

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI
OBJEK**

**MODUL VII
STRUCT & ARRAY**



Oleh :

NAMA : FAISAL KHOIRUDDIN

NIM : 2311102046

Kelas : IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

I. DASAR TEORI

1. Struct

Tipe bentukan memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipe bentukan 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>	type <nama alias> <tipe data>
3		
4	algoritma	func main(){
5
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> <tipe data>	<field 1> <tipe data>
4	<field 2> <tipe data>	<field 2> <tipe data>
5	<field 3> <tipe data>	<field 3> <tipe data>
6	>	}
7		

Berbeda dengan bahasa pemrograman 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

```

2. Array

Array mempunyai ukuran (jumlah elemen) yang tetap (statis) 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.

```

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

1. 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
}

// 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() {
    // array 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])
    }

    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)
    }
}

```

Screenshots Output


```
PS C:\Semester3Go\Praktikum\Modul 6\Guided\No_1> go run No_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:\Semester3Go\Praktikum\Modul 6\Guided\No_1>
```

*Tidak berjudul - Notep... — □ ×

File Edit Lihat ⚙

NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-02

Ln 3, Col 17 100% Windows (CRLF) UTF-8

Deskripsi:

Program tersebut merupakan program melihat peringkat mahasiswa berdasarkan rata-rata nilai. Struct pada program tersebut yaitu untuk menampung data mahasiswa. Pada program tersebut, fungsi `hitungRataRata` untuk menghitung rata-rata nilai tiap mahasiswa

- `package main` → paket utama program golang
- `import` → mengimpor
- `"fmt"` → mengimpor `fmt`
- `"sort"` → mengimpor `sort`
- `type Mahasiswa struct {` → struktur untuk menampung data mahasiswa
- `Nama string` → nama bertipe data string
- `Matematika int` → matematika bertipe data integer
- `Fisika int` → fisika bertipe data integer
- `Kimia int` → kimia bertipe data integer
- `RataRata float64` → rata-rata bertipe data float64
- `func hitungRataRata(m *Mahasiswa) {` → Fungsi untuk menghitung rata-rata nilai tiap mahasiswa
- `total := m.Matematika + m.Fisika + m.Kimia` → menghitung nilai total
- `m.RataRata = float64(total) / 3.0` → menghitung rata – rata dan mengonversi total ke float64
- `func main() {` → merupakan fungsi utama
- `mahasiswa := []Mahasiswa{` → array menampung data mahasiswa

- {"Ali", 85, 90, 80, 0}, ➔ Ali dengan nilai 85, 90, 80, 0}
- {"Budi", 70, 75, 80, 0}, ➔ Budi dengan nilai 70, 75, 80, 0
- {"Cici", 90, 85, 95, 0}, ➔ Cici dengan nilai 90, 85, 95, 0
- {"Doni", 60, 65, 70, 0}, ➔ Doni dengan nilai 60, 65, 70, 0
- {"Eka", 100, 95, 90, 0}, ➔ Eka dengan nilai 100, 95, 90, 0
- for i := range mahasiswa { ➔ menghitung rata-rata nilai tiap mahasiswa
- hitungRataRata(&mahasiswa[i])
- } ➔ memanggil fungsi hitung rata – rata untuk setiap mahasiswa
- sort.Slice(mahasiswa, func(i, j int) bool { ➔ mengurutkan mahasiswa berdasarkan nilai rata - rata
- return mahasiswa[i].RataRata > mahasiswa[j].RataRata ➔ return mahasiswa[i].RataRata > mahasiswa[j].RataRata
- fmt.Println("Peringkat mahasiswa berdasarkan rata-rata nilai:") ➔ menampilkan statement Peringkat mahasiswa berdasarkan rata-rata nilai
- for i, m := range mahasiswa { ➔ iterasi melalui array mahasiswa yang sudah diurutkan
- fmt.Printf("%d.%s - Rata-rata: %.2f (Matematika: %d, Fisika: %d, Kimia: %d)\n", ➔ menampilkan mahasiswa dengan nilai rata – rata matematika, fisika, kimia
- i+1, m>Nama, m.RataRata, m.Matematika, m.Fisika, m.Kimia) ➔ menampilkan peringkat nama, rata-rata, dna nilai untuk setiap mahasiswa

2. 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)
    }
}

```

Screenshots Output

```

PS C:\Semester3Go\Praktikum\Modul 6\Guided\No_2> go run No_1.go
Daftar Mahasiswa:
NIM      Nama
-----
20231001  Andi
20231002  Budi
20231003  Cici
20231004  Dedi
Nama Mahasiswa dengan NIM 20231002 adalah Budi
Daftar Mahasiswa setelah dihapus:
NIM      Nama
-----
20231002  Budi
20231004  Dedi
20231001  Andi
PS C:\Semester3Go\Praktikum\Modul 6\Guided\No_2>

```

Deskripsi:

Program tersebut merupakan program Program tersebut merupakan map menampilkan daftar mahasiswa secara sederhana. program untuk menampilkan daftar masiswa. Pada program tersebut, ditampilkan daftar mahasiswa dan mahasiswa yang sudah dihapus dari map.

