

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK  
MODUL VII  
STRUCK & ARRAY**



Oleh:

OKTAVANIA AYU RAHMADANTY

2311102240

S1IF-11-02

**S1 TEKNIK INFORMATIKA  
UNIVERSITAS TELKOM PURWOKERTO**

**2024**

## I. DASAR TEORI

### I.1 Tipe Bentuk

Tipe bentuk memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipe bentuk ini dapat dibedakan atas dua jenis, yaitu Alias dan Struct.

#### 1) Alias (Type)

Bahasa pemrograman pada umumnya mengizinkan pemrograman untuk mengubah nama suatu tipe data dengan nama baru yang lebih ringkas dan familiar. Sebagai contoh "Integer" dapat dirubah dengan nama alias "bilangan ". Caranya dengan menggunakan kata kunci "type".

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama alias> <tipe data>	
3		type <nama alias> <tipe data>
4	algoritma	
5	...	func main(){
6		...
7		}

Sebagai contoh perhatikan program Go berikut beserta hasil eksekusinya!

```
1 package main
2 import "fmt"
3 type bilangan int
4 type pecahan float64
5 func main(){
6     var a,b bilangan
7     var hasil pecahan
8     a = 9
9     b = 5
10    hasil = pecahan(a) / pecahan(b)
11    fmt.Println(hasil)
12 }
```

```
E:\DEV\GO>go build Demo.go
E:\DEV\GO> Demo.exe
1.8
```

#### 2) Struct atau Record

Structure memungkinkan program untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari structure tersebut.

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama struct> <	type <nama struct> struct {
3	<field 1> <type data>	<field 1> <type data>
4	<field 2> <type data>	<field 2> <type data>
5	<field 3> <type data>	<field 3> <type data>
6	>	}
7		

Berbeda dengan Bahasa pemrograman lain, kesamaan tipe dari dua variable berjenis structure bukan karena namanya tetapi karena strukturnya. Dua variable dengan nama-nama field dan tipe field yang sama (dan dalam urutan yang sama) dianggap mempunyai tipe yang sama. Tentunya akan lebih memudahkan jika structure tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi structure tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

```

1 package main
2 import "fmt"
3 type waktu struct {
4     jam, menit, detik int
5 }
6
7 func main(){
8     var wParkir, wPulang, durasi waktu
9     var dParkir, dPulang, lParkir int
10    fmt.Scan(&wParkir.jam, &wParkir.menit, &wParkir.detik)
11    fmt.Scan(&wPulang.jam, &wPulang.menit, &wPulang.detik)
12    dParkir = wParkir.detik + wParkir.menit*60 + wParkir.jam*3600
13    dPulang = wPulang.detik + wPulang.menit*60 + wPulang.jam*3600
14    lParkir = dPulang - dParkir
15    durasi.jam = lParkir / 3600
16    durasi.menit = lParkir % 3600 / 60
17    durasi.detik = lParkir % 3600 % 60
18    fmt.Printf("Lama parkir: %d jam %d menit %d detik",
19        durasi.jam, durasi.menit, durasi.detik)
20 }

```

```

E:\DEV\GO>go build Demo.go
E:\DEV\GO> Demo.exe
7 30 0
10 45 15
Lama parkir: 3 jam 15 menit 15 detik

```

## I.2 Array

Array mempunyai ukuran (jumlah elemen) yang tetap (statis) selama eksekusi program, sehingga jumlah elemen array menjadi bagian dari deklarasi variable dengan tipe array.

	Notasi dalam bahasa Go
1	var (
2	// array arr mempunyai 73 elemen, masing-masing bertipe CircType2
3	arr [73]CircType
4	
5	// array buf dengan 5 elemen, dengan nilai awal 7, 3, 5, 2, dan 11.
6	buf = [5]byte{7, 3, 5, 2, 11}
7	
8	// mhs adalah array dengan 2000 elemen bertipe NewType
9	mhs [2000]NewType
10	
11	// rec adalah array dari array, yaitu matriks, atau array berdimensi-2
12	rec [20][40]float64
13	)

Jumlah elemen array dapat diminta dengan fungsi len yang tersedia. Sebagai contoh len(arr) akan menghasilkan 73 untuk contoh di atas.

