

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCT & ARRAY**



Disusun Oleh :

Aji Noto Sutrisno (2311102262)

IF 11 05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

A. STRUCT

Struct adalah kumpulan definisi variabel (atau *property*) dan atau fungsi (atau *method*), yang dibungkus sebagai tipe data baru dengan nama tertentu. Property dalam struct, tipe datanya bisa bervariasi. Mirip seperti map, hanya saja key-nya sudah didefinisikan di awal, dan tipe data tiap itemnya bisa berbeda.

Dari sebuah *struct*, kita bisa buat variabel baru, yang memiliki atribut sesuai skema struct tersebut. Kita sepakati dalam buku ini, variabel tersebut dipanggil dengan istilah object atau variabel object.

Deklarasi Struct :

```
type student struct {  
    name string  
    grade int  
}
```

Penerapan Struct Untuk Membuat Object

```
func main() {  
    var s1 student  
    s1.name = "john wick"  
    s1.grade = 2  
  
    fmt.Println("name :", s1.name)  
    fmt.Println("grade :", s1.grade)  
}
```

Inisialisasi Object Struct

```
var s1 = student{}  
s1.name = "wick"  
s1.grade = 2  
  
var s2 = student{"ethan", 2}  
  
var s3 = student{name: "jason"}  
  
fmt.Println("student 1 :", s1.name)
```

```
fmt.Println("student 2 :", s2.name)
fmt.Println("student 3 :", s3.name)
```

Pada kode di atas, variabel `s1` menampung objek cetakan `student`. Variabel tersebut kemudian di-set nilai property-nya. Variabel objek `s2` dideklarasikan dengan metode yang sama dengan `s1`, pembedanya di `s2` nilai propertinya diisi langsung ketika deklarasi. Nilai pertama akan menjadi nilai property pertama (yaitu `name`), dan selanjutnya berurutan.

Pada deklarasi `s3`, dilakukan juga pengisian property ketika pencetakan objek. Hanya saja, yang diisi hanya `name` saja. Cara ini cukup efektif jika digunakan untuk membuat objek baru yang nilai property-nya tidak semua harus disiapkan di awal. Keistimewaan lain menggunakan cara ini adalah penentuan nilai property bisa dilakukan dengan tidak berurutan. Contohnya:

```
var s4 = student{name: "wayne", grade: 2}
var s5 = student{grade: 2, name: "bruce"}
```

B. ARRAY

Array adalah kumpulan data bertipe sama, yang disimpan dalam sebuah variabel. Array memiliki kapasitas yang nilainya ditentukan pada saat pembuatan, menjadikan elemen/data yang disimpan di array tersebut jumlahnya tidak boleh melebihi yang sudah dialokasikan.

Default nilai tiap elemen array pada awalnya tergantung dari tipe datanya. Jika `int` maka tiap element *zero* value-nya adalah 0, jika `bool` maka `false`, dan seterusnya. Setiap elemen array memiliki indeks berupa angka yang merepresentasikan posisi urutan elemen tersebut. Indeks array dimulai dari 0.

Contoh penereapan Array :

```
var names [4]string
names[0] = "trafalgar"
names[1] = "d"
names[2] = "water"
names[3] = "law"
```

```
fmt.Println(names[0], names[1], names[2], names[3])
```

Pengisian Elemen Array yang Melebihi Alokasi Awal. Pengisian elemen array pada indeks yang tidak sesuai dengan jumlah alokasi menghasilkan error. Contoh: jika array memiliki 4 slot, maka pengisian nilai slot 5 seterusnya adalah tidak valid.

```
var names [4]string
names[0] = "trafalgar"
names[1] = "d"
names[2] = "water"
names[3] = "law"
names[4] = "ez" // baris kode ini menghasilkan error
```

Inisialisasi Nilai Awal Array. Pengisian elemen array bisa dilakukan pada saat deklarasi variabel. Caranya dengan menuliskan data elemen dalam kurung kurawal setelah tipe data, dengan pembatas antar elemen adalah tanda koma (.).

```
var fruits = [4]string{"apple", "grape", "banana",
"melon"}

fmt.Println("Jumlah element \t\t", len(fruits))
fmt.Println("Isi semua element \t", fruits)
```

