

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMOGRAMAN II
MODUL VII
STRUCK & ARRAY



Oleh:

NAMA : DWI HESTI ARIANI

NIM : 2311102094

KELAS : 11- IF -02

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. DASAR TEORI

a. Struck

Struktur dalam Golang adalah tipe data yang memungkinkan pengelompokan beberapa nilai dengan tipe yang berbeda ke dalam satu unit. Struktur digunakan untuk membuat representasi data yang lebih kompleks, seperti entitas dalam aplikasi. Deklarasi struktur dilakukan dengan menggunakan kata kunci `type` dan `struct`. Contoh deklarasi struktur adalah sebagai berikut :

```
type Mahasiswa struct {  
    Nama  string  
    Umur  int  
    IPK   float64  
}
```

Dari contoh di atas, Mahasiswa adalah sebuah struktur yang memiliki tiga anggota: Nama, Umur, dan IPK. Penggunaan struktur memudahkan pengelolaan data yang berkaitan, misalnya data mahasiswa.

b. Array

Array adalah kumpulan elemen dengan tipe data yang sama, disimpan dalam memori secara berurutan. Di Golang, array memiliki ukuran tetap yang ditentukan saat deklarasi. Contoh deklarasi array adalah sebagai berikut:

```
var nilai [5]int // Deklarasi array dengan 5 elemen integer
```

Array dapat diakses menggunakan indeks, dimulai dari 0. Untuk menginisialisasi array, seperti contoh berikut :

```
nilai := [5]int{80, 85, 90, 75, 88} // Inisialisasi array dengan  
nilai
```

II. GUIDED

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 Program

```

PS C:\Users\dwih> go run "c:\Users\dwih\OneDrive\Documents\SEMES
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\dwih>

```

Deskripsi Program

Program diatas ialah program sederhana dalam bahasa Go yang berfungsi untuk mengelola data mahasiswa, menghitung rata-rata nilai mereka, dan mengurutkan mahasiswa berdasarkan nilai rata-rata tersebut. Program mulai mendeklarasikan array 'mahasiswa', yang berisi data lima mahasiswa beserta nilai mereka di 3 mata kuliah, kemudian program menghitung rata-rata nilai mahasiswa, setelah itu, program menggunakan sort slice untuk mengurutkan array berdasarkan nilai rata-rata.

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 Program

```
PS C:\Users\dwih> go run "c:\Users\dwih\OneD
Daftar Mahasiswa:
NIM          Nama
-----
20231004     Dedi
20231001     Andi
20231002     Budi
20231003     Cici

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM          Nama
-----
20231001     Andi
20231002     Budi
20231004     Dedi
PS C:\Users\dwih>
```

Deskripsi Program

Program ini merupakan program sederhana Bahasa Golang yang berfungsi untuk mengelola data mahasiswa menggunakan struktur data **map**. Program ini menyimpan NIM (Nomor Induk Mahasiswa) sebagai kunci dan nama mahasiswa sebagai nilai. Program dimulai dengan mendeklarasikan sebuah map yang bervariasi 'mahasiswa'. Program dapat menambah data, menghapus data, mengakses data dan menampilkan isi map setelah penghapusan .

III. UNGUIDED

Source Code

Unguided 1

```
package main

import (
    "fmt"
    "math"
)

type titik struct {
    x int
    y int
}
```

```

type lingkaran struct {
    titikPusat titik
    radius      int
}

func jarak(p, q titik) float64 {
    return math.Sqrt(float64((p.x-q.x)*(p.x-q.x) + (p.y-
q.y)*(p.y-q.y)))
}

func didalam(c lingkaran, p titik) bool {
    return jarak(c.titikPusat, p) <= float64(c.radius)
}

func main() {
    var cx1, cy1, r1, cx2, cy2, r2, zx, zy int
    fmt.Scanln(&cx1, &cy1, &r1)
    fmt.Scanln(&cx2, &cy2, &r2)
    fmt.Scanln(&zx, &zy)

    l1 := lingkaran{titik{cx1, cy1}, r1}
    l2 := lingkaran{titik{cx2, cy2}, r2}
    p := titik{zx, zy}

    if didalam(l1, p) && didalam(l2, p) {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if didalam(l1, p) {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if didalam(l2, p) {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

Output Program

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\Users\dwih\OneDrive\Documents\SE
RO2\MODUL 6\Unguided 1\unguided1.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\Users\dwih\OneDrive\Documents\SE
RO2\MODUL 6\Unguided 1\unguided1.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Users\dwih\OneDrive\Documents\SE
RO2\MODUL 6\Unguided 1\unguided1.go"
115
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
```

