

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 7**

**STRUCK & ARRAY**



Oleh:

Arnanda setya nosa putra

2311102180

IF-11-02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

## **I. DASAR TEORI**

- **Pengertian Array**

Array adalah struktur data yang digunakan untuk menyimpan sejumlah elemen bertipe data yang sama dalam urutan memori yang berkesinambungan. Setiap elemen dalam array memiliki posisi unik yang dikenal sebagai indeks, dimulai dari angka 0. Dengan menggunakan indeks ini, elemen-elemen dalam array dapat diakses dengan cepat. Array memiliki beberapa karakteristik utama yang membedakannya dari struktur data lainnya:

1. Homogenitas: Semua elemen dalam array harus bertipe data yang sama, memastikan konsistensi data.
2. Ukuran Tetap: Saat array dibuat, ukuran atau jumlah elemen di dalamnya ditentukan dan tidak dapat diubah selama program berjalan.
3. Akses Acak: Array memungkinkan akses langsung ke setiap elemen melalui indeksnya, sehingga proses pengambilan atau modifikasi data dapat dilakukan dengan cepat.
4. Banyak Dimensi: Array dapat memiliki lebih dari satu dimensi, seperti array satu dimensi (sering disebut vektor) atau array dua dimensi (sering disebut matriks), yang memungkinkan penyimpanan data lebih kompleks.

- **Pengertian Struct**

Struct, atau struktur, adalah tipe data di dalam bahasa pemrograman yang memungkinkan pengelompokan beberapa variabel dengan tipe data berbeda ke dalam satu entitas logis. Struct sangat berguna untuk mengelola data yang memiliki berbagai tipe namun terkait secara logis, seperti data mahasiswa yang mencakup nama, umur, dan nilai. Variabel yang ada di dalam struct dikenal sebagai *\*field\** atau elemen. Struct memiliki karakteristik yang mencakup:

1. Beragam Tipe Data: Struct dapat mengandung berbagai tipe data dalam satu unit, seperti integer, float, dan string, memungkinkan representasi data yang kompleks.
2. Organisasi Data yang Rapi: Struct memungkinkan penyatuan data terkait ke dalam satu kesatuan yang terstruktur, sehingga lebih mudah dipahami dan diakses.
3. Deklarasi yang Fleksibel: Struct dapat dideklarasikan dalam beberapa cara, baik dengan langsung mendefinisikan dan menginisialisasi field di dalamnya atau dengan menyusunnya secara bertahap, sehingga lebih mudah disesuaikan dengan kebutuhan program.

## II. GUIDED

### Guided 1

```
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
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)
    }
}

```

Screenshot output

```

PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.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\HP>

```

Penjelasan :

Program di atas mengelola dan mengurutkan data mahasiswa berdasarkan nilai rata-rata. Program dimulai dengan mendeklarasikan tipe data `Mahasiswa` yang memiliki atribut nama, nilai matematika, fisika, kimia, dan rata-rata. Fungsi `hitungRataRata` digunakan untuk menghitung rata-rata nilai matematika, fisika, dan kimia tiap mahasiswa. Dalam fungsi `main`, beberapa data mahasiswa dimasukkan ke dalam array, kemudian program menghitung rata-rata nilai tiap mahasiswa. Selanjutnya, data mahasiswa diurutkan berdasarkan rata-rata (dari tertinggi ke terendah) menggunakan fungsi `sort.Slice`. Akhirnya, program menampilkan daftar mahasiswa yang sudah diurutkan sesuai nilai rata-ratanya.

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

Output :

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\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
-----
20231001  Andi
20231002  Budi
20231004  Dedi
PS C:\Users\HP>
```

Penjelasan :

Program di atas mengelola data mahasiswa menggunakan struktur `map`. Program dimulai dengan membuat `map` bernama `mahasiswa` dengan NIM (Nomor Induk Mahasiswa) sebagai kunci dan nama mahasiswa sebagai nilainya. Program kemudian menambahkan data baru ke `map`, menampilkan seluruh data mahasiswa dalam format tabel, dan mengakses data tertentu berdasarkan NIM. Selanjutnya, program menghapus salah satu data mahasiswa berdasarkan NIM tertentu dan menampilkan kembali daftar mahasiswa setelah penghapusan. Program ini menggunakan beberapa fungsi dari package `fmt` untuk mencetak hasil dalam format yang rapi.

### III. UNGUIDED

#### Unguided 1

```
package main

import (
    "fmt"
    "math"
)

type titik struct {
    x, y int
}

type lingkaran struct {
    pusat 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.pusat, p) <= float64(c.radius)
}

func main() {
    var cx1, cy1, r1 int
    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 1:")
    fmt.Scan(&cx1, &cy1, &r1)
    lingkaran1 := lingkaran{pusat: titik{x: cx1, y: cy1}, radius: r1}

