

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 7
STRUCK & ARRAY**



Disusun Oleh:

NAMA : MARIA DWI A

NIM : 2311102228

KELAS : S1- 1F – 11 – 02

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

A. Tipe Bentukkan

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

1. 1. Alias (Type)

Bahasa pemograman pada umumnya mengizinkan pemograman untuk mengubah nama suatu tipe dengan nama baru yang lebih ringkas dan familiar. Sebagai contoh “**integer**” dapat diubah 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 :

```
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 }
```

Apabila diperhatikan subprogram **cetak()** di atas, terlihat pada baris ke-4 terdapat pemanggilan subprogram **cetak()** kembali. Misalnya apabila dieksekusi perintah **cetak()** maka akan menampilkan angka 5 6 7 8 9..dst.

B. Struct atau Record

Structure memungkinkan pemogram untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi satu kesatuan. Masing-masing nilai tersimpan dalam fiels 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		

Beberapa dengan bahasa pemrograman lain, kesamaan tipe dari dua variabel berjenis structure 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 structure tersebut didefinisikan sebagai sebuah tipe baru, sehingga 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 }

```

1. 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 **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]
```

2. Slice (Array dinamik)

Array dalam Go juga dapat mempunyai ukuran yang dinamik. (**Tidak digunakan di kelas Algoritma Pemograman**). 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 bild-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 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.

3. Map

Tipe array lain, sebuah array dinamik, indeksnya (di sini disebut **kunci**) dan 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. Source Code :

```
package main

import (

    "fmt"
    "sort"
)

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
}

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])
    }

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

    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)
    }
}

```

Screenshoot Program

```
PS C:\SEMESTER 3\LATIHAN GOLANG> go run "c:\SEMESTER 3\LATIHAN GOLANG\gui
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)
```

Deskripsi Program

Program diatas merupakan program yang digunakan untuk mengelola data mahasiswa dan menghitung peringkat mahasiswa berdasarkan rata-rata nilai dari tiga mata kuliah yaitu matematika, fisika, dan kimia. Program terdiri dari beberapa langkah yaitu mengurutkan data, menghitung rata-rata nilai, mengurutkan mahasiswa berdasarkan rata-rata tertinggi, dan menampilkan peringkat.

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 {

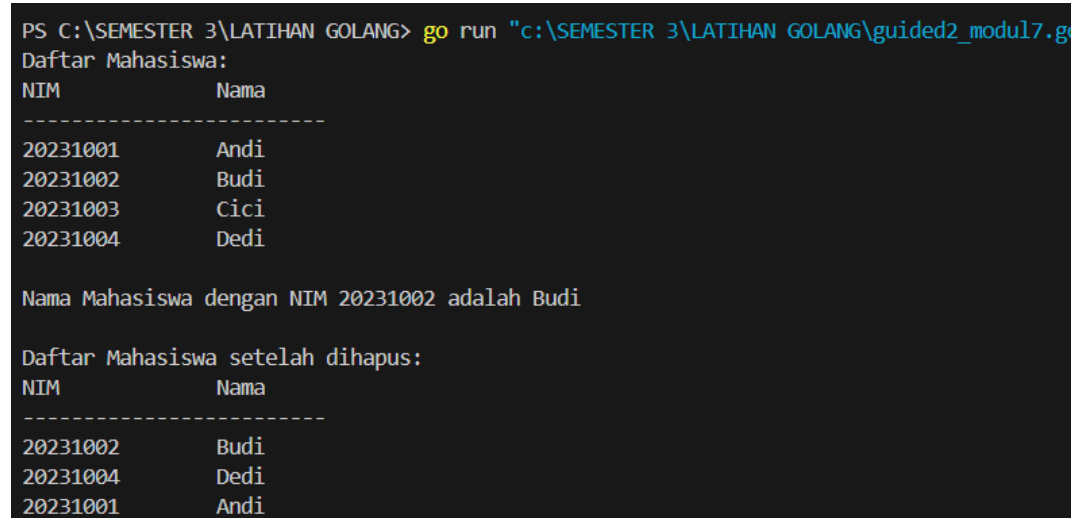
            fmt.Printf("%s\t%s\n", nim, nama)

