

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7
STRUCT DAN ARRAY**



Disusun Oleh :

Shiva Indah Kurnia

2311102035

S1-IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom.,M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

A. ARRAY

Array adalah tipe data dasar yang digunakan untuk menyimpan sekumpulan elemen dengan tipe data yang sama. Di bahasa Go, array memiliki ukuran tetap yang ditetapkan saat deklarasi.

Karakter Array:

1. Ukuran Standar:

Dibandingkan dengan tipe data lain seperti slice, ukuran array di Go ditentukan saat deklarasi dan tidak dapat diubah selama runtime.

2. Data homogen terdiri dari:

Semua elemen dalam array harus memiliki tipe data yang sama—misalnya, int, string, atau float64.

3. Index dimulai dengan zero:

Element awal array berada di indeks 0, elemen kedua berada di indeks 1, dan seterusnya.

4. Akses ke elemen:

Dengan menggunakan notasi `array[index]`, Anda dapat mengakses elemen dalam array dengan menggunakan indeks.

Syarat Array:

Ukuran dan tipe data array dapat dideklarasikan. Dalam Go, ada beberapa cara untuk mendeklarasikan dan menginisialisasi array.

1. Deklarasi Array Kosong
2. Inisialisasi Array dengan Nilai
3. Deklarasi Array dengan Panjang Otomatis
4. Penggunaan Array
5. Array Multidimensi

B. STRUCT

Struktur sering digunakan untuk menunjukkan objek atau entitas dengan beberapa atribut. Ini adalah tipe data komposit yang memungkinkan kita mengelompokkan variabel dengan tipe yang berbeda dalam satu entitas.

Karakter Struktur:

1. Tipe Data Heterogen:

Dalam struct, field dapat memiliki berbagai tipe data, yang memungkinkan penggunaan tipe data yang lebih kompleks.

2. Nama Field:

Setiap field struct memiliki nama sendiri.

3. Notasi Titik:

Kita menggunakan notasi titik (.) untuk mengakses atau mengubah area.

4. Komposisi Struct:

Struct dapat menggabungkan struct lain karena mendukung komposisi.

Deklarasi Struktur:

Deklarasi struktur dimulai dengan kata kunci type, nama struktur, dan definisi field dalam kurung kurawal.

II. GUIDED

1. Guided 1 Soal Studi Case

Program sederhana untuk menghitung lama waktu parkir berdasarkan waktu kedatangan dan waktu pulang

Sourcecode

```
package main

import "fmt"

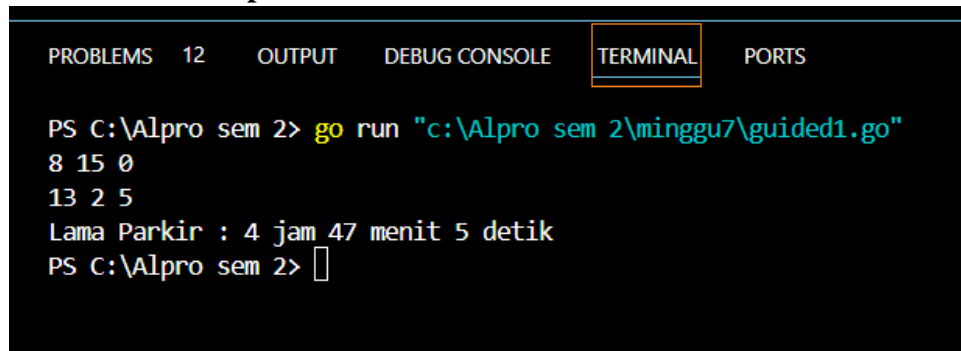
type waktu struct {
    jam, menit, detik int
}

func main() {
    var wParkir, wPulang, durasi waktu
    var dParkir, dPulang, lParkir int

    fmt.Scan(&wParkir.jam, &wParkir.menit, &wParkir.detik)
    fmt.Scan(&wPulang.jam, &wPulang.menit, &wPulang.detik)
    dParkir = wParkir.detik + wParkir.menit*60 +
wParkir.jam*3600 // Konversi ke detik
    dPulang = wPulang.detik + wPulang.menit*60 +
wPulang.jam*3600 // detik
    lParkir = dPulang -
dParkir //detik dari
pulang-datang

    durasi.jam = lParkir / 3600
    durasi.menit = lParkir % 3600 / 60
    durasi.detik = lParkir % 3600 % 60 //17
    fmt.Printf("Lama Parkir : %d jam %d menit %d detik",
durasi.jam, durasi.menit, durasi.detik)
}
```

Screenshoot Output



```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Alpro sem 2> go run "c:\Alpro sem 2\minggu7\guided1.go"
8 15 0
13 2 5
Lama Parkir : 4 jam 47 menit 5 detik
PS C:\Alpro sem 2> 
```