Inisialisasi Nilai Awal Array Tanpa Jumlah Elemen. Deklarasi array yang nilainya diset di awal, boleh tidak dituliskan jumlah lebar array-nya, cukup ganti dengan tanda 3 titik (...). Metode penulisan ini membuat kapasitas array otomatis dihitung dari jumlah elemen array yang ditulis.

```
var numbers = [...]int{2, 3, 2, 4, 3}

fmt.Println("data array \t:", numbers)
fmt.Println("jumlah elemen \t:", len(numbers))
```

Perulangan Elemen Array Menggunakan Keyword for – range. Contohnya:

```
var fruits = [4]string{"apple", "grape", "banana",
"melon"}

for i, fruit := range fruits {
    fmt.Printf("elemen %d : %s\n", i, fruit)
}
```

```
var fruits = [4]string{"apple", "grape", "banana",  
"melon"}  
  
for _, fruit := range fruits {  
    fmt.Printf("nama buah : %s\n", fruit)  
}
```

II. GUIDED

1. Soal Studi Case

Sourcecode

```
package main

import "fmt"

type waktu struct {
    jam, menit, detik int
}

func main() {
    var wParkir, wPulang, durasi waktu
    var dParkir, dPulang, lParkir int

    fmt.Scan(&wParkir.jam, &wParkir.menit,
&wParkir.detik)
    fmt.Scan(&wPulang.jam, &wPulang.menit,
&wPulang.detik)

    dParkir = wParkir.detik + wParkir.menit*60 +
wParkir.jam*3600
    dPulang = wPulang.detik + wPulang.menit*60 +
wPulang.jam*3600
    lParkir = dPulang - dParkir

    durasi.jam = lParkir / 3600
    durasi.menit = lParkir % 3600 / 60
    durasi.detik = lParkir % 3600 % 60

    fmt.Printf("Lama Parkir: %d jam %d detik %d",
durasi.jam, durasi.menit, durasi.detik)
}
```

Output

```
PS D:\Pratikum Alpro 2\Laprak 7> go run "d:\Pratikum Alpro 2\Laprak 7\Guided 1\Guided1.go"
2 43 45
Lama Parkir: 41 jam 0 detik 0
PS D:\Pratikum Alpro 2\Laprak 7> go run "d:\Pratikum Alpro 2\Laprak 7\Guided 1\Guided1.go"
2 43 45
3 59 35
Lama Parkir: 1 jam 15 detik 50
```

Deskripsi

Program dibuat untuk mencari lama parkir dari sebuah kendaraan. Program ini menggunakan deklarasi struct yaitu `type waktu struct` didalam struct tersebut memiliki 3 variabel yaitu `jam`, `menit`, `detik int`. Dalam fungsi `main`, tiga variabel bertipe waktu, yaitu `wParkir`, `wPulang`, `durasi waktu`, digunakan untuk merepresentasikan waktu parkir awal, waktu parkir akhir, dan durasi parkir.

Program meminta input waktu parkir awal (`wParkir`) dan waktu parkir akhir (`wPulang`) dari pengguna dalam bentuk jam, menit, dan detik. Setelah itu, waktu ini dikonversi menjadi detik dengan rumus:

$$\text{Total Detik} = \text{detik} + (\text{menit} \times 60) + (\text{jam} \times 3600)$$

Hasil konversi dari waktu masuk disimpan dalam `dParkir`, dan waktu keluar disimpan dalam `dPulang`. Selisih antara `dPulang` dan `dParkir` dihitung untuk mendapatkan durasi total dalam detik dan disimpan dalam variabel `lParkir`.

Selanjutnya, durasi dalam detik (`lParkir`) dikonversi kembali menjadi jam, menit, dan detik. Jam dihitung dengan membagi `lParkir` dengan 3600. Sisa setelah dikurangi durasi jam dihitung untuk mendapatkan menit, dengan membagi hasil sisa tersebut dengan 60. Terakhir, sisa setelah pengurangan menit dihitung untuk mendapatkan detik. Hasil perhitungan ini disimpan dalam variabel `durasi` dan outputnya "Lama Parkir: X jam Y menit Z detik."

2. Soal Studi Case

Sourcecode

```
// Guided 2 - Slice
package main

import (
    "fmt"
)

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

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

    // Nama-nama baru yang ingin ditambahkan
```

```

        namaBaru := []string{"Dewi", "Budi", "Eka"}

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

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

```

Screenshoot Output

```

PS D:\Pratikum Alpro 2\Laprak 7> go run "d:\Pratikum Alpro 2\Laprak 7\Guided 2\Guided2.go"
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\Pratikum Alpro 2\Laprak 7>

```

Deskripsi Program

Program ini dibuat untuk menambahkan data array kepada array yang sudah ada.