Indeks array dimulai dari 0, sehingga indeks arr pada contoh adalah 0,1..len(arr)-1

Conroh:

```
1 // Mengganti isi elemen ke-0 dengan nilai dari elemen ke-7
2 arr[0] = arr[7]
3
4 // Mengambil data field x dari elemen ke-i
5 currX = arr[i].center.x
6
7 // Mengambil elemen terakhir
8 n := len(arr)
9 buf := arr[n-1]
```

### Slice (Array dinamik)

Array dalam Go juga dapat mempunyai ukuran yang dinamik. (tidak digunakan di keas algoritma pemograman). Deklarasi mirip dengan deklarasi array, tetapi jumlah elemennya dikosongkan.

```

1 // declaring chop as an empty slice of float64
2 var chop []float64
3
4 // declaring sl01 as a slice
5 var sl01 = []int{ 11, 2, 3, 5, 7, 13 }

```

Sebuah slice dapat diprealokasi menggunakan fungsi built-in `make`.

```

1 // Prealokasi 10 elemen untuk sl02 dan sejumlah tempat tambahan
2 var sl02 []int = make([]int, 10, 20)
3
4 // Prealokasi 7 elemen untuk sl03 tanpa tempat tambahan
5 var sl03 []circType = make([]circType, 7)

```

Fungsi built-in `len` dapat digunakan untuk mengetahui ukuran slice. Fungsi lain, `cap`, dapat digunakan untuk mengetahui total tempat yang disediakan untuk slice tersebut.

```

1 // Cetak jumlah elemen dan tempat yang tersedia untuk sl02
2 fmt.Println( len(sl02), cap(sl02) )

```

Fungsi built-in `append` dapat digunakan untuk menambahkan elemen ke suatu slice, dan bila perlu memperbesar tempat untuk slice tersebut.

```

1 /* Append elemen baru, membuat slice baru, dan menyimpan kembali slice baru
   ke variabel semula. Boleh juga disimpan ke variabel lain, sehingga variabel
   semula masih menyimpan slice yang asli. */
2 sl01 = append(sl01, 17)
3 sl01 = append(sl01, 19, 23)

```

Sebuah slice baru juga dapat terbentuk dengan mengambil slice dari suatu array atau slice yang lain.

```

1 // Ambil 3 elemen pertama dari suatu slice atau array
2 sl04 = arr[:4]
3
4 // Ambil beberapa elemen terakhir, dimulai dari indeks 5
5 sl05 = sl01[5:]
6
7 // Salin semua dari slice/array aslinya
8 sl06 = sl05[:]
9
10 // Salin element dari indeks 3 sampai, tapi tidak termasuk, 5.
11 // Jadi dalam contoh hanya 2 elemen sl06[3] dan sl06[4] yang disalin
12 sl07 = sl06[3:5]

```

## Map

Tipe array lain, sebuah array dinamik. Indeksnya (di sini disebut kunci) tidak harus berbentuk integer. Indeks dapat berasal dari tipe apa saja. Struktur ini disebut map.

```
1 // Deklarasi variabel dct sebagai map bilangan bulat dengan kunci string
2 var dct map[string]int
3
4 // Deklarasi map lain dct1 dari elemen string dengan kunci juga string
5 // Mempunyai nilai awal dct1["john"] = "hi", dct1["anne"] = "darling"
6 var dct1 = map[string]string{ "john":"hi", "anne":"darling" }
7
8 // Deklarasi dan prealokasi tempat untuk map dct2
9 var dct2 map[float64]int = make(map[float64]int, 10)
10
11 // Mengambil nilai yang tersimpan dengan kunci "john"
12 fmt.Println( dct1["john"] )
13
14 // Mengganti nilai yang tersimpan pada kunci "anne", dan
15 // Membuat entri baru dengan kunci "boy"
16 dct1["anne"] = "lovely"
17 dct1["boy"] = "runaround"
18
19 // Menghapus entri dengan kunci "john"
20 delete(dct1, "john")
```

