

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCK & ARRAY**



Disusun Oleh :

Marsep Trianto Pakondo / 2311102251

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

Struct adalah tipe data yang digunakan untuk mengelompokkan beberapa nilai dengan tipe data yang berbeda dalam satu unit logis. Struct sangat berguna dalam Golang karena memungkinkan pengorganisasian data yang lebih kompleks dibandingkan tipe data primitif.

Karakteristik Struct dalam Golang:

1. Deklarasi: Struct didefinisikan menggunakan kata kunci `type` diikuti oleh nama struct dan blok berisi field yang merepresentasikan data.
2. Field: Field adalah elemen-elemen di dalam struct yang memiliki tipe data tertentu.
3. Pendekatan OOP: Struct sering digunakan sebagai pengganti kelas dalam paradigma Object-Oriented Programming (OOP) di Golang.
4. Inisialisasi: Struct dapat diinisialisasi dengan dua cara:
 - Secara langsung (literal struct)
 - Dengan pendekatan field-by-field

• Array

Array adalah koleksi elemen-elemen yang memiliki tipe data yang sama dan disimpan di lokasi memori yang berurutan. Array di Golang bersifat statis, artinya ukuran array ditentukan saat deklarasi dan tidak dapat diubah.

Karakteristik Array dalam Golang:

1. Deklarasi: Array didefinisikan dengan menentukan panjangnya dan tipe datanya.
2. Indeks: Elemen dalam array diakses menggunakan indeks, dimulai dari 0.
3. Immutable Size: Ukuran array tidak dapat berubah setelah dideklarasikan.
4. Operasi: Operasi array mencakup inisialisasi, pembaruan nilai elemen, iterasi menggunakan loop, dan pencarian elemen.

II. GUIDED

Soal Studi Case

XXXXXXXXXXXXXXXXXXXX

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 raduis
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 raduis
lingkaran 1 (cx2, cy2, r2): ")
        fmt.Scanln(&cx2, &cy2, &r2)

        lingkaran2 := lingkaran{pusat: titik{x: cx2, y: cy2},
radius: r2}

        var x, y int

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

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

        diDalam1 := titikDidalamLingkaran(Titik, lingkaran1)
        diDalam2 := titikDidalamLingkaran(Titik, lingkaran2)

        if diDalam1 && diDalam2 {
            fmt.Println("Titik didalam lingkaran 1 dan 2")
        } else if diDalam1 {
            fmt.Println("Titik didalam lingkaran 1")
        } else if diDalam2 {
            fmt.Println("Titik didalam lingkaran 2")
        } else {
            fmt.Println("Titik berada di luar kedua
lingkaran")
        }
    }
}

```

Screenshoot Output