- package main → paket utama program golang
- import "fmt" → mengimpor fmt
- func main() { → merupakan fungsi utama

- mahasiswa := map[string]string { ➔ Membuat map dengan NIM sebagai kunci dan Nama sebagai nilai
- "20231001": "Andi", ➔ nim 20231001 dengan nama Andi
- "20231002": "Budi", ➔ nim 20231002 dengan nama Budi
- "20231003": "Cici", ➔ nim 20231003 dengan nama Cici
- mahasiswa["20231004"] = "Dedi" ➔ Menambahkan data baru ke map
- fmt.Println("Daftar Mahasiswa:") ➔ menampilkan statement Daftar Mahasiswa:
- fmt.Println("NIM\t\tNama") ➔ menampilkan NIM dan Nama
- fmt.Println("-----") ➔ menampilkan batas garis
- for nim, nama := range mahasiswa { ➔ perulangan melalui map mahasiswa
- fmt.Printf("%s\t%s\n", nim, nama) ➔ menampilkan NIM dan nama setiap nama mahasiswa
- nim := "20231002" ➔ deklarasi variabel nim dengan nilai 20231002
- fmt.Println("\nNama Mahasiswa dengan NIM", nim, "adalah", mahasiswa[nim]) ➔ menampilkan nama mahasiswa dengan NIM tersebut
- delete(mahasiswa, "20231003") ➔ Menghapus data berdasarkan NIM
- fmt.Println("\nDaftar Mahasiswa setelah dihapus:") ➔ menampilkan nDaftar Mahasiswa setelah dihapus
- fmt.Println("NIM\t\tNama") ➔ menampilkan NIM dan Nama
- fmt.Println("-----") ➔ menampilkan garis batas
- for nim, nama := range mahasiswa { ➔ perulangan melalui map mahasiswa yang telah diubah
- fmt.Printf("%s\t%s\n", nim, nama) ➔ menampilkan NIM dan nama setiap mahasiswa yang tersisa

III. Unguided

1. 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". Contoh

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	Titik di luar lingkaran 1 dan 2
	15 20	

Fungsi untuk menghitung jarak titik (a, b) dan (c, d) dimana rumus jarak adalah:

$$\text{jarak} = \sqrt{(a - c)^2 + (b - d)^2}$$

dan juga fungsi untuk menentukan posisi sebuah titik sembarang

berada di dalam suatu lingkaran atau tidak.

```
function jarak(p, q : titik) -> real
{Mengembalikan jarak antara titik p(x,y) dan titik q(x,y)}

function didalam(c:lingkaran, p:titik) -> boolean
{Mengembalikan true apabila titik p(x,y) berada di dalam lingkaran c yang
memiliki titik pusat (cx,cy) dan radius r}
```

Catatan: Lihat paket math dalam lampiran untuk menggunakan fungsi math.Sqrt() untuk menghitung akar kuadrat.

Source Code

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    titik Titik
    radius int
}

func jarak(t1, t2 Titik) float64 {
    return math.Sqrt(float64((t2.x-
        t1.x)*(t2.x-t1.x) + (t2.y-t1.y)*(t2.y-
        t1.y)))
}
```

```
func posisiTitikTerhadapLingkaran(l
    Lingkaran, t Titik) bool {
    return jarak(l.titik, t) <=
        float64(l.radius)
}

func main() {
    var x1, y1, r1, x2, y2, r2, xt, yt int
    fmt.Scan(&x1, &y1, &r1)
    lingkaran1 := Lingkaran{Titik{x1, y1},
        r1}
    fmt.Scan(&x2, &y2, &r2)
    lingkaran2 := Lingkaran{Titik{x2, y2},
        r2}
    fmt.Scan(&xt, &yt)
    titik := Titik{xt, yt}

    dalamLingkaran1 :=
        posisiTitikTerhadapLingkaran(lingkaran
            1, titik)
    dalamLingkaran2 :=
        posisiTitikTerhadapLingkaran(lingkaran
            2, titik)

    if dalamLingkaran1 && dalamLingkaran2 {
        fmt.Println("Titik di dalam
            lingkaran 1 dan 2")
    } else if dalamLingkaran1 {
        fmt.Println("Titik di dalam
```



```

        lingkaran 1")
    } else if dalamLingkaran2 {
        fmt.Println("Titik di dalam
        lingkaran 2")
    } else {
        fmt.Println("Titik di luar
        lingkaran 1 dan 2")
    }
}
}

```

Screenshots Output