Program ini menambahkan `namaBaru := []string{"Dewi", "Budi", "Eka"}` ke `daftarTeman := []string{"Andi", "Budi", "Cici"}` dengan menggunakan perulangan beserta fungsi *append*, dan ketika ada yang sama maka program akan menampilkan output "Nama budi sudah ada dalam daftar".

3. Soal Studi Case

Sourcecode

```

// Guided 2 - Map
package main

import "fmt"

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

    // Menampilkan harga dari setiap buah

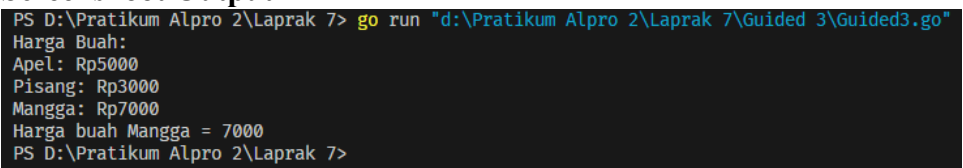
```



```
    fmt.Println("Harga Buah:")
    for buah, harga := range hargaBuah {
        fmt.Printf("%s: Rp%d\n", buah, harga)
    }

    fmt.Print("Harga buah Mangga = ",
hargaBuah["Mangga"])
}
```

Screenshoot Output



```
PS D:\Pratikum Alpro 2\Laparak 7> go run "d:\Pratikum Alpro 2\Laparak 7\Guided 3\Guided3.go"
Harga Buah:
Apel: Rp5000
Pisang: Rp3000
Mangga: Rp7000
Harga buah Mangga = 7000
PS D:\Pratikum Alpro 2\Laparak 7>
```

Deskripsi Program

Program ini merupakan implementasi dari struktur data map yang digunakan untuk menyimpan data harga buah, nama buah menjadi *key*, dan harga menjadi *value*. Definisi map `hargaBuah := map[string]int{ mencakup apel, 5000 pisang 3000 dan mangga 7000.`

III. UNGUIDED

1. Soal Studi Case

Suatu lingkaran didefinisikan dengan koordinat titik pusat (c_x , c_y) 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 radlusnya.

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 15 20	Titik di luar lingkaran 1 dan 2

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

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

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.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

type Lingkaran struct {
    cx, cy, r float64
}

type Titik struct {
    x, y float64
}

func jarak(a, b, c, d float64) float64 {
    return math.Sqrt((a-c)*(a-c) + (b-d)*(b-d))
}

func (l Lingkaran) didalam(t Titik) bool {
    return jarak(l.cx, l.cy, t.x, t.y) <= l.r
}

func main() {
    var l1 Lingkaran
    fmt.Println("Lingkaran 1")
    fmt.Print("(cx) (cy) (r) : ")
    fmt.Scanln(&l1.cx, &l1.cy, &l1.r)

    var l2 Lingkaran
    fmt.Println("Lingkaran 2")
```

```

    fmt.Print("(cx) (cy) (r) : ")
    fmt.Scanln(&l2.cx, &l2.cy, &l2.r)

    var titik Titik
    fmt.Print("Input koordinat sembarang (x) (y) : ")
    fmt.Scanln(&titik.x, &titik.y)

    posisi1 := l1.didalam(titik)
    posisi2 := l2.didalam(titik)

    if posisi1 && posisi2 {
        fmt.Println("Titik di dalam lingkaran 1 dan
2")
    } else if posisi1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if posisi2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

