

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCK & ARRAY**



Disusun Oleh :

Andika Indra Prastawa / 2311102033

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

Struct adalah tipe data komposit yang berguna untuk mengelompokkan beberapa nilai data dari berbagai tipe menjadi satu kesatuan. Ini sangat berguna untuk menunjukkan entitas yang kompleks dengan beberapa sifat. Struct memiliki nama dan tipe data untuk setiap field, sehingga lebih mudah mengorganisasi dan mengelola data yang relevan. Strukturnya sangat membantu dalam pemrograman yang berfokus pada objek karena memungkinkan pengembang membuat objek dengan properti dan metode tertentu, meskipun Go secara tradisional bukanlah bahasa pemrograman yang berfokus pada objek.

Array adalah menggabungkan sekelompok elemen yang memiliki tipe yang sama ke dalam satu variabel. Setiap array memiliki panjang yang tidak dapat diubah yang ditetapkan saat deklarasi. Untuk mengakses elemen dalam array, Anda dapat menggunakan indeks, yang dimulai dengan nol, yang menunjukkan bahwa elemen pertama berada di indeks 0, elemen kedua di indeks 1, dan seterusnya. Akan terjadi kesalahan runtime jika pengguna mencoba mengakses elemen di luar batas yang telah ditentukan.

Ketika kita perlu menyimpan dan mengelola sejumlah data dengan tipe yang sama secara efisien, array sangat bermanfaat. Misalnya, kita dapat menggunakan satu array daripada mendeklarasikan beberapa variabel untuk menyimpan data serupa. Selain itu, array dalam Go mendukung multidimensi, yang memungkinkan pengguna membuat struktur data yang lebih kompleks seperti matriks.

II. GUIDED

1. GUIDED 1

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x int
    y int
}

type Lingkaran struct {
    pusat Titik
    radius int
}

func hitungJarak(a, b Titik) float64 {
    return math.Sqrt(float64((a.x-b.x)*(a.x-b.x) + (a.y-
b.y)*(a.y-b.y)))
}

func titikDiDalamLingkaran(t Titik, l Lingkaran) bool {
    jarak := hitungJarak(t, l.pusat)
    return jarak <= float64(l.radius)
}

func main() {
    var cx1, cy1, r1 int

    fmt.Print("Masukkan koordinat pusat dan radius
lingkaran 1 (cx1 cy1 r1): ")
    fmt.Scanln(&cx1, &cy1, &r1)
    lingkaran1 := Lingkaran{pusat: Titik{x: cx1, y:
cy1}, radius: r1}

    var cx2, cy2, r2 int

    fmt.Print("Masukkan koordinat pusat dan radius
lingkaran 1 (cx1 cy1 r1): ")
    fmt.Scanln(&cx2, &cy2, &r2)
    lingkaran2 := Lingkaran{pusat: Titik{x: cx1, y:
cy1}, radius: r1}

    var x, y int
    fmt.Print("Masukkan koordinat titik sembarang (x y):
")
    fmt.Scanln(&x, &y)
```

```

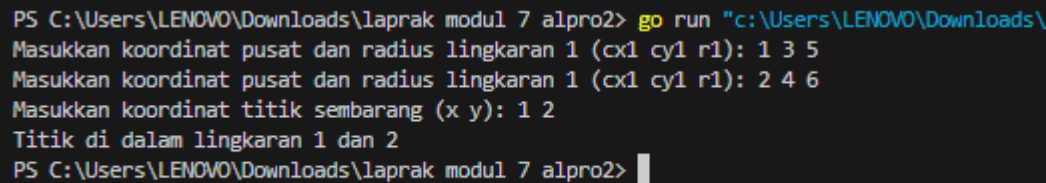
        titik := Titik{x: x, y: y}

        diDalamL1 := titikDiDalamLingkaran(titik,
        lingkaran1)
        diDalamL2 := titikDiDalamLingkaran(titik,
        lingkaran2)

        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 berada di luar kedua
        lingkaran")
        }
    }
}

```

Screenshoot Output



