

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL VII  
STRUCK & ARRAY**



Oleh:

**RIZKULLOH ALPRIYANSAH**

**2311102142**

**IF-11-08**

**PROGRAM STUDI S1 INFORMATIKA  
FAKULTAS INFORMATIKA  
UNIVERSITAS TELKOM PURWOKERTO  
2024**

## I. DASAR TEORI

### 1. Array

Array adalah struktur data yang menyimpan kumpulan elemen dengan tipe data yang sama, di mana setiap elemen dapat diakses melalui indeks. Array dalam Go memiliki ukuran tetap yang didefinisikan saat deklarasi.

Karakteristik Array di Go:

- Tipe Data Homogen: Semua elemen array harus memiliki tipe data yang sama.
- Ukuran Tetap: Ukuran array ditentukan saat deklarasi dan tidak dapat diubah.
- Indeks Dimulai dari Nol: Elemen pertama berada di indeks 0, dan elemen terakhir berada di indeks panjang array - 1.
- Tipe Data: Array di Go termasuk tipe data statik, sehingga ukuran dan tipe elemen array menjadi bagian dari definisinya.

Deklarasi Array:

Deklarasi array dapat dilakukan dengan cara berikut:

```
var namaArray [n]tipeData
```

Contoh

```
var angka [5]int // Array integer dengan 5 elemen
```

Keunggulan Array:

- Struktur data sederhana untuk pengelolaan data homogen.
- Waktu akses cepat karena elemen disimpan secara berurutan di memori.

Keterbatasan Array:

- Ukuran tetap sehingga kurang fleksibel untuk data dinamis.
- Tidak memiliki metode bawaan untuk manipulasi seperti *append* atau *delete* (gunakan slice untuk fleksibilitas).

### 2. Struct

Struct (struktur) adalah tipe data komposit yang digunakan untuk mengelompokkan beberapa elemen yang memiliki tipe data berbeda. Struct digunakan untuk merepresentasikan objek atau entitas dengan berbagai atribut.

Karakteristik Struct di Go:

Tipe Data Komposit: Struct dapat menggabungkan beberapa tipe data menjadi satu kesatuan.

Kustomisasi: Struct memungkinkan pengguna untuk mendefinisikan tipe data baru yang sesuai dengan kebutuhan aplikasi.

Statik: Struktur dan atributnya didefinisikan secara statis dan tidak dapat diubah setelah deklarasi.

Deklarasi Struct:

Struct dideklarasikan dengan kata kunci `type` diikuti dengan nama dan definisi atribut.

```
type Person struct {  
    Name string  
    Age  int  
    Email string  
}
```

Contoh

```
// Menggunakan nilai default  
var p1 Person  
p1.Name = "John"  
p1.Age = 25  
p1.Email = "john@example.com"  
  
// Menggunakan literal  
p2 := Person{Name: "Alice", Age: 30, Email: "alice@example.com"}
```

Keunggulan Struct:

- Membantu mengelompokkan data yang kompleks.
- Mudah diperluas dengan menambahkan atribut atau method.
- Cocok untuk membuat tipe data yang merepresentasikan objek nyata.

Keterbatasan Struct:

- Tidak mendukung pewarisan langsung (inheritance) seperti bahasa OOP lainnya, meskipun dapat diatasi dengan embedding.

Perbandingan Array dan Struct di Go:

Kriteria	Array	Struct
Tipe Data	Homogen (hanya tipe data yang sama)	Heterogen (tipe data berbeda-beda)
Fungsi	Menyimpan data dengan ukuran tetap	Merepresentasikan objek dengan atribut
Fleksibilitas	Ukuran tetap	Atribut dapat disesuaikan
Penerapan	Operasi data homogen	Representasi entitas yang kompleks

