LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL VII

STRUCK & ARRAY

Disusun Oleh :

Ariiq Radhitya Pradana 2311102260

IF 11 06

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dalam bahasa pemrograman Go, array adalah tipe data yang digunakan untuk menyimpan sekumpulan elemen dengan tipe yang sama dalam satu variabel. Setiap array memiliki panjang yang tetap, yang ditentukan saat deklarasi, dan tidak dapat diubah setelahnya. Elemen dalam array diakses menggunakan indeks, yang dimulai dari 0. Ini berarti bahwa elemen pertama berada di indeks 0, elemen kedua di indeks 1, dan seterusnya. Jika pengguna mencoba mengakses elemen di luar batas yang telah ditentukan, akan terjadi kesalahan runtime.

Array sangat berguna ketika kita perlu menyimpan dan mengelola sejumlah data dengan tipe yang sama secara efisien. Misalnya, alih-alih mendeklarasikan beberapa variabel untuk menyimpan data yang serupa, kita dapat menggunakan satu array untuk menyimpan semua data tersebut. Selain itu, array di Go juga mendukung multidimensionality, memungkinkan pengguna untuk membuat struktur data yang lebih kompleks seperti matriks.

Sementara itu, **struck** adalah tipe data yang digunakan untuk mengelompokkan beberapa nilai dengan tipe yang berbeda menjadi satu kesatuan yang lebih kompleks. Struct memungkinkan pengembang untuk mendefinisikan tipe data baru yang mencakup berbagai atribut (field) yang relevan. Setiap field dalam struct memiliki nama dan tipe data tertentu, sehingga memudahkan pengorganisasian dan pengelolaan data yang berkaitan. Struct sangat berguna dalam pemrograman berorientasi objek karena memungkinkan pengembang untuk membuat objek dengan properti dan metode tertentu, meskipun Go bukanlah bahasa pemrograman berorientasi objek secara tradisional.

Secara keseluruhan, baik array maupun struck merupakan komponen penting dalam pengelolaan data di Go, masing-masing dengan karakteristik dan kegunaan spesifik yang mendukung berbagai kebutuhan pemrograman.

II. GUIDED

Guided 1

```
package main

import (
    "fmt"
    "math"
)

// Struktur untuk menyimpan titik dengan koordinat (x,y)
type Titik struct {
    x int
    y int
}

// Struktur untuk menyimpan lingkaran dengan pusat dan
radius
type Lingkaran struct {
    pusat  Titik
    radius int
}

// Fungi untuk menghitung jarak antara dua titik
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)))
}

// Fungsi untuk memeriksa apakah titik berada di dalam
lingkaran
func titikDidalamLingkaran(t Titik, l Lingkaran) bool {
    jarak := hitungjarak(t, l.pusat)
    return jarak <= float64(l.radius)
}

func main() {
    //Input untuk lingkaran 1
    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}

    //Input untuk lingkaran 2
    var cx2, cy2, r2 int
    fmt.Print("Masukkan koordinat pusat dan radius
lingkaran 2 (cx2 cy2 r2): ")
    fmt.Scanln(&cx1, &cy1, &r1)
```

```
    Lingkaran2 := Lingkaran{pusat: Titik{x: cx2, y:
cy2}, radius: r2}

    //Input untuk titik sembarang
    var x, y int
    fmt.Print("Masukkan koordinat titik sembarang (x y):
")
    fmt.Scanln(&x, &y)
    titk := Titik{x: x, y: y}

    //Pengecekan posisi titik
    diDalam1 := titikDidalamLingkaran(titk, Lingkaran1)
    diDalam2 := titikDidalamLingkaran(titk, Lingkaran2)

    // Menampilkan hasil sesuai kondisi
    if diDalam1 && diDalam2 {
        fmt.Print("Titik di dalam lingkaran 1 dan 2")
    } else if diDalam1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diDalam2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar kedua lingkaran")
    }
}
```

Screenshoot Output

```
Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): 1
Masukkan koordinat pusat dan radius lingkaran 2 (cx2 cy2 r2): 2
Masukkan koordinat titik sembarang (x y): 3
Titik di luar kedua lingkaran
```

Deskripsi Program

Kode di atas adalah sebuah program dalam bahasa Go yang digunakan untuk memeriksa posisi sebuah titik sembarang terhadap dua lingkaran. Program ini menggunakan dua struktur data, yaitu Titik untuk menyimpan koordinat titik `(x, y)` dan Lingkaran untuk menyimpan properti lingkaran berupa pusat (dalam bentuk `Titik`) dan radius (dalam bentuk integer).