```

PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_1> go run No_1.go
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_1> go run No_1.go
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_1> go run No_1.go
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_1> go run No_1.go
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_1>

```

Deskripsi:

Program tersebut merupakan program untuk menentukan posisi sebuah titik sembarang (x,y) berdasarkan dua lingkaran tersebut. Pengguna diminta menginputkan pada Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingkaran 2, sedangkan baris ketiga adalah koordinat titik sembarang. Output dari program tersebut yaitu 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".

- package main → paket utama program golang
- import → mengimport

- "fmt" → mengimpor fmt
- "math" → mengimport math
- type Titik struct { → deklarasi type Titik struct
- x, y int → properti variabel x dan y bertipe data int
- type Lingkaran struct { → deklarasi type Lingkaran struct
- titik Titik → lingkaran memiliki properti titik pusat
- radius int → property radius bertipe data integer
- func jarak(t1, t2 Titik) float64 { → deklarasi fungsi jarak dengan (t1, t2 Titik)
- return math.Sqrt(float64((t2.x-t1.x)*(t2.x-t1.x) + (t2.y-t1.y)*(t2.y-t1.y))) → return math.Sqrt(float64((t2.x-t1.x)*(t2.x-t1.x) + (t2.y-t1.y)*(t2.y-t1.y)))
- func posisiTitikTerhadapLingkaran(l Lingkaran, t Titik) bool { → fungsi posisiTitikTerhadapLingkaran dengan parameter (l Lingkaran, t Titik)
- return jarak(l.titik, t) <= float64(l.radius) → mengembalikan jarak(l.titik, t) <= float64(l.radius)
- func main() { → merupakan fungsi utama
- var x1, y1, r1, x2, y2, r2, xt, yt int → deklarasi variabel x1, y1, r1, x2, y2, r2, xt, yt dengan tipe data integer
- fmt.Scan(&x1, &y1, &r1) → membaca input koordinat pusat dan radius lingkaran 1
- lingkaran1 := Lingkaran{Titik{x1, y1}, r1} → membuat objek lingkaran 1 dengan input yang diberikan
- fmt.Scan(&x2, &y2, &r2) → membaca input koordinat pusat

dan radius lingkaran 2

- `lingkaran2 := Lingkaran{Titik{x2, y2}, r2}` → membuat objek lingkaran 2 dengan input yang diberikan
- `fmt.Scan(&xt, &yt)` → membaca input untuk koordinat titik yang dicek
- `titik := Titik{xt, yt}` → membuat objek titik dengan input yang diberikan
- `dalamLingkaran1 := posisiTitikTerhadapLingkaran(lingkaran1, titik)` → mengecek titik apakah berada dalam lingkaran 1
- `dalamLingkaran2 := posisiTitikTerhadapLingkaran(lingkaran2, titik)` → mengecek titik apakah berada dalam lingkaran 2
- `if dalamLingkaran1 && dalamLingkaran2 {` → percabangan if jika `dalamLingkaran1` dan `dalamLingkaran2`
- `fmt.Println("Titik di dalam lingkaran 1 dan 2")` → menampilkan statement "Titik di dalam lingkaran 1 dan 2"
- `} else if dalamLingkaran1 {` → selain itu jika `dalamLingkaran1`
- `fmt.Println("Titik di dalam lingkaran 1")` → menampilkan statement "Titik di dalam lingkaran 1"
- `} else if dalamLingkaran2 {` → selain itu jika `dalamLingkaran2`
- `fmt.Println("Titik di dalam lingkaran 2")` → menampilkan statement "Titik di dalam lingkaran 2"
- `} else {` → kalau tidak
- `fmt.Println("Titik di luar lingkaran 1 dan 2")` → menampilkan statement "Titik di luar lingkaran 1 dan 2"

2. 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:

- a. 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.

Source Code

```
package main

import (
    "fmt"
    "math"
)
```

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

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

func displayIndeksGanjil(arr []int) {
    fmt.Print("Elemen dengan indeks ganji:
    ")
    for i := 1; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}

func displayIndeksGenap(arr []int) {
    fmt.Print("Elemen dengan indeks genap:
    ")
    for i := 0; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}
```

```

}

func displayIndeksKelipatan(arr []int, x
    int) {
    fmt.Printf("Elemen dengan indeks
        kelipatan %d: ", x)
    for i := 0; i < len(arr); i++ {
        if i%x == 0 {
            fmt.Print(arr[i], " ")
        }
    }
    fmt.Println()
}

func deleteElemen(arr []int, indeks int)
    []int {
    return append(arr[:indeks],
        arr[indeks+1:]...)
}

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

func hitungStandarDeviasi(arr []int)
    float64 {

```

```
mean := hitungRataRata(arr)
sum := 0.0
for _, val := range arr {
    sum += math.Pow(float64(val)-mean,
        2)
}
return math.Sqrt(sum /
    float64(len(arr)))
}

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

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

    arr := isiArray(n)

    var pilih int
    for {
```

```

fmt.Println("\n=====
=====")

    fmt.Println("|
Program pada Array          |")

fmt.Println("=====
=====")

    fmt.Println("| 1. | Tampilkan 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 standar
deviasi elemen array           |")
    fmt.Println("| 8. | Hitung
frekuensi elemen tertentu      |")
    fmt.Println("| 9. | keluar
                    |")

fmt.Println("=====
=====")