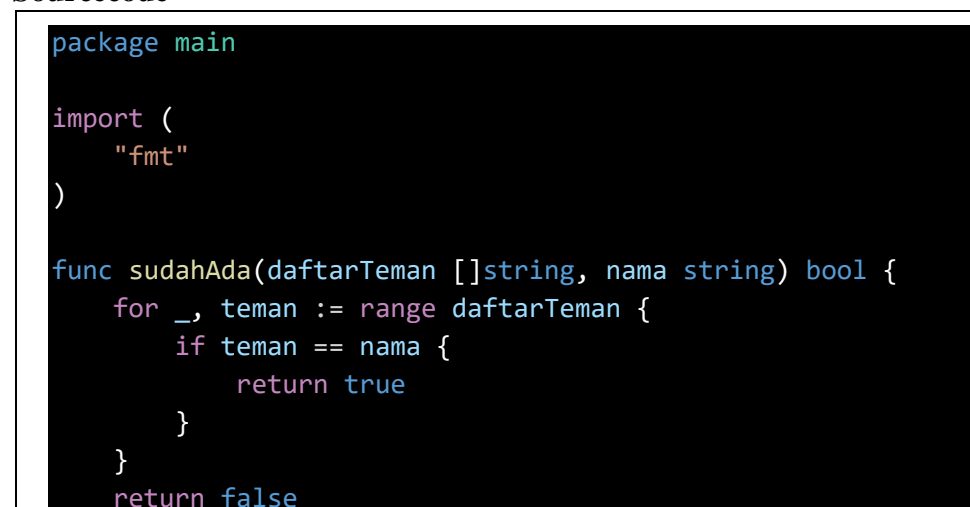
Deskripsi Program :

Program di atas merupakan program Go yang menghitung durasi parkir kendaraan berdasarkan waktu masuk dan waktu keluar. Program menggunakan struktur data `waktu` yang memiliki tiga komponen: jam, menit, dan detik. Program menerima input waktu parkir (masuk) dan waktu pulang (keluar) dari pengguna, kemudian mengkonversi kedua waktu tersebut ke dalam satuan detik untuk memudahkan perhitungan selisih waktu. Setelah mendapatkan selisih waktu dalam detik, program mengkonversi kembali ke format jam, menit, dan detik yang lebih mudah dibaca, lalu menampilkan hasil durasi parkir dalam format "Lama Parkir: X jam X menit X detik". Program ini berguna untuk menghitung berapa lama sebuah kendaraan parkir di suatu tempat dengan tepat hingga satuan detik.

2. Guided 2 Soal Studi Case

Program sederhana untuk validasi duplikasi nama pada daftar teman

Sourcecode



```
package main

import (
    "fmt"
)

func sudahAda(daftarTeman []string, nama string) bool {
    for _, teman := range daftarTeman {
        if teman == nama {
            return true
        }
    }
    return false
}
```

```

}

func main() {
    daftarTeman := []string{"Andi", "Budi", "Cici"}

    namaBaru := []string{"Dewi", "Budi", "Eka"}

    for _, nama := range namaBaru {
        if !sudahAda(daftarTeman, nama) {
            daftarTeman = append(daftarTeman, nama)
        } else {
            fmt.Println("Nama", nama, "sudah ada dalam
daftar.")
        }
    }

    fmt.Println("Daftar Teman:", daftarTeman)
}

```

Screenshoot Output

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Alpro sem 2> go run "c:\Alpro sem 2\minggu7\guided2.go"
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS C:\Alpro sem 2>

```

Deskripsi Program :

Program di atas adalah program Go yang mengelola daftar nama teman dengan mencegah duplikasi nama. Program memiliki fungsi `sudahAda` yang bertugas memeriksa apakah sebuah nama sudah terdapat dalam daftar teman yang ada. Program dimulai dengan daftar awal yang berisi tiga nama ("Andi", "Budi", "Cici"), kemudian mencoba menambahkan tiga nama baru ("Dewi", "Budi", "Eka"). Saat menambahkan nama baru, program memeriksa setiap nama menggunakan fungsi `sudahAda` - jika nama belum ada, maka nama tersebut akan ditambahkan ke daftar, namun jika nama sudah ada (seperti "Budi"), program akan menampilkan pesan bahwa nama tersebut sudah ada dalam daftar. Pada

akhirnya, program menampilkan daftar teman yang sudah diperbarui, yang hanya akan menambahkan nama-nama yang belum ada sebelumnya.

3. Guided 3 Soal Studi Case

Program sederhana untuk menampilkan daftar harga buah

Sourcecode