## II. GUIDED

### Guided 1

#### Source Code

```
package main

import (
    "fmt"
    "sort"
)

//Struktur untuk menampung data mahasiswa
type Mahasiswa struct {
    Nama      string
    Matematika int
    Fisika     int
    Kimia      int
    RataRata   float64
}

func hitungRataRata(m *Mahasiswa) {
    total := m.Matematika + m.Fisika + m.Kimia
    m.RataRata = float64(total) / 3.0
}

//Fungsi utama untuk mengelola dan mengurutkan data
mahasiswa berdasarkan nilai

func main() {
```

```

//Array untuk menampung data mahasiswa
mahasiswa := []Mahasiswa{
    {"Ali", 85, 90, 80, 0},
    {"Budi", 70, 75, 80, 0},
    {"Cici", 90, 85, 95, 0},
    {"Doni", 60, 65, 70, 0},
    {"Eka", 100, 95, 90, 0},
}

//Menghitung rata-rata nilai tiap mahasiswa
for i := range mahasiswa {
    hitungRataRata(&mahasiswa[i])
}

//Mengurutkan mahasiswa berdasarkan nilai rata-
rata (descending)

sort.Slice(mahasiswa, func(i, j int) bool {
    return mahasiswa[i].RataRata >
mahasiswa[j].RataRata
}))

//Menampilkan hasil

fmt.Println("Peringkat mahasiswa berdasarkan rat-
rata nilai: ")

for i, m := range mahasiswa {
    fmt.Printf("%d. %s - Rata-rata: %.2f
(Matematika: %d, Fisika: %d, Kimia: %d)\n", i+1,
m>Nama, m.RataRata, m.Matematika, m.Fisika, m.Kimia)

```



```

    }
}

```

### Screenshot hasil program

```

PS D:\SEMESTER 3\Laprak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array> go r
yu Rahmadanty_2311102240_Modul 7_Struck & Array\guided\guided1\tempCodeRunnerFile.go"
Peringkat mahasiswa berdasarkan rat-rata nilai:
1. Eka - Rata-rata: 95.00 (Matematika: 100, Fisika: 95, Kimia: 90)
2. Cici - Rata-rata: 90.00 (Matematika: 90, Fisika: 85, Kimia: 95)
3. Ali - Rata-rata: 85.00 (Matematika: 85, Fisika: 90, Kimia: 80)
4. Budi - Rata-rata: 75.00 (Matematika: 70, Fisika: 75, Kimia: 80)
5. Doni - Rata-rata: 65.00 (Matematika: 60, Fisika: 65, Kimia: 70)
PS D:\SEMESTER 3\Laprak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array>

```

### Penjelasan:

Program di atas merupakan aplikasi dalam bahasa Go yang digunakan untuk mengelola dan mengurutkan data mahasiswa berdasarkan nilai rata-rata mereka dalam tiga mata pelajaran: Matematika, Fisika, dan Kimia. Pertama, data setiap mahasiswa, seperti nama dan nilai per mata pelajaran, dimasukkan ke dalam array bertipe struct `Mahasiswa`. Kemudian, fungsi `hitungRataRata` menghitung rata-rata nilai setiap mahasiswa dengan membagi total nilai ketiga mata pelajaran dengan 3. Setelah rata-rata nilai dihitung, data mahasiswa diurutkan secara menurun (descending) berdasarkan nilai rata-rata menggunakan fungsi `sort.Slice`. Terakhir, program mencetak daftar mahasiswa yang sudah diurutkan berdasarkan peringkat nilai rata-rata mereka, menampilkan nama, rata-rata, dan nilai tiap mata pelajaran.

## Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    // Membuat map dengan NIM sebagai kunci dan Nama
    sebagai nilai

    mahasiswa := map[string]string{

        "20231001": "Andi",

        "20231002": "Budi",

        "20231003": "Cici",

    }

    // Menambahkan data baru ke map

    mahasiswa["20231004"] = "Dedi"

    // Menampilkan seluruh isi map dalam format kolom
    dan baris

    fmt.Println("Daftar Mahasiswa:")

    fmt.Println("NIM\t\tNama")

    fmt.Println("-----")

    for nim, nama := range mahasiswa {

        fmt.Printf("%s\t%s\n", nim, nama)

    }

    // Mengakses data berdasarkan NIM
```

```

    nim := "20231002"

    fmt.Println("\nNama Mahasiswa dengan NIM", nim,
"adalah", mahasiswa[nim])

    // Menghapus data berdasarkan NIM

    delete(mahasiswa, "20231003")

    // Menampilkan isi map setelah data dihapus dalam
format kolom dan baris

    fmt.Println("\nDaftar Mahasiswa setelah dihapus:")

    fmt.Println("NIM\t\tNama")

    fmt.Println("-----")

    for nim, nama := range mahasiswa {

        fmt.Printf("%s\t%s\n", nim, nama)

    }

}

```

### Screenshot hasil program

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

20231003      Cici
20231004      Dedi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM          Nama
-----
20231001      Andi
20231002      Budi
20231004      Dedi
PS D:\SEMESTER 3\Labrak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array>

```

### Penjelasan:

Program di atas adalah sebuah aplikasi dalam bahasa Go yang menggunakan tipe data map untuk mengelola data mahasiswa dengan

Nomor Induk Mahasiswa (NIM) sebagai kunci dan Nama sebagai nilai. Pertama, program membuat map `mahasiswa` dan mengisinya dengan beberapa data NIM-Nama, lalu menambahkan satu data lagi ke map tersebut. Selanjutnya, program mencetak seluruh isi map dalam format kolom dan baris, menampilkan NIM dan Nama mahasiswa. Setelah itu, program mengakses data mahasiswa berdasarkan NIM tertentu dan menampilkan hasilnya. Kemudian, program menghapus data mahasiswa berdasarkan NIM yang diberikan dan mencetak ulang isi map untuk menampilkan data setelah penghapusan, tetap dalam format kolom dan baris.

### III. UNGUIDED

#### Unguided 1

##### Source Code

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y float64
}

type Lingkaran struct {
    pusat Titik
    radius float64
}

func jarak(p, q Titik) float64 {
    return math.Sqrt(math.Pow(p.x-q.x, 2) +
math.Pow(p.y-q.y, 2))
}

func didalam(c Lingkaran, p Titik) bool {
    return jarak(c.pusat, p) <= c.radius
}

func main() {
    var l1, l2 Lingkaran
    var p Titik

    fmt.Print("Masukkan koordinat pusat dan radius
lingkaran 1: ")
    fmt.Scan(&l1.pusat.x, &l1.pusat.y, &l1.radius)
    fmt.Print("Masukkan koordinat pusat dan radius
lingkaran 2: ")
    fmt.Scan(&l2.pusat.x, &l2.pusat.y, &l2.radius)
    fmt.Print("Masukkan koordinat titik sembarang: ")
    fmt.Scan(&p.x, &p.y)

    inL1 := didalam(l1, p)
    inL2 := didalam(l2, p)
```

```

        if inL1 && inL2 {
            fmt.Println("Titik di dalam lingkaran 1 dan
2")
        } else if inL1 {
            fmt.Println("Titik di dalam lingkaran 1")
        } else if inL2 {
            fmt.Println("Titik di dalam lingkaran 2")
        } else {
            fmt.Println("Titik di luar lingkaran 1 dan 2")
        }
    }
}

```

### Screenshot hasil program

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array> go
yu Rahmadanty_2311102240_Modul 7_Struck & Array\unguided\unguided1\unguided1.go"
Masukkan koordinat pusat dan radius lingkaran 1: 1 1 5
Masukkan koordinat pusat dan radius lingkaran 2: 8 8 4
Masukkan koordinat titik sembarang: 2 2
Titik di dalam lingkaran 1
PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array> go
yu Rahmadanty_2311102240_Modul 7_Struck & Array\unguided\unguided1\unguided1.go"
Masukkan koordinat pusat dan radius lingkaran 1: 1 2 3
Masukkan koordinat pusat dan radius lingkaran 2: 4 5 6
Masukkan koordinat titik sembarang: 7 8
Titik di dalam lingkaran 2
PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array> go
yu Rahmadanty_2311102240_Modul 7_Struck & Array\unguided\unguided1\unguided1.go"
Masukkan koordinat pusat dan radius lingkaran 1: 5 10 15
Masukkan koordinat pusat dan radius lingkaran 2: -15 4 20
Masukkan koordinat titik sembarang: 0 0
Titik di dalam lingkaran 1 dan 2
PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array>

