

**LAPORAN PRAKTIKUM
ALGORITMA PEMOGRAMAN 2**

MODUL 7

MATERI

STRUCK & ARRAY



Oleh:

FEBRIAN FALIH ALWAFI

2311102181

S1F-11-02

**S1 TEKNIK INFORMATIKA
UNIVERSITAS TELKOM PURWOKERTO**

2024

I. DASAR TEORI

➤ 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 data dengan nama baru yang lebih ringkas dan familiar. Sebagai contoh "Integer" dapat dirubah 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>	
3		type <nama alias> <tipe data>
4	algoritma	
5	...	func main(){
6		...
7		}

2) Struct atau Record

Stucture memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari stucture tersebut. Berbeda dengan bahasa pemrograman lain. kesamaan tipe dari dua variabel berjenis stucture bukan karena namanya tetapi karena strukturnya. 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 stucture tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi stucture tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

❖ Array

Array mempunyai urutan (jumlah elemen) yang tetap (statis) selama eksekusi program, sehingga Jumlah elemen array menjadi bagian dari deklarasi variabel dengan tipe array.

Indeks array dimulai dari 0, sehingga indeks air pada contoh adalah 0, 1.. len(arr)-1

```
// Mengganti isi elemen ke-0 dengan nilai dari elemen ke-7
arr[0] = arr[7]

// Mengambil data field x dari elemen ke-i
currX = arr[i].center.x

// Mengambil elemen terakhir
n := len(arr)
buf := arr[n-1]
```

- **Slice (Array dinamik)**

Array dalam Go juga dapat mempunyai ukuran yang dinamik. (Tidak digunakan di kelas Algoritma Pemrograman). Deklarasinya mirip dengan deklarasi array, tetapi jumlah elemennya dikosongkan.

- **Map**

Tipe array lain, sebuah array dinamik. Indeksnya (di sini disebut kunci) tidak harus berbentuk integer. Indeks dapat berasal dari tipe apa saja. Struktur ini disebut map.

```
// Deklarasi variabel dct sebagai map bilangan bulat dengan kunci string
var dct map[string]int

// Deklarasi map lain dct1 dari elemen string dengan kunci juga string
// Mempunyai nilai awal dct1["john"] = "hi", dct1["anne"] = "darling"
var dct1 = map[string]string{ "john":"hi", "anne":"darling" }

// Deklarasi dan prealokasi tempat untuk map dct2
var dct2 map[float64]int = make(map[float64]int, 10)

// Mengambil nilai yang tersimpan dengan kunci "john"
fmt.Println( dct1["john"] )

// Mengganti nilai yang tersimpan pada kunci "anne", dan
// Membuat entri baru dengan kunci "boy"
dct1["anne"] = "lovely"
dct1["boy"] = "runaround"
```

```
// Menghapus entri dengan kunci "john"  
delete(dct1, "john")
```

II. GUIDED

1) 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\lenovo\Documents\GUIDED_Modul 7> go run "c:\Users\lenovo\Documents\GUIDED_Modul 7\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\lenovo\Documents\GUIDED_Modul 7>

```

Deskripsi Program :

Program di atas berfungsi untuk mengelola data mahasiswa dan menghitung rata-rata nilai mereka dalam beberapa mata pelajaran. Program ini menggunakan struct bernama Mahasiswa untuk menampung data seperti nama mahasiswa, nilai matematika, fisika, kimia, dan nilai rata-rata. Setiap data mahasiswa disimpan dalam atribut-atribut yang relevan, yang diatur dalam sebuah array slice.

2) GUIDED 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\lenovo\Documents\GUIDED_Modul 7>
go run "c:\Users\lenovo\Documents\GUIDED_Modul 7\guided2.go"
Daftar Mahasiswa:
NIM      Nama
-----
20231003  Cici
20231004  Dedi
20231001  Andi
20231002  Budi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM      Nama
-----
20231001  Andi
20231002  Budi
20231004  Dedi
PS C:\Users\lenovo\Documents\GUIDED_Modul 7>

