

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCK & ARRAY**



Disusun Oleh :

Daffa Aryaputra / 2311102272

IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Struct adalah tipe data yang digunakan untuk mengelompokkan data yang berhubungan dalam satu kesatuan. Struct memungkinkan programmer untuk mendefinisikan sebuah objek dengan berbagai atribut atau properti. Setiap properti dalam struct memiliki tipe data tertentu, sehingga struct dapat digunakan untuk mewakili data yang lebih kompleks dibandingkan tipe data primitif seperti integer atau string. Struct biasanya digunakan dalam pemrograman berorientasi objek (OOP) meskipun Go tidak sepenuhnya mengikuti paradigma OOP. Dengan struct, programmer dapat mendefinisikan tipe data kustom yang mencerminkan entitas dunia nyata, seperti pengguna, produk, atau transaksi.

Sementara itu, array di Go adalah kumpulan elemen yang memiliki tipe data yang sama dan ukuran tetap. Array digunakan untuk menyimpan beberapa nilai dalam satu variabel yang diakses menggunakan indeks. Dalam Go, array memiliki sifat statis, artinya ukuran array harus ditentukan saat deklarasi dan tidak dapat diubah. Hal ini berbeda dengan slice, yang merupakan versi lebih fleksibel dari array. Array berguna untuk menyimpan data berurutan seperti daftar angka atau kumpulan string. Dalam penggunaannya, array sering dikombinasikan dengan struktur data lain seperti struct untuk membangun sistem yang lebih kompleks dan efisien.

II. 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)

```

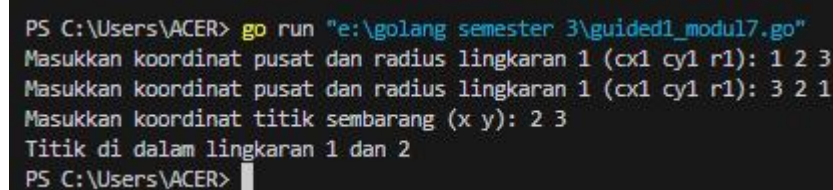
```

        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\ACER> go run "e:\golang semester 3\guided1_modul7.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): 1 2 3
Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): 3 2 1
Masukkan koordinat titik sembarang (x y): 2 3
Titik di dalam lingkaran 1 dan 2
PS C:\Users\ACER>

```

Deskripsi Program

Program di atas adalah untuk menentukan apakah sebuah titik tertentu berada di dalam salah satu atau kedua lingkaran, atau di luar keduanya. Program ini meminta input dari pengguna berupa koordinat pusat dan radius dua lingkaran, serta koordinat titik yang akan diuji. Dengan memanfaatkan rumus jarak Euclidean, program menghitung jarak antara titik tersebut dengan pusat masing-masing lingkaran, lalu membandingkan jarak tersebut dengan radius lingkaran untuk menentukan apakah titik berada di dalam lingkaran atau tidak.

Cara kerja program diawali dengan membaca data masukan berupa koordinat pusat dan radius untuk dua lingkaran. Setelah itu, program membaca koordinat dari titik sembarang yang ingin diuji. Program menggunakan fungsi `hitungJarak` untuk menghitung jarak antara titik tersebut dan pusat lingkaran dengan rumus jarak. Fungsi `titikDiDalamLingkaran` kemudian memeriksa apakah jarak tersebut kurang dari atau sama dengan radius lingkaran. Berdasarkan hasil pemeriksaan ini, program mencetak salah satu dari empat kemungkinan: titik berada di dalam lingkaran pertama, lingkaran kedua, kedua lingkaran, atau di luar keduanya.

III. UNGUIDED

1.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n int

    fmt.Println("Masukkan jumlah elemen array:")
    fmt.Scan(&n)

    // Input array
    array := make([]int, n)

    fmt.Println("Masukkan elemen array:")
    for i := 0; i < n; i++ {
        fmt.Scan(&array[i])
    }

    for {
        fmt.Println("\nPilih operasi:")
```

```

fmt.Println("a. Tampilkan keseluruhan isi array")
fmt.Println("b. Tampilkan elemen dengan indeks ganjil")
fmt.Println("c. Tampilkan elemen dengan indeks genap")
fmt.Println("d. Tampilkan elemen dengan indeks kelipatan
x")

fmt.Println("e. Hapus elemen array pada indeks tertentu")
fmt.Println("f. Tampilkan rata-rata array")
fmt.Println("g. Tampilkan standar deviasi array")
fmt.Println("h. Tampilkan frekuensi bilangan tertentu")
fmt.Println("i. Keluar")

var choice string
fmt.Scan(&choice)

switch choice {
case "a":
    fmt.Println("Isi array:", array)

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