```



```
fmt.Print("Pilih operasi: ")
fmt.Scan(&pilih)

switch pilih {
case 1:
    displayArray(arr)
case 2:
    displayIndeksGanjil(arr)
case 3:
    displayIndeksGenap(arr)
case 4:
    var x int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)
    displayIndeksKelipatan(arr, x)
case 5:
    var indeks int
    fmt.Print("Masukkan indeks yang
dihapus: ")
    fmt.Scan(&indeks)
    arr = deleteElemen(arr, indeks)
    displayArray(arr)
case 6:
    RataRata := hitungRataRata(arr)
    fmt.Println("Rata-Rata pada
array: ", RataRata)
case 7:
    standarDeviasi :=
hitungStandarDeviasi(arr)
    fmt.Println("Standar deviasi
```

```

        dari elemen array: ", standarDeviasi)
    case 8:
        var number int
        fmt.Print("Masukkan bilangan
untuk menghitung frekuensi: ")
        fmt.Scan(&number)
        frekuensi :=
hitungFrekuensi(arr, number)
        fmt.Printf("Frekuensi %d dalam
array: %d\n", number, frekuensi)
    case 9:
        fmt.Println("Keluar dari
program.")
        return
    default:
        fmt.Println("Pilihan tidak
valid.")
    }
}
}

```

Screenshots Output

The screenshot shows a terminal window with the following output:

```

PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_2> go run No_2.go
Masukkan jumlah elemen: 10
Masukkan nilai untuk elemen 0: 0
Masukkan nilai untuk elemen 1: 1
Masukkan nilai untuk elemen 2: 2
Masukkan nilai untuk elemen 3: 3
Masukkan nilai untuk elemen 4: 4
Masukkan nilai untuk elemen 5: 5
Masukkan nilai untuk elemen 6: 6
Masukkan nilai untuk elemen 7: 7
Masukkan nilai untuk elemen 8: 8
Masukkan nilai untuk elemen 9: 9

```

Overlaid on the terminal is a Notepad window titled "*Tidak berjudul - Not...". It contains the following text:

```

NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B

```

The Notepad window's status bar shows "Ln 3, Col 16", "100%", "Windows (CRLF)", and "UTF-8".

```

=====
|                               |
|           Program pada Array |
|                               |
| 1. | Tampilkan 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 standar deviasi elemen array  |
| 8. | Hitung frekuensi elemen tertentu     |
| 9. | keluar                             |
|                               |
=====
Pilih operasi: 1
Isi array: [0 1 2 3 4 5 6 7 8 9]

```

```

=====
|                               |
|           Program pada Array |
|                               |
| 1. | Tampilkan 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 standar deviasi elemen array  |
| 8. | Hitung frekuensi elemen tertentu     |
| 9. | keluar                             |
|                               |
=====
Pilih operasi: 2
Elemen dengan indeks ganji: 1 3 5 7 9

```

```

=====
|                               |
|           Program pada Array |
|                               |
| 1. | Tampilkan 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 standar deviasi elemen array  |
| 8. | Hitung frekuensi elemen tertentu     |
| 9. | keluar                             |
|                               |
=====
Pilih operasi: 3
Elemen dengan indeks genap: 0 2 4 6 8

```

```

=====
|                                     |
|               Program pada Array   |
|=====|
| 1. | Tampilkan 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 standar deviasi elemen array  |
| 8. | Hitung frekuensi elemen tertentu     |
| 9. | keluar                              |
|=====|
Pilih operasi: 4
Masukkan nilai x: 2
Elemen dengan indeks kelipatan 2: 0 2 4 6 8

```

```

=====
|                                     |
|               Program pada Array   |
|=====|
| 1. | Tampilkan 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 standar deviasi elemen array  |
| 8. | Hitung frekuensi elemen tertentu     |
| 9. | keluar                              |
|=====|
Pilih operasi: 5
Masukkan indeks yang dihapus: 0
Isi array: [1 2 3 4 5 6 7 8 9]

```

```

=====
|                                     |
|               Program pada Array   |
|=====|
| 1. | Tampilkan 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 standar deviasi elemen array  |
| 8. | Hitung frekuensi elemen tertentu     |
| 9. | keluar                              |
|=====|
Pilih operasi: 6
Rata-Rata pada array: 5

```

```

=====
|          Program pada Array          |
=====
| 1. | Tampilkan 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 standar deviasi elemen array  |
| 8. | Hitung frekuensi elemen tertentu     |
| 9. | keluar                              |
=====
Pilih operasi: 7
Standar deviasi dari elemen array: 2.581988897471611

