

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 2**  
**MODUL VII**  
**STRUCK & ARRAY**



**Disusun Oleh :**

**Tegar Aji Pangestu / 2311102021**

**IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi, S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

Struktur data adalah cara penyimpanan dan pengorganisasian data untuk mempermudah akses dan manipulasi. Salah satu struktur data yang sering digunakan adalah *struct* (singkatan dari struktur) dalam pemrograman. *Struct* memungkinkan pengelompokan berbagai tipe data yang berbeda ke dalam satu entitas yang disebut struktur. Sebagai contoh, struktur dapat digunakan untuk merepresentasikan entitas dunia nyata, seperti data mahasiswa yang meliputi nama, NIM, dan jurusan. Dengan menggunakan *struct*, data yang terkait dapat dikelompokkan secara logis, sehingga lebih mudah untuk dikelola dan dipahami.

*Struct* terdiri dari beberapa anggota (member) yang masing-masing memiliki tipe data dan nilai tersendiri. Dalam implementasinya, *struct* dapat digunakan untuk membuat tipe data baru yang fleksibel. Di dalam bahasa pemrograman seperti C atau C++, *struct* dideklarasikan menggunakan kata kunci *struct*, diikuti dengan definisi anggota-anggota di dalam blok kurung kurawal {}.

Array adalah kumpulan elemen yang memiliki tipe data yang sama dan disimpan secara berurutan di dalam memori. Elemen-elemen array diakses menggunakan indeks, dengan indeks pertama biasanya dimulai dari nol. Array digunakan untuk menyimpan data dalam jumlah besar secara efisien. Misalnya, array dapat digunakan untuk menyimpan nilai-nilai ujian dari banyak siswa dalam satu struktur data. Hal ini mempermudah pengolahan data seperti iterasi, pencarian, atau modifikasi.

*Struct* dan array sering digunakan secara bersamaan untuk mengelola data kompleks. Sebagai contoh, array dari *struct* dapat digunakan untuk merepresentasikan kumpulan data yang lebih besar, seperti daftar mahasiswa dalam sebuah kelas. Dengan menggunakan array dari *struct*, kita dapat menyimpan data setiap mahasiswa dalam elemen-elemen array dan mengaksesnya dengan mudah menggunakan indeks. Kombinasi ini sering

digunakan dalam aplikasi yang membutuhkan pengolahan data skala besar, seperti sistem informasi akademik, inventarisasi barang, atau database sederhana.

## II. GUIDED

### 1. Guided 1

#### Sourcecode

```
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 l1, l2 Lingkaran
    var t Titik
    fmt.Print("Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): ")
    fmt.Scan(&l1.pusat.x, &l1.pusat.y, &l1.radius)

    fmt.Print("Masukkan koordinat pusat dan radius lingkaran 2 (cx2 cy2 r2): ")
    fmt.Scan(&l2.pusat.x, &l2.pusat.y, &l2.radius)

    fmt.Print("Masukkan koordinat titik sembarang (x y): ")
    fmt.Scan(&t.x, &t.y)
    diDalamL1 := diDalam(l1, t)
    diDalamL2 := diDalam(l2, t)

    if diDalamL1 && diDalamL2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if diDalamL1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diDalamL2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

```
}  
}
```

### Screenshoot Output

```
Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): 1 1 5  
Masukkan koordinat pusat dan radius lingkaran 2 (cx2 cy2 r2): 8 8 4  
Masukkan koordinat titik sembarang (x y): 2 2  
Titik di dalam lingkaran 1
```

### Deskripsi Program

Fungsi jarak Fungsi ini menghitung jarak antara dua titik pada bidang kartesius menggunakan rumus jarak Euclidean, Fungsi ini menerima dua parameter bertipe Titik (p dan q) dan mengembalikan nilai bertipe float64 sebagai hasil perhitungan jarak.

.

### III. UNGUIDED

#### Unguided 1

##### Sourcecode

```
package main
import (
    "fmt"
    "math"
)