## I. GUIDED

### Guided1

#### Source Code

```
//Guided 1 - Alias (Type)
package main

import "fmt"

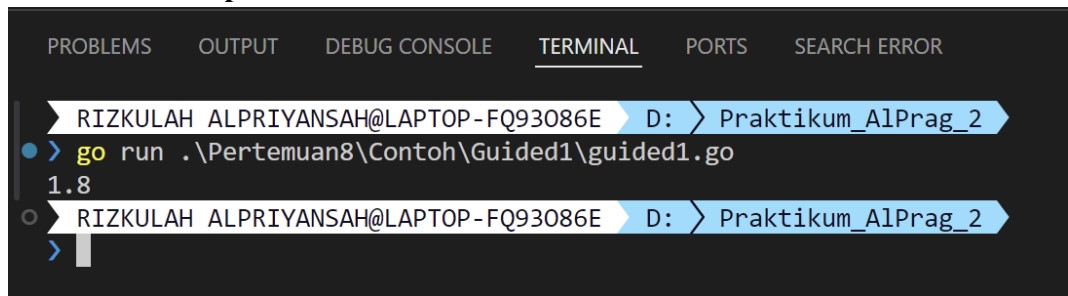
type bilangan int
type pecahan float64

func main() {
    var a, b bilangan
    var hasil pecahan

    a = 9
    b = 5

    hasil = pecahan(a) / pecahan(b)
    fmt.Println(hasil)
}
```

#### ScreenShot Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR
RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2
> go run .\Pertemuan8\Contoh\Guided1\guided1.go
1.8
RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2
>
```

#### Deskripsi

Program di atas mendemonstrasikan penggunaan tipe data buatan (alias) dalam bahasa Go. Tipe bilangan didefinisikan sebagai alias untuk tipe data int, sedangkan pecahan sebagai alias untuk float64. Dalam fungsi main, dua variabel bertipe bilangan dideklarasikan dan diinisialisasi dengan nilai 9 dan 5. Nilai-nilai tersebut kemudian dikonversi ke tipe pecahan untuk melakukan operasi pembagian, yang hasilnya disimpan dalam variabel bertipe pecahan dan ditampilkan menggunakan fungsi `fmt.Println`.

## Guided2

### Source Code

```
//Guided 2 - Slice
package main

import (
    "fmt"
)

// Fungsi untuk mengecek apakah nama sudah ada di dalam slice
func sudahAda(daftarTeman []string, nama string) bool {
    for _, teman := range daftarTeman {
        if teman == nama {
            return true
        }
    }
    return false
}

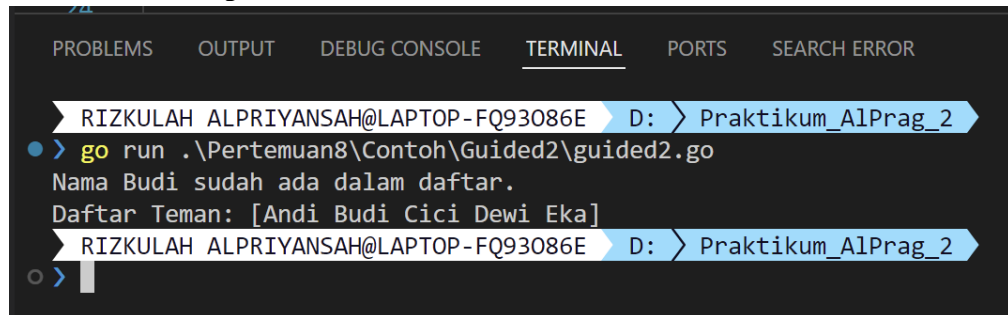
func main() {
    // Slice awal untuk daftar teman dengan beberapa data
    daftarTeman := []string{"Andi", "Budi", "Cici"}

    // Nama-nama baru yang ingin ditambahkan
    namaBaru := []string{"Dewi", "Budi", "Eka"}