```

### Penjelasan:

Program ini menentukan posisi titik relatif terhadap dua lingkaran yang ditentukan oleh pengguna. Fungsi jarak menghitung jarak antara dua titik menggunakan rumus jarak Euclidean. Fungsi didalam memeriksa apakah suatu titik berada dalam lingkaran tertentu dengan membandingkan jarak antara titik pusat lingkaran dan titik tersebut dengan radius lingkaran. Dalam fungsi main, program menerima input untuk dua lingkaran (koordinat pusat dan radius) serta satu titik sembarang. Berdasarkan hasil pengecekan

dengan fungsi didalam, program akan menampilkan pesan apakah titik berada dalam lingkaran pertama, kedua, keduanya, atau di luar keduanya.

## Unguided 2

### Source Code

```
package main

import (
    "fmt"
    "math"
)

func isiArray(arr *[]int, n int) {
    *arr = make([]int, n)
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i)
        fmt.Scan(&(*arr)[i])
    }
}

func tampilkanArray(arr []int) {
    fmt.Println("Isi array:", arr)
}

func tampilkanIndeksGanjil(arr []int) {
    fmt.Print("Elemen di indeks genap: ")
}
```

```

        for i := 1; i < len(arr); i += 2 {
            fmt.Print(arr[i], " ")
        }

        fmt.Println()
    }

func tampilkanIndeksGenap(arr []int) {
    fmt.Print("Elemen di indeks ganjil: ")

    for i := 0; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }

    fmt.Println()
}

func tampilkanIndeksKelipatan(arr []int, x int) {
    fmt.Printf("Elemen di indeks kelipatan %d: ", x)

    for i := x; i < len(arr); i += x {
        fmt.Print(arr[i], " ")
    }

    fmt.Println()
}

func hapusElemen(arr *[]int, indeks int) {
    *arr = append((*arr)[:indeks],
        (*arr)[indeks+1:]...)

    fmt.Println("Isi array setelah penghapusan:",
        *arr)
}

```



```

}

func hitungRataRata(arr []int) float64 {
    total := 0
    for _, v := range arr {
        total += v
    }
    return float64(total) / float64(len(arr))
}

func hitungStandarDeviasi(arr []int) float64 {
    rata := hitungRataRata(arr)
    var total float64
    for _, v := range arr {
        total += math.Pow(float64(v)-rata, 2)
    }
    return math.Sqrt(total / float64(len(arr)))
}

func hitungFrekuensi(arr []int, bilangan int) int {
    count := 0
    for _, v := range arr {
        if v == bilangan {
            count++
        }
    }
}