        }

    }
}

```

Screenshoot Program



PS C:\SEMESTER 3\LATIHAN GOLANG> go run "c:\SEMESTER 3\LATIHAN GOLANG\guided2_modul7.g

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

Deskripsi Program

Program diatas merupakan program yang digunakan untuk mengelola data mahasiswa dengan menggunakan struktur data map, dengan menggunakan NIM sebagai kunci dan Nama sebagai nilai. Program memiliki beberapa fitur, yaitu tambah data, menampilkan data, menampilkan data dalam

bentuk table. Mengakses data berdasarkan NIM, dan menghapus data tertentu.

III. UNGUIDED Soal Modul 7

1. Source code

```
package main

import (

    "fmt"
    "math"
)

func jarak(x1, y1, x2, y2 int) float64 {
    return math.Sqrt(float64((x2-x1)*(y2-1) + (y2-
y1)*(y2-y1)))
}

func didalamLingkaran(x, y, cx, cy, r int) bool{
    distance := jarak(x, y, cx, cy)
    return distance <= float64(r)
}

func main(){
    var cx1, cy1, r1 int
    fmt.Println("Masukkan koordinat x, y, dan radius
dalam lingkaran 1 : ")
    fmt.Scan(&cx1, &cy1, &r1)

    var cx2, cy2, r2 int
    fmt.Println("Masukkan koordinat x, y, dan radius
untuk lingkaran 2 : ")
    fmt.Scan(&cx2, &cy2, &r2)

    var x, y int
    fmt.Println("Masukkan koordinat x dan y untuk
titik sembarang : ")
```

```

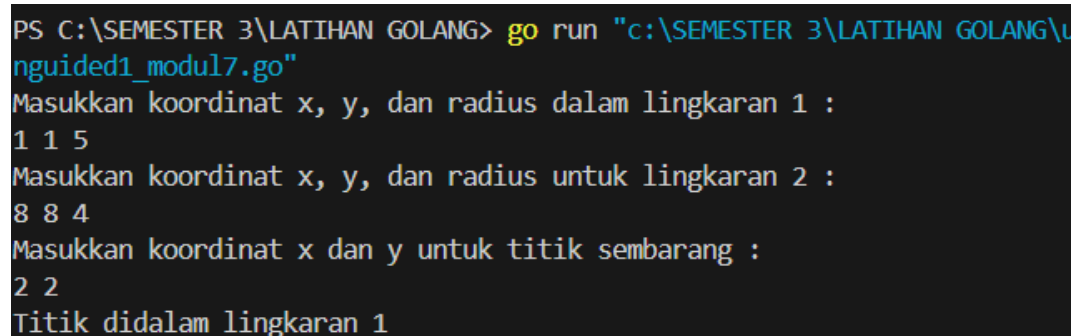
    fmt.Scan(&x, &y)

    didalamLingkaran1 := didalamLingkaran(x, y, cx1,
cy1, r1)
    didalamLingkaran2 := didalamLingkaran(x, y, cx2,
cy2, r2)

    if didalamLingkaran1 && didalamLingkaran2 {
        fmt.Println("Titik didalam lingkaran 1 dan
2")
    } else if didalamLingkaran1{
        fmt.Println("Titik didalam lingkaran 1")
    } else if didalamLingkaran2 {
        fmt.Println("Titik didalam lingkaran 2")
    } else {
        fmt.Println("Titik diluar lingkaran 1 dan 2")
    }
}

```

Screenshoot Program



```

PS C:\SEMESTER 3\LATIHAN GOLANG> go run "c:\SEMESTER 3\LATIHAN GOLANG\unguided1_modul7.go"
Masukkan koordinat x, y, dan radius dalam lingkaran 1 :
1 1 5
Masukkan koordinat x, y, dan radius untuk lingkaran 2 :
8 8 4
Masukkan koordinat x dan y untuk titik sembarang :
2 2
Titik didalam lingkaran 1

```

Deskripsi Program

Program diatas suatu program yang digunakan menentukan suatu titik yang diinputkan user berada didalam lingkaran 1 atau lingkaran 2 atau tidak. Program akan meminta user untuk menginputkan koordinat titik pusat dan

radius lingkaran 1 dan lingkaran 2. Kemudian program juga meminta user untuk menginputkan titik sembarang (x, y) lalu program akan menghitung jarak antara (x, y) dan titik koordinat pusat. Lalu program akan menentukan apakah titik tersebut berada didalam atau diluar lingkaran 1 atau 2.

2. Source code

```
package main

import (
    "fmt"
    "math"
)

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

    //Inisialisasi array
    array := make([]int, N)