Program dimulai dengan meminta input dari pengguna untuk menentukan koordinat pusat dan radius masing-masing dari dua lingkaran. Kemudian, program meminta input koordinat sebuah titik sembarang. Fungsi `hitungjarak` digunakan untuk menghitung jarak Euclidean antara dua titik menggunakan rumus matematika akar kuadrat dari selisih kuadrat koordinatnya.

Selanjutnya, fungsi `titikDidalamLingkaran` digunakan untuk memeriksa apakah jarak antara titik sembarang dan pusat lingkaran lebih kecil atau sama dengan radius lingkaran. Berdasarkan hasil pengecekan ini, program menentukan apakah titik tersebut berada di dalam lingkaran pertama, lingkaran kedua, keduanya, atau di luar kedua lingkaran, dan menampilkan pesan yang sesuai kepada pengguna.

Namun, terdapat kesalahan pada input untuk **lingkaran kedua** karena variabel yang digunakan untuk menyimpan input (`cx1`, `cy1`, `r1`) adalah variabel yang sama dengan lingkaran pertama, sehingga nilai lingkaran pertama akan tertimpa.

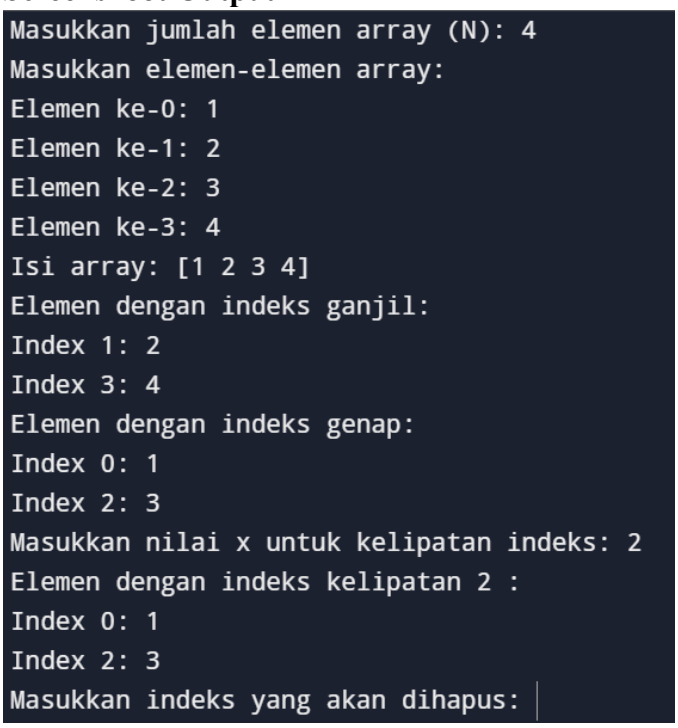
III. UNGUIDED

Unguided 1

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n, x, delIndex, target int
    fmt.Print("Masukkan jumlah elemen array (N): ")
    fmt.Scan(&n)

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

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

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

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

    // d. Menampilkan elemen dengan indeks kelipatan
bilangan x
    fmt.Print("Masukkan nilai x untuk kelipatan indeks:
")
    fmt.Scan(&x)
    fmt.Println("Elemen dengan indeks kelipatan", x,
":")
    for i := 0; i < n; i++ {
        if i%x == 0 {
            fmt.Printf("Index %d: %d\n", i, array[i])
        }
    }
```

```
    // e. Menghapus elemen pada indeks tertentu
    fmt.Print("Masukkan indeks yang akan dihapus: ")
    fmt.Scan(&delIndex)
    array = append(array[:delIndex],
array[delIndex+1:]...)
    fmt.Println("Isi array setelah penghapusan:", array)

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

    // g. Menampilkan standar deviasi
    var varianceSum float64
    for _, v := range array {
        varianceSum += math.Pow(float64(v)-rataRata, 2)
    }
    stdDeviasi := math.Sqrt(varianceSum /
float64(len(array)))
    fmt.Printf("Standar deviasi elemen array: %.2f\n",
stdDeviasi)