```

```
        return count
    }

func main() {
    var n, x, indeks, bilangan int

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

    arr := []int{}
    isiArray(&arr, n)

    fmt.Println("\nMenu:")
    fmt.Println("1. Tampilkan seluruh elemen array")
    fmt.Println("2. Tampilkan elemen dengan indeks genap")
    fmt.Println("3. Tampilkan elemen dengan indeks ganjil")
    fmt.Println("4. Tampilkan elemen dengan indeks kelipatan x")
    fmt.Println("5. Hapus elemen pada indeks tertentu")
    fmt.Println("6. Tampilkan rata-rata elemen")
    fmt.Println("7. Tampilkan standar deviasi elemen")
    fmt.Println("8. Tampilkan frekuensi dari suatu bilangan")
    fmt.Println("9. Keluar")
}
```

```
for {  
    var pilihan int  
    fmt.Print("\nPilih menu: ")  
    fmt.Scan(&pilihan)  
  
    switch pilihan {  
    case 1:  
        tampilkanArray(arr)  
    case 2:  
        tampilkanIndeksGanjil(arr)  
    case 3:  
        tampilkanIndeksGenap(arr)  
    case 4:  
        fmt.Print("Masukkan nilai x: ")  
        fmt.Scan(&x)  
        if x > 0 {  
            tampilkanIndeksKelipatan(arr, x)  
        } else {  
            fmt.Println("Nilai x harus lebih besar  
dari 0.")  
        }  
    case 5:  
        fmt.Print("Masukkan indeks yang ingin  
dihapus: ")  
        fmt.Scan(&indeks)  
        if indeks >= 0 && indeks < len(arr) {  
            hapusElemen(&arr, indeks)
```

```
        } else {  
            fmt.Println("Indeks tidak valid.")  
        }  
        case 6:  
            fmt.Printf("Rata-rata elemen: %.2f\n",  
hitungRataRata(arr))  
        case 7:  
            fmt.Printf("Standar deviasi elemen:  
%.2f\n", hitungStandarDeviasi(arr))  
        case 8:  
            fmt.Print("Masukkan bilangan untuk  
dihitung frekuensinya: ")  
            fmt.Scan(&bilangan)  
            fmt.Printf("Frekuensi %d: %d\n", bilangan,  
hitungFrekuensi(arr, bilangan))  
        case 9:  
            fmt.Println("Keluar program.")  
            return  
        default:  
            fmt.Println("Pilihan tidak valid.")  
    }  
}  
}
```

## Screenshot hasil program

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array>
yu Rahmadanty_2311102240_Modul 7_Struck & Array\unguided\unguided2\unguided2.go"
Masukkan jumlah elemen array: 9
Masukkan elemen ke-0: 1
Masukkan elemen ke-1: 2
Masukkan elemen ke-2: 3
Masukkan elemen ke-3: 4
Masukkan elemen ke-4: 5
Masukkan elemen ke-5: 6
Masukkan elemen ke-6: 7
Masukkan elemen ke-7: 8
Masukkan elemen ke-8: 9

Menu:
1. Tampilkan seluruh elemen array
2. Tampilkan elemen dengan indeks genap
3. Tampilkan elemen dengan indeks ganjil
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi dari suatu bilangan
9. Keluar

Pilih menu: 1
Isi array: [1 2 3 4 5 6 7 8 9]

Pilih menu: 2
Elemen di indeks genap: 2 4 6 8

Pilih menu: 3
Elemen di indeks ganjil: 1 3 5 7 9
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Pilih menu: 3
Elemen di indeks ganjil: 1 3 5 7 9

Pilih menu: 4
Masukkan nilai x: 3
Elemen di indeks kelipatan 3: 4 7

Pilih menu: 5
Masukkan indeks yang ingin dihapus: 7
Isi array setelah penghapusan: [1 2 3 4 5 6 7 9]

Pilih menu: 6
Rata-rata elemen: 4.62

Pilih menu: 7
Standar deviasi elemen: 2.50

Pilih menu: 8
Masukkan bilangan untuk dihitung frekuensinya: 4
Frekuensi 4: 1

Pilih menu: 9
Keluar program.
PS D:\SEMESTER 3\Laprak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array>
```

Penjelasan:

Program ini menggunakan beberapa fungsi untuk memenuhi setiap spesifikasi. Fungsi `isiArray` mengisi array berdasarkan input pengguna. Fungsi `tampilkanArray` mencetak seluruh isi array. Fungsi `tampilkanIndeksGanjil` dan `tampilkanIndeksGenap` mencetak elemen pada indeks ganjil dan genap, sedangkan `tampilkanIndeksKelipatan` mencetak elemen pada indeks kelipatan nilai yang dimasukkan pengguna. Fungsi `hapusElemen` menghapus elemen pada indeks tertentu. Fungsi `hitungRataRata` menghitung rata-rata, `hitungStandarDeviasi` menghitung standar deviasi, dan `hitungFrekuensi` menghitung jumlah kemunculan suatu bilangan dalam array. Program utama menyediakan menu interaktif bagi pengguna untuk memilih operasi yang diinginkan.

## Unguided 3

### Source Code