```

Deskripsi Program :

Program di atas menggunakan map yang berfungsi untuk mengelola data mahasiswa, di mana NIM (Nomor Induk Mahasiswa) berfungsi sebagai kunci dan nama mahasiswa sebagai nilai. Dengan pendekatan ini, program memungkinkan berbagai operasi untuk dilakukan, seperti menambah, mengakses, menampilkan, dan menghapus data mahasiswa. Di awal program, map mahasiswa

dideklarasikan dan diinisialisasi dengan beberapa pasangan data NIM dan nama mahasiswa. Setiap pasangan data ditambahkan dalam format kunci: nilai, dengan tiga data awal yaitu NIM "20231001", "20231002", dan "20231003", yang masing-masing merujuk ke nama "Andi", "Budi", dan "Cici". Setelah itu, data baru dengan NIM "20231004" dan nama "Dedi" ditambahkan secara terpisah ke dalam map dengan menggunakan kunci 20231004 sebagai pengaksesnya.

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
}

// Fungsi untuk memeriksa apakah titik berada di
// dalam lingkaran dengan menghitung jaraknya
func didalam(c lingkaran, p titik) bool {
    // Menghitung jarak antara pusat lingkaran dan
    titik p
```



```

    jarak := math.Sqrt(float64((c.pusat.x-
p.x)*(c.pusat.x-p.x) + (c.pusat.y-p.y)*(c.pusat.y-
p.y)))
    return jarak <= float64(c.radius)
}

func main() {
    var cx1, cy1, r1_2311102181 int
    fmt.Println("Masukkan koordinat pusat dan
radius lingkaran 1:")
    fmt.Scan(&cx1, &cy1, &r1_2311102181)
    lingkaran1 := lingkaran{pusat: titik{x: cx1, y:
cy1}, radius: r1_2311102181}

    var cx2, cy2, r2 int
    fmt.Println("Masukkan koordinat pusat dan
radius lingkaran 2:")
    fmt.Scan(&cx2, &cy2, &r2)
    lingkaran2 := lingkaran{pusat: titik{x: cx2, y:
cy2}, radius: r2}

    var x, y int
    fmt.Println("Masukkan koordinat titik
sembarang:")
    fmt.Scan(&x, &y)
    titikSembarang := titik{x: x, y: y}

    diLingkaran1 := didalam(lingkaran1,
titikSembarang)
    diLingkaran2 := didalam(lingkaran2,
titikSembarang)

    if diLingkaran1 && diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if diLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {

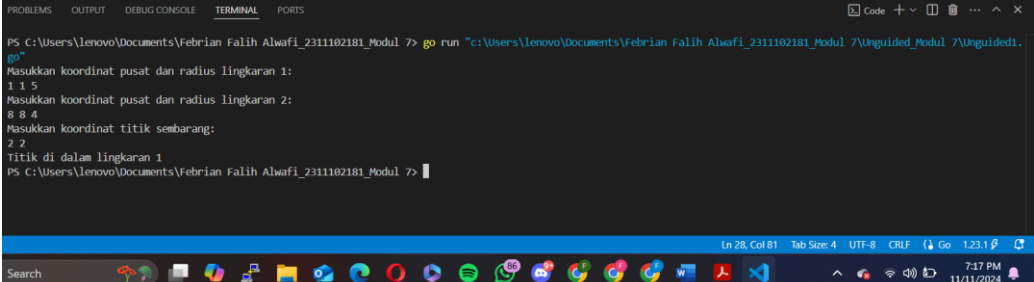
```

```

        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

Output :



```

PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7> go run "c:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7\Unguided_Modul 7\Unguided1.go"
Masukkan koordinat pusat dan radius lingkaran 1:
1 1 5
Masukkan koordinat pusat dan radius lingkaran 2:
8 8 4
Masukkan koordinat titik sembarang:
2 2
Titik di dalam lingkaran 1
PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7>