    // h. Menampilkan frekuensi suatu bilangan
    fmt.Print("Masukkan bilangan untuk mengetahui
frekuensinya: ")
    fmt.Scan(&target)
    count := 0
    for _, v := range array {
        if v == target {
            count++
        }
    }
    fmt.Printf("Frekuensi bilangan %d dalam array:
%d\n", target, count)
}
```

Screenshoot Output



Deskripsi Program

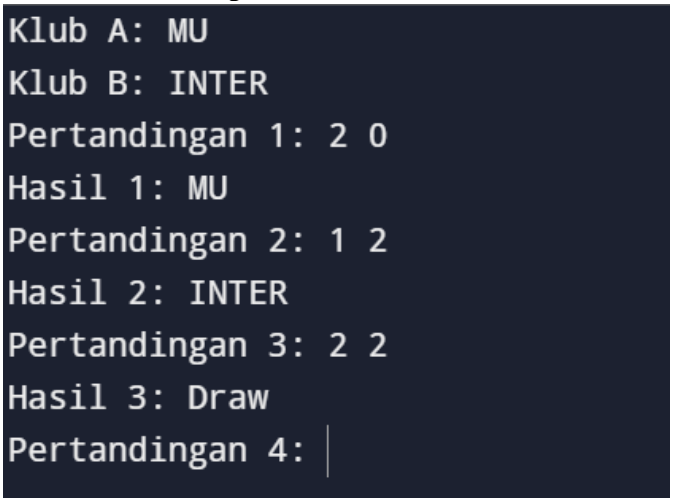
Program ini ditulis dalam bahasa Go untuk mengelola dan menganalisis data pada sebuah array integer dengan berbagai fitur. Awalnya, pengguna diminta memasukkan jumlah elemen array (`n`) dan mengisi array dengan nilai-nilai yang dimasukkan satu per satu. Program kemudian menampilkan isi array dan memisahkan elemen-elemen berdasarkan indeks ganjil maupun genap. Selanjutnya, pengguna dapat menentukan bilangan `x` untuk menampilkan elemen pada indeks yang merupakan kelipatan `x`. Program juga menyediakan fitur penghapusan elemen pada indeks tertentu, di mana elemen pada indeks yang dipilih akan dihapus, dan array diperbarui. Setelah itu, program menghitung rata-rata elemen dalam array dan menggunakan rata-rata tersebut untuk menghitung standar deviasi, yang mencerminkan penyebaran nilai-nilai dalam array. Terakhir, program meminta pengguna memasukkan angka tertentu untuk menghitung frekuensinya dalam array. Dengan kombinasi manipulasi dan analisis ini, program memberikan alat interaktif untuk bekerja dengan array dalam berbagai skenario.

Unguided 2

```
package main

import (
	"fmt"
)

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

	// Input nama klub
	fmt.Print("Klub A: ")
	fmt.Scanln(&klubA)
	fmt.Print("Klub B: ")
	fmt.Scanln(&klubB)

	// Proses input skor pertandingan
	pertandingan := 1
	for {
		fmt.Printf("Pertandingan %d: ", pertandingan)
		fmt.Scan(&skorA, &skorB)

		// Validasi skor
		if skorA < 0 || skorB < 0 {
			break
		}

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

		pertandingan++
	}

	fmt.Println("Pertandingan selesai")

	// Menampilkan daftar klub pemenang
	fmt.Println("Daftar klub yang memenangkan
pertandingan:")
	for _, klub := range pemenang {
```

```
            fmt.Println(klub)
        }
}
```

Screenshoot Output



Deskripsi Program

Kode ini adalah program untuk merekap hasil pertandingan antara dua klub sepak bola berdasarkan skor yang diberikan. Pertama, program meminta pengguna untuk memasukkan nama kedua klub yang akan bertanding, lalu mencatat nama klub tersebut dalam variabel `klubA` dan `klubB`. Selanjutnya, program memasuki sebuah loop untuk menerima input skor pertandingan antara kedua klub. Setiap skor yang dimasukkan diverifikasi agar tidak negatif; jika skor negatif terdeteksi, loop akan berhenti, menandai akhir dari proses input.

Dalam setiap iterasi, skor kedua klub dibandingkan. Jika skor klub A lebih besar dari klub B, program menyimpan nama klub A ke dalam array `pemenang` dan mencetak bahwa klub A memenangkan pertandingan. Jika skor klub B lebih besar, nama klub B disimpan ke dalam array `pemenang`, dan program mencetak bahwa klub B menang. Jika kedua skor sama, program mencetak hasil "Draw" untuk pertandingan tersebut.

Ketika input skor negatif dimasukkan, program keluar dari loop dan mencetak pesan bahwa pertandingan telah selesai. Akhirnya, program menampilkan daftar nama klub yang memenangkan pertandingan dalam urutan kronologis berdasarkan skor yang dimasukkan.

Unguided 3