```
package main

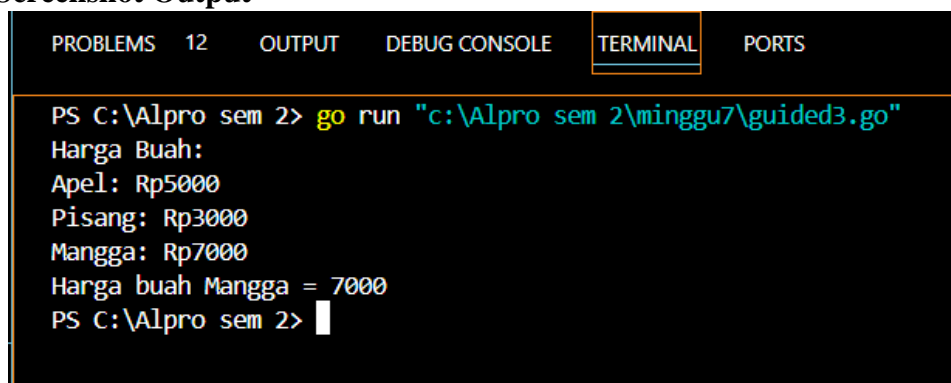
import "fmt"

func main() {
    hargaBuah := map[string]int{
        "Apel": 5000,
        "Pisang": 3000,
        "Mangga": 7000,
    }

    fmt.Println("Harga Buah:")
    for buah, harga := range hargaBuah {
        fmt.Printf("%s: Rp%d\n", buah, harga)
    }

    fmt.Print("Harga buah Mangga = ", hargaBuah["Mangga"])
}
```

Screenshot Output



```
PS C:\Alpro sem 2> go run "c:\Alpro sem 2\minggu7\guided3.go"
Harga Buah:
Apel: Rp5000
Pisang: Rp3000
Mangga: Rp7000
Harga buah Mangga = 7000
PS C:\Alpro sem 2> 
```

Deskripsi Program :

Program di atas adalah program Go yang menggunakan struktur data map untuk menyimpan daftar harga buah-buahan. Program membuat sebuah map bernama `hargaBuah` yang memetakan nama buah (string) dengan harganya (integer), di mana terdapat tiga jenis buah yaitu Apel seharga Rp5000, Pisang seharga Rp3000, dan Mangga seharga Rp7000. Program kemudian menampilkan seluruh daftar harga buah menggunakan perulangan `for range` yang mengiterasi setiap pasangan key-value dalam map, dan mencetak nama buah beserta harganya. Terakhir, program menampilkan harga spesifik untuk buah Mangga dengan mengakses langsung nilai dalam map menggunakan key "Mangga".

III. UNGUIDED

1. Unguided 1 Soal Studi Case

Suatu lingkaran didefinisikan dengan koordinat titik pusat (cx, cy) dengan radius r . Apabila diberikan dua buah lingkaran, maka tentukan posisi sebuah titik sembarang (x, y) berdasarkan dua lingkaran tersebut. Gunakan tipe bentukan titik untuk menyimpan koordinat, dan tipe bentukan lingkaran untuk menyimpan titik pusat lingkaran dan radiusnya.

Masukan terdiri dari beberapa tiga baris. Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingkaran 2, sedangkan baris ketiga adalah koordinat titik sembarang. Asumsi sumbu x dan y dari semua titik dan juga radius direpresentasikan dengan bilangan bulat.

Keluaran berupa string yang menyatakan posisi titik "Titik di dalam lingkaran 1 dan 2", "Titik di dalam lingkaran 1", "Titik di dalam lingkaran 2", atau "Titik di luar lingkaran 1 dan 2".

No	Masukan	Keluaran
1	1 1 5 8 8 4 2 2	Titik di dalam lingkaran 1
2	1 2 3 4 5 6 7 8	Titik di dalam lingkaran 2
3	5 10 15 -15 4 20 0 0	Titik di dalam lingkaran 1 dan 2
4	1 1 5 8 8 4 15 20	Titik di luar lingkaran 1 dan 2

Sourcecode

```
package main

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

// Struct untuk menyimpan koordinat titik
type Point struct {
    x, y float64
}

// Struct untuk menyimpan informasi lingkaran
type Circle struct {
    center Point
    radius float64
}

// Fungsi untuk menghitung jarak antara dua titik
func distance(p1, p2 Point) float64 {
    dx := p1.x - p2.x
    dy := p1.y - p2.y
    return math.Sqrt(dx*dx + dy*dy)
}

// Fungsi untuk mengecek apakah titik berada di dalam
lingkaran
func isPointInCircle(p Point, c Circle) bool {
    return distance(p, c.center) <= c.radius
}