Deskripsi Program

Program diatas ialah program sederhana dalam bahasa Go menggunakan struct dan type yang berfungsi untuk menentukan posisi suatu titik relative terhadap dua lingkaran . Program menyimpan dua atribut yaitu x dan y yang mewakili koordinat titik dalam system koordinat kartesian. Pada variable 'didalam' terdapat suatu titik (p) berada dalam lingkaran (c). User diminta untuk memasukkan bilangan bulat positif, setelah menerima input program akan membuat dua objek 'lingkaran' dan satu objek titik (p), kemudian program akan memeriksa posisi titik apakah berada dalam lingkaran atau tidak, dan menghitung jarak antara pusat lingkaran dan titik tersebut.

Unguided 2

Source Code

```
package main

import (
    "fmt"
    "math"
)

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

    // Membuat array dengan kapasitas N
    arr := make([]int, N)

    // Mengisi array dengan nilai
    fmt.Println("Masukkan elemen array:")
    for i := 0; i < N; i++ {
        fmt.Printf("Elemen [%d]: ", i)
        fmt.Scan(&arr[i])
    }

    // Menampilkan keseluruhan isi dari array
    fmt.Println("\nIsi keseluruhan array:")
    displayArray(arr)

    fmt.Println("\nElemen dengan indeks ganjil:")
    displayOddIndex(arr)

    fmt.Println("\nElemen dengan indeks genap:")
    displayEvenIndex(arr)

    var x int
    fmt.Print("\nMasukkan bilangan x untuk indeks kelipatan: ")
    fmt.Scan(&x)
    fmt.Printf("\nElemen dengan indeks kelipatan %d:\n", x)
    displayMultipleOfX(arr, x)

    var indexToRemove int
    fmt.Print("\nMasukkan indeks yang ingin dihapus: ")
```

```

    fmt.Scan(&indexToRemove)
    arr = removeElement(arr, indexToRemove)
    fmt.Println("\nIsi array setelah penghapusan:")
    displayArray(arr)

    average := calculateAverage(arr)
    fmt.Printf("\nRata-rata: %.2f\n", average)

    stdDev := calculateStdDev(arr, average)
    fmt.Printf("Standar deviasi: %.2f\n", stdDev)

    var numberToFind int
    fmt.Print("\nMasukkan bilangan untuk mencari frekuensinya: ")
    fmt.Scan(&numberToFind)
    frequency := countFrequency(arr, numberToFind)
    fmt.Printf("Frekuensi dari bilangan %d: %d\n", numberToFind, frequency)
}

func displayArray(arr []int) {
    for _, v := range arr {
        fmt.Println(v)
    }
}

func displayOddIndex(arr []int) {
    for i := 1; i < len(arr); i += 2 {
        fmt.Println(arr[i])
    }
}

func displayEvenIndex(arr []int) {
    for i := 0; i < len(arr); i += 2 {
        fmt.Println(arr[i])
    }
}

func displayMultipleOfX(arr []int, x int) {
    for i := 0; i < len(arr); i++ {
        if i%x == 0 {
            fmt.Println(arr[i])
        }
    }
}
}