```

Deskripsi Program :

Program di atas memeriksa apakah sebuah titik tertentu berada di dalam satu atau dua lingkaran berdasarkan jaraknya dari pusat lingkaran. Program ini meminta input dari pengguna untuk menentukan pusat dan radius dua lingkaran, serta koordinat titik yang akan diperiksa posisinya. Berdasarkan data ini, program akan mengidentifikasi apakah titik tersebut berada di dalam lingkaran pertama, kedua, keduanya, atau di luar keduanya. Program menggunakan struktur if-else untuk menentukan posisi titik relatif terhadap kedua lingkaran, dan menampilkan pesan hasilnya. Jika titik berada di dalam kedua lingkaran, program menampilkan pesan "Titik di dalam lingkaran 1 dan 2." Jika titik hanya berada di dalam salah satu lingkaran, ditampilkan pesan yang relevan, dan jika titik berada di luar keduanya, program menampilkan "Titik di luar lingkaran 1 dan 2."

2. UNGUIDED 2

Source Code :

```

package main

import (
    "fmt"

```

```

    "math"
)

// Fungsi untuk menghapus elemen pada indeks tertentu
dalam slice
func hapusItem(slice []int, pos int) []int {
    return append(slice[:pos], slice[pos+1:]...)
}

// Fungsi untuk menghitung rata-rata dari elemen-
elemen dalam slice
func hitungRatarata_2311102181(slice []int) float64 {
    total := 0
    for _, nilai := range slice {
        total += nilai
    }
    return float64(total) / float64(len(slice))
}

// Fungsi untuk menghitung standar deviasi dari elemen-
elemen dalam slice
func hitungStandarDeviasi(slice []int) float64 {
    mean := hitungRatarata_2311102181(slice)
    var varian float64
    for _, nilai := range slice {
        varian += math.Pow(float64(nilai)-mean, 2)
    }
    return math.Sqrt(varian / float64(len(slice)))
}

// Fungsi untuk menghitung jumlah kemunculan elemen
tertentu dalam slice
func hitungFrekuensi(slice []int, target int) int {
    jumlah := 0
    for _, nilai := range slice {
        if nilai == target {
            jumlah++
        }
    }
}

```

```
    return jumlah
}

func main() {
    // Menerima jumlah elemen dari pengguna dan
    mengisi slice
    var jumlahElemen int
    fmt.Print("Masukkan jumlah elemen dalam array: ")
    fmt.Scan(&jumlahElemen)

    data := make([]int, jumlahElemen)
    fmt.Println("Masukkan elemen-elemen array:")
    for i := 0; i < jumlahElemen; i++ {
        fmt.Scan(&data[i])
    }

    // a. Menampilkan seluruh isi dari array
    fmt.Println("Isi lengkap array:", data)

    // b. Menampilkan elemen pada indeks ganjil
    fmt.Print("Elemen di indeks ganjil: ")
    for i := 1; i < len(data); i += 2 {
        fmt.Print(data[i], " ")
    }
    fmt.Println()

    // c. Menampilkan elemen pada indeks genap
    fmt.Print("Elemen di indeks genap: ")
    for i := 0; i < len(data); i += 2 {
        fmt.Print(data[i], " ")
    }
    fmt.Println()

    // d. Menampilkan elemen pada indeks kelipatan
    bilangan tertentu
    var kelipatan int
    fmt.Print("Masukkan nilai kelipatan untuk indeks: ")
    fmt.Scan(&kelipatan)
```

```

        fmt.Printf("Elemen pada indeks kelipatan %d: ",
kelipatan)
        for i := kelipatan; i < len(data); i += kelipatan {
            fmt.Print(data[i], " ")
        }
        fmt.Println()

// e. Menghapus elemen pada indeks tertentu
var posisi int
fmt.Print("Masukkan indeks yang ingin dihapus: ")
fmt.Scan(&posisi)
if posisi >= 0 && posisi < len(data) {
    data = hapusItem(data, posisi)
    fmt.Println("Isi array setelah penghapusan:", data)
} else {
    fmt.Println("Indeks tidak valid!")
}

// f. Menampilkan rata-rata dari elemen-elemen dalam
array
fmt.Printf("Rata-rata elemen dalam array: %.2f\n",
hitungRatarata_2311102181(data))

// g. Menampilkan standar deviasi dari elemen-elemen
dalam array
fmt.Printf("Standar deviasi elemen dalam array:
%.2f\n", hitungStandarDeviasi(data))

// h. Menghitung frekuensi dari elemen tertentu dalam
array
var elemenCari int
fmt.Print("Masukkan elemen yang ingin dihitung
frekuensinya: ")
fmt.Scan(&elemenCari)
fmt.Printf("Frekuensi kemunculan elemen %d dalam
array: %d\n", elemenCari, hitungFrekuensi(data,
elemenCari))
}