// Fungsi untuk menentukan posisi titik terhadap dua
lingkaran
func determinePosition(p Point, c1, c2 Circle) string {
    inCircle1 := isPointInCircle(p, c1)
    inCircle2 := isPointInCircle(p, c2)

    switch {
```

```

        case inCircle1 && inCircle2:
            return "Titik di dalam lingkaran 1 dan 2"
        case inCircle1:
            return "Titik di dalam lingkaran 1"
        case inCircle2:
            return "Titik di dalam lingkaran 2"
        default:
            return "Titik di luar lingkaran 1 dan 2"
    }
}

func main() {
    scanner := bufio.NewScanner(os.Stdin)

    // Membaca data lingkaran 1
    circle1 := readCircle(scanner)

    // Membaca data lingkaran 2
    circle2 := readCircle(scanner)

    // Membaca koordinat titik yang akan dicek
    point := readPoint(scanner)

    // Menentukan dan mencetak hasil
    result := determinePosition(point, circle1, circle2)
    fmt.Println(result)
}

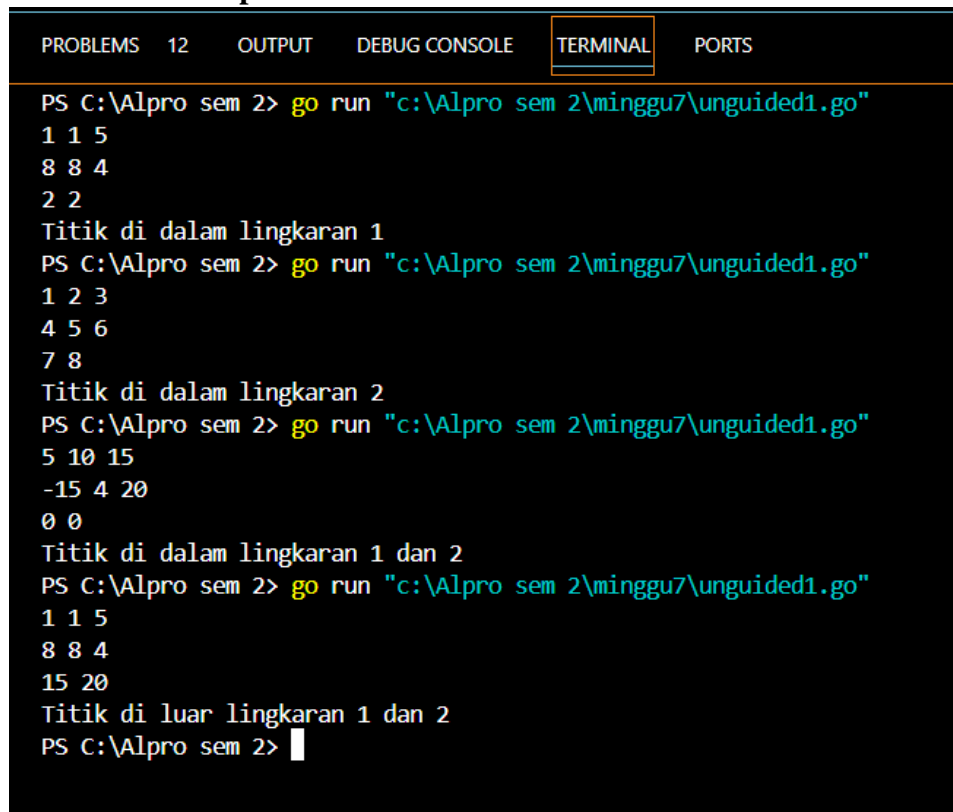
// Fungsi untuk membaca data lingkaran dari input
func readCircle(scanner *bufio.Scanner) Circle {
    scanner.Scan()
    line := strings.Fields(scanner.Text())
    x, _ := strconv.ParseFloat(line[0], 64)
    y, _ := strconv.ParseFloat(line[1], 64)
    radius, _ := strconv.ParseFloat(line[2], 64)
    return Circle{Point{x, y}, radius}
}

// Fungsi untuk membaca koordinat titik dari input
func readPoint(scanner *bufio.Scanner) Point {
    scanner.Scan()
    line := strings.Fields(scanner.Text())
    x, _ := strconv.ParseFloat(line[0], 64)
    y, _ := strconv.ParseFloat(line[1], 64)

```

