

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7
STRUCT & ARRAY**



**Universitas
Telkom**

Oleh:

HENDWI SAPUTRA

2311102218

IF-11-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

I. DASAR TEORI

7.1 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

Stucture memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari stucture 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 yang lain. Kesamaan tipe dari dua variabel berjenis stucture bukan karena namanya tetapi karena strukturnya. Dua variabel dengan nama-nama field dan tipe field yang sama (dan dalam urutan yang sama) dianggap mempunyai tipe yang sama. Tentunya akan lebih memudahkan jika stucture tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi stucture 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

```

7.2 Array

Array mempunyai ukuran jumlah elemen) yang tetap (statls) selama eksekusi program, sehingga jumlah elemen array menjadi bagian dari deklarasi variabel 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

Contoh:

```

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 kelas Algoritma Pemrograman). Deklarasinya 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.

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"

```

Halaman 64 | Modul Praktikum Algoritma dan Pemrograman 2

```

18
19 // Menghapus entri dengan kunci "john"
20 delete(dct1, "john")

```

II. GUIDED

GUIDED I

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
}

// Fungsi untuk menghitung rata-rata nilai tiap mahasiswa
func hitungRataRata(m *Mahasiswa) {
    total := m.Matematika + m.Fisika + m.Kimia
    m.Ratarata = float64(total) / 3.0
}

func main() {
    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},
    }

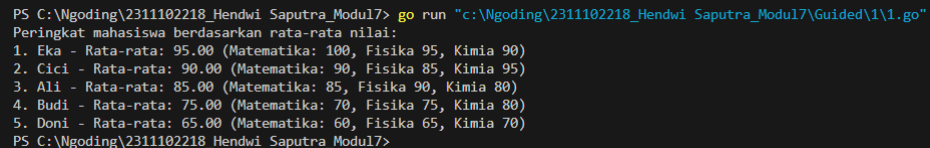
    for i := range mahasiswa {
        hitungRataRata(&mahasiswa[i])
    }

    // 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 rata-
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



```
PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7> go run "c:\Ngoding\2311102218_Hendwi Saputra_Modul7\Guided\1\1.go"
Peringkat mahasiswa berdasarkan rata-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 C:\Ngoding\2311102218_Hendwi Saputra_Modul7>
```

Deskripsi:

Program ini menghitung dan mengurutkan rata-rata nilai beberapa mahasiswa berdasarkan nilai mereka dalam mata pelajaran Matematika, Fisika, dan Kimia. Setiap mahasiswa diwakili oleh struktur Mahasiswa yang menyimpan nama dan nilai mereka dalam tiga mata pelajaran tersebut serta rata-rata nilai. Fungsi `hitungRataRata` digunakan untuk menghitung rata-rata nilai setiap mahasiswa. Dalam fungsi `main`, daftar mahasiswa diinisialisasi dengan beberapa data contoh, dan kemudian program menghitung rata-rata nilai tiap mahasiswa. Selanjutnya, program mengurutkan mahasiswa berdasarkan nilai rata-rata mereka secara menurun (descending) menggunakan fungsi `sort.Slice`. Akhirnya, program mencetak peringkat mahasiswa berdasarkan rata-rata nilai mereka, menampilkan nama, rata-rata nilai, dan nilai individu dalam setiap mata pelajaran.

GUIDED II

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

```
PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7> go run "c:\Ngoding\2311102218_Hendwi Saputra_Modul7\Guided\2\2.go"
Daftar Mahasiswa:
NIM      Nama
-----
Daftar Mahasiswa:
NIM      Nama
Daftar Mahasiswa:
Daftar Mahasiswa:
NIM      Nama
-----
20231003  Cici
20231004  Dedi
20231001  Andi
20231002  Budi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM      Nama
-----
20231001  Andi
20231002  Budi
20231004  Dedi
PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7>
```

Deskripsi:

Program ini mengelola data mahasiswa menggunakan struktur data map. Map ini menyimpan NIM (Nomor Induk Mahasiswa) sebagai kunci dan nama mahasiswa sebagai nilai. Pada awalnya, program menginisialisasi map dengan tiga entri dan kemudian menambahkan satu entri baru. Program ini menampilkan seluruh isi map dalam format kolom dan baris. Selain itu, program menunjukkan cara mengakses data berdasarkan NIM tertentu dan menghapus data berdasarkan NIM. Setelah penghapusan, isi map ditampilkan kembali untuk menunjukkan perubahan.

III. UNGUIDED

UNGUIDED I

Source Code