    // Menambahkan nama baru hanya jika belum ada di daftar
    for _, nama := range namaBaru {
        if !sudahAda(daftarTeman, nama) {
            daftarTeman = append(daftarTeman, nama)
        } else {
            fmt.Println("Nama", nama, "sudah ada dalam daftar.")
        }
    }

    // Menampilkan daftar teman akhir
    fmt.Println("Daftar Teman:", daftarTeman)
}
```

## ScreenShot Output



```
RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2
> go run .\Pertemuan8\Contoh\Guided2\guided2.go
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2
> 
```

## Deskripsi

Program di atas menggunakan konsep slice dalam bahasa Go untuk mengelola daftar teman. Slice awal berisi daftar nama, sementara ada array lain berisi nama-nama baru yang ingin ditambahkan. Sebuah fungsi bernama `sudahAda` dibuat untuk memeriksa apakah suatu nama sudah ada dalam slice. Pada bagian `main`, nama-nama baru akan ditambahkan ke slice hanya jika belum ada di dalam daftar. Jika nama sudah ada, program akan mencetak pemberitahuan. Hasil akhirnya adalah daftar teman yang diperbarui ditampilkan menggunakan fungsi `fmt.Println`.

## Guided3

### Source Code

```
//Guided 3 - Map
package main

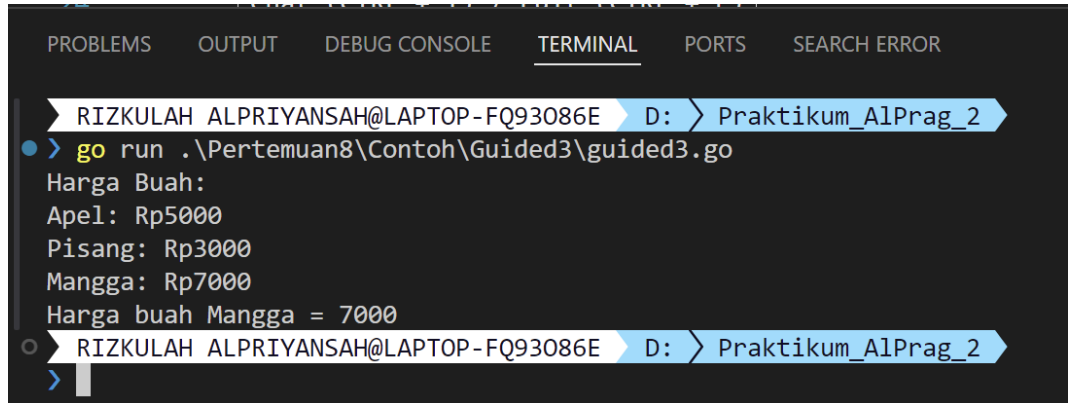
import (
    "fmt"
)

func main() {
    // Membuat map dengan nama buah sebagai kunci dan harga sebagai nilai
    hargaBuah := map[string]int{
        "Apel": 5000,
        "Pisang": 3000,
        "Mangga": 7000,
    }

    // Menampilkan harga dari setiap buah
    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

A screenshot of a terminal window with a dark background. The terminal shows the execution of a Go program. The prompt is 'RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: >'. The command entered is 'go run .\Pertemuan8\Contoh\Guided3\guided3.go'. The output of the program is: 'Harga Buah:', 'Apel: Rp5000', 'Pisang: Rp3000', 'Mangga: Rp7000', and 'Harga buah Mangga = 7000'. The terminal window has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), 'PORTS', and 'SEARCH ERROR'.

### Deskripsi

Program di atas mendemonstrasikan penggunaan map dalam bahasa Go untuk menyimpan pasangan data berupa nama buah sebagai kunci (key) dan harga sebagai nilai (value). Map hargaBuah diinisialisasi dengan beberapa data. Dalam fungsi main, program mencetak daftar harga buah dengan menggunakan loop for range, yang iterasi melalui setiap pasangan kunci dan nilai. Selain itu, harga buah tertentu seperti Mangga dapat diakses langsung dengan menyebutkan kuncinya. Hasilnya ditampilkan menggunakan fungsi `fmt.Printf` dan `fmt.Print`.