```
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul7\guided\guided1.go"
Masukkan koordinat pusat dan raduis lingkaran 1 (cx1, cy1, r1): 1 1 5
Masukkan koordinat pusat dan raduis lingkaran 1 (cx2, cy2, r2): 8 8 4
Masukkan koordinat titik sembarang (x, y): 2 2
Titik didalam lingkaran 1
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul7\guided\guided1.go"
Masukkan koordinat pusat dan raduis lingkaran 1 (cx1, cy1, r1): 1 2 3
Masukkan koordinat pusat dan raduis lingkaran 1 (cx2, cy2, r2): 4 5 6
Masukkan koordinat titik sembarang (x, y): 7 8
Titik didalam lingkaran 2
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul7\guided\guided1.go"
Masukkan koordinat pusat dan raduis lingkaran 1 (cx1, cy1, r1): 1 1 5
Masukkan koordinat pusat dan raduis lingkaran 1 (cx2, cy2, r2): 8 8 4
Masukkan koordinat titik sembarang (x, y): 15 20
Titik berada di luar kedua lingkaran
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul7\guided\guided1.go"
Masukkan koordinat pusat dan raduis lingkaran 1 (cx1, cy1, r1): 5 10 15
Masukkan koordinat pusat dan raduis lingkaran 1 (cx2, cy2, r2): -15 4 20
Masukkan koordinat titik sembarang (x, y): 0 0
Titik didalam lingkaran 1 dan 2
PS E:\alpro 2\src>
```

Deskripsi Program

Program diatas digunakan untuk menentukan apakah sebuah titik berada di dalam satu, dua, atau di luar dua lingkaran yang diberikan. Program ini pertama-tama meminta pengguna untuk memasukkan koordinat pusat dan jari-jari dari dua lingkaran, serta koordinat sebuah titik. Kemudian, program menghitung jarak antara titik tersebut dengan pusat masing-masing lingkaran. Jika jarak antara titik dengan pusat lingkaran lebih kecil atau sama dengan jari-jari lingkaran, maka titik tersebut berada di dalam lingkaran. Program akan mencetak pesan sesuai dengan hasil perhitungan, yaitu apakah titik tersebut berada di dalam lingkaran pertama, kedua, keduanya, atau di luar kedua lingkaran.

III. UNGUIDED

Soal Studi Case

XXXXXXXXXXXXXXXXXXXX

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func isianArray(bilBul[]int, n int) {
    fmt.Println("\nIsian dari array : \n")
    for i := 0; i < n; i++ {
        fmt.Printf("Nilai indeks ke-%v : %v\n", i,
bilBul[i])
    }
}

func nilaiIndeksGanjil(bilBul[]int, n int) {
    fmt.Printf("\nNilai indeks Ganjil : \n")

    for i := 0; i < n; i++ {
        if i % 2 == 1 {
            fmt.Printf("Indeks ke-%v. %v\n", i,
bilBul[i])
        }
    }
}

func nilaiIndeksGenap(bilBul[]int, n int) {
    fmt.Printf("\nNilai indeks Genap : \n")

    for i := 0; i < n; i++ {
        if i % 2 == 0 {
            fmt.Printf("Indeks ke-%v. %v\n", i,
bilBul[i])
        }
    }
}
```

```

    }

    func nilaiKelipatanBilangan(bilBul[]int, n int) {
        var kelipatan int

        fmt.Print("\nMasukkan kelipatan berapa : ")
        fmt.Scan(&kelipatan)

        fmt.Printf("\nNilai kelipatan %v :\n", kelipatan)

        for i := 0; i < n; i++ {
            if bilBul[i] % kelipatan == 0 {
                fmt.Printf("%v. %v\n", i+1, bilBul[i])
            }
        }
    }

    func haspusElemenTertentu(bilBul[]int, n int) {
        var elemen int

        fmt.Print("\nHapus elemen indeks ke berapa : ")
        fmt.Scan(&elemen)

        fmt.Printf("\nNilai elemen %v :\n", elemen)

        for i := 0; i < n; i++ {
            if i == elemen {
                bilBul[i] = ''
            }
        }
        for i := 0; i < n; i++ {
            if bilBul[i] == '' {
                continue
            }
            fmt.Printf("%v. %v\n", i+1, bilBul[i])
        }
    }

    func rataRata(bilBul[]int, n int) float64 {
        jumlah := 0
        bagi := 0
        for i := 0; i < n; i++ {
            if bilBul[i] == '' {

```

```

        continue
    }
    jumlah += bilBul[i]
    bagi += 1
}

return float64(jumlah) / float64(bagi)
}

func standarDevasi(bilBul[]int, n int) float64 {
    rata := rataRata(bilBul[:], n)
    jumlah := 0.0
    bagi := 0

    for i := 0; i < n; i++ {
        if bilBul[i] == '' {
            continue
        }
        jumlah += math.Pow(float64(bilBul[i])-rata, 2)
        bagi += 1
    }
    return math.Sqrt(jumlah / float64(bagi))
}

func frekuensi(bilBul[]int, n, cari int) int {
    var count int

    for i := 0; i < n; i++ {
        if bilBul[i] == cari {
            count++
        }
    }

    return count
}

func main() {
    var n, cari int
    var bilBul[20]int // Asumsi panjang array 20

    fmt.Print("Masukkan panjang array : ")
    fmt.Scan(&n)