```
    return Point{x, y}  
}
```

Screenshoot Output



```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Alpro sem 2> go run "c:\Alpro sem 2\minggu7\unguided1.go"  
1 1 5  
8 8 4  
2 2  
Titik di dalam lingkaran 1  
PS C:\Alpro sem 2> go run "c:\Alpro sem 2\minggu7\unguided1.go"  
1 2 3  
4 5 6  
7 8  
Titik di dalam lingkaran 2  
PS C:\Alpro sem 2> go run "c:\Alpro sem 2\minggu7\unguided1.go"  
5 10 15  
-15 4 20  
0 0  
Titik di dalam lingkaran 1 dan 2  
PS C:\Alpro sem 2> go run "c:\Alpro sem 2\minggu7\unguided1.go"  
1 1 5  
8 8 4  
15 20  
Titik di luar lingkaran 1 dan 2  
PS C:\Alpro sem 2> 
```

Deskripsi Program :

Program Go di atas berfungsi untuk menentukan posisi sebuah titik relatif terhadap dua lingkaran. Program ini menggunakan dua struktur data, yaitu "point" yang menyimpan koordinat titik (x, y) dan "circle" yang menyimpan informasi lingkaran, yaitu pusat dan jari-jari. Program ini memiliki beberapa fungsi utama, yaitu "distance" untuk menghitung jarak antara dua titik menggunakan rumus Pythagoras, dan "isPointInCircle"

2. Unguided 2 Soal Studi Case

Sebuah array digunakan untuk menampung sekumpulan bilangan bulat. Buatlah program yang digunakan untuk mengisi array tersebut sebanyak N elemen nilai. Asumsikan array memiliki kapasitas penyimpanan data sejumlah elemen tertentu. Program dapat menampilkan beberapa informasi berikut:

- Menampilkan keseluruhan isi dari array.

- b. Menampilkan elemen-elemen array dengan indeks ganjil saja.
- c. Menampilkan elemen-elemen array dengan indeks genap saja (asumsi indek ke-0 adalah genap).
- d. Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x. x bisa diperoleh dari masukan pengguna.
- e. Menghapus elemen array pada indeks tertentu, asumsi indeks yang hapus selalu valid. Tampilkan keseluruhan isi dari arraynya, pastikan data yang dihapus tidak tampil
- f. Menampilkan rata-rata dari bilangan yang ada di dalam array.
- g. Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut.
- h. Menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi tersebut.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

type ArrayOperations struct {
    data []float64
}

// Membuat array baru
func NewArrayOperations(capacity int) *ArrayOperations {
    return &ArrayOperations{data: make([]float64, 0,
capacity)}
}

// Menambah elemen ke array
func (a *ArrayOperations) Add(value float64) {
    if len(a.data) < cap(a.data) {
        a.data = append(a.data, value)
    }
}

// Menampilkan seluruh isi array
func (a *ArrayOperations) DisplayAll() {
    fmt.Println("Seluruh isi array:")
}
```

```

    for i, v := range a.data {
        fmt.Printf("Index %d: %.2f\n", i, v)
    }
}

// Menampilkan elemen dengan indeks ganjil atau genap
func (a *ArrayOperations) DisplayIndices(isEven bool) {
    if isEven {
        fmt.Println("Elemen dengan indeks genap:")
    } else {
        fmt.Println("Elemen dengan indeks ganjil:")
    }
    for i, v := range a.data {
        if (isEven && i%2 == 0) || (!isEven && i%2 != 0) {
            fmt.Printf("Index %d: %.2f\n", i, v)
        }
    }
}

// Menampilkan elemen dengan indeks kelipatan x
func (a *ArrayOperations) DisplayMultipleIndices(x int) {
    fmt.Printf("Elemen dengan indeks kelipatan %d:\n", x)
    for i, v := range a.data {
        if i%x == 0 {
            fmt.Printf("Index %d: %.2f\n", i, v)
        }
    }
}

// Menghapus elemen pada indeks tertentu
func (a *ArrayOperations) DeleteAtIndex(index int) {
    if index >= 0 && index < len(a.data) {
        a.data = append(a.data[:index], a.data[index+1:]...)
    }
}

// Menghitung rata-rata
func (a *ArrayOperations) CalculateAverage() float64 {
    sum := 0.0
    for _, v := range a.data {
        sum += v
    }
    return sum / float64(len(a.data))
}

```

```

// Menghitung standar deviasi
func (a *ArrayOperations) CalculateStandardDeviation()
float64 {
    if len(a.data) < 2 {
        return 0
    }
    mean := a.CalculateAverage()
    sumSquaredDiff := 0.0
    for _, v := range a.data {
        diff := v - mean
        sumSquaredDiff += diff * diff
    }
    return math.Sqrt(sumSquaredDiff / float64(len(a.data)-
1))
}

// Menghitung frekuensi suatu bilangan
func (a *ArrayOperations) CalculateFrequency(number float64)
int {
    frequency := 0
    for _, v := range a.data {
        if v == number {
            frequency++
        }
    }
    return frequency
}

func main() {
    arr := NewArrayOperations(10)

    values := []float64{1.0, 2.0, 3.0, 4.0, 5.0, 2.0, 3.0}
    for _, value := range values {
        arr.Add(value)
    }

    fmt.Println("\n=== Demonstrasi Operasi Array ===")
    arr.DisplayAll()
    fmt.Println()
    arr.DisplayIndices(false) // Indeks ganjil
    fmt.Println()
    arr.DisplayIndices(true) // Indeks genap
    fmt.Println()
}