    //Mengisi array dengan nilai dari pengguna
    for i := 0; i < N; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i)
        fmt.Scan(&array[i])
    }

    //Menampilkan keseluruhan isi dari array
    fmt.Println("\nIsi array:")
    fmt.Println(array)

    //Menampilkan elemen dengan indeks ganjil
```

```

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

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

    //Menampilkan elemen dengan indeks kelipatan
    bilangan x
    var x int
    fmt.Println("\n\nElemen dengan kelipatan x:")
    fmt.Print("Masukkan bilangan x: ")
    fmt.Scan(&x)
    fmt.Println("Elemen dengan indeks kelipatan", x,
":")
    for i := 0; i < N; i++ {
        if i%x == 0 {
            fmt.Print(array[i], " ")
        }
    }

    //Menghapus elemen pada indeks tertentu
    var indeksHapus int
    fmt.Print("\n\nMasukkan indeks yang akan dihapus:
")
    fmt.Scan(&indeksHapus)
    array = append(array[:indeksHapus],
array[indeksHapus+1:]...)
    fmt.Println("Array setelah dihapus elemen pada
indeks", indeksHapus, ":")

```

```

fmt.Println(array)

//Menghitung dan menampilkan rata-rata
var total int
for _, value := range array {
    total += value
}
rataRata := float64(total) / float64(len(array))
fmt.Printf("\nRata-rata: %.2f\n", rataRata)

//Menghitung dan menampilkan standar deviasi
var variance float64
for _, value := range array {
    variance += math.Pow(float64(value)-rataRata,
2)
}
variance /= float64(len(array))
standarDeviasi := math.Sqrt(variance)
    fmt.Printf("Standar    deviasi:    %.2f\n",
standarDeviasi)

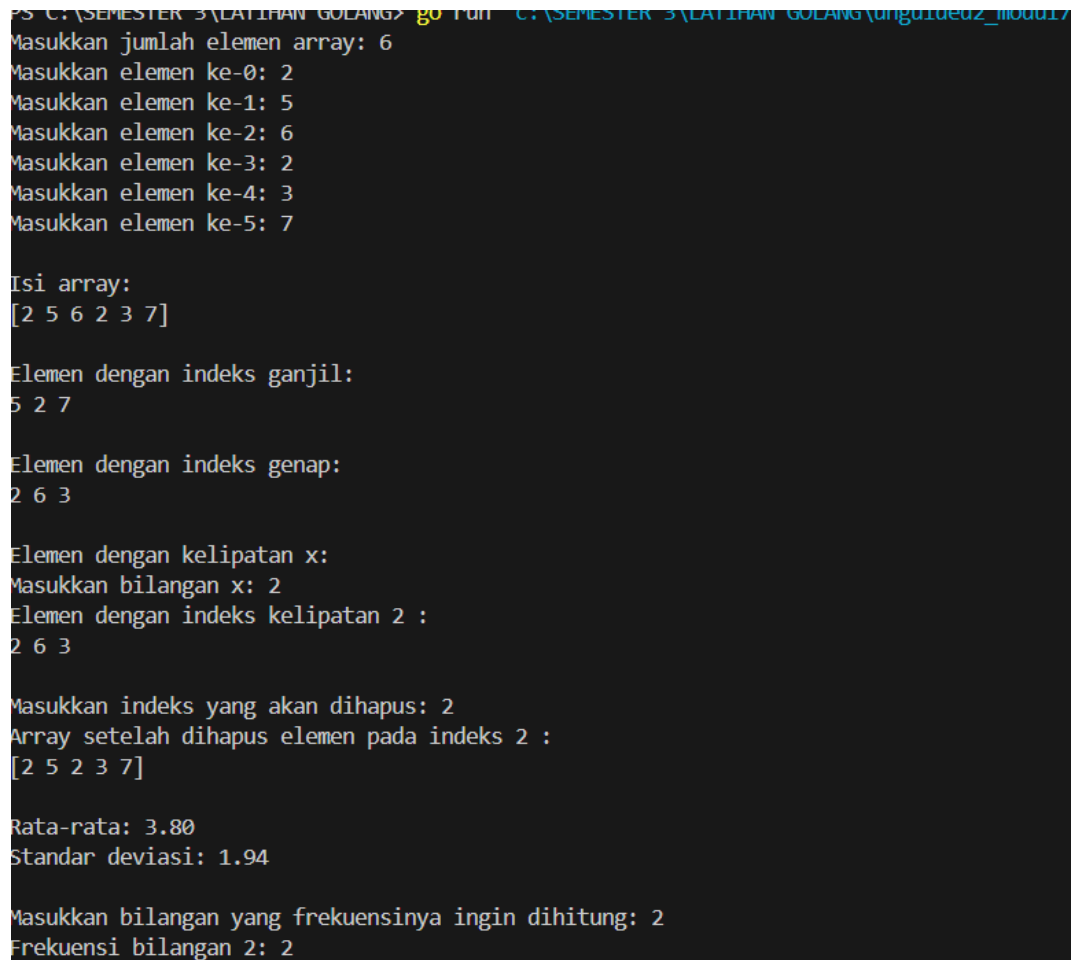
    //Menampilkan frekuensi dari suatu bilangan
tertentu
var bilangan int
    fmt.Print("\nMasukkan bilangan yang frekuensinya
ingin dihitung: ")
    fmt.Scan(&bilangan)
    frekuensi := 0
    for _, value := range array {
        if value == bilangan {
            frekuensi++
        }
    }
}

```