```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7> go run "c:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7\Unguided_Modul 7\Unguided2.go"
Masukkan jumlah elemen dalam array: 10
Masukkan elemen-elemen array:
0 1 2 3 4 5 6 7 8 9
Isi lengkap array: [0 1 2 3 4 5 6 7 8 9]
Elemen di indeks ganjil: 1 3 5 7 9
Elemen di indeks genap: 0 2 4 6 8
Masukkan nilai kelipatan untuk indeks: 6
Elemen pada indeks kelipatan 6: 6
Masukkan indeks yang ingin dihapus: 5
Isi array setelah penghapusan: [0 1 2 3 4 6 7 8 9]
Rata-rata elemen dalam array: 4.44
Standar deviasi elemen dalam array: 3.02
Masukkan elemen yang ingin dihitung frekuensinya: 5
Frekuensi kemunculan elemen 5 dalam array: 0
PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7> 
```

Deskripsi Program :

Program di atas adalah yang berfungsi untuk mengelola data dalam bentuk array (atau slice) integer. Program ini memulai dengan meminta pengguna untuk memasukkan sejumlah elemen integer, kemudian menyimpannya dalam sebuah slice. Setelah data terkumpul, program menyediakan berbagai fungsi untuk melakukan operasi dan analisis dasar pada data tersebut, termasuk menampilkan elemen-elemen array pada indeks tertentu, menghitung rata-rata, menghitung standar deviasi, dan mencari frekuensi kemunculan elemen tertentu dalam array. Fungsi dimulai dengan menerima input jumlah elemen dari pengguna, lalu mengisi array sesuai dengan jumlah elemen yang dimasukkan. Program kemudian menampilkan seluruh isi array dan menampilkan elemen pada indeks ganjil serta genap secara terpisah. Selain itu, program juga memiliki fitur untuk menampilkan elemen pada indeks yang merupakan kelipatan dari bilangan yang dimasukkan oleh pengguna.

3. UNGUIDED 3

Source Code :

```
package main

import "fmt"
```

```

type Pertandingan struct {
    klub1, klub2 string
    hasil []string
}

func main() {
    var p Pertandingan
    var skorA_2311102181, skorB int

    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&p.klub1)
    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scanln(&p.klub2)

    // Memulai loop untuk input skor pertandingan
    for j := 1; ; j++ {
        fmt.Printf("Masukkan skor pertandingan %v : ", j)
        fmt.Scanln(&skorA_2311102181, &skorB)

        // Mengakhiri loop jika skor negatif ditemukan
        if skorA_2311102181 < 0 || skorB < 0 {
            break
        }

