

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

MODUL 7

STRUCK & ARRAY



Oleh:

YASVIN SYAHGANA

2311102065

IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

(*)Pengertian Array

Array adalah struktur data yang digunakan untuk menyimpan kumpulan elemen dengan tipe data yang sama dalam memori secara berurutan. Array memiliki posisi khusus untuk setiap elemen, yang dimulai dari 0. Dengan menggunakan indeks ini, kita dapat dengan cepat mengakses elemen-elemen array. Array memiliki beberapa karakteristik utama yang membedakannya dari struktur data lainnya:

1. **Homogenitas:** Setiap elemen dalam array harus memiliki tipe data yang sama, menjaga konsistensi data.
2. **Ukuran Tetap:** Ukuran array ditentukan saat pembuatan dan tidak dapat diubah selama program berjalan.
3. **Akses Acak:** Array memungkinkan akses langsung ke elemen melalui indeks, sehingga pengambilan atau modifikasi data dapat dilakukan dengan cepat.
4. **Banyak Dimensi:** Array dapat berbentuk satu dimensi (vektor) atau lebih, seperti array dua dimensi (matriks), yang mendukung penyimpanan data yang lebih kompleks.

(*)Pengertian Struct

Struct, atau struktur, adalah tipe data yang memungkinkan pengelompokan variabel dari berbagai tipe data dalam satu kesatuan logis. Struct sangat berguna dalam mengelola data kompleks yang memiliki beberapa tipe data berbeda namun saling terkait, seperti data mahasiswa yang terdiri dari nama, umur, dan nilai. Variabel dalam struct disebut sebagai *field* atau elemen. Struct memiliki karakteristik sebagai berikut:

1. **Beragam Tipe Data:** Struct dapat mengandung berbagai tipe data dalam satu unit, seperti integer, float, dan string, sehingga cocok untuk merepresentasikan data yang kompleks.
2. **Organisasi Data yang Terstruktur:** Struct menyatukan data terkait ke dalam satu kesatuan yang terorganisir, membuatnya lebih mudah dipahami dan diakses.
3. **Deklarasi yang Fleksibel:** Struct dapat dideklarasikan dengan berbagai cara, baik dengan mendefinisikan dan menginisialisasi field secara langsung, atau menyusunnya bertahap sesuai 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 E:\LAPORAN PRAKTIKUM> go run "e:\LAPORAN PRAKTIKUM\YASVIN SYAHGANA_2311102065_Laprak 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. Aldi - 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 E:\LAPORAN PRAKTIKUM>
```

Penjelasan :

Program Data siswa diurus dan diurutkan berdasarkan nilai rata-rata. Program dimulai dengan mendeklarasikan tipe data "Mahasiswa", yang mencakup atribut nama, nilai matematika, fisika, dan kimia, dan nilai rata-rata. Kemudian, dalam fungsi "main", beberapa data siswa dimasukkan ke dalam array, dan program menghitung rata-rata nilai matematika, fisika, dan kimia tiap siswa. Selanjutnya, fungsi "sort.Slice" digunakan untuk mengurutkan data siswa berdasarkan rata-rata, diurutkan dari tertinggi ke terendah. Pada akhirnya, program menampilkan daftar siswa yang telah diurutkan berdasarkan nilai rata-ratanya.

Guided 2

```
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 :

```
PS E:\LAPORAN PRAKTIKUM> go run "e:\LAPORAN PRAKTIKUM\YASVIN SYAHGANA_2311102065_Laprak 7\guided 2.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 E:\LAPORAN PRAKTIKUM>
```