```
package main

import (
    "fmt"
)

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

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

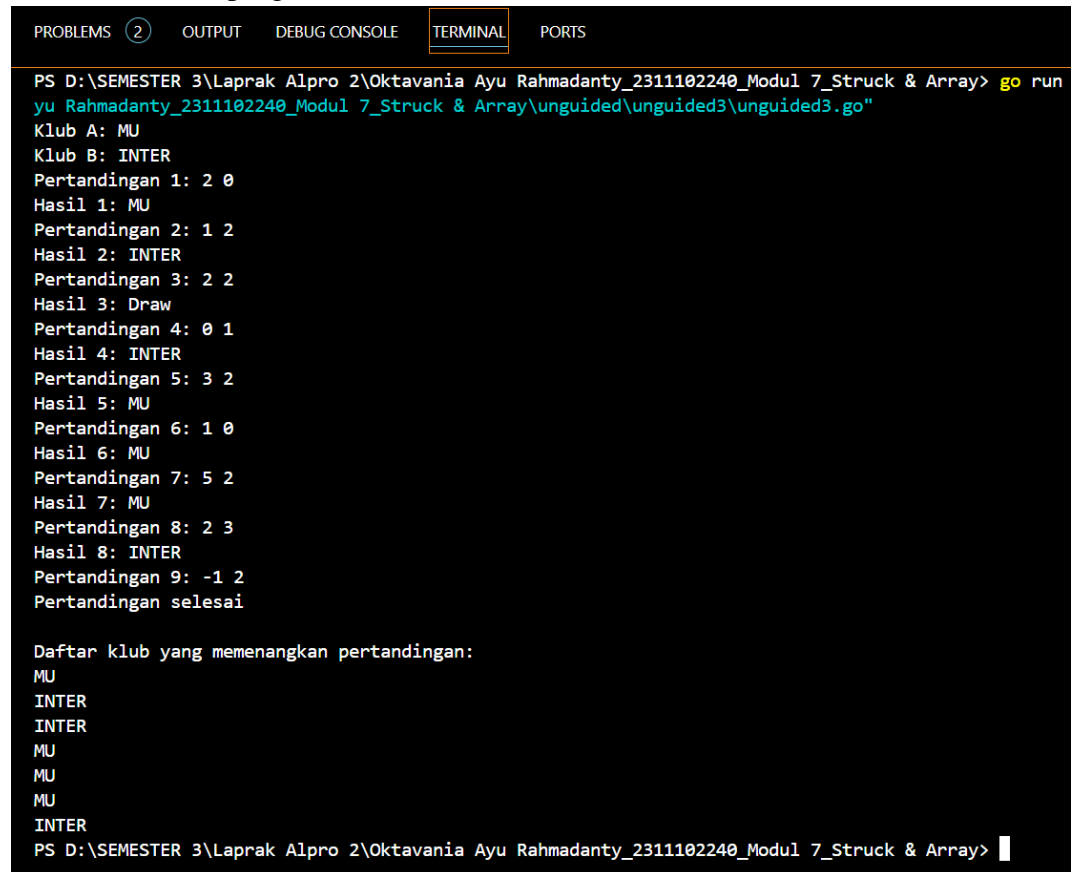
    for i := 1; ; i++ {
        fmt.Printf("Pertandingan %d: ", i)
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            fmt.Println("Pertandingan selesai")
            break
        }

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

    fmt.Println("\nDaftar klub yang memenangkan pertandingan:")
    for _, klub := range pemenang {
        fmt.Println(klub)
    }
}
```

## Screenshot hasil program