```



```

        for i := 0; i < n; i++ {
            fmt.Printf("Masukkan bilangan bulat indeks ke-
%v : ", i)
            fmt.Scan(&bilBul[i])
        }

// Isi Data Array
isianArray(bilBul[:], n)

// Isi Nilai Indeks Ganjil
nilaiIndeksGanjil(bilBul[:], n)

// Isi Nilai Indeks Genap
nilaiIndeksGenap(bilBul[:], n)

// Mencari Nilai Kelipatan X
nilaiKelipatanBilangan(bilBul[:], n)

// Hapus Element di Indeks Tertentu
haspusElemenTertentu(bilBul[:], n)

// Rata-rata
fmt.Print("\nRata-ratanya adalah ", rataRata(bilBul[:], n))

// Standar Deviasi
fmt.Print("\n\nStandar deviasinya adalah ",
standarDeviasi(bilBul[:], n))

// Frekuensi
fmt.Print("\n\nMasukkan bilangan yang ingin dicari
frekuensinya : ")
fmt.Scan(&cari)
fmt.Printf("\nFrekuensi %v muncul %v kali", cari,
frekuensi(bilBul[:], n, cari))
}

```

Screenshoot Output

```

PS E:\alpro 2\src> go run "e:\alpro 2\src\modul7\unguided\unguided1.go"
Masukkan panjang array : 7
Masukkan bilangan bulat indeks ke-0 : 5
Masukkan bilangan bulat indeks ke-1 : 15
Masukkan bilangan bulat indeks ke-2 : 25
Masukkan bilangan bulat indeks ke-3 : 35
Masukkan bilangan bulat indeks ke-4 : 45
Masukkan bilangan bulat indeks ke-5 : 55
Masukkan bilangan bulat indeks ke-6 : 65

Isian dari array :
Nilai indeks ke-0 : 5
Nilai indeks ke-1 : 15
Nilai indeks ke-2 : 25
Nilai indeks ke-3 : 35
Nilai indeks ke-4 : 45
Nilai indeks ke-5 : 55
Nilai indeks ke-6 : 65

Nilai indeks Ganjil :
Indeks ke-1. 15
Indeks ke-3. 35
Indeks ke-5. 55

Nilai indeks Genap :
Indeks ke-0. 5
Indeks ke-2. 25
Indeks ke-4. 45
Indeks ke-6. 65

Masukkan kelipatan berapa : 3

Nilai kelipatan 3 :
2. 15
5. 45

Hapus elemen indeks ke berapa : 4

Nilai elemen 4 :
1. 5
2. 15
3. 25
4. 35
6. 55
7. 65

Rata-ratanya adalah 33.333333333333336

Standar deviasinya adalah 21.147629234082533

Masukkan bilangan yang ingin dicari frekuensinya : 25

Frekuensi 25 muncul 1 kali
PS E:\alpro 2\src>

```

Deskripsi Program

Program diatas digunakan untuk mengolah data berupa array bilangan bulat dan menyediakan berbagai fungsi manipulasi, seperti menampilkan isi array, nilai pada indeks ganjil dan genap, mencari nilai kelipatan tertentu, menghapus elemen berdasarkan indeks, menghitung rata-rata dan standar deviasi, serta menentukan frekuensi kemunculan suatu nilai. Pertama pengguna disuruh memasukkan panjang array dan nilai-nilai elemennya. Kemudian menjalankan fungsi-fungsi seperti mencetak elemen array (isianArray), menampilkan elemen pada indeks ganjil dan genap (nilaiIndeksGanjil dan nilaiIndeksGenap), mencari kelipatan bilangan tertentu (nilaiKelipatanBilangan), menghapus elemen pada indeks tertentu (hapusElemenTertentu), menghitung rata-rata (rataRata), menghitung standar deviasi (standarDeviasi), serta menghitung frekuensi kemunculan nilai tertentu (frekuensi). Setiap hasil dari operasi ini akan ditampilkan ke layer.