```

```

arr.DisplayMultipleIndices(2)
fmt.Println()

fmt.Println("Menghapus elemen index 2:")
arr.DeleteAtIndex(2)
arr.DisplayAll()
fmt.Println()

fmt.Printf("Rata-rata: %.2f\n", arr.CalculateAverage())
fmt.Printf("Standar deviasi: %.2f\n",
arr.CalculateStandardDeviation())
fmt.Printf("Frekuensi angka 2.0: %d\n",
arr.CalculateFrequency(2.0))
}

```

Screenshot Output

```

=== Demonstrasi Operasi Array ===
Seluruh isi array:
Index 0: 1.00
Index 1: 2.00
Index 2: 3.00
Index 3: 4.00
Index 4: 5.00
Index 5: 2.00
Index 6: 3.00

Elemen dengan indeks ganjil:
Index 1: 2.00
Index 3: 4.00
Index 5: 2.00

Elemen dengan indeks genap:
Index 0: 1.00
Index 2: 3.00
Index 4: 5.00
Index 6: 3.00

```

```
Elemen dengan indeks kelipatan 2:
```

```
Index 0: 1.00
```

```
Index 2: 3.00
```

```
Index 4: 5.00
```

```
Index 6: 3.00
```

```
Menghapus elemen index 2:
```

```
Seluruh isi array:
```

```
Index 0: 1.00
```

```
Index 1: 2.00
```

```
Index 2: 4.00
```

```
Index 3: 5.00
```

```
Index 4: 2.00
```

```
Index 5: 3.00
```

```
Rata-rata: 2.83
```

```
Standar deviasi: 1.47
```

```
Frekuensi angka 2.0: 2
```

```
PS C:\Alpro sem 2>
```

Deskripsi Program :

Program di atas adalah implementasi Go yang mengelola array floating-point menggunakan struktur data `ArrayOperations`. Program ini menyediakan berbagai fungsi untuk memanipulasi dan menganalisis array, termasuk menambah elemen (`Add`), menampilkan seluruh isi array (`DisplayAll`), menampilkan elemen dengan indeks ganjil dan genap, menampilkan elemen dengan indeks kelipatan tertentu, menghapus elemen pada indeks tertentu (`DeleteAtIndex`), serta menghitung rata-rata, standar deviasi, dan frekuensi kemunculan suatu bilangan. Dengan menggunakan konsep `struct` dan `method`, program mendemonstrasikan semua operasi tersebut dalam fungsi `main` dengan membuat array contoh dan menjalankan setiap operasi secara berurutan.

3. Unguided 3 Soal Studi Case

Sebuah program digunakan untuk menyimpan dan menampilkan nama-nama klub yang memenangkan pertandingan bola pada suatu grup pertandingan. Buatlah program yang digunakan untuk merekap skor pertandingan bola 2 buah klub bola yang berlaga. Pertama-tama program meminta masukan nama-nama klub yang bertanding, kemudian program meminta masukan skor hasil pertandingan kedua klub tersebut. Yang disimpan dalam array adalah nama-nama klub yang menang saja. Proses input skor berhenti ketika skor salah satu atau kedua klub tidak valid (negatif). Di akhir program, tampilkan daftar klub yang memenangkan pertandingan.

Perhatikan sesi interaksi pada contoh berikut ini (teks bergaris bawah adalah input/read)

```
Klub A : MU
Klub B : Inter
Pertandingan 1 : 2 0 // MU = 2 sedangkan Inter = 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
Pertandingan selesai
```