```
PROBLEMS (2) OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array> go run
yu Rahmadanty_2311102240_Modul 7_Struck & Array\unguided\unguided3\unguided3.go"
Klub A: MU
Klub B: INTER
Pertandingan 1: 2 0
Hasil 1: MU
Pertandingan 2: 1 2
Hasil 2: INTER
Pertandingan 3: 2 2
Hasil 3: Draw
Pertandingan 4: 0 1
Hasil 4: INTER
Pertandingan 5: 3 2
Hasil 5: MU
Pertandingan 6: 1 0
Hasil 6: MU
Pertandingan 7: 5 2
Hasil 7: MU
Pertandingan 8: 2 3
Hasil 8: INTER
Pertandingan 9: -1 2
Pertandingan selesai

Daftar klub yang memenangkan pertandingan:
MU
INTER
INTER
MU
MU
MU
INTER
PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array> |
```

### Penjelasan:

Program di atas adalah aplikasi dalam bahasa Go yang mencatat hasil pertandingan antara dua klub. Pertama, program meminta nama kedua klub (klubA dan klubB). Kemudian, program menjalankan loop untuk meminta input skor pertandingan bagi kedua klub. Jika salah satu skor negatif, input dihentikan, dan program mengakhiri pertandingan. Program mencatat pemenang setiap pertandingan ke dalam array pemenang, mencetak hasil setiap pertandingan (nama pemenang atau "Draw" jika skorimbang), dan menampilkan daftar klub yang memenangkan pertandingan di akhir.



## Unguided 4

### Source Code

```
package main

import (
    "fmt"
)

const NMAX int = 127

type tabel struct {
    tab [NMAX]rune
    m    int
}

func isiArray(t *tabel, n *int) {
    var input rune
    fmt.Print("Masukkan karakter (akhiri dengan '.'):")
    for {
        fmt.Scanf("%c", &input)
        if input == '.' || *n >= NMAX {
            break
        }
        t.tab[*n] = input
        *n++
    }
}
```

```

        t.m = *n
    }

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

func balikkanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        t.tab[i], t.tab[n-i-1] = t.tab[n-i-1],
t.tab[i]
    }
}

func palindrome(t tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t.tab[i] != t.tab[n-i-1] {
            return false
        }
    }
    return true
}

func main() {

```

```

var tab tabel

var n int

isiArray(&tab, &n)

fmt.Print("Teks: ")

cetakArray(tab, n)

fmt.Print("Reverse teks: ")

balikkanArray(&tab, n)

cetakArray(tab, n)

if palindrome(tab, n) {
    fmt.Println("Palindrome: true")
} else {
    fmt.Println("Palindrome: false")
}
}

```

### Screenshot hasil program

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
yu Rahmadanty_2311102240_Modul 7_Struck & Array\unguided\unguided4\unguided4.go"
Masukkan karakter (akhiri dengan '.'): KATAK.
Teks: KATAK
Reverse teks: KATAK
Palindrome: true
PS D:\SEMESTER 3\Laprak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array> go run
yu Rahmadanty_2311102240_Modul 7_Struck & Array\unguided\unguided4\unguided4.go"
Masukkan karakter (akhiri dengan '.'): SENANG.
Teks: SENANG
Reverse teks: GNANES
Palindrome: false
PS D:\SEMESTER 3\Laprak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 7_Struck & Array>

```

Penjelasan:

Program di atas menggunakan struktur tabel untuk menyimpan array karakter dan jumlah elemen. Fungsi `isiArray` mengisi array dengan karakter dari input pengguna hingga batas tertentu atau karakter titik (.). Fungsi `cetakArray` mencetak isi array, sementara `balikkanArray` membalik urutan karakter dalam array. Fungsi `palindrome` memeriksa apakah array merupakan palindrom dengan membandingkan karakter dari awal dan akhir secara berpasangan. Pada fungsi `main`, program meminta input, menampilkan array asli dan array yang sudah dibalik, serta memeriksa dan menampilkan hasil apakah array tersebut palindrom atau tidak.