## II. UNGUIDED

### Unguided1

#### Source Code

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x_142, y_142 int
}

type Lingkaran struct {
    titikPusat_142 Titik
    radius_142     int
}

func jarak_142(a_142, b_142 Titik) float64 {
    return math.Sqrt(float64((a_142.x_142-b_142.x_142)*(a_142.x_142-
b_142.x_142) + (a_142.y_142-b_142.y_142)*(a_142.y_142-b_142.y_142)))
}

func dalamLingkaran_142(t_142 Titik, l_142 Lingkaran) bool {
    return jarak_142(t_142, l_142.titikPusat_142) <=
float64(l_142.radius_142)
}

func main() {
    var cx1_142, cy1_142, r1_142 int
    var cx2_142, cy2_142, r2_142 int
    var px_142, py_142 int

    fmt.Println("Koordinat titik pusat & radius lingkaran 1 (cx cy r):")
    fmt.Scan(&cx1_142, &cy1_142, &r1_142)

    fmt.Println("Koordinat titik pusat & radius lingkaran 2 (cx cy r):")
    fmt.Scan(&cx2_142, &cy2_142, &r2_142)

    fmt.Println("Koordinat titik sembarang (x y):")
    fmt.Scan(&px_142, &py_142)

    lingkaran1_142 := Lingkaran{Titik{cx1_142, cy1_142}, r1_142}
```



```

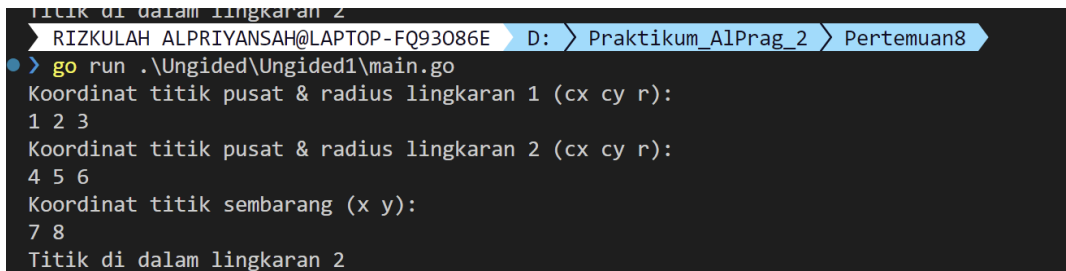
lingkaran2_142 := Lingkaran{Titik{cx2_142, cy2_142}, r2_142}
titik_142 := Titik{px_142, py_142}

inLingkaran1_142 := dalamLingkaran_142(titik_142, lingkaran1_142)
inLingkaran2_142 := dalamLingkaran_142(titik_142, lingkaran2_142)

if inLingkaran1_142 && inLingkaran2_142 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if inLingkaran1_142 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if inLingkaran2_142 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

## Screenshot



```

Titik di dalam lingkaran 2
RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2 > Pertemuan8
> go run .\Ungided\Ungided1\main.go
Koordinat titik pusat & radius lingkaran 1 (cx cy r):
1 2 3
Koordinat titik pusat & radius lingkaran 2 (cx cy r):
4 5 6
Koordinat titik sembarang (x y):
7 8
Titik di dalam lingkaran 2

```

## Deskripsi

Program di atas menggunakan tipe data struct dalam bahasa Go untuk merepresentasikan koordinat titik dan lingkaran. Struct Titik menyimpan koordinat x dan y, sedangkan struct Lingkaran menyimpan titik pusat dan radius lingkaran. Fungsi jarak\_142 menghitung jarak antara dua titik, sedangkan fungsi dalamLingkaran\_142 memeriksa apakah suatu titik berada di dalam lingkaran berdasarkan jaraknya dari titik pusat. Program membaca input untuk dua lingkaran dan satu titik sembarang, kemudian menentukan apakah titik tersebut berada di dalam salah satu atau kedua lingkaran. Hasilnya ditampilkan melalui konsol.