Screenshoot Output

```

PS D:\Pratikum Alpro 2\Laprak 7> go run "d:\Pratikum Alpro 2\Laprak 7\Unguguided 1\Unguguided1.go"
Lingkaran 1
Lingkaran 2
(cx)(cy)(r) : 8 8 4
Input koordinat sembarang (x)(y) : 2 2
Titik di dalam lingkaran 1
PS D:\Pratikum Alpro 2\Laprak 7> go run "d:\Pratikum Alpro 2\Laprak 7\Unguguided 1\Unguguided1.go"
Lingkaran 1
(cx)(cy)(r) : 1 2 3
Lingkaran 2
(cx)(cy)(r) : 4 5 6
Input koordinat sembarang (x)(y) : 7 8
Titik di dalam lingkaran 2
PS D:\Pratikum Alpro 2\Laprak 7> go run "d:\Pratikum Alpro 2\Laprak 7\Unguguided 1\Unguguided1.go"
Lingkaran 1
(cx)(cy)(r) : 5 10 15
Lingkaran 2
(cx)(cy)(r) : -15 4 20
Input koordinat sembarang (x)(y) : 0 0
Titik di dalam lingkaran 1 dan 2
PS D:\Pratikum Alpro 2\Laprak 7> go run "d:\Pratikum Alpro 2\Laprak 7\Unguguided 1\Unguguided1.go"
Lingkaran 1
(cx)(cy)(r) : 1 1 5
Lingkaran 2
(cx)(cy)(r) : 8 8 4
Input koordinat sembarang (x)(y) : 15 20
Titik di luar lingkaran 1 dan 2

```

Deskripsi Program

Program ini dibuat untuk menentukan apakah titik koordinat berada pada lingkaran 1 atau lingkaran 2, atau di tengah keduanya. Program ini memiliki beberapa fungsi yaitu func jarak(a, b, c, d float64) dan func (l Lingkaran) didalam(t Titik) bool. Program ini juga memiliki struct

berupa type Lingkaran struct dan type Titik struct. Cara kerja Program adalah

- User menginputkan (cx, cy, r) dan (x,y) untuk lingkaran 1 dan lingkaran 2
- Kemudian program akan mengecek apakah koordinat masih didalam radius lingkaran "(cx, cy, x, y) <= l.r"
- Setelah true, maka program akan menjalankan fungsi jarak(a, b, c, d float64) untuk mencari jarak setiap lingkaran.
- Kemudian program akan menjalankan hasilnya dengan percabangan yang dimana jika lingkaran1 && lingkaran2, maka ("Titik di dalam lingkaran 1 dan 2"), jika lingkaran1 ("Titik di dalam lingkaran 1"), jika lingkaran2("Titik di dalam lingkaran 2") jika tidak ketiganya maka ("Titik di luar lingkaran 1 dan 2")

2. Soal Studi Case

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.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

type NilaiBil struct {
    data []int
}

func (jia *NilaiBil) tampilkanSeluruhIsi() {
    fmt.Println("Isi array:", jia.data)
}

func (jia *NilaiBil) elemenIndeksGanjil() {
    fmt.Print("Elemen array dengan indeks ganjil: ")
    for i := 1; i < len(jia.data); i += 2 {
        fmt.Print(jia.data[i], " ")
    }
    fmt.Println()
}

func (jia *NilaiBil) elemenIndeksGenap() {
    fmt.Print("Elemen array dengan indeks genap: ")
    for i := 0; i < len(jia.data); i += 2 {
        fmt.Print(jia.data[i], " ")
    }
    fmt.Println()
}

func (jia *NilaiBil) tampilkanKelipatanX() {
    var x int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)
    fmt.Print("Elemen array yang merupakan kelipatan ",
x, ": ")
    for _, v := range jia.data {
        if v%x == 0 {
            fmt.Print(v, " ")
        }
    }
    fmt.Println()
}

func (jia *NilaiBil) hapusElemenIndeks() {
    var idx int
    fmt.Print("Masukkan indeks yang akan dihapus: ")
    fmt.Scan(&idx)
    if idx < 0 || idx >= len(jia.data) {
        fmt.Println("Indeks tidak valid.")
    }
}
```

```

        return
    }
    jia.data = append(jia.data[:idx],
jia.data[idx+1:]...)
    fmt.Println("Array setelah penghapusan:", jia.data)
}

func (jia *NilaiBil) hitungRataRata() {
    total := 0
    for _, v := range jia.data {
        total += v
    }
    rata := float64(total) / float64(len(jia.data))
    fmt.Printf("Rata-rata isi array: %.2f\n", rata)
}

func (jia *NilaiBil) hitungSimpanganBaku() {
    total := 0
    for _, v := range jia.data {
        total += v
    }
    rata := float64(total) / float64(len(jia.data))
    var totalKuadrat float64
    for _, v := range jia.data {
        totalKuadrat += math.Pow(float64(v)-rata, 2)
    }
    simpanganBaku := math.Sqrt(totalKuadrat /
float64(len(jia.data)))
    fmt.Printf("Simpangan baku dari array: %.2f\n",
simpanganBaku)
}