```

PS C:\Users\LENOVO\Downloads\laprak modul 7 alpro2> go run "c:\Users\LENOVO\Downloads\
Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): 1 3 5
Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): 2 4 6
Masukkan koordinat titik sembarang (x y): 1 2
Titik di dalam lingkaran 1 dan 2
PS C:\Users\LENOVO\Downloads\laprak modul 7 alpro2>

```

Deskripsi Program

Program digunakan untuk menentukan apakah sebuah titik berada di dalam satu atau dua lingkaran tertentu. Program ini meminta pengguna untuk memasukkan koordinat pusat dan radius dari dua lingkaran, serta koordinat sebuah titik sembarang. Dengan menggunakan fungsi untuk menghitung jarak antara dua titik dan membandingkannya dengan radius lingkaran, program akan menyiarkan apakah titik tersebut berada di dalam lingkaran pertama, kedua, keduanya, atau di luar keduanya.

III. UNGUIDED

1. Unguided 1

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {

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

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

    for {

        fmt.Println("\nPilih operasi:")
        fmt.Println("a. Menampilkan keseluruhan isi
array")
        fmt.Println("b. Menampilkan elemen-elemen array
dengan indeks ganjil")
        fmt.Println("c. Menampilkan elemen-elemen array
dengan indeks genap")
        fmt.Println("d. Menampilkan elemen array dengan
indeks kelipatan bilangan tertentu")
        fmt.Println("e. Menghapus elemen array pada
indeks tertentu")
        fmt.Println("f. Menampilkan rata-rata dari
bilangan di dalam array")
        fmt.Println("g. Menampilkan standar deviasi dari
bilangan di dalam array")
        fmt.Println("h. Menampilkan frekuensi dari suatu
bilangan tertentu")
        fmt.Println("q. Keluar")

        var pilihan string
        fmt.Print("Masukkan pilihan: ")
        fmt.Scanln(&pilihan)

        switch pilihan {
            case "a":
```

```

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

        case "b":

            fmt.Println("Elemen dengan indeks ganjil:")
            for i := 1; i < len(array); i += 2 {
                fmt.Printf("array[%d] = %d\n", i,
array[i])
            }

            case "c":

                fmt.Println("Elemen dengan indeks genap:")
                for i := 0; i < len(array); i += 2 {
                    fmt.Printf("array[%d] = %d\n", i,
array[i])
                }

                case "d":

                    var x int
                    fmt.Print("Masukkan bilangan x: ")
                    fmt.Scanln(&x)
                    fmt.Printf("Elemen dengan indeks kelipatan
%d:\n", x)
                    for i := 0; i < len(array); i++ {
                        if i%x == 0 {
                            fmt.Printf("array[%d] = %d\n", i,
array[i])
                        }
                    }

                    case "e":

                        var idx int
                        fmt.Print("Masukkan indeks yang akan
dihapus: ")
                        fmt.Scanln(&idx)
                        if idx >= 0 && idx < len(array) {
                            array = append(array[:idx],
array[idx+1:]...)
                            fmt.Println("Array setelah
penghapusan:", array)
                        } else {
                            fmt.Println("Indeks tidak valid!")
                        }

                        case "f":

                            sum := 0
                            for _, val := range array {
                                sum += val

```

```

    }
    average := float64(sum) /
float64(len(array))
    fmt.Printf("Rata-rata: %.2f\n", average)

    case "g":
        sum := 0
        for _, val := range array {
            sum += val
        }
        mean := float64(sum) / float64(len(array))
        var varianceSum float64
        for _, val := range array {
            varianceSum += math.Pow(float64(val) -
mean, 2)
        }
        stdDev := math.Sqrt(varianceSum /
float64(len(array)))
        fmt.Printf("Standar deviasi: %.2f\n",
stdDev)

    case "h":

        var target int
        fmt.Print("Masukkan bilangan yang dicari
frekuensinya: ")
        fmt.Scanln(&target)
        count := 0
        for _, val := range array {
            if val == target {
                count++
            }
        }
        fmt.Printf("Frekuensi bilangan %d: %d
kali\n", target, count)

    case "q":

        fmt.Println("Keluar dari program.")
        return

    default:
        fmt.Println("Pilihan tidak valid! Silakan
coba lagi.")
    }
}