## Unguided2

### Source Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n_142 int
    fmt.Print("Masukkan jumlah elemen array: ")
    fmt.Scan(&n_142)

    array_142 := make([]int, n_142)
    fmt.Println("Masukkan elemen array:")
    for i_142 := 0; i_142 < n_142; i_142++ {
        fmt.Printf("Elemen ke-%d: ", i_142)
        fmt.Scan(&array_142[i_142])
    }

    for {
        fmt.Println("\nMenu:")
        fmt.Println("1. Tampilkan keseluruhan isi array")
        fmt.Println("2. Tampilkan elemen dengan indeks ganjil")
        fmt.Println("3. Tampilkan elemen dengan indeks genap")
        fmt.Println("4. Tampilkan elemen dengan indeks kelipatan x")
        fmt.Println("5. Hapus elemen pada indeks tertentu")
        fmt.Println("6. Hitung rata-rata elemen array")
        fmt.Println("7. Hitung simpangan baku elemen array")
        fmt.Println("8. Hitung frekuensi bilangan tertentu")
        fmt.Println("9. Keluar")
        fmt.Print("Pilih menu: ")

        var pilihan_142 int
        fmt.Scan(&pilihan_142)

        switch pilihan_142 {
        case 1:
            fmt.Println("Isi array:")
            for _, val_142 := range array_142 {
                fmt.Print(val_142, " ")
            }
            fmt.Println()
        }
```

case 2:

```
fmt.Println("Elemen dengan indeks ganjil:")
for i_142 := 1; i_142 < len(array_142); i_142 += 2 {
    fmt.Print(array_142[i_142], " ")
}
fmt.Println()
```

case 3:

```
fmt.Println("Elemen dengan indeks genap:")
for i_142 := 0; i_142 < len(array_142); i_142 += 2 {
    fmt.Print(array_142[i_142], " ")
}
fmt.Println()
```

case 4:

```
var x_142 int
fmt.Print("Masukkan nilai x: ")
fmt.Scan(&x_142)
fmt.Print("Elemen dengan indeks kelipatan ", x_142, ": ")
if x_142 > 0 {
    for i_142 := x_142; i_142 < len(array_142); i_142 += x_142 {
        fmt.Print(array_142[i_142], " ")
    }
} else {
    fmt.Println("Nilai x harus lebih besar dari 0.")
}
fmt.Println()
```

case 5:

```
var idx_142 int
fmt.Print("Masukkan indeks yang ingin dihapus: ")
fmt.Scan(&idx_142)
if idx_142 >= 0 && idx_142 < len(array_142) {
    array_142 = append(array_142[idx_142:], array_142[idx_142+1:]...)
    fmt.Println("Array setelah penghapusan:")
    for _, val_142 := range array_142 {
        fmt.Print(val_142, " ")
    }
    fmt.Println()
} else {
    fmt.Println("Indeks tidak valid!")
}
```

case 6:

```
sum_142 := 0
```

```

        for _, val_142 := range array_142 {
            sum_142 += val_142
        }
        rataRata_142 := float64(sum_142) / float64(len(array_142))
        fmt.Printf("Rata-rata elemen array: %.2f\n", rataRata_142)

    case 7:
        sum_142 := 0
        for _, val_142 := range array_142 {
            sum_142 += val_142
        }
        mean_142 := float64(sum_142) / float64(len(array_142))

        var variance_142 float64
        for _, val_142 := range array_142 {
            variance_142 += math.Pow(float64(val_142)-mean_142, 2)
        }
        variance_142 /= float64(len(array_142))
        stdDev_142 := math.Sqrt(variance_142)
        fmt.Printf("Simpangan baku elemen array: %.2f\n", stdDev_142)

    case 8:
        var target_142 int
        fmt.Print("Masukkan bilangan yang ingin dihitung frekuensinya: ")
        fmt.Scan(&target_142)
        count_142 := 0
        for _, val_142 := range array_142 {
            if val_142 == target_142 {
                count_142++
            }
        }
        fmt.Printf("Frekuensi bilangan %d: %d kali\n", target_142, count_142)

    case 9:
        // Keluar dari program
        fmt.Println("Keluar dari program.")
        return