    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}

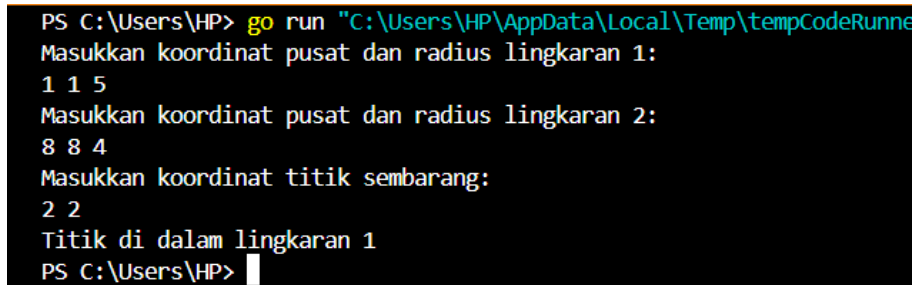
    diLingkaran1 := didalam(lingkaran1, titikSembarang)
    diLingkaran2 := didalam(lingkaran2, titikSembarang)
```

```

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")
}
}

```

Screenshot output



```

PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunne
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\HP>

```

### Penjelasan

Program di atas menentukan posisi sebuah titik sembarang berdasarkan dua lingkaran. Program dimulai dengan mendefinisikan struktur `titik` untuk koordinat `(x, y)` dan `lingkaran` untuk menyimpan pusat lingkaran serta radiusnya. Fungsi `jarak` digunakan untuk menghitung jarak antara dua titik, sementara fungsi `didalam` mengecek apakah sebuah titik berada di dalam lingkaran tertentu. Program menerima input untuk koordinat pusat dan radius kedua lingkaran serta koordinat titik sembarang, lalu mengevaluasi posisi titik tersebut apakah berada di dalam lingkaran 1, lingkaran 2, keduanya, atau di luar keduanya, dan menampilkan hasilnya.



## Unguided 2

```
package main

import (
    "fmt"
    "math"
)

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

    arr := make([]int, N)

    for i := 0; i < N; i++ {
        fmt.Printf("Masukkan nilai untuk elemen ke-%d: ", i)
        fmt.Scanln(&arr[i])
    }

    fmt.Println("a. Isi keseluruhan array:", arr)

    fmt.Print("b. Elemen dengan indeks ganjil: ")
    for i := 1; i < N; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    fmt.Print("c. Elemen dengan indeks genap: ")
    for i := 0; i < N; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    var x int
    fmt.Print("Masukkan nilai x untuk menampilkan elemen dengan indeks kelipatan x: ")
    fmt.Scanln(&x)
    fmt.Print("d. Elemen dengan indeks kelipatan ", x, ": ")
    for i := x; i < N; i += x {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}
```

```

var delIndex int
fmt.Print("Masukkan indeks elemen yang ingin dihapus: ")
fmt.Scanln(&delIndex)
if delIndex >= 0 && delIndex < N {
    arr = append(arr[:delIndex], arr[delIndex+1:]...)
}
fmt.Println("e. Isi array setelah penghapusan:", arr)

sum := 0
for _, val := range arr {
    sum += val
}
average := float64(sum) / float64(len(arr))
fmt.Printf("f. Rata-rata elemen: %.2f\n", average)

var deviationSum float64
for _, val := range arr {
    deviationSum += math.Pow(float64(val)-average, 2)
}
standardDeviation := math.Sqrt(deviationSum / float64(len(arr)))
fmt.Printf("g. Standar deviasi: %.2f\n", standardDeviation)

var searchValue int
fmt.Print("Masukkan bilangan untuk menghitung frekuensinya di
dalam array: ")
fmt.Scanln(&searchValue)
frequency := 0
for _, val := range arr {
    if val == searchValue {
        frequency++
    }
}
fmt.Printf("h. Frekuensi bilangan %d di dalam array: %d\n",
searchValue, frequency)
}

```

### Screenshot output

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan jumlah elemen array: 10
Masukkan nilai untuk elemen ke-0: 1
Masukkan nilai untuk elemen ke-1: 2
Masukkan nilai untuk elemen ke-2: 3
Masukkan nilai untuk elemen ke-3: 4
Masukkan nilai untuk elemen ke-4: 5
Masukkan nilai untuk elemen ke-5: 6
Masukkan nilai untuk elemen ke-6: 7
Masukkan nilai untuk elemen ke-7: 8
Masukkan nilai untuk elemen ke-8: 9
Masukkan nilai untuk elemen ke-9: 10
a. Isi keseluruhan array: [1 2 3 4 5 6 7 8 9 10]
b. Elemen dengan indeks ganjil: 2 4 6 8 10
c. Elemen dengan indeks genap: 1 3 5 7 9
Masukkan nilai x untuk menampilkan elemen dengan indeks kelipatan x: 3
d. Elemen dengan indeks kelipatan 3: 4 7 10
Masukkan indeks elemen yang ingin dihapus: 2
e. Isi array setelah penghapusan: [1 2 4 5 6 7 8 9 10]
f. Rata-rata elemen: 5.78
g. Standar deviasi: 2.90
Masukkan bilangan untuk menghitung frekuensinya di dalam array: 6
h. Frekuensi bilangan 6 di dalam array: 1
PS C:\Users\HP>
```