```
package main

import (
	"fmt"
)

const NMAX int = 127
type tabel [NMAX]rune

// Prosedur untuk mengisi array dengan karakter
func isiArray(t *tabel, n *int) {
	var ch rune
	fmt.Println("Masukkan teks (akhiri dengan TITIK):")
	*n = 0
	for {
		fmt.Scanf("%c", &ch)
		if ch == '.' || *n >= NMAX {
			break
		}
		t[*n] = ch
		*n++
	}
}

// Prosedur untuk mencetak isi array
func cetakArray(t tabel, n int) {
	for i := 0; i < n; i++ {
		fmt.Printf("%c", t[i])
	}
	fmt.Println()
}

// Prosedur untuk membalikkan isi array
func balikanArray(t *tabel, n int) {
	for i := 0; i < n/2; i++ {
		t[i], t[n-i-1] = t[n-i-1], t[i]
	}
}

// Fungsi untuk mengecek apakah array membentuk
palindrom
func palindrom(t tabel, n int) bool {
	var reversed tabel
	copy(reversed[:n], t[:n])
	balikanArray(&reversed, n)
	for i := 0; i < n; i++ {
		if t[i] != reversed[i] {
			return false
		}
	}
	return true
```

```
}

func main() {
	var tab tabel
	var m int

	// Isi array dengan input dari user
	isiArray(&tab, &m)

	// Cetak teks asli
	fmt.Print("Teks: ")
	cetakArray(tab, m)

	// Balikkan teks dan cetak hasilnya
	var reversed tabel
	copy(reversed[:m], tab[:m])
	balikanArray(&reversed, m)
	fmt.Print("Reverse teks: ")
	cetakArray(reversed, m)

	// Cek apakah palindrom dan tampilkan hasilnya
	if palindrom(tab, m) {
		fmt.Println("Palindrom? true")
	} else {
		fmt.Println("Palindrom? false")
	}
}
```

Screenshoot Output

```
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
Palindrom? false
```

```
Masukkan teks (akhiri dengan TITIK):
k a t a k.
Teks: k a t a k
Reverse teks: k a t a k
Palindrom? true
```

Deskripsi Program

Program di atas ditulis dalam bahasa Go dan dirancang untuk membaca sebuah teks dari pengguna, memprosesnya dalam sebuah array karakter, membalikkan urutan teks, dan memeriksa apakah teks tersebut adalah palindrom. **Palindrom** adalah teks yang terbaca sama baik dari awal maupun dari akhir, misalnya "KATAK" atau "RADAR".

Pertama, prosedur `isiArray` membaca input karakter dari pengguna satu per satu hingga menemukan karakter titik (`.`) atau jumlah karakter mencapai batas maksimal (127). Karakter yang dibaca disimpan dalam array `tabel`. Selanjutnya, prosedur `cetakArray` digunakan untuk mencetak seluruh elemen array ke layar.

Prosedur `balikanArray` membalikkan urutan elemen array dengan cara menukar elemen pertama dengan elemen terakhir, elemen kedua dengan elemen kedua dari akhir, dan seterusnya hingga mencapai titik tengah array. Fungsi `palindrom` mengecek apakah teks dalam array membentuk palindrom dengan membandingkan array asli dengan versi array yang telah dibalik menggunakan `balikanArray`.

Dalam fungsi `main`, program membaca teks dari pengguna, mencetak teks asli, mencetak teks setelah dibalik, dan mengecek apakah teks tersebut adalah palindrom. Hasil akhir berupa informasi teks asli, teks yang telah dibalik, serta status apakah teks tersebut palindrom atau tidak.

Contoh interaksi dengan program: Jika pengguna memasukkan teks "KATAK.", program akan mencetak teks asli ("KATAK"), teks yang telah dibalik ("KATAK"), dan menyatakan bahwa teks tersebut adalah palindrom (true). Sebaliknya, jika pengguna memasukkan teks "SENANG.", program akan mencetak teks asli ("SENANG"), teks yang telah dibalik ("GNANES"), dan menyatakan bahwa teks tersebut bukan palindrom (false). Program ini menggunakan pendekatan modular dengan membagi fungsionalitas ke dalam prosedur dan fungsi terpisah untuk memudahkan pembacaan dan pemeliharaan kode.