```

```

=====
|          Program pada Array          |
=====
| 1. | Tampilkan 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 standar deviasi elemen array  |
| 8. | Hitung frekuensi elemen tertentu     |
| 9. | keluar                              |
=====
Pilih operasi: 8
Masukkan bilangan untuk menghitung frekuensi: 4
Frekuensi 4 dalam array: 1

```

```

=====
|          Program pada Array          |
=====
| 1. | Tampilkan 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 standar deviasi elemen array  |
| 8. | Hitung frekuensi elemen tertentu     |
| 9. | keluar                              |
=====
Pilih operasi: 9
Keluar dari program.
PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_2> 

```

Deskripsi:

Program tersebut merupakan program array untuk melakukan beberapa operasi. Program tersebut diawali dengan meminta pengguna untuk memasukkan jumlah elemen dan memasukkan nilai dari elemen, kemudian memilih menu. Program tersebut

menampilkan output sesuai dengan menu yang dipilih.

- package main → paket utama pada program golang
- import (→ menimpor
- "fmt" → impor fmt
- "math" → impor math
- func isiArray(n int) []int { fungsi untuk mengisi array
- arr := make([]int, n) → arr := make([]int, n)
- for i := 0; i < n; i++ { → perulangan for i := 0; i < n; i++
(increment)
- fmt.Printf("Masukkan nilai untuk elemen %d: ", i) → Masukkan
nilai untuk elemen
- fmt.Scan(&arr[i]) → membaca input arr[i]
- return arr → return array
- func displayArray(arr []int) { → fungsi menampilkan array
- fmt.Println("Isi array:", arr) → menampilkan Isi array
- func displayIndeksGanjil(arr []int) { → fungsi
displayIndeksGanjil
- fmt.Print("Elemen dengan indeks ganji: ") → menampilkan
Elemen dengan indeks ganjil statement
- for i := 1; i < len(arr); i += 2 { → perulangan i := 1; i < len(arr);
i += 2
- fmt.Print(arr[i], " ") → menampilkan arr[i]
- fmt.Println() → fmt.Println()
- func displayIndeksGenap(arr []int) { → fungsi

displayIndeksGenap(arr []int)

- `fmt.Print("Elemen dengan indeks genap: ")` → menampilkan statement Elemen dengan indeks genap
- `for i := 0; i < len(arr); i += 2 {` → perulangan `for i := 0; i < len(arr); i += 2`
- `fmt.Print(arr[i], " ")` → menampilkan `arr[i]`
- `fmt.Println()` → `fmt.Println()`
- `func displayIndeksKelipatan(arr []int, x int) {` → fungsi `displayIndeksKelipatan(arr []int, x int)`
- `fmt.Printf("Elemen dengan indeks kelipatan %d: ", x)` → menampilkan Elemen dengan indeks kelipatan
- `for i := 0; i < len(arr); i++ {` → perulangan `for i := 0; i < len(arr); i++`
- `if i%x == 0 {` → percabangan `if i%x == 0`
- `fmt.Print(arr[i], " ")` → menampilkan `arr[i]`
- `fmt.Println()` → `fmt.Println()`
- `func deleteElemen(arr []int, indeks int) []int {` → fungsi `deleteElemen(arr []int, indeks int) []int`
- `return append(arr[:indeks], arr[indeks+1:]...)` → `return append(arr[:indeks], arr[indeks+1:]...)`
- `func hitungRataRata(arr []int) float64 {` → fungsi `hitungRataRata(arr []int) float64`
- `sum := 0` → sum sama dengan 0
- `for _, val := range arr {` → perulangan `for _, val := range arr`
- `sum += val` → `sum += val`

- `return float64(sum) / float64(len(arr))` ➔ mengembalikan `float64(sum) / float64(len(arr))`
- `func hitungStandarDeviasi(arr []int) float64 {` ➔ fungsi `hitungStandarDeviasi(arr []int) float64`
- `mean := hitungRataRata(arr)` ➔ `mean := hitungRataRata(arr)`
- `sum := 0.0` ➔ `sum := 0.0`
- `for _, val := range arr {` ➔ perulangan `for _, val := range arr`
- `sum += math.Pow(float64(val)-mean, 2)` ➔ `sum += math.Pow(float64(val)-mean, 2)`
- `return math.Sqrt(sum / float64(len(arr)))` ➔ mengembalikan `math.Sqrt(sum / float64(len(arr)))`
- `func hitungFrekuensi(arr []int, number int) int {` ➔ fungsi `hitungFrekuensi(arr []int, number int) int`
- `count := 0` ➔ `count := 0`
- `for _, val := range arr {` ➔ perulangan `for _, val := range arr`
- `if val == number {` ➔ percabangan `if val == number`
- `count++` ➔ `count++(increment)`
- `return count` ➔ mengembalikan `count`
- `func main() {` ➔ merupakan fungsi utama
- `var n int` ➔ deklarasi variabel `n` bertipe data integer
- `fmt.Print("Masukkan jumlah elemen: ")` ➔ menampilkan statement `Masukkan jumlah elemen`
- `fmt.Scan(&n)` ➔ membaca input `n`
- `arr := isiArray(n)` ➔ `arr := isiArray(n)`