```

```

        case "c":

            fmt.Print("Elemen dengan indeks genap: ")

            for i := 0; i < len(array); i += 2 {

                fmt.Print(array[i], " ")

            }

            fmt.Println()

        case "d":

            var x int

            fmt.Println("Masukkan nilai x:")

            fmt.Scan(&x)

            fmt.Print("Elemen dengan indeks kelipatan ", x, ":")

            for i := 0; i < len(array); i++ {

                if i%x == 0 {

                    fmt.Print(array[i], " ")

                }

            }

            fmt.Println()

        case "e":

            var index int

            fmt.Println("Masukkan indeks elemen yang ingin")
            fmt.Scan(&index)
            dihapus:")

```

```

        if index >= 0 && index < len(array) {
            array = append(array[:index],
array[index+1:]...)
            fmt.Println("Array setelah penghapusan:",
array)
        } else {
            fmt.Println("Indeks tidak valid.")
        }

    case "f":
        sum := 0
        for _, val := range array {
            sum += val
        }
        rataRata := float64(sum) / float64(len(array))
        fmt.Println("Rata-rata:", rataRata)

    case "g":
        sum := 0
        for _, val := range array {
            sum += val
        }
        rataRata := float64(sum) / float64(len(array))
        var variance float64
        for _, val := range array {

```



```

                variance += math.Pow(float64(val)-rataRata,
2)
            }

            stdDeviasi := math.Sqrt(variance /
float64(len(array)))

            fmt.Println("Standar deviasi:", stdDeviasi)

        case "h":

            var target int

            fmt.Println("Masukkan bilangan yang ingin dicari
frekuensinya:")

            fmt.Scan(&target)

            frequency := 0

            for _, val := range array {

                if val == target {

                    frequency++

                }

            }

            fmt.Println("Frekuensi bilangan", target, ":",
frequency)

        case "i":

            fmt.Println("Program selesai.")

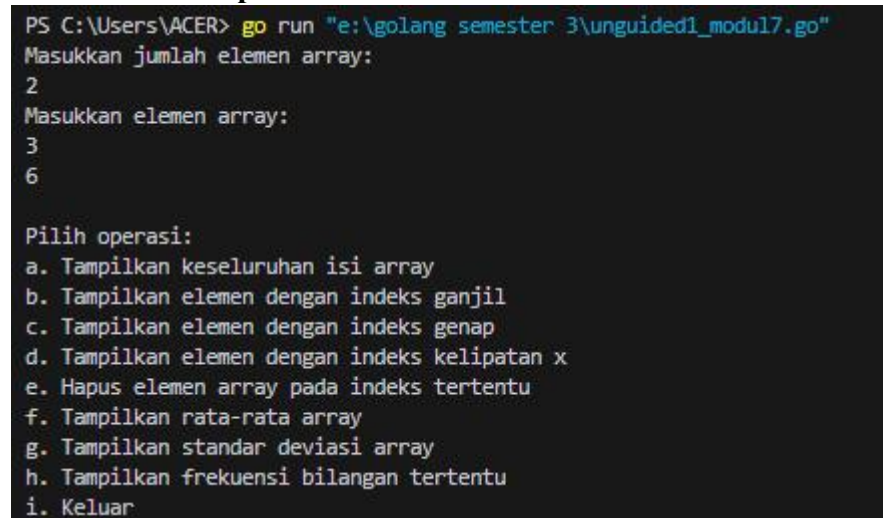
            return

        default:

```

```
        fmt.Println("Pilihan tidak valid.")
    }
}
}
```

Screenshoot Output



```
PS C:\Users\ACER> go run "e:\golang semester 3\unguided1_modul7.go"
Masukkan jumlah elemen array:
2
Masukkan elemen array:
3
6

Pilih operasi:
a. Tampilkan keseluruhan isi array
b. Tampilkan elemen dengan indeks ganjil
c. Tampilkan elemen dengan indeks genap
d. Tampilkan elemen dengan indeks kelipatan x
e. Hapus elemen array pada indeks tertentu
f. Tampilkan rata-rata array
g. Tampilkan standar deviasi array
h. Tampilkan frekuensi bilangan tertentu
i. Keluar
```

Deskripsi Program

Program di atas adalah aplikasi berbasis konsol yang memungkinkan pengguna untuk melakukan berbagai operasi pada sebuah array integer. Program diawali dengan meminta pengguna memasukkan jumlah elemen array dan elemen-elemen array tersebut. Setelah array dibuat, pengguna dapat memilih beberapa operasi seperti menampilkan isi array, menampilkan elemen berdasarkan indeks tertentu (ganjil, genap, atau kelipatan suatu bilangan), menghitung rata-rata, menghitung standar deviasi, mencari frekuensi kemunculan sebuah bilangan, atau menghapus elemen berdasarkan indeks. Program ini dirancang untuk terus berjalan hingga pengguna memilih opsi untuk keluar.

Cara kerja program dimulai dengan pembacaan input array. Setelah itu, program memasuki sebuah loop di mana setiap iterasi menawarkan berbagai pilihan operasi kepada pengguna. Misalnya, untuk menampilkan elemen dengan indeks ganjil atau genap, program menggunakan loop untuk memeriksa indeks elemen dan mencetaknya jika memenuhi kriteria. Untuk menghitung rata-rata, program menjumlahkan semua elemen array lalu membaginya dengan jumlah elemen. Standar deviasi dihitung dengan

rumus matematika menggunakan rata-rata dan selisih kuadrat rata-rata. Jika pengguna ingin menghapus elemen, program memanfaatkan fungsi `append` untuk merekonstruksi array tanpa elemen pada indeks yang dihapus. Semua hasil operasi akan dicetak ke konsol sesuai dengan pilihan pengguna.

2.

Sourcecode

```
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 pemenang []string
    var pertandingan int = 1

    for {
        var skorA, skorB int
        fmt.Printf("Pertandingan %d (Skor %s vs %s): ",
pertandingan, klubA, klubB)
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            break
        }

        if skorA > skorB {
            pemenang = append(pemenang, klubA)
            fmt.Printf("Hasil %d: %s\n", pertandingan, klubA)
        } else if skorA < skorB {
            pemenang = append(pemenang, klubB)
            fmt.Printf("Hasil %d: %s\n", pertandingan, klubB)
        } else {
            pemenang = append(pemenang, "Draw")
            fmt.Printf("Hasil %d: Draw\n", pertandingan)
        }
    }
}
```

```
        pertandingan++
    }

    fmt.Println("\nDaftar klub yang memenangkan pertandingan:")
    for i, hasil := range pemenang {
        fmt.Printf("Pertandingan %d: %s\n", i+1, hasil)
    }

    fmt.Println("Pertandingan selesai.")
}
```

Screenshoot Output

```
PS C:\Users\ACER> go run "e:\golang semester 3\unguided2_modul7.go"
Klub A: MU
Klub B: Inter
Pertandingan 1 (Skor MU vs Inter): 2 0
Hasil 1: MU
Pertandingan 2 (Skor MU vs Inter): 1 2
Hasil 2: Inter
Pertandingan 3 (Skor MU vs Inter): 2 2
Hasil 3: Draw
Pertandingan 4 (Skor MU vs Inter): 0 1
Hasil 4: Inter
Pertandingan 5 (Skor MU vs Inter): 3 2
Hasil 5: MU
Pertandingan 6 (Skor MU vs Inter): 1 0
Hasil 6: MU
Pertandingan 7 (Skor MU vs Inter): 5 2
Hasil 7: MU
Pertandingan 8 (Skor MU vs Inter): 2 3
Hasil 8: Inter
Pertandingan 9 (Skor MU vs Inter): -1 2

Daftar klub yang memenangkan pertandingan:
Pertandingan 1: MU
Pertandingan 2: Inter
Pertandingan 3: Draw
Pertandingan 4: Inter
Pertandingan 5: MU
Pertandingan 6: MU
Pertandingan 7: MU
Pertandingan 8: Inter
Pertandingan selesai.
PS C:\Users\ACER>
```

