

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL VII  
STRUCK & ARRAY**



Oleh:

NAMA : KARTIKA PRINGGO HUTOMO

NIM : 2311102196

KELAS ; IF-11-02

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

### Array

Kumpulan elemen data dengan tipe yang sama dan disimpan secara berurutan dalam memori disebut array.

Ciri:

- Homogen: Semua elemen memiliki jenis data yang sama.
- Statik: Ukuran array ditetapkan saat deklarasi dan tidak dapat diubah selama runtime.
- Terstruktur: Anda dapat mengakses elemen dalam array dengan menggunakan indeks.

### Struck

Definisi: Struktur adalah jenis data yang menggabungkan beberapa variabel dari berbagai jenis menjadi satu kesatuan logis.

Ciri:

- Heterogen: Tipe data yang berbeda dapat dimiliki oleh anggota struktur.
- Terorganisir: Memungkinkan pengelompokan data yang saling terkait, seperti informasi siswa, staf, atau produk.

## 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) / 30
}

// Fungsi utama untuk mengelola dan mengurutkan data mahasiswa
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 D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7> go run "d:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7\guided1.go"
Peringkat mahasiswa berdasarkan rata-rata nilai:
1. Eka - Rata-Rata: 9.500000 (Matematika: 100, Fisika: 95, Kimia: 90)
2. Cici - Rata-Rata: 9.000000 (Matematika: 90, Fisika: 85, Kimia: 95)
3. Ali - Rata-Rata: 8.500000 (Matematika: 85, Fisika: 90, Kimia: 80)
4. Budi - Rata-Rata: 7.500000 (Matematika: 70, Fisika: 75, Kimia: 80)
5. Doni - Rata-Rata: 6.500000 (Matematika: 60, Fisika: 65, Kimia: 70)
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7>
```

### Penjelasan

Program ini dimaksudkan untuk mengelola dan mengurutkan data nilai siswa dari Matematika, Fisika, dan Kimia. Pertama, program mendefinisikan struktur siswa, di mana nama siswa dan nilai masing-masing mata pelajaran disimpan. Selanjutnya, program menampilkan daftar peringkat siswa dengan informasi tentang nama, rata-rata, dan nilai masing-masing mata pelajaran. Hasilnya ditampilkan secara terurut, dengan siswa dengan nilai tertinggi rata-rata berada di posisi pertama dan seterusnya.

## 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\ttNama")
    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\ttNama")
    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }
}
```

## Output

```
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7> go run "d:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7\guided2.go"
Daftar Mahasiswa:
NIM      Nama
-----
20231001  Andi
20231002  Budi
20231003  Cici
20231004  Dedi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM      Nama
-----
20231001  Andi
20231002  Budi
20231004  Dedi
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7>
```

## Penjelasan

Kode ini adalah program yang dirancang untuk mengelola data siswa menggunakan struktur data map Go, dengan NIM sebagai kunci dan nama siswa sebagai nilai. Pertama, program membuat map siswa dan memasukkan data seperti NIM dan nama siswa. Setelah penghapusan, program mencetak kembali seluruh isi map untuk menunjukkan daftar siswa yang tersisa dalam format tabel.

### III. UNGUIDED

#### 1.

##### Source code

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    pusat Titik
    r    int
}

func cekTitikDiDalamLingkaran(t Titik, l Lingkaran) bool {
    jarak := math.Sqrt(math.Pow(float64(t.x-l.pusat.x), 2) + math.Pow(float64(t.y-l.pusat.y), 2))
    return jarak <= float64(l.r)
}

func main() {

    var x1, y1, r1 int

    fmt.Print("Masukkan lingkaran 1 (x y r): ")

    fmt.Scan(&x1, &y1, &r1)

    lingkaran1 := Lingkaran{Titik{x1, y1}, r1}

    var x2, y2, r2 int
```

```
    fmt.Print("Masukkan lingkaran 2 (x y r): ")
    fmt.Scan(&x2, &y2, &r2)
    lingkaran2 := Lingkaran{Titik{x2, y2}, r2}

    var x, y int
    fmt.Print("Masukkan titik (x y): ")
    fmt.Scan(&x, &y)
    titik := Titik{x, y}

    diLingkaran1 := cekTitikDiDalamLingkaran(titik, lingkaran1)
    diLingkaran2 := cekTitikDiDalamLingkaran(titik, lingkaran2)