- var pilih int ➔ deklarasi variabel var bertipe data integer
- for { ➔ perulangan for
- fmt.Println("\n===== \n =====") ➔ menampilkan garis
- fmt.Println("| Program pada Array |") ➔ menampilkan judul
- fmt.Println("===== \n =====") ➔ menampilkan garis
- fmt.Println("| 1. | Tampilkan array |") ➔ menampilkan pilihan 1. Tampilkan array
- fmt.Println("| 2. | Tampilkan elemen dengan indeks ganjil |") ➔ menampilkan pilihan 2. Tampilkan elemen dengan indeks ganjil
- fmt.Println("| 3. | Tampilkan elemen dengan indeks genap |") ➔ menampilkan pilihan 3. Tampilkan elemen dengan indeks genap
- fmt.Println("| 4. | Tampilkan elemen dengan indeks kelipatan x|") ➔ menampilkan pilihan 4. Tampilkan elemen dengan indeks kelipatan
- fmt.Println("| 5. | Hapus elemen pada indeks tertentu |") ➔ menampilkan pilihan 5. Hapus elemen pada indeks tertentu
- fmt.Println("| 6. | Hitung rata-rata elemen array |") ➔ menampilkan pilihan 6. Hitung rata-rata elemen array
- fmt.Println("| 7. | Hitung standar deviasi elemen array |") ➔ menampilkan pilihan 7. Hitung standar deviasi elemen array
- fmt.Println("| 8. | Hitung frekuensi elemen tertentu |") ➔

menampilkan pilihan 8. Hitung frekuensi elemen tertentu

- `fmt.Println("| 9. | keluar")` → menampilkan pilihan 9. keluar
- `fmt.Println("=====`
`=====")` → menampilkan garis
- `fmt.Print("Pilih operasi: ")` → menampilkan Pilih operasi
- `fmt.Scan(&pilih)` → membaca input user
- `switch pilih {` → switch pilih
- `case 1:` → pada case 1
- `displayArray(arr)` → menampilkan array
- `case 2:` → pada case 2
- `displayIndeksGanjil(arr)` → menampilkan Elemen dengan indeks ganjil
- `case 3:` → pada case 3
- `displayIndeksGenap(arr)` → → menampilkan Elemen dengan indeks genap
- `case 4:` → pada case 4
- `var x int` → deklarasi var x
- `fmt.Print("Masukkan nilai x: ")` → menampilkan statement Masukkan nilai x
- `fmt.Scan(&x)` → membaca input user
- `displayIndeksKelipatan(arr, x)` → menampilkan indeks kelipatan dari input
- `case 5:` → pada case 5

- `var indeks int` → deklarasi variabel indeks
- `fmt.Print("Masukkan indeks yang dihapus: ")` → menampilkan statement Masukkan indeks yang dihapus
- `fmt.Scan(&indeks)` → membaca input pengguna
- `arr = deleteElemen(arr, indeks)` → `arr = deleteElemen(arr, indeks)`
- `displayArray(arr)` → menampilkan array
- case 6: → pada case 6
- `RataRata := hitungRataRata(arr)` → `RataRata := hitungRataRata(arr)`
- `fmt.Println("Rata-Rata pada array: ", RataRata)` → menampilkan Rata-Rata pada array
- case 7: → pada case 7
- `standarDeviasi := hitungStandarDeviasi(arr)` → `standarDeviasi := hitungStandarDeviasi(arr)`
- `fmt.Println("Standar deviasi dari elemen array: ", standarDeviasi)` → menampilkan Standar deviasi dari elemen array
- case 8: → pada case 8
- `var number int` → deklarasi variabel number
- `fmt.Print("Masukkan bilangan untuk menghitung frekuensi: ")` → menampilkan statement untuk Masukkan bilangan untuk menghitung frekuensi:
- `fmt.Scan(&number)` → membaca input pengguna
- `frekuensi := hitungFrekuensi(arr, number)` → `frekuensi :=`

hitungFrekuensi(arr, number)

- `fmt.Printf("Frekuensi %d dalam array: %d\n", number, frekuensi)` → menampilkan Frekuensi %d dalam array: %d\n", number, frekuensi
- case 9: → pada case 9
- `fmt.Println("Keluar dari program.")` → menampilkan statement Keluar dari program
- `return` → return
- default: → jika input tidak sesuai dengan pilihan menu
- `fmt.Println("Pilihan tidak valid.")` → menampilkan statement Pilihan tidak valid