```

Screenshoot Output

Memasukan jumlah elemen array

```
PS C:\Users\LENOVO\Downloads\laprak modul 7 alpro2> go run "c:\Users\LENOVO\Downloads\laprak modul 7 alpro2\main.go"
Masukkan jumlah elemen array (N): 2
Masukkan elemen-elemen array:
Elemen ke-0: 1
Elemen ke-1: 2
```

a. Menampilkan keseluruhan isi array

```
Pilih operasi:
a. Menampilkan keseluruhan isi array
b. Menampilkan elemen-elemen array dengan indeks ganjil
c. Menampilkan elemen-elemen array dengan indeks genap
d. Menampilkan elemen array dengan indeks kelipatan bilangan tertentu
e. Menghapus elemen array pada indeks tertentu
f. Menampilkan rata-rata dari bilangan di dalam array
g. Menampilkan standar deviasi dari bilangan di dalam array
h. Menampilkan frekuensi dari suatu bilangan tertentu
q. Keluar
Masukkan pilihan: a
Isi array: [1 2]
```

b. Menampilkan elemen elemen array dengan indeks ganjil

```
Pilih operasi:
a. Menampilkan keseluruhan isi array
b. Menampilkan elemen-elemen array dengan indeks ganjil
c. Menampilkan elemen-elemen array dengan indeks genap
d. Menampilkan elemen array dengan indeks kelipatan bilangan tertentu
e. Menghapus elemen array pada indeks tertentu
f. Menampilkan rata-rata dari bilangan di dalam array
g. Menampilkan standar deviasi dari bilangan di dalam array
h. Menampilkan frekuensi dari suatu bilangan tertentu
q. Keluar
Masukkan pilihan: b
Elemen dengan indeks ganjil:
array[1] = 2
```

c. Menampilkan elemen-elemen array dengan indeks genap


```
Pilih operasi:
a. Menampilkan keseluruhan isi array
b. Menampilkan elemen-elemen array dengan indeks ganjil
c. Menampilkan elemen-elemen array dengan indeks genap
d. Menampilkan elemen array dengan indeks kelipatan bilangan tertentu
e. Menghapus elemen array pada indeks tertentu
f. Menampilkan rata-rata dari bilangan di dalam array
g. Menampilkan standar deviasi dari bilangan di dalam array
h. Menampilkan frekuensi dari suatu bilangan tertentu
q. Keluar
Masukkan pilihan: c
Elemen dengan indeks genap:
array[0] = 1
```

d. Menampilkan elemen array dengan indeks kelipatan bilangan tertentu

```
Pilih operasi:
a. Menampilkan keseluruhan isi array
b. Menampilkan elemen-elemen array dengan indeks ganjil
c. Menampilkan elemen-elemen array dengan indeks genap
d. Menampilkan elemen array dengan indeks kelipatan bilangan tertentu
e. Menghapus elemen array pada indeks tertentu
f. Menampilkan rata-rata dari bilangan di dalam array
g. Menampilkan standar deviasi dari bilangan di dalam array
h. Menampilkan frekuensi dari suatu bilangan tertentu
q. Keluar
Masukkan pilihan: d
Masukkan bilangan x: 2
Elemen dengan indeks kelipatan 2:
array[0] = 1
```

e. Menghapus elemen array pada indeks tertentu

```
Pilih operasi:
a. Menampilkan keseluruhan isi array
b. Menampilkan elemen-elemen array dengan indeks ganjil
c. Menampilkan elemen-elemen array dengan indeks genap
d. Menampilkan elemen array dengan indeks kelipatan bilangan tertentu
e. Menghapus elemen array pada indeks tertentu
f. Menampilkan rata-rata dari bilangan di dalam array
g. Menampilkan standar deviasi dari bilangan di dalam array
h. Menampilkan frekuensi dari suatu bilangan tertentu
q. Keluar
Masukkan pilihan: e
Masukkan indeks yang akan dihapus: 1
Array setelah penghapusan: [1]
```

f. Menampilkan rata-rata dari bilangan di dalam array

```
Pilih operasi:
a. Menampilkan keseluruhan isi array
b. Menampilkan elemen-elemen array dengan indeks ganjil
c. Menampilkan elemen-elemen array dengan indeks genap
d. Menampilkan elemen array dengan indeks kelipatan bilangan tertentu
e. Menghapus elemen array pada indeks tertentu
f. Menampilkan rata-rata dari bilangan di dalam array
g. Menampilkan standar deviasi dari bilangan di dalam array
h. Menampilkan frekuensi dari suatu bilangan tertentu
q. Keluar
Masukkan pilihan: f
Rata-rata: 1.00
```