    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 D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7> go run "d:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7\unguided1.go"
Masukkan lingkaran 1 (x y r): 1 1 5
Masukkan lingkaran 2 (x y r): 8 8 4
Masukkan titik (x y): 2 2
Titik di dalam lingkaran 1
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7> █
```

### Penjelasan

Kode ini adalah program Go yang digunakan untuk menentukan apakah suatu titik berada di dalam satu atau dua lingkaran. Pertama, program mendefinisikan dua struktur: titik, yang menunjukkan koordinat (x, y), dan lingkaran, yang menunjukkan pusat lingkaran (sebagai objek titik) dan jari-jari lingkaran. Kemudian, program memanggil fungsi untuk memeriksa apakah titik tersebut berada di dalam masing-masing lingkaran dan menampilkan hasilnya: apakah titik tersebut ada di dalam lingkaran



## 2.

```
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.Printf("Elemen [%d]: ", i)
        fmt.Scan(&arr[i])
    }

    fmt.Println("\nIsi seluruh array:")
    tampilkanArray(arr)

    fmt.Println("\nElemen pada indeks ganjil:")
    tampilkanIndeksGanjil(arr)

    fmt.Println("\nElemen pada indeks genap:")
    tampilkanIndeksGenap(arr)

    var x int

    fmt.Print("\nMasukkan angka x untuk indeks kelipatan: ")
    fmt.Scan(&x)

    fmt.Printf("\nElemen pada indeks kelipatan %d:\n", x)
```

```

    tampilkanKelipatanX(arr, x)

    var indeksHapus int

    fmt.Print("\nMasukkan indeks yang ingin dihapus: ")

    fmt.Scan(&indeksHapus)

    arr = hapusElemen(arr, indeksHapus)

    fmt.Println("\nIsi array setelah penghapusan:")

    tampilkanArray(arr)

    rataRata := hitungRataRata(arr)

    fmt.Printf("\nRata-rata: %.2f\n", rataRata)

    stdDev := hitungStandarDeviasi(arr, rataRata)

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

    var angkaCari int

    fmt.Print("\nMasukkan angka untuk mencari frekuensinya: ")

    fmt.Scan(&angkaCari)

    frekuensi := hitungFrekuensi(arr, angkaCari)

    fmt.Printf("Frekuensi dari angka %d: %d\n", angkaCari, frekuensi)
}

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

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

```

```

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

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

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

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

func hitungStandarDeviasi(arr []int, rataRata float64) float64 {
    if len(arr) == 0 {
        return 0
    }
    var total float64

```

```

        for _, v := range arr {
            total += math.Pow(float64(v)-rataRata, 2)
        }

        return math.Sqrt(total / float64(len(arr)))
    }

func hitungFrekuensi(arr []int, angka int) int {
    jumlah := 0

    for _, v := range arr {
        if v == angka {
            jumlah++
        }
    }

    return jumlah
}

```

## Output

```

3
4
5
6
7

Elemen pada indeks ganjil:
1
3
5
7

Elemen pada indeks genap:
2
4
6

Masukkan angka x untuk indeks kelipatan: 2

Elemen pada indeks kelipatan 2:
1
3
5
7

Masukkan indeks yang ingin dihapus: 3

Isi array setelah penghapusan:
1
2
3
5
6
7

Rata-rata: 4.00
Standar deviasi: 2.16

Masukkan angka untuk mencari frekuensinya: 5
Frekuensi dari angka 5: 1
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7>

```

## Penjelasan

Program meminta pengguna untuk memasukkan jumlah elemen dalam array dan kemudian mengisi nilai elemen. Kemudian, program menampilkan isi seluruh array, elemen pada indeks ganjil, dan elemen pada indeks genap. Pengguna kemudian dapat memasukkan angka untuk menampilkan elemen pada indeks yang merupakan kelipatan dari angka tersebut, dan program memungkinkan mereka untuk menghapus elemen berdasarkan indeks yang diinginkan. Program ini tidak hanya menghitung dan menampilkan standar deviasi dan rata-rata elemen array setelah penghapusan, tetapi juga menampilkan array yang telah diperbarui. Terakhir, pengguna dapat mencari frekuensi suatu angka dalam array, dan hasilnya akan ditampilkan. Kode ini membuat logika lebih terorganisir dan mudah dibaca dengan menggunakan berbagai fungsi.

### 3.