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

```

## Screenshot

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2 > Pertemuan8 > Ungided
RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2 > Pertemuan8 > Ungided
> go run .\Unguided2\main.go
Masukkan jumlah elemen array: 6
Masukkan elemen array:
Elemen ke-0: 2
Elemen ke-1: 7
Elemen ke-2: 9
Elemen ke-3: 6
Elemen ke-4: 4
Elemen ke-5: 0

Menu:
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata elemen array
7. Hitung simpangan baku elemen array
8. Hitung frekuensi bilangan tertentu
9. Keluar
Pilih menu: 1
Isi array:
2 7 9 6 4 0

Menu:
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata elemen array
7. Hitung simpangan baku elemen array
8. Hitung frekuensi bilangan tertentu
9. Keluar
Pilih menu: 2
Elemen dengan indeks ganjil:
7 6 0
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

```
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata elemen array
7. Hitung simpangan baku elemen array
8. Hitung frekuensi bilangan tertentu
9. Keluar
```

Pilih menu: 3

Elemen dengan indeks genap:

2 9 4

Menu:

```
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata elemen array
7. Hitung simpangan baku elemen array
8. Hitung frekuensi bilangan tertentu
9. Keluar
```

Pilih menu: 4

Masukkan nilai x: 2

Elemen dengan indeks kelipatan 2: 9 4

Menu:

```
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata elemen array
7. Hitung simpangan baku elemen array
8. Hitung frekuensi bilangan tertentu
9. Keluar
```

Pilih menu: 5

Masukkan indeks yang ingin dihapus: 0

Array setelah penghapusan:

7 9 6 4 0

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

```
Menu:
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata elemen array
7. Hitung simpangan baku elemen array
8. Hitung frekuensi bilangan tertentu
9. Keluar
Pilih menu: 1
Isi array:
7 9 6 4 0
```

```
Menu:
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata elemen array
7. Hitung simpangan baku elemen array
8. Hitung frekuensi bilangan tertentu
9. Keluar
Pilih menu: 6
Rata-rata elemen array: 5.20
```

```
Menu:
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata elemen array
7. Hitung simpangan baku elemen array
8. Hitung frekuensi bilangan tertentu
9. Keluar
Pilih menu: 7
Simpangan baku elemen array: 3.06
```

```
Menu:
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata elemen array
7. Hitung simpangan baku elemen array
8. Hitung frekuensi bilangan tertentu
9. Keluar
Pilih menu: 8
Masukkan bilangan yang ingin dihitung frekuensinya: 0
Frekuensi bilangan 0: 1 kali
```

```
Menu:
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata elemen array
7. Hitung simpangan baku elemen array
8. Hitung frekuensi bilangan tertentu
9. Keluar
Pilih menu: 9
Keluar dari program.
```

RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum\_AIPrag 2 > Pertemuan8 > Ungided

## Deskripsi

Program ini adalah aplikasi interaktif dalam bahasa Go (Golang) untuk mengelola array dengan berbagai operasi. Pengguna dapat memasukkan elemen array dan memilih menu untuk menampilkan seluruh isi array, elemen berdasarkan indeks ganjil, genap, atau kelipatan tertentu, serta menghapus elemen pada indeks tertentu. Selain itu, program juga dapat menghitung rata-rata nilai elemen, simpangan baku, dan frekuensi kemunculan suatu bilangan dalam array. Dengan loop tak terbatas, program akan terus berjalan hingga pengguna memilih opsi untuk keluar. Semua operasi dilakukan secara dinamis berdasarkan input dari pengguna.

## Unguided3

### Source Code