```
package main

import (
    "fmt"
    "math"
)

type lingkaran struct {
    titikX, titikY, cx, cy, r float64
}

func cek(titikX, titikY float64, lingkaran1, lingkaran2
lingkaran) string {
    jarakLingkaran1 := math.Sqrt(math.Pow(titikX-
lingkaran1.cx, 2) + math.Pow(titikY-lingkaran1.cy, 2))
    jarakLingkaran2 := math.Sqrt(math.Pow(titikX-
lingkaran2.cx, 2) + math.Pow(titikY-lingkaran2.cy, 2))

    if jarakLingkaran1 <= lingkaran1.r && jarakLingkaran2
<= lingkaran2.r {
        return "titik didalam lingkaran 1 dan 2"
    } else if jarakLingkaran1 <= lingkaran1.r {
        return "titik didalam lingkaran 1"
    } else if jarakLingkaran2 <= lingkaran2.r {
        return "titik didalam lingkaran 2"
    } else {
        return "titik diluar lingkaran 1 dan 2"
    }
}

func main() {
    var lingkaran1, lingkaran2 lingkaran

    fmt.Print("Masukkan pusat koordinat dan radius
lingkaran 1 (x y radius): ")
    fmt.Scan(&lingkaran1.cx, &lingkaran1.cy,
&lingkaran1.r)

    fmt.Print("Masukkan pusat koordinat dan radius
lingkaran 2 (x y radius): ")
    fmt.Scan(&lingkaran2.cx, &lingkaran2.cy,
&lingkaran2.r)

    fmt.Print("Masukkan titik koordinat (x y): ")
    fmt.Scan(&lingkaran1.titikX, &lingkaran1.titikY)