```
package main

import "fmt"

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

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

    pertandingan := 1
    for {
        fmt.Printf("Skor Pertandingan %d (Klub A Klub B): ", pertandingan)
        fmt.Scanln(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            break
        }

        if skorA > skorB {
            hasil = append(hasil, klubA)
        } else if skorA < skorB {
            hasil = append(hasil, klubB)
        }
    }
}
```

```

        } else {

            hasil = append(hasil, "Draw")

        }

        pertandingan++

    }

    fmt.Println("Pertandingan selesai")

    for i, p := range hasil {

        fmt.Printf("Hasil Pertandingan %d: %s\n", i+1, p)

    }

}

```

## Output

```

PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7> go run "d:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7\unguided3.go"
Nama Klub A: mu
Nama Klub B: inter
Skor Pertandingan 1 (Klub A Klub B): 2 0
Skor Pertandingan 2 (Klub A Klub B): 1 2
Skor Pertandingan 3 (Klub A Klub B): 2 2
Skor Pertandingan 4 (Klub A Klub B): 0 1
Skor Pertandingan 5 (Klub A Klub B): 3 2
Skor Pertandingan 6 (Klub A Klub B): 1 0
Skor Pertandingan 7 (Klub A Klub B): 5 2
Skor Pertandingan 8 (Klub A Klub B): 2 3
Skor Pertandingan 9 (Klub A Klub B): -1 2
Pertandingan selesai
Hasil Pertandingan 1: mu
Hasil Pertandingan 2: inter
Hasil Pertandingan 3: Draw
Hasil Pertandingan 4: inter
Hasil Pertandingan 5: mu
Hasil Pertandingan 6: mu
Hasil Pertandingan 7: mu
Hasil Pertandingan 8: inter
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7> █

```

## Penjelasan

berfungsi untuk mencatat hasil pertandingan sepak bola antara dua tim. Dimulai dengan meminta pengguna memberikan nama kedua klub mereka. Setelah itu, program memulai loop yang meminta skor untuk setiap pertandingan. Di dalam loop ini, pengguna harus memasukkan skor untuk Klub A dan Klub B; loop akan berhenti jika ada skor negatif. Kemudian, program mencatat perbandingan skor dan hasilnya. Jika Klub A menang, namanya dimasukkan ke daftar hasil, jika Klub B menang, dan jika keduanya memiliki skor yang sama, hasilnya disebut "Draw". Setelah semua pertandingan berakhir, program mencetak semua hasil yang telah dicatat.

#### 4.

##### Sourcode

```
package main

import "fmt"

const NMAX int = 127

type tabel struct {
    tab [NMAX]rune
    m int
}

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

    fmt.Print("Masukkan karakter (akhiri dengan titik): ")
    var input rune
    for *n = 0; *n < NMAX; *n++ {
        fmt.Scanf("%c", &input)
        if input == '.' {
            break
        }
        t.tab[*n] = input
    }
    t.m = *n
}

func cetakArray(t tabel, n int) {

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

```

func balikanArray(t *tabel, n int) {

    for i := 0; i < n/2; i++ {

        t.tab[i], t.tab[n-i-1] = t.tab[n-i-1], t.tab[i]

    }

}

```

```

func cekPalindrom(t tabel, n int) bool {

    for i := 0; i < n/2; i++ {

        if t.tab[i] != t.tab[n-i-1] {

            return false

        }

    }

    return true

}

```

```

func main() {

    var tab tabel

    var m int

    isiArray(&tab, &m)

    fmt.Print("Teks: ")

    cetakArray(tab, m)

    fmt.Print("Teks Terbalik: ")

    balikanArray(&tab, m)

    cetakArray(tab, m)

    if cekPalindrom(tab, m) {

        fmt.Println("Palindrom: true")

    } else {

```



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

## Output

```
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7> go run "d:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7\umguided4.go"
Masukkan karakter (akhiri dengan titik): KATAK.
Teks: KATAK
Teks Terbalik: KATAK
Palindrom: true
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7> go run "d:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7\umguided4.go"
Masukkan karakter (akhiri dengan titik): SENANG.
Teks: SENANG
Teks Terbalik: GNANES
Palindrom: false
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 7> |
```

## Penjelasan

Dimaksudkan untuk menangani karakter yang dimasukkan oleh pengguna. Pertama, program meminta pengguna untuk memasukkan karakter hingga mereka mengetik titik (.) sebagai tanda akhir input. Karakter yang dimasukkan disimpan dalam struktur data yang disebut tabel, yang memiliki array maksimal berukuran 127. Setelah input selesai, program mencetak teks yang dimasukkan dan kemudian membalikkan urutan karakter dalam array sebelum mencetaknya lagi. Program juga memeriksa apa karakter yang dimasukkan oleh pengguna sebelum mencetaknya lagi.