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 {
    cx, cy, r int
}

func jarak(p titik, c lingkaran) float64 {

    return math.Sqrt(float64((p.x-c.cx)*(p.x-c.cx) + (p.y-c.cy)*(p.y-c.cy)))
}

func posisiTitik(p titik, l1, l2 lingkaran) string {
    jarak1 := jarak(p, l1)
    jarak2 := jarak(p, l2)

    dalam1 := jarak1 < float64(l1.r)
    dalam2 := jarak2 < float64(l2.r)

    if dalam1 && dalam2 {
        return "Titik di dalam lingkaran 1 dan 2"
    } else if dalam1 {
        return "Titik di dalam lingkaran 1"
    } else if dalam2 {
        return "Titik di dalam lingkaran 2"
    } else {
        return "Titik di luar lingkaran 1 dan 2"
    }
}

func main() {
    var l1, l2 lingkaran
    var p titik

    fmt.Print("Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r): ")
    fmt.Scan(&l1.cx, &l1.cy, &l1.r)
```

```
fmt.Print("Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r): ")
fmt.Scan(&l2.cx, &l2.cy, &l2.r)

fmt.Print("Masukkan koordinat titik sembarang (x y): ")
//fahrur059
fmt.Scan(&p.x, &p.y)
fmt.Print("\n")

hasil := posisiTitik(p, l1, l2)
fmt.Println(hasil)
}
```


Screenshot output

```
PS E:\LAPORAN PRAKTIKUM> go run "e:\LAPORAN PRAKTIKUM\YASVIN SYAHKANA_2311162065_Laprak 7\unguided 1.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r): 1 2 3
Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r): 4 5 6
Masukkan koordinat titik sembarang (x y): 7 8

Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r): 1 2 3
Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r): 4 5 6
Masukkan koordinat titik sembarang (x y): 7 8

Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r): 4 5 6
Masukkan koordinat titik sembarang (x y): 7 8

Masukkan koordinat titik sembarang (x y): 7 8

Titik di dalam lingkaran 2
PS E:\LAPORAN PRAKTIKUM>
```

Penjelasan

Program menentukan posisi sebuah titik berdasarkan dua lingkaran di atas. Program dimulai dengan menciptakan struktur "titik" untuk koordinat "(x, y)" dan "lingkaran" untuk menyimpan pusat dan radius lingkaran. "Jarak" menghitung jarak antara dua titik, dan "di dalam" mengevaluasi 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, x, delIndex, searchValue int

    fmt.Print("Masukkan jumlah elemen array: ")
    fmt.Scan(&n)
    array := make([]int, n)

    fmt.Println("Masukkan elemen-elemen array:")
    for i := 0; i < n; i++ {
        fmt.Printf("Elemen ke-%d: ", i+1)
        fmt.Scan(&array[i])
    }

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

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

    fmt.Print("c. Elemen dengan indeks genap: ")
    for i := 0; i < n; i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
    fmt.Print("Masukkan nilai x untuk menampilkan elemen dengan indeks kelipatan x: ")
    fmt.Scan(&x)
    fmt.Printf("d. Elemen dengan indeks kelipatan %d: ", x)
    for i := x; i < n; i += x {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
    fmt.Print("Masukkan indeks elemen yang ingin dihapus: ")
    fmt.Scan(&delIndex)
    if delIndex >= 0 && delIndex < n {
        array = append(array[:delIndex], array[delIndex+1:]...)
        fmt.Println("e. Isi array setelah penghapusan:", array)
    } else {
        fmt.Println("Indeks tidak valid.")
    }

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

    var deviationSum float64
    for _, num := range array {
        deviationSum += math.Pow(float64(num)-average, 2)
    }
}
```

```

    }
    standardDeviation := math.Sqrt(deviationSum / float64(len(array)))
    fmt.Printf("g. Standar deviasi: %.2f\n", standardDeviation)

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