g. Menampilkan standar deviasi dari bilangan di dalam array

```
Pilih operasi:
a. Menampilkan keseluruhan isi array
b. Menampilkan elemen-elemen array dengan indeks ganjil
c. Menampilkan elemen-elemen array dengan indeks genap
d. Menampilkan elemen array dengan indeks kelipatan bilangan tertentu
e. Menghapus elemen array pada indeks tertentu
f. Menampilkan rata-rata dari bilangan di dalam array
g. Menampilkan standar deviasi dari bilangan di dalam array
h. Menampilkan frekuensi dari suatu bilangan tertentu
q. Keluar
Masukkan pilihan: g
Standar deviasi: 0.00
```

h. Menampilkan frekuensi dari suatu bilangan tertentu

```
Pilih operasi:
a. Menampilkan keseluruhan isi array
b. Menampilkan elemen-elemen array dengan indeks ganjil
c. Menampilkan elemen-elemen array dengan indeks genap
d. Menampilkan elemen array dengan indeks kelipatan bilangan tertentu
e. Menghapus elemen array pada indeks tertentu
f. Menampilkan rata-rata dari bilangan di dalam array
g. Menampilkan standar deviasi dari bilangan di dalam array
h. Menampilkan frekuensi dari suatu bilangan tertentu
q. Keluar
Masukkan pilihan: h
Masukkan bilangan yang dicari frekuensinya: 1
Frekuensi bilangan 1: 1 kali
```

Deskripsi Program

Program memungkinkan pengguna untuk mengolah data berupa array bilangan bulat melalui berbagai operasi interaktif. Pertama, pengguna diminta memasukkan jumlah elemen array dan nilai-nilai elemen array tersebut. Program kemudian menyediakan menu dengan berbagai fitur,

termasuk menampilkan seluruh isi array, menampilkan elemen dengan indeks ganjil atau genap, menampilkan elemen berdasarkan kelipatan indeks tertentu, menghapus elemen pada indeks tertentu, menghitung rata-rata dan standar deviasi dari elemen array, serta mencari frekuensi kemunculan nilai tertentu.

2. Unguided 2

Sourcecode

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

func main() {
    var clubA, clubB string
    var results []string

    scanner := bufio.NewScanner(os.Stdin)

    fmt.Print("Klub A: ")
    scanner.Scan()
    clubA = scanner.Text()

    fmt.Print("Klub B: ")
    scanner.Scan()
    clubB = scanner.Text()

    matchCount := 1
    for {
        fmt.Printf("Pertandingan %d : ", matchCount)
        scanner.Scan()
        input := scanner.Text()
        scores := strings.Split(input, " ")

        if len(scores) != 2 {
            fmt.Println("Input tidak valid. Harus terdiri dari 2 skor.")
            continue
        }

        scoreA, err1 := strconv.Atoi(scores[0])
        scoreB, err2 := strconv.Atoi(scores[1])
```

```

        if err1 != nil || err2 != nil || scoreA < 0 ||
scoreB < 0 {
            fmt.Println("Skor tidak valid. Program
berhenti.")
            break
        }

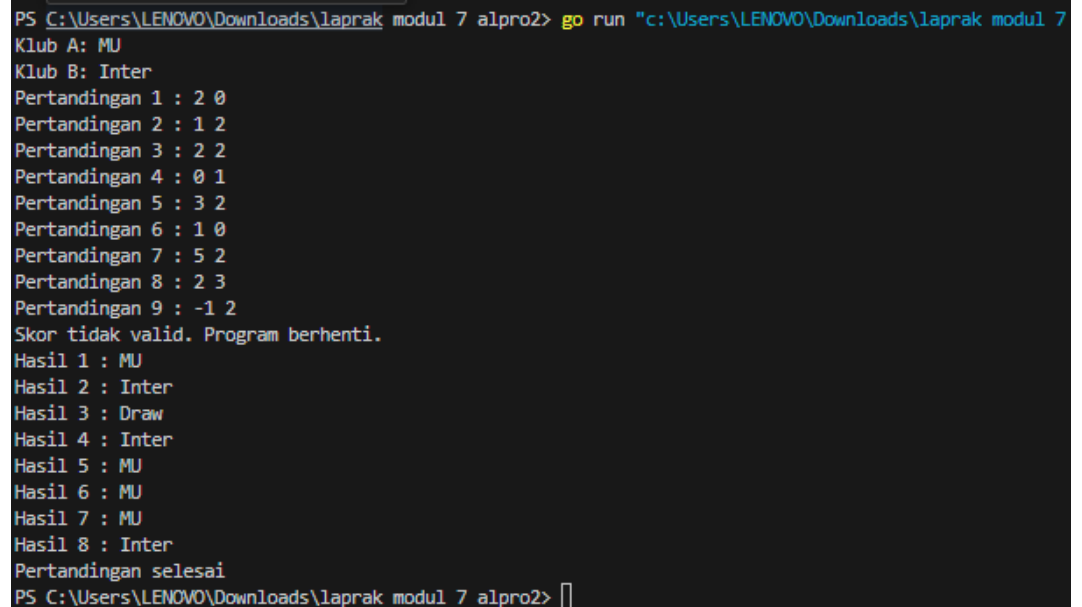
        if scoreA > scoreB {
            results = append(results, clubA)
        } else if scoreA < scoreB {
            results = append(results, clubB)
        } else {
            results = append(results, "Draw")
        }

