

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL VII

STRUCK & ARRAY



Disusun Oleh :

Aji Tri Prasetyo / 2311102064

IF-11-02

Dosen Pengampu :

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Struct adalah tipe data komposit (struktur data) yang digunakan untuk mengelompokkan beberapa data dengan tipe yang berbeda di dalam satu kesatuan. Struct memungkinkan kita untuk mengelompokkan berbagai atribut menjadi satu kesatuan logis.

Penggunaan Struct

Struct biasanya digunakan untuk merepresentasikan data yang memiliki beberapa atribut atau properti. Contoh yang umum adalah merepresentasikan data mahasiswa yang memiliki atribut nama, umur, dan jurusan.

Array adalah struktur data yang digunakan untuk menyimpan sejumlah elemen dengan tipe yang sama di dalam satu kesatuan. Array di Go memiliki panjang tetap yang didefinisikan pada saat deklarasi, dan elemen di dalam array diakses melalui indeks.

Deklarasi Array

Array dapat dideklarasikan dengan menentukan ukuran dan tipe elemen yang akan disimpan di dalamnya.

Kesimpulan

Struct digunakan untuk mengelompokkan berbagai data yang memiliki tipe berbeda menjadi satu unit logis.

Array digunakan untuk menyimpan sekumpulan data yang memiliki tipe yang sama dengan ukuran tetap.

Struct dan array dapat dikombinasikan untuk membuat struktur data yang lebih kompleks, seperti daftar objek yang memiliki atribut tertentu.

Pemahaman tentang struct dan array sangat penting dalam Go karena keduanya merupakan dasar untuk mengelola data secara efisien.

II. GUIDED

1. Guided 1

Sourcecode

```
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 berdasarkan nilai rata-rata
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)
    }
}

```

Screenshoot Output

```

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)

```

Deskripsi Program

Fungsi `hitungRataRata` digunakan untuk menghitung rata-rata nilai Matematika, Fisika, dan Kimia dari setiap mahasiswa. Fungsi ini menerima parameter berupa pointer ke Mahasiswa, menghitung total nilai dari ketiga mata pelajaran, lalu membagi total tersebut dengan 3 untuk mendapatkan rata-rata. Hasil perhitungan disimpan ke dalam bidang `RataRata` dari struct Mahasiswa.

2. Guided 2

Sourcecode

```

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)
}
}

```

Screenshoot Output

```

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

```

Deskripsi Program

Program membuat sebuah map bernama mahasiswa yang memiliki tipe `map[string]string`, di mana string pertama adalah NIM sebagai kunci, dan string kedua adalah nama mahasiswa sebagai nilai. Dalam map ini, tiga data awal mahasiswa dimasukkan: Andi, Budi, dan Cici dengan NIM masing-masing.

III. UNGUIDED

Unguided 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 diDalam: Memeriksa apakah jarak antara titik p dan pusat lingkaran c lebih kecil atau sama dengan radius lingkaran. Jika ya, titik berada di dalam lingkaran.

Unguided 2

Sourcecode

```
package main  
  
import (  
    "fmt"  
    "math"  
)  
  
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): 5
Masukkan elemen array:
Elemen ke-0: 1
Elemen ke-1: 2
Elemen ke-2: 3
Elemen ke-3: 4
Elemen ke-4: 5
Isi array: [1 2 3 4 5]
Elemen dengan indeks ganjil: 2 4
Elemen dengan indeks genap: 1 3 5
Masukkan nilai x untuk menampilkan elemen dengan indeks kelipatan x: 5
Elemen dengan indeks kelipatan 5:
Masukkan indeks yang ingin dihapus: 2
Menghapus elemen pada indeks 2
Isi array: [1 2 4 5]
Rata-rata elemen array: 3.00
Standar deviasi elemen array: 1.58
Masukkan bilangan yang ingin dicari frekuensinya: 4
Frekuensi 4 di dalam array: 1
```

Deskripsi Program

Fungsi Tampilan: tampilSemua: Menampilkan seluruh elemen array.
tampilGanjil: Menampilkan elemen-elemen dengan indeks ganjil.
tampilGenap: Menampilkan elemen-elemen dengan indeks genap.
tampilKelipatan: Menampilkan elemen dengan indeks kelipatan bilangan x.

Unguided 3

Sourcecode

```
package main

import (
    "fmt"
)

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: MU  
Masukkan nama Klub B: Inter  
Pertandingan 1 - Masukkan skor MU: 2 0  
Pertandingan 1 - Masukkan skor Inter: Hasil 1: MU  
Pertandingan 2 - Masukkan skor MU: 1 2  
Pertandingan 2 - Masukkan skor Inter: Hasil 2: Inter  
Pertandingan 3 - Masukkan skor MU: 2 2  
Pertandingan 3 - Masukkan skor Inter: Hasil 3: Draw  
Pertandingan 4 - Masukkan skor MU: 0 1  
Pertandingan 4 - Masukkan skor Inter: Hasil 4: Inter  
Pertandingan 5 - Masukkan skor MU: 3 2  
Pertandingan 5 - Masukkan skor Inter: Hasil 5: MU  
Pertandingan 6 - Masukkan skor MU: 1 0  
Pertandingan 6 - Masukkan skor Inter: Hasil 6: MU  
Pertandingan 7 - Masukkan skor MU: 5 2  
Pertandingan 7 - Masukkan skor Inter: Hasil 7: MU  
Pertandingan 8 - Masukkan skor MU: 2 3  
Pertandingan 8 - Masukkan skor Inter: Hasil 8: Inter  
Pertandingan 9 - Masukkan skor MU: 1 2  
Pertandingan 9 - Masukkan skor Inter: Hasil 9: Inter  
Pertandingan 10 - Masukkan skor MU: -1 2  
Pertandingan 10 - Masukkan skor Inter: Skor tidak valid. Pertandingan selesai.  
  
Daftar klub yang memenangkan pertandingan:  
MU  
Inter  
Inter  
MU  
MU  
MU  
Inter  
Inter
```

Deskripsi Program

Program ini menangani input skor yang valid, menyimpan hanya nama klub pemenang ke dalam array, dan berhenti jika ditemukan skor negatif.

Unguided 4

Sourcecode

```
package main

import (
    "fmt"
)

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)

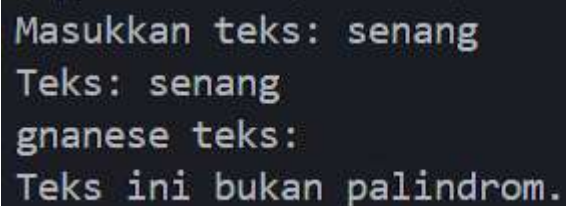
        // Mencetak array asli
        fmt.Print("Teks: ")
        cetakArray(tab, n)

        // 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: senang
Teks: senang
gnanese teks:
Teks ini bukan palindrom.

```

Deskripsi Program

Fungsi palindrom: Memeriksa apakah isi array t membentuk palindrom dengan membandingkan karakter dari awal dan akhir. Fungsi main: Mengelola input dan output, serta memanggil fungsi-fungsi di atas untuk menjalankan keseluruhan logika program.