```

```

func removeElement(arr []int, index int) []int {
    return append(arr[:index], arr[index+1:]...)
}

func calculateAverage(arr []int) float64 {
    if len(arr) == 0 {
        return 0
    }
    sum := 0
    for _, v := range arr {
        sum += v
    }
    return float64(sum) / float64(len(arr))
}

func calculateStdDev(arr []int, average float64) float64 {
    if len(arr) == 0 {
        return 0
    }
    var sum float64
    for _, v := range arr {
        sum += math.Pow(float64(v)-average, 2)
    }
    return math.Sqrt(sum / float64(len(arr)))
}

func countFrequency(arr []int, number int) int {
    count := 0
    for _, v := range arr {
        if v == number {
            count++
        }
    }
    return count
}

```

Output Program

```
Masukkan jumlah elemen array: 6
Masukkan elemen array:
Elemen [0]: 12
Elemen [1]: 13
Elemen [2]: 14
Elemen [3]: 15
Elemen [4]: 16
Elemen [5]: 17

Isi keseluruhan array:
12
13
14
15
16
17

Elemen dengan indeks ganjil:
13
15
17

Elemen dengan indeks genap:
12
14
16

Masukkan bilangan x untuk indeks kelipatan: 2
```

```
Elemen dengan indeks kelipatan 2:
12
14
16

Masukkan indeks yang ingin dihapus: 4

Isi array setelah penghapusan:
12
13
14
15
17

Rata-rata: 14.20
Standar deviasi: 1.72

Masukkan bilangan untuk mencari frekuensinya: 12
Frekuensi dari bilangan 12: 1
PS C:\Users\dwih\OneDrive\Documents\SEMESTER 3\PRAKTIKUM>
```

Deskripsi Program

Program ini merupakan program sederhana Bahasa Golang yang berfungsi untuk mengelola dan menganalisis data dalam bentuk array. Program ini memungkinkan pengguna untuk memasukkan elemen array, melakukan berbagai operasi pada array, dan menghitung statistik dasar.

User diminta untuk memasukkan jumlah elemen array lalu program akan meminta user untuk memasukkan nilai untuk setiap elemen array menggunakan loop, setelah itu, program akan menampilkan elemen dengan indeks ganjil dan indeks genap, program juga meminta user untuk memasukkan nilai kelipatan x dan elemen yang akan di hapus dalam array. Kemudian program akan menghitung nilai rata-rata elemen dalam array, menghitung standar deviasi dan menghitung frekuensi elemen tertentu.

Unguided 3

Source Code

```
package main

import (
    "fmt"
)

func main() {
    var klubA, klubB string
    var skorA, skorB int
    var winners []string
    var results []string

    fmt.Print("Klub A : ")
    fmt.Scanln(&klubA)

    fmt.Print("Klub B : ")
    fmt.Scanln(&klubB)

    pertandinganKe := 1

    for {
        fmt.Printf("Pertandingan %d (skor A B): ", pertandinganKe)
        fmt.Scanln(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
```

```

        break
    }

    if skorA > skorB {
        winners = append(winners, klubA)
        results = append(results, fmt.Sprintf("Pertandingan
%d: %s %d - %d %s (Pemenang: %s)", pertandinganKe, klubA, skorA,
skorB, klubB, klubA))
    } else if skorB > skorA {
        winners = append(winners, klubB)
        results = append(results, fmt.Sprintf("Pertandingan
%d: %s %d - %d %s (Pemenang: %s)", pertandinganKe, klubA, skorA,
skorB, klubB, klubB))
    } else {
        results = append(results, fmt.Sprintf("Pertandingan
%d: %s %d - %d %s (Hasil Seri)", pertandinganKe, klubA, skorA,
skorB, klubB))
    }

    pertandinganKe++
}

fmt.Println("Pertandingan selesai")
fmt.Println("Hasil setiap pertandingan:")
for _, result := range results {
    fmt.Println(result)
}
}