    hasil := cek(lingkaran1.titikX, lingkaran1.titikY,
lingkaran1, lingkaran2)
```

```
        fmt.Println(hasil)
    }
```

Screenshot

```
PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7> go run "c:\Ngoding\2311102218_Hendwi Saputra_Modul7\Unguided\1\1.go"
Masukkan pusat koordinat dan radius lingkaran 1 (x y radius): 1 1 5
Masukkan pusat koordinat dan radius lingkaran 2 (x y radius): 8 8 4
Masukkan titik koordinat (x y): 2 2
titik didalam lingkaran 1
PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7> go run "c:\Ngoding\2311102218_Hendwi Saputra_Modul7\Unguided\1\1.go"
Masukkan pusat koordinat dan radius lingkaran 1 (x y radius): 1 2 3
Masukkan pusat koordinat dan radius lingkaran 2 (x y radius): 4 5 6
Masukkan titik koordinat (x y): 7 8
titik didalam lingkaran 2
PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7> go run "c:\Ngoding\2311102218_Hendwi Saputra_Modul7\Unguided\1\1.go"
Masukkan pusat koordinat dan radius lingkaran 1 (x y radius): 5 10 15
Masukkan pusat koordinat dan radius lingkaran 2 (x y radius): -15 4 20
Masukkan titik koordinat (x y): 0 0
titik didalam lingkaran 1 dan 2
PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7> go run "c:\Ngoding\2311102218_Hendwi Saputra_Modul7\Unguided\1\1.go"
Masukkan pusat koordinat dan radius lingkaran 1 (x y radius): 1 1 5
Masukkan pusat koordinat dan radius lingkaran 2 (x y radius): 8 8 4
Masukkan titik koordinat (x y): 15 20
titik diluar lingkaran 1 dan 2
PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7> █
```

Deskripsi:

Program ini bertujuan untuk menentukan apakah sebuah titik berada di dalam dua lingkaran, salah satu lingkaran, atau di luar keduanya. Program meminta pengguna untuk memasukkan pusat koordinat (x, y) dan radius dari dua lingkaran, kemudian meminta koordinat (x, y) dari titik yang ingin diperiksa. Fungsi cek menghitung jarak titik ke pusat masing-masing lingkaran dan membandingkannya dengan radius lingkaran tersebut. Berdasarkan hasil perhitungan, fungsi akan mengembalikan informasi apakah titik tersebut berada di dalam lingkaran pertama, kedua, keduanya, atau di luar keduanya. Hasil ini kemudian ditampilkan kepada pengguna.

UNGUIDED II

Source Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var N int

    fmt.Print("Masukkan jumlah elemn array: ")
    fmt.Scan(&N)

    arr := make([]int, N)

    fmt.Println("Masukkan nilai-nilai array:")
    for i := 0; i < N; i++ {
        fmt.Printf("Elemen ke-%d: ", i+1)
        fmt.Scan(&arr[i])
    }

    fmt.Println("\nSeluruh isi array:")
    for _, v := range arr {
        fmt.Print(v, " ")
    }
    fmt.Println()

    fmt.Println("\nElemen-elemen array dengan indeks ganjil:")
    for i := 1; i < N; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    fmt.Println("\nElemen-elemen array dengan indeks gendap:")
    for i := 0; i < N; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    var x int
    fmt.Print("\nMasukkan nilai x: ")
    fmt.Scan(&x)
    fmt.Printf("Elemen-elemen array dengan indeks kelipatan %d: \n", x)
    for i := 0; i < N; i++ {
        if i%x == 0 {
            fmt.Print(arr[i], " ")
        }
    }
    fmt.Println()
}
```

```

var hapusElemen int
fmt.Print("\nMasukkan indeks yang ingin dihapus: ")
fmt.Scan(&hapusElemen)
if hapusElemen >= 0 && hapusElemen < len(arr) {
    arr = append(arr[:hapusElemen],
arr[hapusElemen+1:]...)
    fmt.Println("Array setelah penghapusan:")
    for _, v := range arr {
        fmt.Print(v, " ")
    }
    fmt.Println()
} else {
    fmt.Println("Indeks tidak valid")
}

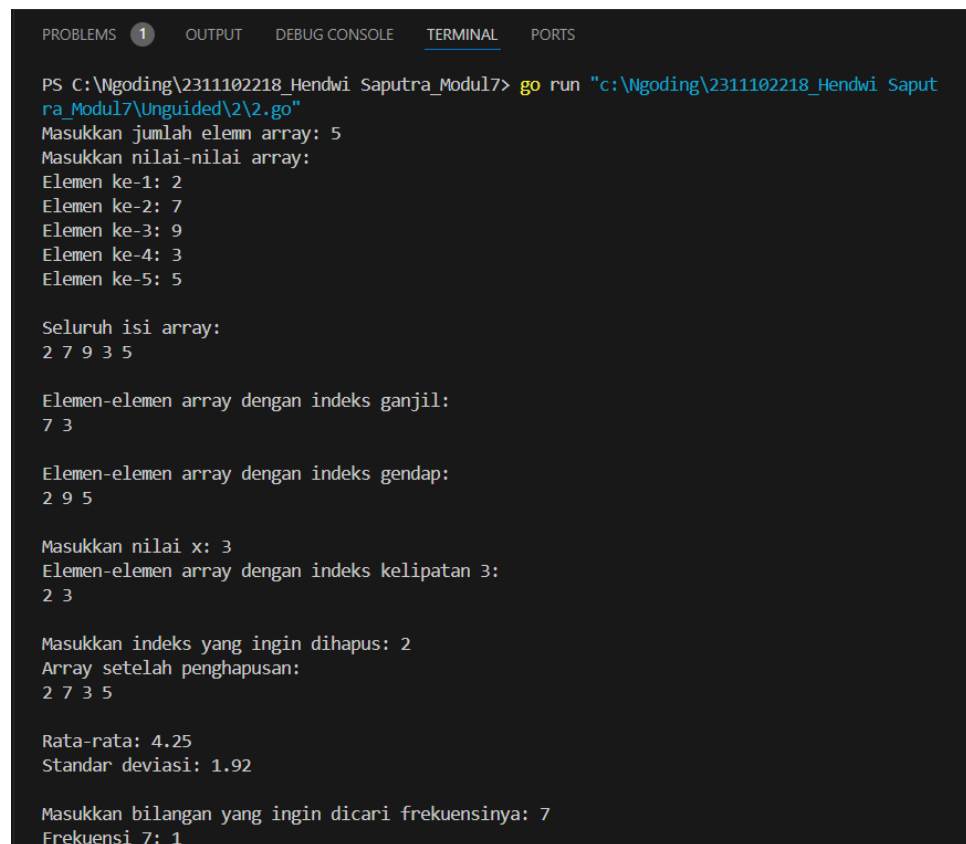
var sum float64
for _, v := range arr {
    sum += float64(v)
}
rataRata := sum / float64(len(arr))
fmt.Printf("\nRata-rata: %.2f\n", rataRata)

var sumKuadrat float64
for _, v := range arr {
    sumKuadrat += float64(v * v)
}
varian := (sumKuadrat / float64(len(arr))) -
math.Pow(rataRata, 2)
standarDeviasi := math.Sqrt(varian)
fmt.Printf("Standar deviasi: %.2f\n", standarDeviasi)

var bilangan int
fmt.Print("\nMasukkan bilangan yang ingin dicari
frekuensinya: ")
fmt.Scan(&bilangan)
var frekuensi int
for _, v := range arr {
    if v == bilangan {
        frekuensi++
    }
}
fmt.Printf("Frekuensi %d: %d\n", bilangan, frekuensi)
}