```
package main

import (
    "fmt"
)

func main() {
    var klubA_142, klubB_142 string
    var skorA_142, skorB_142 int
    var hasil_142 []string

    fmt.Print("Klub A: ")
    fmt.Scanln(&klubA_142)
    fmt.Print("Klub B: ")
    fmt.Scanln(&klubB_142)

    pertandingan_142 := 1
    for {
        fmt.Printf("Pertandingan %d : ", pertandingan_142)
        fmt.Scan(&skorA_142, &skorB_142)

        if skorA_142 < 0 || skorB_142 < 0 {
            break
        }

        if skorA_142 > skorB_142 {
```



```

        hasil_142 = append(hasil_142, fmt.Sprintf("Hasil %d: %s",
pertandingan_142, klubA_142))
    } else if skorB_142 > skorA_142 {
        hasil_142 = append(hasil_142, fmt.Sprintf("Hasil %d: %s",
pertandingan_142, klubB_142))
    } else {
        hasil_142 = append(hasil_142, fmt.Sprintf("Hasil %d: Draw",
pertandingan_142))
    }

    pertandingan_142++
}

fmt.Println("\nDaftar Hasil Pertandingan:")
for _, h_142 := range hasil_142 {
    fmt.Println(h_142)
}

fmt.Println("Pertandingan selesai")
}

```

## Screenshot

```

RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2 > Pertemuan8 > Ungided > Unguided3
> go run .\main.go
Klub A: Indonesia
Klub B: Korea
Pertandingan 1 : 6 3
Pertandingan 2 : 2 4
Pertandingan 3 : 3 3
Pertandingan 4 : 3 4
Pertandingan 5 : 4 2
Pertandingan 6 : 4 3
Pertandingan 7 : 2 5
Pertandingan 8 : -0 7
Pertandingan 9 : -1 7

Daftar Hasil Pertandingan:
Hasil 1: Indonesia
Hasil 2: Korea
Hasil 3: Draw
Hasil 4: Korea
Hasil 5: Indonesia
Hasil 6: Indonesia
Hasil 7: Korea
Hasil 8: Korea
Pertandingan selesai
RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2 > Pertemuan8 > Ungided > Unguided3

```

## Deskripsi

Program ini adalah aplikasi interaktif sederhana dalam bahasa Go (Golang) untuk mencatat hasil pertandingan antara dua klub sepak bola. Pengguna memasukkan nama kedua klub, lalu menginput skor untuk setiap pertandingan. Program akan mencatat hasil setiap pertandingan dengan menentukan pemenang berdasarkan skor, atau mencatat "Draw" jika skornya sama. Input akan berhenti jika skor yang dimasukkan negatif. Setelah itu, program menampilkan daftar hasil semua pertandingan yang telah dimainkan dan mengakhiri program dengan pesan penutup.

## Unguided4

### Source Code

```
package main

import (
    "fmt"
)

const NMAX_142 = 127

type tabel_142 [NMAX_142]rune

func isiArray_142(t_142 *tabel_142, n_142 *int) {
    *n_142 = 0
    fmt.Println("Masukkan teks (akhiri dengan '.'): ")
    for {
        var c_142 rune
        fmt.Scanf("%c", &c_142)
        if c_142 == '.' {
            break
        }
        t_142[*n_142] = c_142
        (*n_142)++
    }
}

func cetakArray_142(t_142 tabel_142, n_142 int) {
    for i_142 := 0; i_142 < n_142; i_142++ {
        fmt.Printf("%c", t_142[i_142])
    }
    fmt.Println()
}
```