        matchCount++
    }

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

```

Screenshot Output



```

PS C:\Users\LENOVO\Downloads\laprak modul 7 alpro2> go run "c:\Users\LENOVO\Downloads\laprak modul 7
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
Skor tidak valid. Program berhenti.
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\LENOVO\Downloads\laprak modul 7 alpro2>

```

Deskripsi Program

Program Go ini dirancang untuk mensimulasikan pertandingan antara dua klub sepak bola, clubA dan clubB. Pengguna secara bergantian memasukkan skor setiap pertandingan, dan program akan mencatat hasil setiap pertandingan. Setelah semua pertandingan selesai, program akan menampilkan hasil akhir dari setiap pertandingan.

3. Unguided 3

Sourcecode

```
package main

import (
    "bufio"
    "fmt"
    "os"
)

const NMAX int = 127

type tabel struct {
    tab [NMAX]rune
    m    int
}

func isiArray(t *tabel, n *int) {
    reader := bufio.NewReader(os.Stdin)
    fmt.Println("Masukkan teks (akhiri dengan titik):")

    i := 0
    for i < NMAX {
        r, _, _ := reader.ReadRune()
        if r == '.' {
            break
        }
        t.tab[i] = r
        i++
    }
    t.m = i
    *n = i
}

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-1-i] = t.tab[n-1-i], t.tab[i]
    }
}

func palindrom(t tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t.tab[i] != t.tab[n-1-i] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var m int

    isiArray(&tab, &m)

    fmt.Println("Teks:")
    cetakArray(tab, m)

    balikanArray(&tab, m)
    fmt.Println("Reverse teks:")
    cetakArray(tab, m)

    isPalindrome := palindrom(tab, m)
    fmt.Println("Palindrome?")
    fmt.Println(isPalindrome)
}
```

Screenshot Output

```

PS C:\Users\LENOVO\Downloads\laprak modul 7 alpro2> go run "c:\Users\LENOVO\Downloads\laprak modul 7 alpro2\main.go"
Masukkan teks (akhiri dengan titik):
s e n a n g.
Teks:
s e n a n g
Reverse teks:
g n a n e s
Palindrome?
false
PS C:\Users\LENOVO\Downloads\laprak modul 7 alpro2> go run "c:\Users\LENOVO\Downloads\laprak modul 7 alpro2\main.go"
Masukkan teks (akhiri dengan titik):
k a t a k.
Teks:
k a t a k
Reverse teks:
k a t a k
Palindrome?
true
PS C:\Users\LENOVO\Downloads\laprak modul 7 alpro2> 

```

Deskripsi Program

Program ini memanipulasi dan menganalisis teks yang dimasukkan pengguna hingga karakter titik (.). Teks disimpan dalam sebuah array karakter, kemudian program menyediakan fungsi untuk mencetak teks, membalik urutan teks, dan memeriksa apakah teks tersebut merupakan palindrome (teks yang terbaca sama dari depan dan belakang). Program dimulai dengan membaca input menggunakan fungsi `isiArray`, mencetak teks asli dengan `cetakArray`, membalikkan teks menggunakan `balikanArray`, dan memeriksa apakah teks adalah palindrome dengan fungsi `palindrom`. Hasilnya ditampilkan secara berurutan, termasuk teks asli, teks yang telah dibalik, dan status palindrome (true/false).