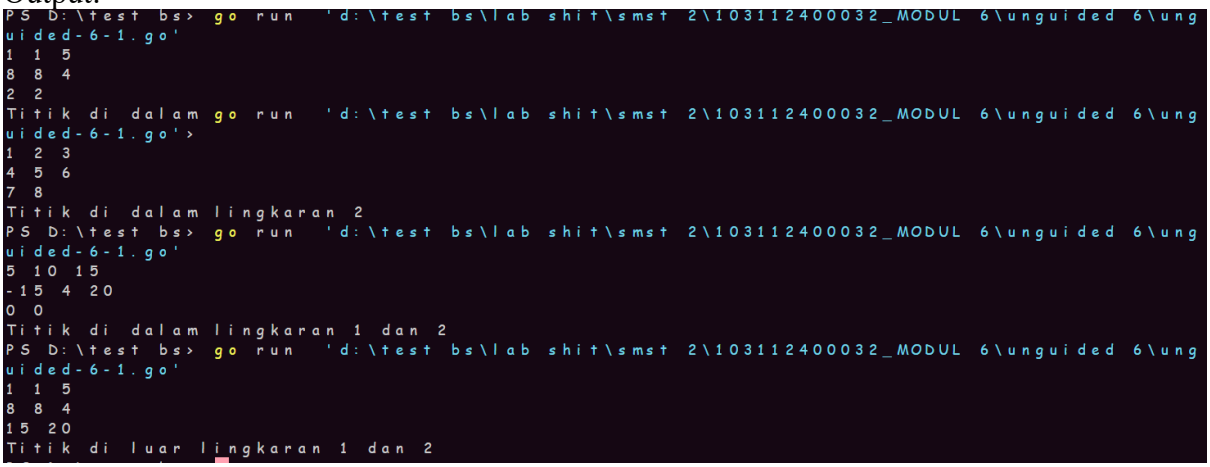
Source Code + Screenshot hasil program beserta penjelasan

1. UNGUIDED 1

Source Code:

```
package main

import (
    "fmt"
    "math"
)

type coor struct {
    x, y float64
}
type lingkaran struct {
    titik  coor
    radius float64
}

func jarak(a, b, c, d float64) float64 {
    return math.Sqrt(math.Pow(a-c, 2) + math.Pow(b-d, 2))
}
func isInside(circle lingkaran, point coor) bool {
    d := jarak(circle.titik.x, circle.titik.y, point.x, point.y)
    return d <= circle.radius
}
func main() {
    var cx1, cx2, cy1, cy2, r1, r2, x, y float64
    fmt.Scan(&cx1, &cy1, &r1)
    fmt.Scan(&cx2, &cy2, &r2)
    fmt.Scan(&x, &y)
    l1 := lingkaran{titik: coor{cx1, cy1}, radius: r1}
    l2 := lingkaran{titik: coor{cx2, cy2}, radius: r2}
    titik := coor{x, y}
    in1 := isInside(l1, titik)
    in2 := isInside(l2, titik)
    switch {
    case in1 && in2:
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    case in1:
        fmt.Println("Titik di dalam lingkaran 1")
    case in2:
        fmt.Println("Titik di dalam lingkaran 2")
    default:
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

Output:



Deskripsi Program:

Program ini digunakan untuk menentukan posisi suatu titik terhadap dua buah lingkaran menggunakan pendekatan struktur data (struct). Struct coor mewakili koordinat titik 2D, sedangkan lingkaran berisi pusat dan jari-jarinya. Program menghitung jarak titik terhadap pusat lingkaran menggunakan rumus Euclidean, lalu memeriksa apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, dan mencetak hasil sesuai kondisinya.

2. UNGUIDED 2

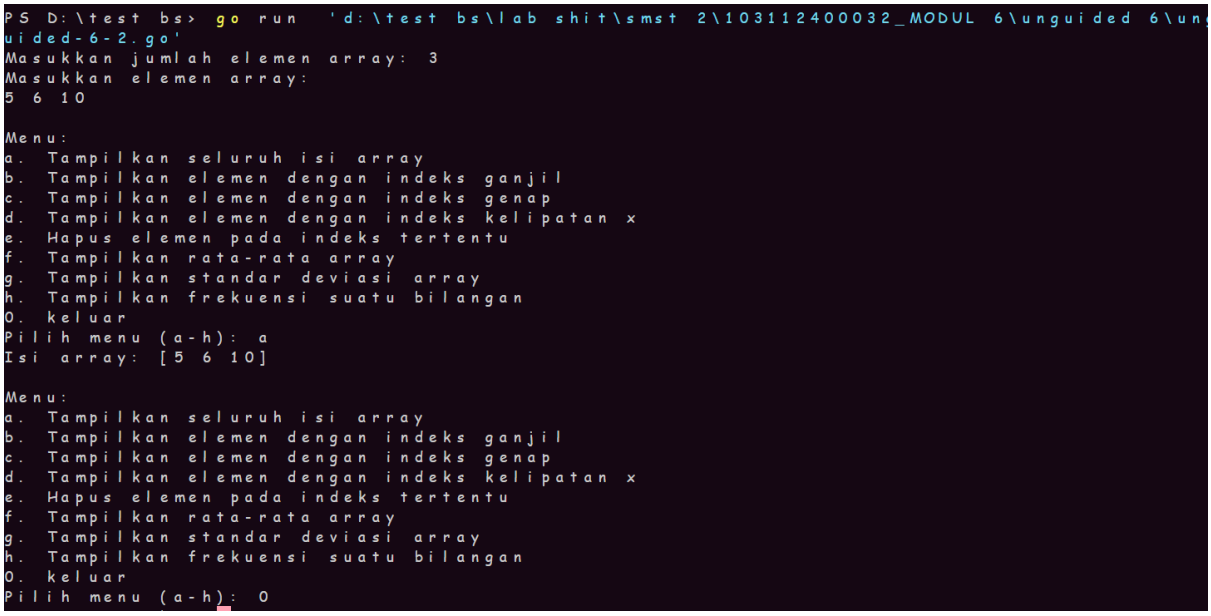
Source Code:

```
package main

import (
    "fmt"
    "math"
)

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

    arr := make([]int, n)

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

    for {
        fmt.Println("\nMenu:")
        fmt.Println("a. Tampilkan seluruh isi array")
        fmt.Println("b. Tampilkan elemen dengan indeks ganjil")
```

```
fmt.Println("c. Tampilkan elemen dengan indeks genap")
fmt.Println("d. Tampilkan elemen dengan indeks kelipatan x")
fmt.Println("e. Hapus elemen pada indeks tertentu")
fmt.Println("f. Tampilkan rata-rata array")
fmt.Println("g. Tampilkan standar deviasi array")
fmt.Println("h. Tampilkan frekuensi suatu bilangan")
fmt.Println("0. keluar")
var opsi string
fmt.Print("Pilih menu (a-h): ")
fmt.Scan(&opsi)

switch opsi {
case "a":
   fmt.Println("Isi array:", arr)

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

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

case "d":
   var x int
   fmt.Print("Masukkan nilai x: ")
   fmt.Scan(&x)
   fmt.Print("Indeks kelipatan ", x, ": ")
   for i := 0; i < len(arr); i++ {
      if i%x == 0 {
         fmt.Print(arr[i], " ")
      }
   }
   fmt.Println()

case "e":
   var index int
   fmt.Print("Masukkan indeks yang ingin dihapus: ")
   fmt.Scan(&index)
   if index >= 0 && index < len(arr) {
      arr = append(arr[:index], arr[index+1:]...)
      fmt.Println("Array setelah penghapusan:", arr)
   } else {
      fmt.Println("Indeks tidak valid")
```

```
            }

        case "f":
            sum := 0
            for _, v := range arr {
                sum += v
            }
            rata := float64(sum) / float64(len(arr))
            fmt.Println("Rata-rata:", rata)

        case "g":
            sum := 0
            for _, v := range arr {
                sum += v
            }
            rata := float64(sum) / float64(len(arr))
            var std float64
            for _, v := range arr {
                std += math.Pow(float64(v)-rata, 2)
            }
            std = math.Sqrt(std / float64(len(arr)))
            fmt.Println("Standar deviasi:", std)

        case "h":
            var target int
            fmt.Print("Masukkan bilangan yang ingin dicari frekuensinya: ")
            fmt.Scan(&target)
            count := 0
            for _, v := range arr {
                if v == target {
                    count++
                }
            }
            fmt.Println("Frekuensi:", count)
        case "0":
            return

        default:
            fmt.Println("Opsi tidak dikenali.")
        }
    }
}
```

Output:



Deskripsi Program:

Program ini digunakan untuk memanipulasi dan menganalisis isi array berdasarkan pilihan menu interaktif. Setelah pengguna mengisi array dengan sejumlah elemen, program menyediakan berbagai opsi: menampilkan seluruh isi array, menampilkan elemen berdasarkan indeks (ganjil, genap, kelipatan tertentu), menghapus elemen di indeks tertentu, menghitung rata-rata, standar deviasi, serta mencari frekuensi kemunculan suatu bilangan. Program berjalan dalam perulangan hingga pengguna memilih keluar melalui opsi menu.

3. UNGUIDED 3

Source Code:

```
package main

import "fmt"

func main() {
    var tim1, tim2 string
    var score1, score2, x int
    var result []string
    x = 0
    fmt.Print("nama tim 1: ")
    fmt.Scan(&tim1)
    fmt.Print("nama tim 2: ")
    fmt.Scan(&tim2)
    for {
        fmt.Print("score pertandingan = ")
        _, err := fmt.Scan(&score1, &score2)
        if err != nil || score1 < 0 || score2 < 0 {
            fmt.Println("Pertandingan selesai")
            break
        }
```

```
        if score1 > score2 {
            result = append(result, tim1)
        } else if score1 < score2 {
            result = append(result, tim2)
        } else if score1 == score2 {
            result = append(result, "draw")
        }
        x += 1
    }
    for i := 0; i < x; i++ {
        fmt.Printf("Hasil %d : %s\n", i+1, result[i])
    }
}
```

Output:

```
PS D:\test bs> go run 'd:\test bs\lab shit\smst 2\103112400032_MODUL 6\unguided 6\unguided-6-3.go'
nama tim 1: MU
nama tim 2: Inter
score pertandingan = 2 0
score pertandingan = 1 2
score pertandingan = 2 2
score pertandingan = 0 1
score pertandingan = 3 2
score pertandingan = 1 0
score pertandingan = 5 2
score pertandingan = 2 3
score pertandingan = -1 2
Pertandingan selesai
Hasil 1 : MU
Hasil 2 : Inter
Hasil 3 : draw
Hasil 4 : Inter
Hasil 5 : MU
Hasil 6 : MU
Hasil 7 : MU
Hasil 8 : Inter
```

Deskripsi Program:

Program ini digunakan untuk mencatat dan menampilkan hasil dari sejumlah **pertandingan antara dua tim** berdasarkan skor yang dimasukkan pengguna. Pengguna memasukkan nama dua tim, lalu secara berulang memasukkan skor pertandingan. Untuk setiap pertandingan, program mencatat pemenang atau hasil **seri** ke dalam slice. Jika input skor tidak valid atau bernilai negatif, program menganggap pertandingan telah selesai dan mencetak seluruh hasil pertandingan secara berurutan.

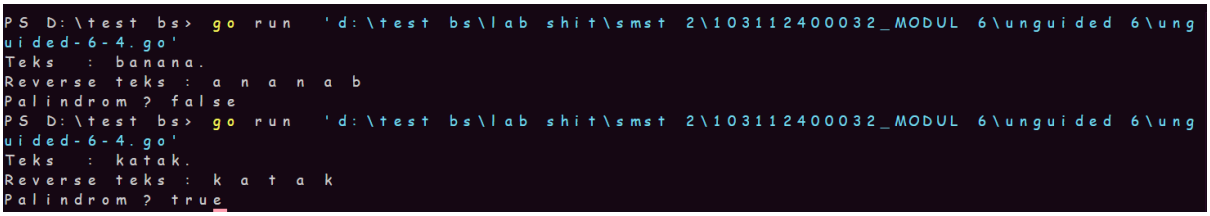
4. UNGUIDED 4

Source Code:

```
package main

import "fmt"

const Nmax int = 127

type table [Nmax]rune

func isiArray(t *table, n *int) {
    var c rune
    *n = 0
```

```
    fmt.Print("Teks  : ")
    for {
        fmt.Scanf("%c", &c)
        if c == '.' || *n >= Nmax {
            break
        }
        t[*n] = c
        *n++
    }
}

func cetakArray(t table, n int) {
    for i := 0; i < n; i++ {
        fmt.Printf("%c ", t[i])
    }
    fmt.Println()
}

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

func palindrom(t table, 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 table
    var m int

    isiArray(&tab, &m)

    fmt.Print("Reverse teks : ")
    var reversedTab table = tab // salin isi
    balikanArray(&reversedTab, m)
    cetakArray(reversedTab, m)

    fmt.Println("Palindrom ?", palindrom(tab, m))
}
```

Output:



Deskripsi Program:

Program ini digunakan untuk memproses teks dalam bentuk array karakter dan melakukan beberapa operasi terhadapnya. Teks dibaca karakter per karakter hingga karakter . atau batas maksimum tercapai, lalu disimpan ke dalam array. Program kemudian:

1. Mencetak versi terbalik dari teks tersebut.
2. Memeriksa apakah teks adalah palindrom (membaca sama dari depan dan belakang). Semua operasi dilakukan menggunakan array statis dan manipulasi indeks, tanpa menggunakan string atau slice bawaan.