```

Screenshot



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7> go run "c:\Ngoding\2311102218_Hendwi Saputra_Modul7\Unguided\2\2.go"
Masukkan jumlah elemen array: 5
Masukkan nilai-nilai array:
Elemen ke-1: 2
Elemen ke-2: 7
Elemen ke-3: 9
Elemen ke-4: 3
Elemen ke-5: 5

Seluruh isi array:
2 7 9 3 5

Elemen-elemen array dengan indeks ganjil:
7 3

Elemen-elemen array dengan indeks genap:
2 9 5

Masukkan nilai x: 3
Elemen-elemen array dengan indeks kelipatan 3:
2 3

Masukkan indeks yang ingin dihapus: 2
Array setelah penghapusan:
2 7 3 5

Rata-rata: 4.25
Standar deviasi: 1.92

Masukkan bilangan yang ingin dicari frekuensinya: 7
Frekuensi 7: 1
```

Deskripsi:

Program ini meminta pengguna untuk memasukkan jumlah elemen dalam array dan nilai-nilai array tersebut. Program kemudian menampilkan semua elemen array, elemen dengan indeks ganjil, elemen dengan indeks genap, dan elemen yang memiliki indeks kelipatan x, di mana x adalah nilai yang dimasukkan oleh pengguna. Pengguna juga dapat menghapus elemen array berdasarkan indeks yang diberikan. Program menghitung dan menampilkan rata-rata dan standar deviasi dari elemen array yang tersisa setelah penghapusan. Selain itu, program juga mencari dan menampilkan frekuensi kemunculan nilai tertentu dalam array.

UNGUIDED III

Source Code

```
package main

import "fmt"

type nama struct {
    Nama string
    Skor int
}

func main() {
    var klubA, klubB nama
    var pemenang []string
    var tanding int = 1

    fmt.Print("Masukkan nama Klub pertama: ")
    fmt.Scanln(&klubA.Nama)
    fmt.Print("Masukkan nama Klub kedua: ")
    fmt.Scanln(&klubB.Nama)

    for {
        fmt.Printf("Pertandingan %v: ", tanding)
        fmt.Scanln(&klubA.Skor, &klubB.Skor)
        tanding++

        if klubA.Skor < 0 || klubB.Skor < 0 {
            break
        }

        if klubA.Skor > klubB.Skor {
            pemenang = append(pemenang, klubA.Nama)
        } else if klubB.Skor == klubA.Skor {
            pemenang = append(pemenang, "Draw")
        } else if klubB.Skor > klubA.Skor {
            pemenang = append(pemenang, klubB.Nama)
        }
    }