func (jia *NilaiBil) tampilkanFrekuensi() {
    var target int
    fmt.Print("Masukkan bilangan yang ingin dicari
frekuensinya: ")
    fmt.Scan(&target)
    frekuensi := 0
    for _, v := range jia.data {
        if v == target {
            frekuensi++
        }
    }
    fmt.Printf("Frekuensi bilangan %d dalam array: %d
kali\n", target, frekuensi)
}

func (jia *NilaiBil) pilihanmenu() {

    fmt.Println("\n-----
-----")
    jia.tampilkanSeluruhIsi()
}

```

```

        fmt.Println("\n-----")
        -----")
        jia.elemenIndeksGanjil()
        fmt.Println("\n-----")
        -----")
        jia.elemenIndeksGenap()
        fmt.Println("\n-----")
        -----")
        jia.tampilkanKelipatanX()
        fmt.Println("\n-----")
        -----")
        jia.hapusElemenIndeks()
        fmt.Println("\n-----")
        -----")
        jia.hitungRataRata()
        fmt.Println("\n-----")
        -----")
        jia.hitungSimpanganBaku()
        fmt.Println("\n-----")
        -----")
        jia.tampilkanFrekuensi()
        fmt.Println("\n-----")
        -----")

    }

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

    bilBul := NilaiBil{data: make([]int, N)}

    for i := 0; i < N; i++ {
        fmt.Print("Masukkan bilangan bulat indeks ke-
", i, " : ")
        fmt.Scan(&bilBul.data[i])
    }

    bilBul.pilihanmenu()
}

```


Screenshoot Output

```
PS D:\Pratikum Alpro 2\Laparak 7> go run "d:\Pratikum Alpro 2\Laparak 7\Unguided 2\Unguided2.go"
Masukkan ukuran array: 12
Masukkan bilangan bulat indeks ke-0 : 3
Masukkan bilangan bulat indeks ke-1 : 4
Masukkan bilangan bulat indeks ke-2 : 7
Masukkan bilangan bulat indeks ke-3 : 4
Masukkan bilangan bulat indeks ke-4 : 234
Masukkan bilangan bulat indeks ke-5 : 4
Masukkan bilangan bulat indeks ke-6 :
8
Masukkan bilangan bulat indeks ke-7 : 6
Masukkan bilangan bulat indeks ke-8 : 4
Masukkan bilangan bulat indeks ke-9 : 2
Masukkan bilangan bulat indeks ke-10 : 7
Masukkan bilangan bulat indeks ke-11 : 9

-----
Isi array: [3 4 7 4 234 4 8 6 4 2 7 9]

-----
Elemen array dengan indeks ganjil: 4 4 4 6 2 9

-----
Elemen array dengan indeks genap: 3 7 234 8 4 7

-----
Masukkan nilai x: 3
Elemen array yang merupakan kelipatan 3: 3 234 6 9

-----
Masukkan indeks yang akan dihapus: 4
Array setelah penghapusan: [3 4 7 4 8 6 4 2 7 9]

-----
Rata-rata isi array: 5.27

-----
Simpangan baku dari array: 2.14

-----
Masukkan bilangan yang ingin dicari frekuensinya: 4
Frekuensi bilangan 4 dalam array: 4 kali

-----
PS D:\Pratikum Alpro 2\Laparak 7> |
```

Deskripsi Program

Program ini dirancang untuk mengolah data berupa array bilangan bulat dengan menggunakan *Struct* bernama *NilaiBil* digunakan untuk menyimpan data array. Cara kerja program dengan meminta *user* untuk menentukan ukuran array dan memasukkan elemen-elemennya. Setelah itu, program akan menampilkan berbagai operasi pada array dengan sebuah fungsi. Seperti :

- `jia.tampilkanSeluruhIsi()` Menampilkan seluruh elemen array.
- `jia.elemenIndeksGanjil()` Menampilkan elemen-elemen array yang berada pada indeks ganjil.
- `jia.elemenIndeksGenap()` Menampilkan elemen-elemen array yang berada pada indeks genap.

- `jia.tampilkanKelipatanX()` Menampilkan elemen array yang merupakan kelipatan dari bilangan tertentu (x) yang diinputkan oleh pengguna.
- `jia.hapusElemenIndeks()` Menghapus elemen array berdasarkan indeks yang diinputkan oleh *user*.
- `jia.hitungRataRata()` Menghitung rata-rata dari seluruh elemen array.
- `jia.hitungSimpanganBaku()` Menghitung simpangan baku dari elemen-elemen array.
- `jia.tampilkanFrekuensi()` Menampilkan jumlah kemunculan bilangan tertentu yang diinputkan oleh pengguna dalam array.

Fungsi fungsi diatas memiliki kegunaan kegunaan yang sesuai dengan nama fungsinya

3. Soal Studi Case

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