//Tegar Aji Pangestu 2311102021//

func tampilSemua(array []int) {
    fmt.Println("Isi array:", array)
}

func tampilGanjil(array []int) {
    fmt.Print("Elemen dengan indeks ganjil: ")
    for i := 1; i < len(array); i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}

func tampilGenap(array []int) {
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 0; i < len(array); i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}

func tampilKelipatan(array []int, x int) {
    fmt.Printf("Elemen dengan indeks kelipatan %d: ", x)
    for i := x; i < len(array); i += x {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}

func hapusIndeks(array []int, indeks int) []int {
    fmt.Printf("Menghapus elemen pada indeks %d\n",
    indeks)
    return append(array[:indeks], array[indeks+1:]...)
}

func rataRata(array []int) float64 {
    total := 0
    for _, val := range array {
        total += val
    }
    return float64(total) / float64(len(array))
}
```

```

}

func standarDeviasi(array []int) float64 {
    mean := rataRata(array)
    var sum float64
    for _, val := range array {
        sum += math.Pow(float64(val)-mean, 2)
    }
    return math.Sqrt(sum / float64(len(array)))
}

func frekuensi(array []int, nilai int) int {
    count := 0
    for _, val := range array {
        if val == nilai {
            count++
        }
    }
    return count
}

func main() {
    var N, x, hapusIdx, cariFrekuensi int

    fmt.Print("Masukkan jumlah elemen array (N): ")
    fmt.Scan(&N)

    array := make([]int, N)

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

    tampilSemua(array)
    tampilGanjil(array)
    tampilGenap(array)

    fmt.Print("Masukkan nilai x untuk menampilkan elemen
dengan indeks kelipatan x: ")
    fmt.Scan(&x)
    tampilKelipatan(array, x)

    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&hapusIdx)
    if hapusIdx >= 0 && hapusIdx < len(array) {
        array = hapusIndeks(array, hapusIdx)
        tampilSemua(array)
    } else {
        fmt.Println("Indeks yang dimasukkan tidak valid!")
    }
}

```

```

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

        fmt.Printf("Standar deviasi elemen array: %.2f\n",
standarDeviasi(array))

        fmt.Print("Masukkan bilangan yang ingin dicari
frekuensinya: ")
        fmt.Scan(&cariFrekuensi)
        fmt.Printf("Frekuensi %d di dalam array: %d\n",
cariFrekuensi, frekuensi(array, cariFrekuensi))
    }

```

### Screenshoot Output

```

Masukkan jumlah elemen array (N): 4
Masukkan elemen array:
Elemen ke-0: 4
Elemen ke-1: 6
Elemen ke-2: 8
Elemen ke-3: 2
Isi array: [4 6 8 2]
Elemen dengan indeks ganjil: 6 2
Elemen dengan indeks genap: 4 8
Masukkan nilai x untuk menampilkan elemen dengan indeks kelipatan x: 2
Elemen dengan indeks kelipatan 2: 8
Masukkan indeks yang ingin dihapus: 3
Menghapus elemen pada indeks 3
Isi array: [4 6 8]
Rata-rata elemen array: 6.00
Standar deviasi elemen array: 1.63
Masukkan bilangan yang ingin dicari frekuensinya: 6
Frekuensi 6 di dalam array: 1

```

### Deskripsi Program

Program meminta input data array dan menampilkan semua elemen serta elemen dengan indeks tertentu. Setelah input nilai x, program menampilkan elemen dengan indeks kelipatan x. Program kemudian menghapus elemen pada indeks yang dipilih dan menampilkan array hasilnya. Rata-rata dan standar deviasi dihitung berdasarkan elemen array setelah penghapusan. Frekuensi kemunculan nilai tertentu dalam array dihitung dan ditampilkan.

### Unguided 2



## Sourcecode

```
package main
import (
    "fmt"
)

//Tegar Aji Pangestu 2311102021//

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

    // Meminta nama klub yang bertanding
    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&klubA)
    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scanln(&klubB)

    for i := 1; ; i++ {
        fmt.Printf("Pertandingan %d - Masukkan skor %s: ", i, klubA)
        fmt.Scan(&skorA)
        fmt.Printf("Pertandingan %d - Masukkan skor %s: ", i, klubB)
        fmt.Scan(&skorB)

        // Jika skor negatif, keluar dari loop
        if skorA < 0 || skorB < 0 {
            fmt.Println("Skor tidak valid. Pertandingan selesai.")
            break
        }