```

Screenshot output

```

PS E:\LAPORAN PRAKTIKUM> go run "e:\LAPORAN PRAKTIKUM\YASVIN SYAHGANA_2311102065_Laprak 7\unguided 2.go"
Masukkan jumlah elemen array: 6
Masukkan elemen-elemen array:
Elemen ke-1: 1
Elemen ke-2: 2
Elemen ke-3: 3
Elemen ke-4: 4
Elemen ke-5: 5
Elemen ke-6: 6
a. Isi keseluruhan array: [1 2 3 4 5 6]
b. Elemen dengan indeks ganjil: 2 4 6
c. Elemen dengan indeks genap: 1 3 5
Masukkan nilai x untuk menampilkan elemen dengan indeks kelipatan x: 3
d. Elemen dengan indeks kelipatan 3: 4
Masukkan indeks elemen yang ingin dihapus: 2
e. Isi array setelah penghapusan: [1 2 4 5 6]
f. Rata-rata elemen: 3.60
g. Standar deviasi: 1.85
Masukkan nilai untuk menghitung frekuensinya di dalam array: 5
h. Frekuensi bilangan 5 di dalam array: 1
PS E:\LAPORAN PRAKTIKUM>

```

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 clubA, clubB string
    var scoreA, scoreB int
    var results []string

    fmt.Print("Klub A: ")
    fmt.Scanln(&clubA)
    fmt.Print("Klub B: ")
    fmt.Scanln(&clubB)

    matchCount := 1

    for {
        fmt.Printf("Pertandingan %d: ", matchCount)
        fmt.Scan(&scoreA, &scoreB)

        if scoreA < 0 || scoreB < 0 {
            break
        }

        if scoreA > scoreB {
            results = append(results, fmt.Sprintf("Hasil %d: %s", matchCount, clubA))
        } else if scoreB > scoreA {
            results = append(results, fmt.Sprintf("Hasil %d: %s", matchCount, clubB))
        } else {
            results = append(results, fmt.Sprintf("Hasil %d: Draw", matchCount))
        }
        matchCount++
    }

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

Screenshot output

```
PS E:\LAPORAN PRAKTIKUM> go run "e:\LAPORAN PRAKTIKUM\YASVIN SYAHGANA_2311102065_Laprak 7\unguided 2.go"
Klub A: munchen
Klub B: barca
Pertandingan 1: 8 0
Pertandingan 2: 1 0
Pertandingan 3: 1 2
Pertandingan 4: 2 1
Pertandingan 5: 3 0
Pertandingan 6: 4 2
Pertandingan 7: 3 1
Pertandingan 8: 1 2
Pertandingan 9: -1 2
Hasil 1: munchen
Hasil 2: munchen
Hasil 3: barca
Hasil 4: munchen
Hasil 5: munchen
Hasil 6: munchen
Hasil 7: munchen
Hasil 8: barca
Pertandingan selesai
PS E:\LAPORAN PRAKTIKUM> |
```

Penjelasan

Program di atas merekam hasil pertandingan antara dua tim sepak bola. Setelah pengguna memasukkan nama kedua klub, program meminta skor untuk setiap pertandingan dalam format angka. Sampai salah satu atau kedua skor bernilai negatif, input pertandingan berakhir. Setiap pertandingan disimpan dalam bagian "hasil", di mana klub yang memenangkan pertandingan dicatat, atau "Draw" jika hasilnya imbang. Setelah loop berakhir, program menampilkan pesan "Pertandingan selesai" dan mencetak semua hasil pertandingan yang telah direkam.

UNGUIDED 4

```
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 palindrom(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("Reverse teks: ")
balikanArray(&tab, m)
cetakArray(tab, m)

isPalindrom := palindrom(tab, m)
if isPalindrom {
    fmt.Println("Palindrom: true")
} else {
    fmt.Println("Palindrom: false")
}
}
```

Screenshot output

```
PS E:\LAPORAN PRAKTIKUM> go run "e:\LAPORAN PRAKTIKUM\unguided 4.go"
Masukkan karakter (akhiri dengan titik): monyet.
Teks: monyet
Reverse teks: teynom
Palindrom: false
PS E:\LAPORAN PRAKTIKUM> 
```

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, membaliknya, 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`.