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


Screenshot

```
PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7> go run "c:\Ngoding\2311102218_Hendwi Saputra_Modul7\Unguided\3\3.go"
Masukkan nama Klub pertama: Indo
Masukkan nama Klub kedua: Jepang
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
Daftar klub yang memenangkan pertandingan:
Indo
Jepang
Draw
Jepang
Indo
Indo
Indo
Jepang
PS C:\Ngoding\2311102218_Hendwi Saputra_Modul7>
```

Deskripsi:

Program ini bertujuan untuk mencatat dan menampilkan hasil pertandingan antara dua klub sepak bola berdasarkan skor yang diinput oleh pengguna. Pengguna diminta memasukkan nama dua klub dan skor mereka untuk setiap pertandingan. Program ini kemudian menentukan pemenang dari setiap pertandingan atau mencatat hasil seri jika skor kedua klub sama. Proses ini diulang terus menerus hingga pengguna memasukkan skor negatif, yang menjadi tanda untuk berhenti. Setelah itu, program menampilkan daftar klub yang memenangkan setiap pertandingan.

UNGUIDED IV

Source Code

```
package main

import (
    "fmt"
)

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var char rune
    *n = 0
    fmt.Print("Masukkan karakter (gunakan .) untuk berhenti): ")

    for *n < NMAX {
        fmt.Scanf("%c", &char)
        if char == '.' {
            break
        }

        if char != '\n' {
            t[*n] = char
            *n++
        }
    }
}

func balikanArray(t tabel, n int) tabel {
    var hasil tabel
    for i := 0; i < n; i++ {
        hasil[i] = t[n-1-i]
    }
    return hasil
}

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

func cetakArray(t tabel, n int) {
    var balik tabel = balikanArray(t, n)
    isPalindrome := palindrom(t, n)

    fmt.Print("Teks\t\t:")
    for i := 0; i < n; i++ {
```

```

        fmt.Printf(" %c", t[i])
    }
    fmt.Println()

    fmt.Print("Reverse teks\t:")
    for i := 0; i < n; i++ {
        fmt.Printf(" %c", balik[i])
    }
    fmt.Println()

    fmt.Print("Palindrom\t? ")
    if isPalindrome {
        fmt.Print("true")
    } else {
        fmt.Print("false")
    }
}

func main() {
    var tab tabel
    var m int

    isiArray(&tab, &m)
    cetakArray(tab, m)
}

```

Screenshot

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Ngoding\2311102218_Hendwi_Saputra_Modul7> go run "c:\Ngoding\2311102218_Hendwi_Saputra_Modul7\Unguided\4\4.go"
Masukkan karakter (gunakan .) untuk berhenti: S E N A N G.
Teks      : S E N A N G
Reverse teks : G N A N E S
Palindrom  ? false
PS C:\Ngoding\2311102218_Hendwi_Saputra_Modul7> go run "c:\Ngoding\2311102218_Hendwi_Saputra_Modul7\Unguided\4\4.go"
Masukkan karakter (gunakan .) untuk berhenti: K A T A K.
Teks      : K A T A K
Reverse teks : K A T A K
Palindrom  ? true

```

Deskripsi:

Program ini adalah implementasi sederhana yang bertujuan untuk memeriksa apakah sebuah teks adalah palindrom, yaitu teks yang dibaca sama dari depan maupun dari belakang. Program ini dimulai dengan mendefinisikan konstanta NMAX sebesar 127 dan tipe array tabel yang berisi karakter (rune). Fungsi isiArray bertugas untuk mengisi array tabel dengan karakter yang dimasukkan oleh pengguna sampai karakter titik (.) dimasukkan atau sampai array mencapai ukuran maksimal. Setelah itu, fungsi balikanArray akan membuat array baru yang merupakan kebalikan

dari array asli. Kemudian, fungsi palindrom digunakan untuk memeriksa apakah array tabel adalah palindrom dengan membandingkan elemen pertama dengan elemen terakhir, elemen kedua dengan elemen kedua dari akhir, dan seterusnya. Fungsi cetakArray akan mencetak array asli, array yang dibalik, dan hasil pengecekan apakah array tersebut adalah palindrom. Akhirnya, fungsi main menginisialisasi array tabel, memanggil isiArray untuk mengisi array, dan memanggil cetakArray untuk menampilkan hasilnya.