```

}

func balikkanArray_142(t_142 *tabel_142, n_142 int) {
    for i_142, j_142 := 0, n_142-1; i_142 < j_142; i_142, j_142 = i_142+1,
j_142-1 {
        t_142[i_142], t_142[j_142] = t_142[j_142], t_142[i_142]
    }
}

func palindrom_142(t_142 tabel_142, n_142 int) bool {
    for i_142 := 0; i_142 < n_142/2; i_142++ {
        if t_142[i_142] != t_142[n_142-i_142-1] {
            return false
        }
    }
    return true
}

func main() {
    var tab_142 tabel_142
    var n_142 int

    isiArray_142(&tab_142, &n_142)

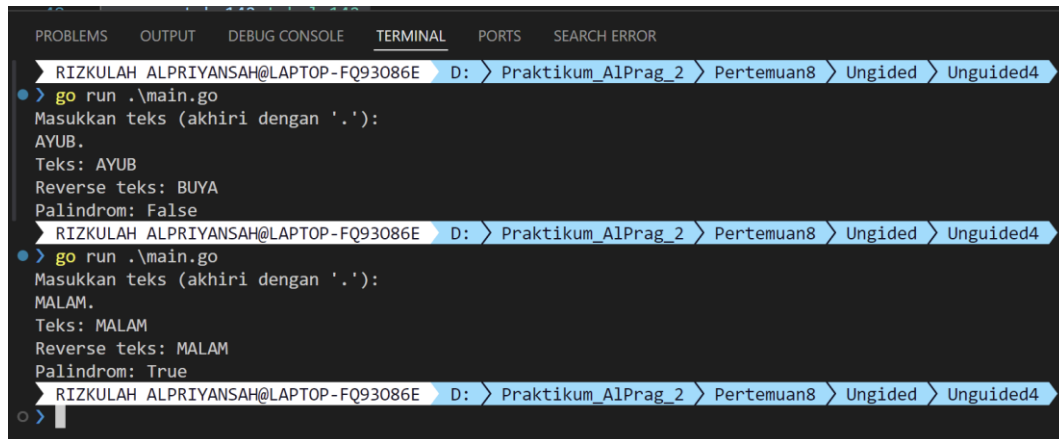
    fmt.Print("Teks: ")
    cetakArray_142(tab_142, n_142)

    balikkanArray_142(&tab_142, n_142)
    fmt.Print("Reverse teks: ")
    cetakArray_142(tab_142, n_142)

    if palindrom_142(tab_142, n_142) {
        fmt.Println("Palindrom: True")
    } else {
        fmt.Println("Palindrom: False")
    }
}

```

## Screenshot



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2 > Pertemuan8 > Ungided > Unguided4
> go run .\main.go
Masukkan teks (akhiri dengan '.'):
AYUB.
Teks: AYUB
Reverse teks: BUYA
Palindrom: False
RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2 > Pertemuan8 > Ungided > Unguided4
> go run .\main.go
Masukkan teks (akhiri dengan '.'):
MALAM.
Teks: MALAM
Reverse teks: MALAM
Palindrom: True
RIZKULAH ALPRIYANSAH@LAPTOP-FQ93086E D: > Praktikum_AlPrag_2 > Pertemuan8 > Ungided > Unguided4
>
```

## Deskripsi

Program ini adalah aplikasi dalam bahasa Go (Golang) yang memproses teks berbasis array karakter hingga 127 karakter. Pengguna dapat memasukkan teks secara bertahap, diakhiri dengan karakter `.`. Setelah itu, program mencetak teks asli, membalik urutannya, dan memeriksa apakah teks tersebut merupakan palindrom (teks yang terbaca sama dari depan maupun belakang). Operasi dilakukan menggunakan array statis dan algoritma sederhana untuk pembalikan teks serta deteksi palindrom dengan membandingkan karakter di awal dan akhir array secara berpasangan.

### **III. DAFTAR PUSTAKA**

- 1) Asisten praktikum, Akmelia Zahara dan Kyla Azzahra Kinan “Modul VII STRUCK & ARRAY” Learning Management System, 2024