```
        fmt.Printf("Frekuensi bilangan %d: %d\n", bilangan, frekuensi)
    }
}
```

Screenshoot Program



```
PS C:\SEMESTER 3\LATIHAN GOLANG> go run C:\SEMESTER 3\LATIHAN GOLANG\unguided2_modul17.go
Masukkan jumlah elemen array: 6
Masukkan elemen ke-0: 2
Masukkan elemen ke-1: 5
Masukkan elemen ke-2: 6
Masukkan elemen ke-3: 2
Masukkan elemen ke-4: 3
Masukkan elemen ke-5: 7

Isi array:
[2 5 6 2 3 7]

Elemen dengan indeks ganjil:
5 2 7

Elemen dengan indeks genap:
2 6 3

Elemen dengan kelipatan x:
Masukkan bilangan x: 2
Elemen dengan indeks kelipatan 2 :
2 6 3

Masukkan indeks yang akan dihapus: 2
Array setelah dihapus elemen pada indeks 2 :
[2 5 2 3 7]

Rata-rata: 3.80
Standar deviasi: 1.94

Masukkan bilangan yang frekuensinya ingin dihitung: 2
Frekuensi bilangan 2: 2
```

Deskripsi program

Program diatas merupakan program yang digunakan untuk menampung sekumpulan bilangan bulat. Program akan menampilkan beberapa informasi seperti menampilkan isi dari array, menampilkan elemen-elemen array dengan indeks ganjil dan genap, menampilkan elemen array dengan indeks kelipatan bilangan x(inputan pengguna), menghapus elemen-elemen array dengan indeks tertentu, menampilkan rata-rata dari bilangan

yang ada di dalam array, menampilkan standar deviasi atau simpangan baku dari bilangan array dan menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi.

3. Source code

```
package main

import "fmt"

func cariFaktor(n, i int) {
    if i > n {
        return
    }
    if n % i == 0 {
        fmt.Print(i, " ") // jika n dibagi habis, maka
i adalah faktor dari
    }
    cariFaktor(n, i+1)
    fmt.Println()
}

func main(){
    var n int

    fmt.Println("\n*PROGRAM    MENAMPILKAN    FAKTOR
BILANGAN*")

    fmt.Print("\nMasukkan bilangan : ")
    fmt.Scan(&n)

    fmt.Println("\nFaktor dari", n, " adalah:")
    cariFaktor(n, 1)
    fmt.Println()
```

```
}
```

Screenshoot Program

```
PS C:\BELAJAR GO-LANG\PRAKTIKUM ALPRO MODUL 3> go run "c:\BELAJAR GO-L
*PROGRAM MENAMPILKAN FAKTOR BILANGAN*

Masukkan bilangan : 12

Faktor dari 12 adalah:
1 2 3 4 6 12
```

Deskripsi Program

Program diatas merupakan program implementasi dari fungsi rekursif yang digunakan untuk menampilkan faktor dari sebuah bilangan. Program akan meminta user untuk menginputkan sebuah bilangan lalu fungsi rekursif `cariFaktor` akan dipanggil untuk menentukan faktor dari bilangan yang diinputkan dengan syarat rekursif akan berjalan selama $i > n$, jika bilangan yang diinputkan (n) habis membagi i maka (i dimulai dari 1) tersebut termasuk faktor dari bilangan n . Untuk inputan program diatas user menginputkan bilangan 12, lalu program akan memanggil fungsi rekursif `cariFaktor` lalu menampilkan faktor dari 12 yaitu 1 2 3 4 6 12.

4. Source Code