Sourcecode

```
package main
import "fmt"

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

    fmt.Print("Masukkan Klub A : ")
    fmt.Scan(&klubA)

    fmt.Print("Masukkan Klub B : ")
    fmt.Scan(&klubB)

    i := 1
    for {
        fmt.Printf("Pertandingan %v :", i)
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            break
        } else if skorA > skorB {
            hasil = append(hasil, klubA)
        } else if skorB > skorA {
            hasil = append(hasil, klubB)
        } else {
            hasil = append(hasil, "Draw")
        }
        i++
    }

    for i := 0; i < len(hasil); i++ {
        fmt.Println("Hasil ", i+1, " : ", hasil[i])
    }
}
```

Screenshot Output

```

PS E:\alpro 2\src> go run "e:\alpro 2\src\modul7\unguided\unguided2.go"
Masukkan Klub A : MU
Masukkan 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
Hasil 1 : MU
Hasil 2 : Inter
Hasil 3 : Draw
Hasil 4 : Inter
Hasil 5 : MU
Hasil 6 : MU
Hasil 7 : MU
Hasil 8 : Inter
PS E:\alpro 2\src>

```

Deskripsi Program

Program diatas digunakan untuk mensimulasikan sebuah turnamen sederhana antara dua klub, yaitu Klub A dan Klub B. Program ini akan terus meminta input skor untuk setiap pertandingan hingga pengguna memasukkan skor negatif. Setelah itu, program akan mencetak hasil dari setiap pertandingan, yaitu klub mana yang menang atau apakah pertandingan berakhir imbang. Program ini menggunakan struktur perulangan for untuk terus meminta input skor, dan menggunakan slice (array yang dinamis) untuk menyimpan hasil setiap pertandingan sehingga program dapat mencatat dan menampilkan hasil dari beberapa pertandingan secara berurutan.

Sourcecode

```

package main

import "fmt"

const NMAX int = 127
type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var kar rune
    fmt.Print("Masukkan Teks : ")
    for i := 0; i < NMAX; i++ {
        fmt.Scanf("%c", &kar)
        if kar == '.' {
            break
        }
        t[i] = kar
        *n++
    }
}

```

```

    }
}

func cetakArray(t tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Printf("%c ", t[i])
    }
    fmt.Print("\n")
}

func balikkanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        temp := t[i]
        t[i] = t[n-1-i]
        t[n-1-i] = temp
    }
    for i := 0; i < n; i++ {
        fmt.Printf("%c ", t[i])
    }
    fmt.Print("\n")
}

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 m int

    isiArray(&tab, &m)

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

    fmt.Print("Reverse teks: ")
    balikkanArray(&tab, m)
}

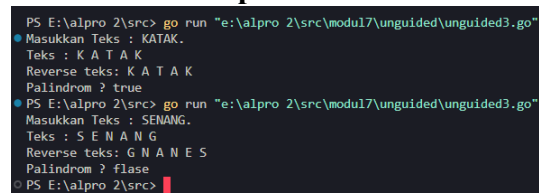
```

```

        if palindrom(tab, m) {
            fmt.Println("Palindrom ? true")
        } else {
            fmt.Println("Palindrom ? flase")
        }
    }
}

```

Screenshoot Output



```

PS E:\alpro 2\src> go run "e:\alpro 2\src\modul7\unguided\unguided3.go"
Masukkan Teks : KATAK.
Teks : K A T A K
Reverse teks: K A T A K
Palindrom ? true
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul7\unguided\unguided3.go"
Masukkan Teks : SENANG.
Teks : S E N A N G
Reverse teks: G N A N E S
Palindrom ? flase
PS E:\alpro 2\src>

```

Deskripsi Program

Program diatas digunakan untuk memproses teks yang diinputkan oleh pengguna yaitu membalikan sebuah kata dan apakah kata tersebut merupakan polindrom atau bukan. Fungsi *isiArray* berfungsi untuk memasukkan teks satu karakter dalam satu waktu hingga menekan tombol titik (.), dengan setiap karakter disimpan dalam array *tab*. Fungsi *cetakArray* untuk mencetak isi array karakter demi karakter. Fungsi *balikkanArray* untuk membalik urutan karakter dalam array dan menampilkan hasilnya. Fungsi *palindrom* digunakan untuk memeriksa apakah teks dalam array *tab* merupakan palindrom, yaitu teks yang terbaca sama dari depan maupun belakang. Program utama (*main*) mengoordinasikan semua fungsi.