        // Tentukan pemenang dan simpan ke dalam array
        jika ada pemenang
        if skorA > skorB {
            pemenang = append(pemenang, klubA)
            fmt.Printf("Hasil %d: %s\n", i, klubA)
        } else if skorB > skorA {
            pemenang = append(pemenang, klubB)
            fmt.Printf("Hasil %d: %s\n", i, klubB)
        } else {
            fmt.Printf("Hasil %d: Draw\n", i)
        }
    }

    // Tampilkan daftar klub yang memenangkan pertandingan
    fmt.Println("\nDaftar klub yang memenangkan pertandingan:")
    for _, klub := range pemenang {
```

```
        fmt.Println(klub)
    }
}
```

### Screenshoot Output

```
Masukkan nama Klub A: Indonesia
Masukkan nama Klub B: Japan
Pertandingan 1 - Masukkan skor Indonesia: 0 4
Pertandingan 1 - Masukkan skor Japan: Hasil 1: Japan
Pertandingan 2 - Masukkan skor Indonesia: 0 4
Pertandingan 2 - Masukkan skor Japan: Hasil 2: Japan
Pertandingan 3 - Masukkan skor Indonesia: 0 4
Pertandingan 3 - Masukkan skor Japan: Hasil 3: Japan
Pertandingan 4 - Masukkan skor Indonesia: -1 4
Pertandingan 4 - Masukkan skor Japan: Skor tidak valid. Pertandingan selesai.

Daftar klub yang memenangkan pertandingan:
Japan
Japan
Japan
```

### Deskripsi Program

Program menggunakan perulangan tanpa batas (for i := 1; ;) untuk meminta skor dari setiap pertandingan. Setiap iterasi merepresentasikan satu pertandingan: Pengguna memasukkan skor untuk klubA dan klubB (disimpan dalam skorA dan skorB). Jika salah satu skor negatif, program menghentikan loop dengan pesan "Skor tidak valid. Pertandingan selesai."

### Unguided 3

#### Sourcecode

```
package main
import (
    "fmt"
)

//Tegar Aji Pangestu 2311102021//

const NMAX int = 127

type tabel [NMAX]rune

// Fungsi untuk mengisi array dengan karakter dari input
user
func isiArray(t *tabel, n *int) {
```

```

var ch rune
*n = 0
for {
    fmt.Scanf("%c", &ch)
    if ch == '\n' || ch == 'T' {
        break
    }
    t[*n] = ch
    *n++
    if *n >= NMAX {
        break
    }
}

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

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

// Fungsi untuk memeriksa apakah array membentuk palindrom
func palindrom(t tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t[i] != t[n-1-i] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var n int

    // Mengisi array dengan input dari user
    fmt.Print("Masukkan teks: ")
    isiArray(&tab, &n)

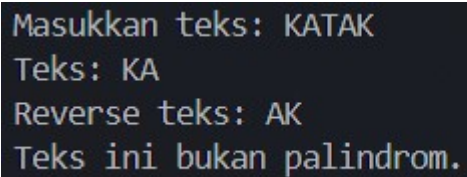
    // Mencetak array asli
    fmt.Print("Teks: ")
    cetakArray(tab, n)
}

```

```
// Membalikkan isi array
balikanArray(&tab, n)
fmt.Print("Reverse teks: ")
cetakArray(tab, n)

// Memeriksa apakah array membentuk palindrom
if palindrom(tab, n) {
    fmt.Println("Teks ini adalah palindrom.")
} else {
    fmt.Println("Teks ini bukan palindrom.")
}
}
```

### Screenshoot Output



```
Masukkan teks: KATAK
Teks: KA
Reverse teks: AK
Teks ini bukan palindrom.
```

### Deskripsi Program

balikanArray Fungsi ini membalikkan isi array dengan menukar elemen pertama dengan elemen terakhir, elemen kedua dengan elemen kedua terakhir, dan seterusnya hingga mencapai tengah array. cetakArray Fungsi ini mencetak semua elemen array hingga indeks n ke layar sebagai sebuah teks. palindrom Fungsi ini memeriksa apakah teks yang ada dalam array merupakan *palindrom* (teks yang terbaca sama dari depan dan belakang).