```
package main
import "fmt"

func turun(n int) {
    if n < 1 {
        return
    }
}
```

```
        fmt.Print(n, " ")
        turun(n - 1)
    }

func naik(n, i int) {
    if i > n {
        return
    }
    fmt.Print(i, " ")
    naik(n, i + 1)

    fmt.Println()
}

func main(){

    var n int

        fmt.Println("\n*PROGRAM    MENGHITUNG    BARIS
BILANGAN*")

    fmt.Print("\nMasukkan angka : ")
    fmt.Scan(&n)

    fmt.Println()

    fmt.Println("Barisan bilangan : \n")
    turun(n)
    naik(n, 2)

    fmt.Println()
}
```

```
}
```

Screenshoot Program

```
PS C:\SEMESTER 5\DATIMAN GOLANG> go run C:\SEMESTER 5\DATIMAN GOLANG\unguided3.go
Klub A : MU
Klub B : Inter
Pertandingan 1 : 2 0
Pertandingan 2 : 1 2
Pertandingan 3 : 2 2
Pertandingan 4 : 0 1
Pertandingan 5 : 3 2
Pertandingan 6 : 1 0
Pertandingan 7 : 5 2
Pertandingan 8 : 2 3
Pertandingan 9 : -1 2

Hasil 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!
```

Deskripsi Program

Program diatas merupakan program yang digunakan untuk merekap skor pertandingan bola 2 buah klub yang berlaga. Pertama program akan meminta user memasukkan nama-nama klub yang bertanding, kemudian program akan meminta user memasukkan skor hasil pertandingan kedua klub tersebut. Array dalam program akan menyimpan nama-nama yang menang saja. Proses input skor berhenti ketika skor salah satu atau kedua klub bernilai negative. Lalu pada akhir program akan menampilkan daftar daftar klub yang memenangkan pertandingan.

5. Souce Code

```
package main

import "fmt"

const NMAX int = 127
```

```

type tabel [NMAX] rune

func isiArray(t *tabel, n *int) {
    var input string

    fmt.Print("Masukkan teks : ")
    fmt.Scanln(&input)
    *n = len(input)
    for i:= 0; i < *n; i++){
        t[i] = rune(input[i])
    }
}

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-i-1] = t[n-i-1], t[i]
    }
}

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

```

```
        return true
    }

func main(){
    var tab tabel
    var m int

    isiArray(&tab, &m)

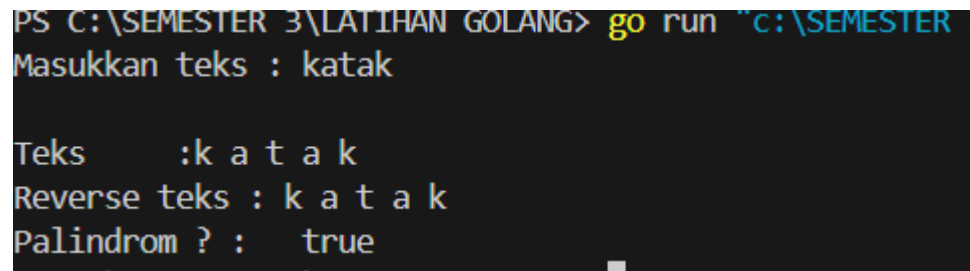
    fmt.Print("\nTeks      :")
    cetakArray(tab, m)

    // balikan array
    fmt.Print("Reverse teks : ")
    balikanArray(&tab, m)
    cetakArray(tab, m)

    if palindrom(tab, m) {
        fmt.Println("Palindrom ? :   true")

    } else {
        fmt.Println("Palindrom ? :   false")
    }
}
```

Screenshoot Program



```
PS C:\SEMESTER 3\LATIHAN GOLANG> go run "c:\SEMESTER 3\LATIHAN GOLANG\main.go"
Masukkan teks : katak

Teks      :k a t a k
Reverse teks : k a t a k
Palindrom ? :  true
```

Deskripsi Program

Program diatas merupakan program yang menerima input teks dari user, lalu program akan melakukan operasi dasar pada teks tersebut, program akan menampilkan teks asli, lalu akan akan memablik urutan teks dan memeriksa apakah teks yang diinputkan palindrom atau bukan (teks yang sama jika dibaca dari depan ataupun belakang).