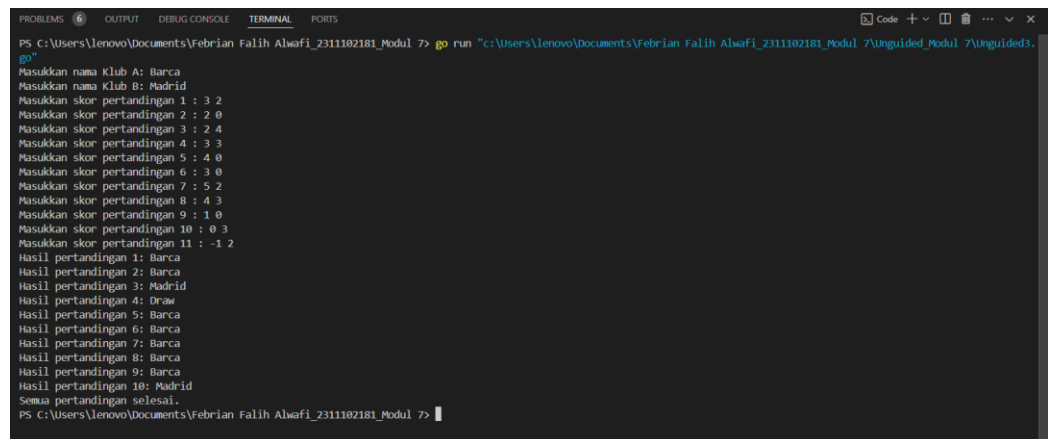
        // Menentukan hasil pertandingan berdasarkan skor
        switch {
        case skorA_2311102181 > skorB:
            p.hasil = append(p.hasil, p.klub1)
        case skorA_2311102181 < skorB:
            p.hasil = append(p.hasil, p.klub2)
        default:
            p.hasil = append(p.hasil, "Draw")
        }
    }

    // Menampilkan hasil setiap pertandingan
    for k, hasil := range p.hasil {
        fmt.Printf("Hasil pertandingan %d: %v\n", k+1,
hasil)
    }
}

```

```
}  
  
    fmt.Println("Semua pertandingan selesai.")  
}
```

Output :



```
PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7> go run "c:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7\Unguided_Modul 7\Unguided3.go"  
Masukkan nama Klub A: Barca  
Masukkan nama Klub B: Madrid  
Masukkan skor pertandingan 1 : 3 2  
Masukkan skor pertandingan 2 : 2 0  
Masukkan skor pertandingan 3 : 2 4  
Masukkan skor pertandingan 4 : 3 3  
Masukkan skor pertandingan 5 : 4 0  
Masukkan skor pertandingan 6 : 3 0  
Masukkan skor pertandingan 7 : 5 2  
Masukkan skor pertandingan 8 : 4 3  
Masukkan skor pertandingan 9 : 1 0  
Masukkan skor pertandingan 10 : 0 3  
Masukkan skor pertandingan 11 : -1 2  
Hasil pertandingan 1: Barca  
Hasil pertandingan 2: Barca  
Hasil pertandingan 3: Madrid  
Hasil pertandingan 4: Draw  
Hasil pertandingan 5: Barca  
Hasil pertandingan 6: Barca  
Hasil pertandingan 7: Barca  
Hasil pertandingan 8: Barca  
Hasil pertandingan 9: Barca  
Hasil pertandingan 10: Madrid  
Semua pertandingan selesai.  
PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7>
```

Deskripsi Program :

Program di atas adalah aplikasi yang digunakan untuk mencatat hasil pertandingan antara dua klub sepak bola yang ditentukan oleh pengguna. Program dimulai dengan mendeklarasikan struktur Pertandingan, yang menyimpan nama kedua klub (klub1 dan klub2) serta slice hasil, yang nantinya digunakan untuk menyimpan hasil setiap pertandingan. Pertama-tama program meminta pengguna memasukkan nama untuk kedua klub yang akan bertanding. Setelah itu, program memasuki loop untuk menerima input skor dari setiap pertandingan. Dalam loop ini, pengguna diminta memasukkan skor untuk klub1 dan klub2 di setiap pertandingan, yang disimpan masing-masing dalam variabel skorA dan skorB. Loop ini terus berjalan hingga pengguna memasukkan skor negatif untuk salah satu klub, yang digunakan sebagai sinyal untuk menghentikan proses input pertandingan.

4. UNGUIDED 4

Source Code :

