

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 7  
STRUCK & ARRAY**



Oleh:

**CHRIST DANIEL SANTOSO**

**2311102305**

**IF-11-02**

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

## I. DASAR TEORI

Array adalah struktur data dasar yang menyimpan kumpulan elemen dengan tipe data yang sama secara berurutan dalam memori, memungkinkan akses cepat dan mudah melalui indeks yang dimulai dari 0. Elemen-elemen dalam array bersifat homogen, artinya semua elemen harus memiliki tipe data yang sama. Array juga bersifat statis, di mana ukurannya ditentukan saat deklarasi dan tidak dapat diubah selama program berjalan. Struktur data ini mendukung akses acak (random access), memungkinkan setiap elemen untuk diakses langsung melalui indeks tanpa perlu traversal berurutan, yang membuatnya sangat efisien untuk operasi pengambilan dan pengubahan data. Array juga dapat memiliki lebih dari satu dimensi, seperti array satu dimensi (vektor) atau dua dimensi (matriks), yang memungkinkannya untuk menyimpan data dalam bentuk tabel atau grid. Sementara itu, struct adalah tipe data kompleks yang memungkinkan pengelompokan variabel-variabel dengan tipe data yang berbeda ke dalam satu kesatuan logis, memungkinkan pengelolaan data yang lebih terstruktur. Struct digunakan untuk mengelompokkan data yang terkait, sehingga memungkinkan penyimpanan berbagai elemen yang mewakili sifat objek yang berbeda, seperti `int`, `float`, atau `string`, dalam satu entitas. Struct sangat fleksibel dalam deklarasinya, dapat diinisialisasi secara langsung atau melalui variabel, yang menjadikannya pilihan tepat untuk representasi data yang kompleks, seperti objek nyata dengan berbagai sifat.

## II. GUIDED

### Guided 1

```
package main

// NAMA : CHRIST DANIEL SANTOSO
// NIM : 2311102305

import (
    "fmt"
    "sort"
)

type Mahasiswa struct {
    Nama      string
    Matematika int
    Fisika    int
    Kimia     int
    RataRata  float64
}
```

```

}

func hitungRataRata(m *Mahasiswa) {
    total := m.Matematika + m.Fisika + m.Kimia
    m.RataRata = float64(total) / 3.0
}

func main() {

    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},
    }

    for i := range mahasiswa {
        hitungRataRata(&mahasiswa[i])
    }

    sort.Slice(mahasiswa, func(i, j int) bool {
        return mahasiswa[i].RataRata > mahasiswa[j].RataRata
    })

    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)
    }
}

```

### Screenshot output

```

PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\GUIDED MODUL 7\GUIDED 1.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\cynzw> 

```

Penjelasan :

Mengelola data mahasiswa, menghitung rata-rata nilai, dan mengurutkannya berdasarkan nilai rata-rata secara menurun. Setiap mahasiswa memiliki nama dan nilai untuk mata pelajaran Matematika, Fisika, dan Kimia. Program menghitung rata-rata nilai dari ketiga mata pelajaran untuk masing-masing mahasiswa, lalu mengurutkan mereka dalam urutan menurun berdasarkan rata-rata tersebut. Hasil akhirnya ditampilkan dalam bentuk peringkat, dengan detail nilai dan rata-rata masing-masing mahasiswa dicetak ke layar.

## Guided 2

```
package main

// NAMA : CHRIST DANIEL SANTOSO
// NIM : 2311102305

import "fmt"

func main() {

    mahasiswa := map[string]string{
        "20231001": "Andi",
        "20231002": "Budi",
        "20231003": "Cici",
    }

    mahasiswa["20231004"] = "Dedi"

    fmt.Println("Daftar Mahasiswa:")
    fmt.Println("NIM\t\tNama")
    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }

    nim := "20231002"
    fmt.Println("\nNama Mahasiswa dengan NIM", nim, "adalah", mahasiswa[nim])

    delete(mahasiswa, "20231003")

    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\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\GUIDED MODUL 7\tempCodeRunnerFile.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
-----
20231002  Budi
20231004  Dedi
20231001  Andi
PS C:\Users\cynzw> 

```

Penjelasan :

Membuat map untuk menyimpan data mahasiswa, di mana NIM (Nomor Induk Mahasiswa) digunakan sebagai kunci dan nama mahasiswa sebagai nilai. Program dimulai dengan data awal beberapa mahasiswa, kemudian menambahkan data baru ke dalam map. Program menampilkan seluruh isi map dalam format tabel yang rapi, mencetak daftar mahasiswa dengan NIM dan nama mereka. Program juga menunjukkan bagaimana mengakses data mahasiswa berdasarkan NIM tertentu serta menghapus data mahasiswa tertentu dari map. Setelah data dihapus, daftar diperbarui dan ditampilkan kembali untuk mencerminkan perubahan tersebut.