### Penjelasan

Program di atas mengelola array bilangan bulat dengan berbagai fungsi. Pertama, program meminta pengguna untuk memasukkan jumlah elemen yang diinginkan untuk array. Setelah itu, pengguna diminta untuk mengisi nilai untuk setiap elemen array. Program kemudian menampilkan seluruh isi array, elemen dengan indeks ganjil, elemen dengan indeks genap, dan elemen pada indeks kelipatan bilangan yang dimasukkan pengguna. Selain itu, program juga memungkinkan pengguna untuk menghapus elemen pada indeks tertentu dan menampilkan isi array setelah penghapusan. Program menghitung rata-rata dan standar deviasi dari elemen-elemen dalam array, serta menghitung frekuensi kemunculan bilangan tertentu yang diminta pengguna dalam array.

### Unguided 3

```
package main

import (
    "fmt"
)

func main() {
    var klubA, klubB string
    fmt.Print("Klub A: ")
    fmt.Scanln(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scanln(&klubB)

    var hasil []string

    var skorA, skorB int
    pertandingan := 1

    for {
        fmt.Printf("Pertandingan %d: ", pertandingan)
        fmt.Scanln(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            fmt.Println("Pertandingan selesai")
            break
        }

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

        pertandingan++
    }

    for i, result := range hasil {
        fmt.Printf("Hasil %d: %s\n", i+1, result)
    }
    fmt.Println("Pertandingan selesai")
}
```

### Screenshot output

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunner"
Klub A: MU
Klub B: Inter
Pertandingan 1: 2 0
Pertandingan 2: 1 2
Pertandingan 3: 2 2
Pertandingan 4: 0 1
Pertandingan 5: 3 2
Pertandingan 6: 1 0
Pertandingan 7: 5 2
Pertandingan 8: 2 3
Pertandingan 9: -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
Pertandingan selesai
PS C:\Users\HP>
```

### Penjelasan

Program di atas merekam hasil pertandingan antara dua klub sepak bola. Pengguna diminta untuk memasukkan nama kedua klub, setelah itu program meminta input skor untuk setiap pertandingan dalam format angka. Skor akan diinput dalam sebuah loop hingga salah satu atau kedua skor bernilai negatif, yang menandakan akhir dari input pertandingan. Hasil dari setiap pertandingan disimpan dalam slice `hasil`, di mana klub yang menang dicatat, atau "Draw" jika hasilnya imbang. Setelah loop berakhir, program mencetak semua hasil pertandingan yang telah direkam dan menampilkan pesan "Pertandingan selesai."

## Unguided 4

```
package main
import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var char rune
    fmt.Println("Masukkan karakter (akhiri dengan titik .): ")
    for *n = 0; *n < NMAX; *n++ {
        fmt.Scanf("%c", &char)
        if char == '.' {
            break
        }
        t[*n] = char
    }
}

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

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

func palindrom(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() {  
    var tab tabel  
    var m int  
  
    isiArray(&tab, &m)  
  
    fmt.Print("Teks: ")  
    cetakArray(tab, m)  
  
    balikanArray(&tab, m)  
  
    fmt.Print("Reverse teks: ")  
    cetakArray(tab, m)  
  
    if palindrom(tab, m) {  
        fmt.Println("Palindrom: true")  
    } else {  
        fmt.Println("Palindrom: false")  
    }  
}
```

## Screenshot output

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerF
Masukkan karakter (akhiri dengan titik.):
KATAK.
Teks: KATAK
Reverse teks: KATAK
Palindrom: true
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerF
Masukkan karakter (akhiri dengan titik.):
SENANG.
Teks: SENANG
Reverse teks: GNANES
Palindrom: false
PS C:\Users\HP> █
```

## Penjelasan

Program di atas mengelola array karakter (`tabel`) dan memiliki beberapa fungsi untuk mengisi array, mencetaknya, membalik urutan elemen, dan memeriksa apakah array tersebut adalah palindrom. Pertama, pengguna diminta untuk memasukkan karakter satu per satu, yang akan disimpan dalam array `tabel` hingga maksimal `NMAX` karakter atau sampai pengguna memasukkan titik (`.`) untuk mengakhiri input. Program kemudian mencetak teks asli, membalikinya, dan mencetak teks hasil balikkannya. Terakhir, program memeriksa apakah teks yang dimasukkan adalah palindrom (teks yang sama jika dibaca dari depan atau belakang) dan mencetak hasilnya sebagai `true` atau `false`.