Sourcecode

```
package main

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

func main() {
    scanner := bufio.NewScanner(os.Stdin)
    var winners []string

    // Input nama klub
    fmt.Print("Klub A: ")
    scanner.Scan()
    clubA := scanner.Text()

    fmt.Print("Klub B: ")
```

```

scanner.Scan()
clubB := scanner.Text()

// Proses input skor pertandingan
for matchNumber := 1; ; matchNumber++ {
    fmt.Printf("Pertandingan %d: ", matchNumber)
    scanner.Scan()
    input := scanner.Text()

    scores := strings.Fields(input)
    if len(scores) != 2 {
        break
    }

    var scoreA, scoreB int
    if _, err1 := fmt.Sscanf(scores[0], "%d",
&scoreA); err1 != nil || scoreA < 0 {
        break
    }
    if _, err2 := fmt.Sscanf(scores[1], "%d",
&scoreB); err2 != nil || scoreB < 0 {
        break
    }

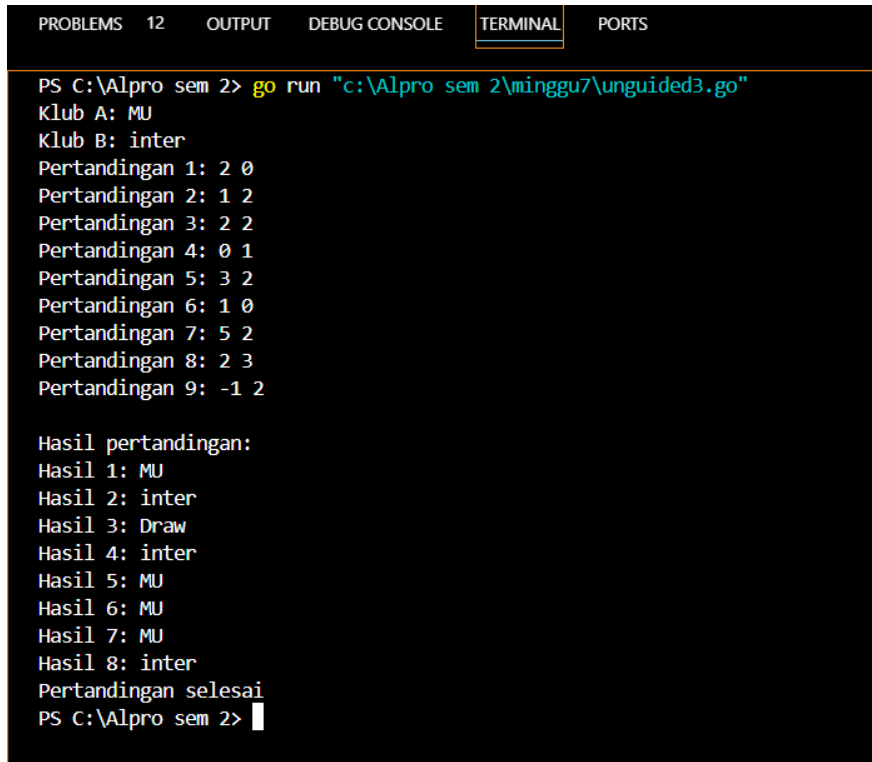
    var winner string
    if scoreA > scoreB {
        winner = clubA
    } else if scoreB > scoreA {
        winner = clubB
    } else {
        winner = "Draw"
    }

    winners = append(winners, winner)
}

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

```

Screenshot Output

A screenshot of a terminal window with a dark background. The terminal shows the execution of a Go program. The prompt is 'PS C:\Alpro sem 2>'. The user enters 'go run "c:\Alpro sem 2\minggu7\unguided3.go"'. The program outputs the names of two clubs: 'Klub A: MU' and 'Klub B: inter'. It then displays nine match results in the format 'Pertandingan X: skor_A skor_B'. The results are: 1: 2 0, 2: 1 2, 3: 2 2, 4: 0 1, 5: 3 2, 6: 1 0, 7: 5 2, 8: 2 3, and 9: -1 2. After the matches, it shows 'Hasil pertandingan:' followed by the result for each match: 'Hasil 1: MU', 'Hasil 2: inter', 'Hasil 3: Draw', 'Hasil 4: inter', 'Hasil 5: MU', 'Hasil 6: MU', 'Hasil 7: MU', 'Hasil 8: inter', and 'Pertandingan selesai'. The prompt returns to 'PS C:\Alpro sem 2>'.

```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Alpro sem 2> go run "c:\Alpro sem 2\minggu7\unguided3.go"
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

Hasil pertandingan:
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:\Alpro sem 2> 
```

Deskripsi Program :

Program di atas adalah program Go yang mencatat hasil pertandingan antara dua klub olahraga. Program dimulai dengan meminta nama kedua klub, kemudian secara berulang meminta skor pertandingan dalam format "skor_A skor_B". Input akan terus diminta hingga pengguna memasukkan skor negatif atau format yang salah. Setiap skor yang valid akan menentukan pemenang, yang disimpan dalam slice winners. Ketika input tidak valid diterima, program menampilkan hasil pertandingan yang dicatat dan mengakhiri dengan pesan "Pertandingan selesai".

1. Unguided 4 Soal Studi Case

Sebuah array digunakan untuk menampung sekumpulan karakter, Anda diminta untuk membuat subprogram untuk melakukan membalikkan urutan isi array dan memeriksa apakah membentuk palindrom. Lengkapi potongan algoritma berikut ini!

```

package main
import "fmt"
const NMAX int = 127
type tabel [NMAX]rune
    tab : tabel
    m : integer

func isiArray(t *tabel, n *int)
/*I.S. Data tersedia dalam piranti masukan
 F.S. Array t berisi sejumlah n karakter yang dimasukkan user,
 Proses input selama karakter bukanlah TITIK dan n <= NMAX */

func cetakArray(t tabel, n int)
/*I.S. Terdefinisi array t yang berisi sejumlah n karakter
 F.S. n karakter dalam array muncul di layar */

func balikanArray(t *tabel, n int)
/*I.S. Terdefinisi array t yang berisi sejumlah n karakter
 F.S. Urutan isi array t terbalik */

func main(){
    var tab tabel
    var m int
    // si array tab dengan memanggil prosedur isiArray

    // Balikian isi array tab dengan memanggil balikanArray

    // Cetak isi array tab
}