### III. UNGUIDED

#### Unguided 1

Screenshot output

```
package main

// NAMA : CHRIST DANIEL SANTOSO
// NIM : 2311102305

import (
    "fmt"
    "math"
)

type Titik struct {
    X, Y int
}

type Lingkaran struct {
    Pusat Titik
    JariJari int
}

func hitungJarak(t Titik, l Lingkaran) float64 {
    return math.Sqrt(math.Pow(float64(t.X-l.Pusat.X), 2) +
math.Pow(float64(t.Y-l.Pusat.Y), 2))
}

func titikDiDalamLingkaran(t Titik, l Lingkaran) bool {
```

```

    return hitungJarak(t, l) <= float64(l.JariJari)
}

func main() {
    var x1, y1, r1 int
    fmt.Print("Masukkan untuk lingkaran 1: ")
    fmt.Scan(&x1, &y1, &r1)
    lingkaran1 := Lingkaran{Pusat: Titik{X: x1, Y: y1}, JariJari: r1}

    var x2, y2, r2 int
    fmt.Print("Masukkan untuk lingkaran 2: ")
    fmt.Scan(&x2, &y2, &r2)
    lingkaran2 := Lingkaran{Pusat: Titik{X: x2, Y: y2}, JariJari: r2}

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

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

    switch {
    case diLingkaran1 && diLingkaran2:

```

```

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

    case diLingkaran1:

        fmt.Println("Titik di dalam lingkaran 1")

    case diLingkaran2:

        fmt.Println("Titik di dalam lingkaran 2")

    default:

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

    }
}

```

output :

```

PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\New folder\tempCodeRunnerFile.go"
Masukkan untuk lingkaran 1: 1 1 5
Masukkan untuk lingkaran 2: 8 8 4
Masukkan untuk titik sembarang: 2 2
Titik di dalam lingkaran 1
PS C:\Users\cynzw>

```

## Penjelasan

Menentukan apakah sebuah titik sembarang berada di dalam salah satu atau kedua lingkaran yang ditentukan. Lingkaran didefinisikan dengan koordinat titik pusat dan jari-jari, sedangkan titik sembarang didefinisikan dengan koordinat x dan y. Program menggunakan struktur data Titik dan Lingkaran serta fungsi matematika untuk menghitung jarak antara titik pusat lingkaran dan titik sembarang. Hasil keluaran menyatakan apakah titik berada di dalam lingkaran 1, lingkaran 2, keduanya, atau di luar kedua lingkaran.



## Unguided 2

```
package main

// NAMA : CHRIST DANIEL SANTOSO
// NIM : 2311102305

import (
    "fmt"
    "math"
)

func tampilkanArray(arr []int) {
    fmt.Println("Seluruh elemen array:", arr)
}

func tampilkanElemenIndeksGanjil(arr []int) {
    fmt.Print("Elemen dengan indeks ganjil: ")
    for i := 0; i < len(arr); i++ {
        if i%2 == 0 {
            fmt.Print(arr[i], " ")
        }
    }
    fmt.Println()
}

func tampilkanElemenIndeksGenap(arr []int) {
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 1; i < len(arr); i++ {
        if i%2 != 0 {
            fmt.Print(arr[i], " ")
        }
    }
    fmt.Println()
}

func tampilkanElemenIndeksKelipatan(arr []int, x int) {
    fmt.Printf("Elemen dengan indeks kelipatan %d: ", x)
    for i := 0; i < len(arr); i++ {
        if i%x == 0 {
            fmt.Print(arr[i], " ")
        }
    }
}
```

```

    }
}
fmt.Println()
}

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

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

func hitungStandarDeviasi(arr []int) float64 {
    rata := hitungRataRata(arr)
    var totalKuadrat float64
    for _, v := range arr {
        totalKuadrat += math.Pow(float64(v)-rata, 2)
    }
    return math.Sqrt(totalKuadrat / float64(len(arr)))
}

func hitungFrekuensi(arr []int, elemen int) int {
    jumlah := 0
    for _, v := range arr {
        if v == elemen {
            jumlah++
        }
    }
    return jumlah
}

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

    arr := make([]int, n)