3. 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
```

Source Code

```
package main

import "fmt"

func rekapPertandingan(klubA, klubB string,
    pertandingan int, pemenang *[]string) {
    var skorA, skorB int
    fmt.Printf("Pertandingan %d: ",
        pertandingan)
    fmt.Scanln(&skorA, &skorB)

    if skorA < 0 || skorB < 0 {
        return
    }

    if skorA > skorB {
        *pemenang = append(*pemenang,
```

```

        klubA)
    } else if skorA < skorB {
        *pemenang = append(*pemenang,
            klubB)
    } else {
        *pemenang = append(*pemenang,
            "Draw")
    }
    rekapPertandingan(klubA, klubB,
        pertandingan+1, pemenang)
}

func main() {
    var klubA, klubB string
    fmt.Print("Klub A: ")
    fmt.Scanln(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scanln(&klubB)

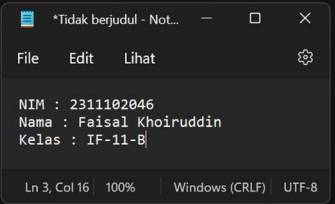
    var pemenang []string
    rekapPertandingan(klubA, klubB, 1,
        &pemenang)

    for i, p := range pemenang {
        fmt.Printf("Hasil %d: %s\n", i+1,
            p)
    }
    fmt.Println("Pertandingan Selesai")
}

```

Screenshots Output

```
PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_3> go run No_3.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 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:\Semester3Go\Praktikum\Modul 6\Unguided\No_3>
```



Deskripsi:

Program tersebut merupakan program merekap skor pertandingan bola 2 buah klub bola yang berlaga. Program tersebut meminta pengguna menginput nama-nama klub yang bertanding, kemudian program meminta masukan skor hasil pertandingan kedua klub tersebut. Program tersebut menampilkan daftar klub yang memenangkan pertandingan.

- package main → paket utama pada program golang
- import "fmt" → import fmt
- func rekapPertandingan(klubA, klubB string, pertandingan int, pemenang *[]string) { → fungsi rekapPertandingan(klubA, klubB string, pertandingan int, pemenang *[]string)
- var skorA, skorB int → deklarasi var skorA, skorB
- fmt.Printf("Pertandingan %d: ", pertandingan) → menampilkan statement Pertandingan
- fmt.Scanln(&skorA, &skorB) → membaca input pengguna
- if skorA < 0 || skorB < 0 { → percabangan if skorA < 0 atau skorB < 0
- Return → return

- `if skorA > skorB {` → percabangan `if skorA > skorB`
- `*pemenang = append(*pemenang, klubA)` → `*pemenang = append(*pemenang, klubA)`
- `}` else `if skorA < skorB {` → kalau tidak jika `skorA < skorB`
- `*pemenang = append(*pemenang, klubB)` → `*pemenang = append(*pemenang, klubB)`
- `}` else `{` → kalau tidak
- `*pemenang = append(*pemenang, "Draw")` → `*pemenang = append(*pemenang, "Draw")`
- `rekapPertandingan(klubA, klubB, pertandingan+1, pemenang)`
→ `rekapPertandingan(klubA, klubB, pertandingan+1, pemenang)`
- `func main() {` → merupakan fungsi utama
- `var klubA, klubB string` → deklarasi `var klubA, klubB`
- `fmt.Print("Klub A: ")` → menginput klub A
- `fmt.Scanln(&klubA)` → membaca input pengguna
- `fmt.Print("Klub B: ")` → menginput klub B
- `fmt.Scanln(&klubB)` → membaca input pengguna
- `var pemenang []string` → `var pemenang []string`
- `rekapPertandingan(klubA, klubB, 1, &pemenang)` →
`rekapPertandingan(klubA, klubB, 1, &pemenang)`
- `for i, p := range pemenang {` → `for i, p := range pemenang`
- `fmt.Printf("Hasil %d: %s\n", i+1, p)` → menampilkan hasil
- `fmt.Println("Pertandingan Selesai")` → menampilkan statement

Pertandingan Selesai

4. Sebuah array digunakan untuk menampung sekumpulan karakter, Anda diminta untuk membuat sebuah 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 is 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 I A K .
Reverse teks	: K A T A K

```
package main

import "fmt"
```

```
const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var ch rune
    fmt.Print("Teks : ")
    for {
        fmt.Scanf("%c", &ch)
        if ch == '.' || *n >= NMAX {
            break
        }
        t[*n] = ch
        *n++
    }
}

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

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

```

func main() {
    var tab tabel
    var m int

    isiArray(&tab, &m)

    fmt.Print("Reverse teks: ")
    balikanArray(&tab, m)
    cetakArray(tab, m)
}

```

Screenshots Output

```

PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_4> go run No_4.go
Teks : S E N A N G .
Reverse teks:  G N A N E S
PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_4> go run No_4.go
Teks : K A T A K .
Reverse teks:  K A T A K
PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_4>

```

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	: <u>K A T A K</u>
Palindrom	? true

Teks	: <u>S</u> <u>E</u> <u>N</u> <u>A</u> <u>N</u> <u>G</u>
Palindrom	? false

Source Code

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "unicode"
)