```
package main

import "fmt"

const MaxLength int = 127

type Tabel struct {
    data [MaxLength]rune
    length_2311102181 int
}

// Fungsi untuk menerima input teks dan mengisinya ke
// dalam array hingga titik atau batas panjang
func inputTeks(t *Tabel) {
    fmt.Println("Masukkan teks (akhiri dengan TITIK):")
    var input rune
    for i := 0; ; i++ {
        fmt.Scanf("%c", &input)
        if input == '.' || i >= MaxLength {
            break
        }
        if input != ' ' {
            t.data[i] = input
            t.length_2311102181++
        }
    }
}

// Fungsi untuk mencetak seluruh teks yang ada dalam
// array
func tampilkanTeks(t Tabel) {
    for i := 0; i < t.length_2311102181; i++ {
        fmt.Print(string(t.data[i]))
    }
    fmt.Println()
}
```

```

// Fungsi untuk membalikkan teks dalam array
func balikTeks(t *Tabel) {
    for i := 0; i < t.length_2311102181/2; i++ {
        t.data[i], t.data[t.length_2311102181-i-1] =
t.data[t.length_2311102181-i-1], t.data[i]
    }
}

// Fungsi untuk memeriksa apakah teks adalah
palindrome
func cekPalindrome(t Tabel) bool {
    for i := 0; i < t.length_2311102181/2; i++ {
        if t.data[i] != t.data[t.length_2311102181-i-1] {
            return false
        }
    }
    return true
}

func main() {
    var t Tabel

    // Memasukkan dan mengisi array dengan teks
    inputTeks(&t)

    // Menampilkan teks asli
    fmt.Print("Teks: ")
    tampilkanTeks(t)

    // Menampilkan teks setelah dibalik
    fmt.Print("Reverse teks: ")
    balikTeks(&t)
    tampilkanTeks(t)

    // Mengecek apakah teks adalah palindrome
    if cekPalindrome(t) {
        fmt.Println("Palindrome: true")
    } else {

```

```
        fmt.Println("Palindrome: false")
    }
}
```

Output :

```
PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7> go run "c:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 7\Unguided_Modul 7\Unguided4.go"
Masukkan teks (akhiri dengan TITIK):
SENANG.
Teks: SENANG
Reverse teks: GNWANS
Palindrome: false
KATAK.
Teks: KATAK
Reverse teks: KATAK
Palindrome: true
```

Deskripsi Program :

Program ini digunakan untuk menerima input teks dari pengguna, melakukan pembalikan terhadap teks tersebut, serta memeriksa apakah teks yang dimasukkan merupakan palindrome (teks yang dapat dibaca sama dari depan dan belakang). Struktur utama dalam program ini adalah Tabel, yang berfungsi untuk menyimpan array karakter (dalam bentuk rune) dan panjang teks yang valid. Program dimulai dengan meminta pengguna untuk memasukkan teks, yang kemudian disimpan dalam array dengan batas maksimal 127 karakter, dan diakhiri dengan tanda titik (.). Program memastikan karakter spasi tidak disertakan dalam array. program menampilkan teks asli, kemudian membalikkan urutan teks dengan cara menukar elemen array dari indeks pertama dengan indeks terakhir dan seterusnya hingga mencapai tengah array. Teks yang telah dibalik kemudian ditampilkan. Selanjutnya, program memeriksa apakah teks tersebut merupakan palindrome dengan membandingkan karakter di setiap sisi array secara berpasangan. Jika teks adalah palindrome, maka hasilnya adalah true, jika tidak, maka hasilnya adalah false.