```

Perhatikan sesi Interaksi pada contoh berikut ini (**teks bergaris bawah adalah Input/read**)

Teks	: S E N A N G .
Reverse teks	: G N A N E S
Teks	: K A T A K .
Reverse teks	: K A T A K

Modifikasi program tersebut dengan menambahkan fungsi palindrom. Tambahkan instruksi untuk memanggil fungsi tersebut dan menampilkan hasilnya pada program utama.

***Palindrom adalah teks yang dibaca dari awal atau akhir adalah sama, contoh: KATAK, APA, KASUR_RUSAK.**

```

func palindrom(t tabel, n int) bool
/* Mengembalikan true apabila susunan karakter di dalam t membentuk palindrom,
 dan false apabila sebaliknya. Petunjuk: Manfaatkan prosedur balikanArray */

```

Perhatikan sesi interaksi pada contoh berikut ini (**teks bergaris bawah adalah Input/read**)

Teks	: K A T A K
Palindrom	? true
Teks	: S E N A N G
Palindrom	? false

Sourcecode

```
package main

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

const NMAX int = 127

type tabel [NMAX]rune

// Fungsi untuk mengisi array dengan karakter
func isiArray(t *tabel, n *int) {
    scanner := bufio.NewScanner(os.Stdin)
    fmt.Print("Teks: ")
    scanner.Scan()
    text :=
strings.ReplaceAll(strings.ReplaceAll(strings.ReplaceAll(scanner.Text(), " ", ""), ".", ""), ",", "")
    *n = len([]rune(text))
    for i, char := range text {
        if i < NMAX {
            t[i] = char
        }
    }
}

// Fungsi untuk mencetak isi array
func cetakArray(t tabel, n int) {
    fmt.Print("Reverse teks: ")
    for i := n - 1; i >= 0; i-- {
        fmt.Printf("%c", t[i])
    }
    fmt.Println()
}

// Fungsi untuk mengecek 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
}
```

```

    }
}
return true
}

func main() {
    var tab tabel
    var m int

    // Input dan proses array
    isiArray(&tab, &m)

    // Tampilkan array yang dibalik
    cetakArray(tab, m)

    // Cek palindrom
    fmt.Print("Palindrom? ")
    fmt.Println(palindrom(tab, m))
}

```

Screenshot Output

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Alpro sem 2> go run "c:\Alpro sem 2\minggu7\unguided4.go"
Teks: SENANG
Reverse teks: GNANES
Palindrom? false
PS C:\Alpro sem 2> go run "c:\Alpro sem 2\minggu7\unguided4.go"
Teks: KATAK
Reverse teks: KATAK
Palindrom? true
PS C:\Alpro sem 2> 

```

Deskripsi Program :

Program di atas adalah program Go yang menganalisis apakah teks merupakan palindrom (dibaca sama dari depan dan belakang). Menggunakan array bertipe rune dengan kapasitas maksimal 127 karakter, program memiliki beberapa fungsi utama: `isiArray` untuk menerima dan menyimpan input teks (menghilangkan spasi dan tanda baca), `cetakArray` untuk menampilkan isi array, `balikan Array` untuk membalik urutan karakter, dan `palindrom` untuk memeriksa apakah teks adalah palindrom dengan membandingkan array asli dan yang dibalik. Dalam fungsi `main`, program menerima input dari pengguna, menampilkan teks terbalik, dan mengoutputkan `true` jika teks adalah palindrom atau `false` jika tidak.

IV. KESIMPULAN

Pembelajaran tentang struct dan array adalah aspek penting dalam pemrograman yang mendukung pengelolaan data secara efisien. Struct memungkinkan pengembang mengelompokkan berbagai tipe data menjadi satu kesatuan terorganisir, memudahkan representasi objek kompleks. Sementara itu, array menyimpan sekumpulan data dengan tipe yang sama dalam urutan tertentu, memungkinkan akses dan manipulasi data dengan cepat. Keduanya saling melengkapi; struct memberikan fleksibilitas dalam mendefinisikan tipe data kompleks, sedangkan array membantu mengelola koleksi data secara sistematis. Dengan memahami kedua konsep ini, programmer dapat merancang program yang lebih terstruktur dan efisien serta menyelesaikan masalah dengan lebih baik.

V. REFERENSI

- [1] Modul 7 Praktikum Algoritma 2
- [2] A. A. A. Donovan and B. W. Kernighan, *The Go Programming Language*. Boston, MA: Addison-Wesley, 2015.