```

Output Program

```

PS C:\Users\dwihe\OneDrive\Documents\SEMESTER 3\
RO2\MODUL 6\Unguided 2\Unguided 3\unguided3.go"
Klub A : MU
Klub B : MC
Pertandingan 1 (skor A B): 1 2
Pertandingan 2 (skor A B): 2 0
Pertandingan 3 (skor A B): 2 2
Pertandingan 4 (skor A B): 1 0
Pertandingan 5 (skor A B): 2 1
Pertandingan 6 (skor A B): 3 2
Pertandingan 7 (skor A B): -1 2
Pertandingan selesai
Hasil setiap pertandingan:
Pertandingan 1: MU 1 - 2 MC (Pemenang: MC)
Pertandingan 2: MU 2 - 0 MC (Pemenang: MU)
Pertandingan 3: MU 2 - 2 MC (Hasil Seri)
Pertandingan 4: MU 1 - 0 MC (Pemenang: MU)
Pertandingan 5: MU 2 - 1 MC (Pemenang: MU)
Pertandingan 6: MU 3 - 2 MC (Pemenang: MU)
PS C:\Users\dwihe\OneDrive\Documents\SEMESTER 3\

```

Deskripsi program

Program diatas ialah program sederhana dalam bahasa Go yang berfungsi untuk mencatat hasil pertandingan antara dua klub sepak bola. Program ini memungkinkan pengguna untuk memasukkan nama klub, mencatat skor pertandingan, menentukan pemenang, dan menampilkan hasil pertandingan setelah selesai. User diminta untuk memasukkan nama klub A dan klub B, kemudian program meminta user untuk memasukkan skor hasil pertandingan kedua klub menggunakan loop, jika salah satu skor yang dimasukkan kurang dari 0 maka loop akan berhenti, dari kedua skor pertandingan terserbut akan di bandingkan dan dicatat.

Unguided 4

Source Code

```
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

var tab tabel
var m int

func isiArray(t *tabel, n *int) {

    var i int
    var ch rune
    i = 0
    fmt.Scanf("%c", &ch)
    for ch != '.' && i < NMAX {
        t[i] = ch
        i++
        fmt.Scanf("%c", &ch)
    }
    *n = i
}

func cetakArray(t tabel, n int) {
```

```

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

func balikanArray(t *tabel, n int) {

    var i, j int
    var temp rune
    for i = 0; i < n/2; i++ {
        j = n - i - 1
        temp = t[i]
        t[i] = t[j]
        t[j] = temp
    }
}

func isPalindrome(t tabel, n int) bool {

    for i := 0; i < n/2; i++ {
        if t[i] != t[n-i-1] {
            return false
        }
    }
    return true
}

func main() {
    isiArray(&tab, &m)
    fmt.Println("Array yang dimasukkan:")
    cetakArray(tab, m)

    if isPalindrome(tab, m) {
        fmt.Println("Array tersebut adalah palindrom.")
    } else {
        fmt.Println("Array tersebut bukan palindrom.")
    }
}

```


Output Program

```
PS C:\Users\dwihe\OneDrive\Documents\SEMESTER 3\RO2\MODUL 6\Unguided 2\Unguided 3\Unguided4\ungu
katak.
Array yang dimasukkan:
katak
Array tersebut adalah palindrom.
PS C:\Users\dwihe\OneDrive\Documents\SEMESTER 3\RO2\MODUL 6\Unguided 2\Unguided 3\Unguided4\ungu
senang.
Array yang dimasukkan:
senang
Array tersebut bukan palindrom.
PS C:\Users\dwihe\OneDrive\Documents\SEMESTER 3\
```

Deskripsi Program

Program diatas ialah program sederhana dalam bahasa Go yang berfungsi untuk membaca karakter dari input, menyimpannya dalam array, dan kemudian memeriksa apakah array tersebut merupakan palindrom. Program akan meminta user untuk memasukkan array dalam bentuk karakter yang diakhiri dengan tanda ' . ' , lalu program akan memeriksa array dan membandingkan karakter dari awal dan akhir array hingga mencapai tengah array, setelah pengecekan dan perbandingan selesai, program akan menampilkan array tersebut palindrom atau bukan .