Deskripsi Program

Program ini bertujuan untuk mencatat hasil pertandingan antara dua klub sepak bola yang dimasukkan oleh pengguna. Pertama, program meminta nama kedua klub yang akan bertanding, kemudian masuk ke proses utama berupa input skor untuk setiap pertandingan. Pengguna dapat memasukkan skor kedua klub untuk setiap pertandingan, dan program akan menentukan pemenangnya berdasarkan skor yang diberikan. Jika salah satu skor negatif, program menghentikan input dan menampilkan hasil rekapitulasi dari seluruh pertandingan.

Algoritma program bekerja dengan membaca skor setiap pertandingan dalam sebuah loop, kemudian membandingkan skor kedua tim. Jika skor klub pertama lebih besar, maka klub pertama dinyatakan menang, dan sebaliknya untuk klub kedua. Jika skor kedua klub sama, hasilnya adalah "Draw". Semua hasil pertandingan disimpan dalam array pemenang, yang

kemudian ditampilkan ke layar secara berurutan setelah proses input selesai. Program ini memberikan pengguna gambaran jelas tentang hasil setiap pertandingan dan daftar pemenang pertandingan hingga proses input dihentikan. Output mencakup pemenang untuk setiap pertandingan, serta konfirmasi bahwa pertandingan selesai.

3.

Sourcecode

```
package main

import (
    "fmt"
)

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var input string
    fmt.Println("Masukkan teks (akhiri dengan titik '.'): ")
    fmt.Scanln(&input)

    *n = 0
    for _, char := range input {
        if char == '.' {
            break
        }
        t[*n] = char
        *n++
    }
}

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

func balikkanArray(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 n int

    isiArray(&tab, &n)

    fmt.Print("Teks: ")
    cetakArray(tab, n)

    balikkanArray(&tab, n)

    fmt.Print("Reverse teks: ")
    cetakArray(tab, n)

    isPalindrom := palindrom(tab, n)
    fmt.Print("Palindrom: ")
    fmt.Println(isPalindrom)
}

```

Screenshoot Output

```
PS C:\Users\ACER> go run "e:\golang semester 3\unguided3_modul7.go"
Masukkan teks (akhiri dengan titik '.'):
katak
Teks: katak
Reverse teks: katak
Palindrom: true
PS C:\Users\ACER> go run "e:\golang semester 3\unguided3_modul7.go"
Masukkan teks (akhiri dengan titik '.'):
senang
Teks: senang
Reverse teks: ganes
Palindrom: false
PS C:\Users\ACER> |
```

Deskripsi Program

Program di atas adalah implementasi algoritma untuk memproses string yang dimasukkan oleh pengguna, membalikkan string tersebut, dan memeriksa apakah string tersebut merupakan palindrom. Sebuah palindrom adalah teks yang dapat dibaca sama baik dari depan maupun dari belakang, seperti "KATAK". Program ini bekerja dengan mendefinisikan array untuk menampung karakter string, kemudian menyediakan fungsi-fungsi untuk mengisi array, mencetak array, membalikkan isi array, dan mengecek apakah array tersebut membentuk palindrom.

Cara kerja program dimulai dengan pengguna memasukkan string yang diakhiri oleh titik (.). Fungsi isiArray akan membaca input dan menyimpan setiap karakter ke dalam array hingga menemukan titik, kemudian menghitung panjang string. Setelah itu, fungsi cetakArray menampilkan string asli. Kemudian, fungsi balikkanArray membalik urutan karakter dalam array. String yang telah dibalikkan ini juga ditampilkan dengan fungsi cetakArray. Akhirnya, program memanggil fungsi palindrom untuk memeriksa apakah string tersebut sama dengan kebalikannya. Hasilnya, program akan mencetak status "true" jika string adalah palindrom, atau "false" jika bukan. Output program memberikan informasi lengkap mulai dari string asli, string yang dibalik, hingga hasil pengecekan palindrom.