```

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

tampilkanArray(arr)
tampilkanElemenIndeksGanjil(arr)
tampilkanElemenIndeksGenap(arr)

var x int
fmt.Print("Masukkan bilangan x untuk indeks kelipatan: ")
fmt.Scan(&x)
tampilkanElemenIndeksKelipatan(arr, x)

var idxHapus int
fmt.Print("Masukkan indeks yang akan dihapus: ")
fmt.Scan(&idxHapus)
arr = hapusPadaIndeks(arr, idxHapus)
fmt.Println("Array setelah penghapusan elemen:")
tampilkanArray(arr)

fmt.Printf("Rata-rata array: %.2f\n", hitungRataRata(arr))
fmt.Printf("Standar deviasi array: %.2f\n", hitungStandarDeviasi(arr))

var elemenCari int
fmt.Print("Masukkan elemen untuk mencari frekuensi: ")
fmt.Scan(&elemenCari)
fmt.Printf("Frekuensi elemen %d: %d kali\n", elemenCari,
hitungFrekuensi(arr, elemenCari))
}
```

## Screenshot output

```
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\New folder\UNGUIDED 2.go"
Masukkan jumlah elemen array: 10
Masukkan elemen-elemen array:
1 2 3 4 5 6 7 8 9 10
Seluruh elemen array: [1 2 3 4 5 6 7 8 9 10]
Elemen dengan indeks ganjil: 1 3 5 7 9
Elemen dengan indeks genap: 2 4 6 8 10
Masukkan bilangan x untuk indeks kelipatan: 3
Elemen dengan indeks kelipatan 3: 1 4 7 10
Masukkan indeks yang akan dihapus: 2
Array setelah penghapusan elemen:
Seluruh elemen array: [1 2 4 5 6 7 8 9 10]
Rata-rata array: 5.78
Standar deviasi array: 2.90
Masukkan elemen untuk mencari frekuensi: 6
Frekuensi elemen 6: 1 kali
PS C:\Users\cynzw>
```

## Penjelasan

Program ini meminta pengguna untuk memasukkan jumlah elemen dan elemen array itu sendiri, kemudian menampilkan semua elemen, elemen dengan indeks ganjil dan genap, serta elemen dengan indeks kelipatan tertentu yang ditentukan pengguna. Selain itu, program memungkinkan pengguna untuk menghapus elemen dari array berdasarkan indeks yang diberikan, menghitung rata-rata dan standar deviasi dari elemen array, serta menghitung frekuensi kemunculan elemen tertentu dalam array.

## Unguided 3

```
package main

// NAMA : CHRIST DANIEL SANTOSO
// NIM : 2311102305

import "fmt"

func main() {
    var team1, team2 string
    var score1, score2 int
    var results []string

    fmt.Print("Nama Klub 1: ")
    fmt.Scanln(&team1)
    fmt.Print("Nama Klub 2: ")
    fmt.Scanln(&team2)

    gameCount := 1
```

```

for {
    fmt.Printf("Skor Pertandingan %d (Format: Skor1 Skor2): ", gameCount)
    fmt.Scanln(&score1, &score2)

    if score1 < 0 || score2 < 0 {
        break
    }

    switch {
    case score1 > score2:
        results = append(results, team1)
    case score1 < score2:
        results = append(results, team2)
    default:
        results = append(results, "Draw")
    }

    gameCount++
}

fmt.Println("Semua pertandingan telah selesai")
for i, result := range results {
    fmt.Printf("Hasil pertandingan %d: %s\n", i+1, result)
}
}

```

### Screenshot output