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var input string
    fmt.Print("Teks: ")
    reader := bufio.NewReader(os.Stdin)
    input, _ = reader.ReadString('\n')

    if len(input) > 0 && input[len(input)-1] == '-' {
        input = input[:len(input)-1]
    }

    for _, ch := range input {
        if unicode.IsSpace(ch) {
            continue
        }
    }
}
```

```

        }
        if *n >= NMAX {
            break
        }
        t[*n] = ch
        *n++
    }
}

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

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

func main() {
    var tab tabel
    var m int

```

```

isiArray(&tab, &m)
balikanArray(&tab, m)

if palindrom(tab, m) {
    fmt.Println("Palindrom ? true")
} else {
    fmt.Println("Palindrom ? false")
}
}

```

Screenshots Output

The screenshot shows a terminal window with the following output:

```

PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_4_Modifikasi> go run No_4_Modifikasi.go
Teks: K A T A K
Palindrom ? true
PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_4_Modifikasi> go run No_4_Modifikasi.go
Teks: S E N A E N G
Palindrom ? false
PS C:\Semester3Go\Praktikum\Modul 6\Unguided\No_4_Modifikasi>

```

Overlaid on the terminal is a Notepad window titled '*Tidak berjudul - Notep...'. It contains the following text:

```

NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-02

```

The Notepad window also shows status information at the bottom: 'Ln 3, Col 17 | 100% | Windows (CRLF) | UTF-8'.

Deskripsi:

Program tersebut merupakan program untuk mengecek kata yang termasuk palindrom atau bukan. Program tersebut meminta pengguna menginput kata dengan huruf kapital semua dan dengan spasi setiap huruf. Program tersebut menampilkan kata yang diinput termasuk palindrom ? true atau palindrom ? false.

- package main ➔ paket utama pada program golang
- import (➔ mengimpor
- "bufio" ➔ mengimpor bufio
- "fmt" ➔ mengimpor fmt
- "os" ➔ mengimpor os
- "unicode" ➔ mengimpor unicode

- `const NMAX int = 127` → menetapkan konstanta `NMAX int = 127`
- `type tabel [NMAX]rune` → mendefinisikan tipe table `[NMAX]rune`
- `func isiArray(t *tabel, n *int) {` → fungsi untuk mengisi array
- `var input string` → deklarasi variabel input bertipe data string
- `fmt.Print("Teks: ")` → menampilkan statement untuk menginput teks
- `reader := bufio.NewReader(os.Stdin)` → `reader := bufio.NewReader(os.Stdin)`
- `input, _ = reader.ReadString('\n')` → membaca baris teks dari input
- `if len(input) > 0 && input[len(input)-1] == '-' {` → percabangan if jika `len(input) > 0 && input[len(input)-1] == '-'`
- `input = input[:len(input)-1]` → input sama dengan `input[:len(input)-1]`
- `for _, ch := range input {` → perulangan for melalui setiap karakter dalam string input
- `if unicode.IsSpace(ch) {` → percabangan if jika `unicode.IsSpace(ch)`
- `continue` → continue
- `if *n >= NMAX {` → memeriksa apakah array sudah mencapai ukuran maksimum
- `Break` → berhenti dan keluar dari perulangan
- `t[*n] = ch` → menambahkan karakter ke array t

- `*n++` → `n++(increment)`
- `func balikanArray(t *tabel, n int) {` → fungsi `balikanArray`
- `for i := 0; i < n/2; i++ {` → perulangan `for` untuk `for i := 0; i < n/2; i++(increment)`
- `t[i], t[n-1-i] = t[n-1-i], t[i]` → menukar elemen
- `func palindrom(t tabel, n int) bool {` → fungsi `palindrom`
- `var reversed table` → deklarasi variabel `reversed`
- `copy(reversed[:], t[:n])` → menyalin elemen `t` ke `reversed`
- `balikanArray(&reversed, n)` → membalikkan array `reversed`
- `for i := 0; i < n; i++ {` → perulangan `for` untuk `for i := 0; i < n; i++(increment)`
- `if t[i] != reversed[i] {` → jika ada karakter tidak cocok
- `return false` → return false
- `return true` → jika semua karakter cocok return true
- `func main() {` → merupakan fungsi utama
- `var tab table` → deklarasi array `tab` dengan tipe `tabel`
- `var m int` → deklarasi variabel `m` bertipe data integer
- `isiArray(&tab, &m)` → memanggil fungsi `isiArray` untuk mengisi array
- `balikanArray(&tab, m)` → memanggil fungsi `balikanArray`
- `if palindrom(tab, m) {` → percabangan `if` jika `palindrom`
- `fmt.Println("Palindrom ? true")` → menampilkan output `Palindrom ? true`
- `} else {` → kalau tidak

- `fmt.Println("Palindrom ? false")` ➔ menampilkan output
Palindrom ? false