```

PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\New folder\UNGUIDED 3.go"
Nama Klub 1: MADRID
Nama Klub 2: CITY
Skor Pertandingan 1 (Format: Skor1 Skor2): 2 1
Skor Pertandingan 2 (Format: Skor1 Skor2): 4 5
Skor Pertandingan 3 (Format: Skor1 Skor2): 9 0
Skor Pertandingan 4 (Format: Skor1 Skor2): 4 1
Skor Pertandingan 5 (Format: Skor1 Skor2): 5 2
Skor Pertandingan 6 (Format: Skor1 Skor2): 7 3
Skor Pertandingan 7 (Format: Skor1 Skor2): 6 0
Skor Pertandingan 8 (Format: Skor1 Skor2): 1 1
Skor Pertandingan 9 (Format: Skor1 Skor2): 1 4
Skor Pertandingan 10 (Format: Skor1 Skor2): 4 1
Skor Pertandingan 11 (Format: Skor1 Skor2): -2 5
Semua pertandingan telah selesai
Hasil pertandingan 1: MADRID
Hasil pertandingan 2: CITY
Hasil pertandingan 3: MADRID
Hasil pertandingan 4: MADRID
Hasil pertandingan 5: MADRID
Hasil pertandingan 6: MADRID
Hasil pertandingan 7: MADRID
Hasil pertandingan 8: Draw
Hasil pertandingan 9: CITY
Hasil pertandingan 10: MADRID

```

### Penjelasan

Program meminta input nama dua klub sepak bola dan skor pertandingan untuk menentukan pemenang dari setiap pertandingan. Program akan terus meminta skor untuk setiap pertandingan dan mencatat pemenang hingga skor negatif dimasukkan, yang menandakan akhir dari input. Skor pertandingan kemudian dievaluasi: jika skor klub A lebih tinggi, maka klub A dinyatakan pemenang, begitu pula sebaliknya untuk klub B. Jika skornya seri, hasilnya dicatat sebagai "Draw". Pada akhirnya, program menampilkan hasil pemenang dari semua pertandingan yang telah dimainkan.

#### Unguided 4

```
package main

// NAMA : CHRIST DANIEL SANTOSO
// NIM : 2311102305

import "fmt"

const maxSize int = 127

type ArrayChar struct {
    data [maxSize]rune
    size int
}

func inputArray(arr *ArrayChar, length *int) {
    fmt.Print("Masukkan karakter (akhiri dengan titik): ")
    var char rune
    for *length = 0; *length < maxSize; *length++ {
        fmt.Scanf("%c", &char)
        if char == '.' {
            break
        }
        arr.data[*length] = char
    }
    arr.size = *length
}

func printArray(arr ArrayChar, length int) {
    for i := 0; i < length; i++ {
        fmt.Printf("%c", arr.data[i])
    }
}
```

```

    fmt.Println()
}

func reverseArray(arr *ArrayChar, length int) {
    for i := 0; i < length/2; i++ {
        arr.data[i], arr.data[length-i-1] = arr.data[length-i-1], arr.data[i]
    }
}

func isPalindrome(arr ArrayChar, length int) bool {
    for i := 0; i < length/2; i++ {
        if arr.data[i] != arr.data[length-i-1] {
            return false
        }
    }
    return true
}

func main() {
    var arr ArrayChar
    var length int

    inputArray(&arr, &length)

    fmt.Print("Teks: ")
    printArray(arr, length)

    fmt.Print("Reverse teks: ")
    reverseArray(&arr, length)
    printArray(arr, length)

    if isPalindrome(arr, length) {
        fmt.Println("Palindrom: true")
    } else {
        fmt.Println("Palindrom: false")
    }
}

```

Screenshot output

```
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\New folder\tempCodeRunnerFile.go"
Masukkan karakter (akhiri dengan titik): DINO.
Teks: DINO
Reverse teks: ONID
Palindrom: false
PS C:\Users\cynzw> go run "c:\Users\cynzw\OneDrive\Documents\New folder\tempCodeRunnerFile.go"
Masukkan karakter (akhiri dengan titik): KATAK.
Teks: KATAK
Reverse teks: KATAK
Palindrom: true
PS C:\Users\cynzw> █
```

## Penjelasan

Program meminta input karakter dari pengguna hingga karakter titik (.) dimasukkan, kemudian menampilkan isi array karakter, membalik susunan array, dan mencetak hasilnya. Selain itu, program memeriksa apakah array karakter tersebut membentuk palindrom. Program menggunakan prosedur isiArray untuk pengisian, cetakArray untuk mencetak, balikanArray untuk membalik array, dan palindrom untuk mengecek kondisi palindrom.