```

Sourcecode

```

package main

import "fmt"

type Pertandingan struct {
    clubA    string
    clubB    string
    SkorA    int
    SkorB    int
    Pemenang string
}

func main() {
    var clubA, clubB string
    var skorA, skorB int
    var hasil []Pertandingan

    fmt.Print("Klub A: ")
    fmt.Scan(&clubA)
    fmt.Print("Klub B: ")
    fmt.Scan(&clubB)

    i := 1

```

```

    for {
        fmt.Printf("Pertandingan %d : ", i)
        fmt.Scan(&skorA, &skorB)
        if skorA < 0 || skorB < 0 {
            break
        }

        var pemenang string
        if skorA > skorB {
            pemenang = clubA
        } else if skorA < skorB {
            pemenang = clubB
        } else {
            pemenang = "Draw"
        }

        hasil = append(hasil, Pertandingan{
            clubA:    clubA,
            clubB:    clubB,
            SkorA:    skorA,
            SkorB:    skorB,
            Pemenang: pemenang,
        })
        i++
    }

    for j, jia := range hasil {
        fmt.Printf("Pertandingan %d: %s %d - %d %s,
Pemenang: %s\n",
            j+1, jia.clubA, jia.SkorA, jia.SkorB,
            jia.clubB, jia.Pemenang)
    }
}

```

Screenshoot Output

```
PS D:\Pratikum Alpro 2\Laprak 7> go run "d:\Pratikum Alpro 2\Laprak 7\Unguided 3\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
Pertandingan 1: MU 2 - 0 Inter, Pemenang: MU
Pertandingan 2: MU 1 - 2 Inter, Pemenang: Inter
Pertandingan 3: MU 2 - 2 Inter, Pemenang: Draw
Pertandingan 4: MU 0 - 1 Inter, Pemenang: Inter
Pertandingan 5: MU 3 - 2 Inter, Pemenang: MU
Pertandingan 6: MU 1 - 0 Inter, Pemenang: MU
Pertandingan 7: MU 5 - 2 Inter, Pemenang: MU
Pertandingan 8: MU 2 - 3 Inter, Pemenang: Inter
PS D:\Pratikum Alpro 2\Laprak 7> █
```

Deskripsi Program

Program di atas digunakan untuk melihat hasil pertandingan sepak bola antara dua klub. Program ini memiliki struct type `Pertandingan struct`. Cara kerja Program ini sebagai berikut :

- *User* menginputkan nama dua klub, yaitu Klub A dan Klub B, kemudian secara berulang memasukkan skor pertandingan mereka kedalam array `hasil []Pertandingan`.
- Kemudian data inputan *user* di setiap pertandingan akan dicatat dalam `hasil []Pertandingan`, yang mencakup nama kedua klub, skor masing-masing, dan pemenang pertandingan (atau hasil "Draw" jika skor seri).
- Proses pencatatan skor berlangsung hingga pengguna memasukkan skor negatif sebagai penanda untuk berhenti.
- Kemudian Program akan menampilkan semua hasil pertandingan disimpan dalam slice bertipe `hasil []Pertandingan`. Yang berisi daftar semua pertandingan yang telah dimainkan, termasuk skor masing-masing klub dan pemenang untuk setiap pertandingan.

4. Soal Studi Case

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
    // isi array tab dengan memanggil prosedur isiArray

    // Balikian isi array tab dengan memanggil balikanArray

    // Cetak isi 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 T A K .
Reverse teks	: K A T A K

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	: K A T A K
Palindrom	? true
Teks	: S E N A N G
Palindrom	? false

Sourcecode

```
package main

import (
    "fmt"
    "unicode"
)

const NMAX int = 127

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

func (jia *Ajinoto) InputArr() {
    fmt.Println("Masukkan karakter (tuliskan '.' untuk berhenti):")
    for jia.m < NMAX {
        var input rune
        fmt.Scanf("%c", &input)
        if unicode.IsSpace(input) {
            continue
        }
        if input == '.' {
            break
        }
        jia.tabel[jia.m] = unicode.ToLower(input)
        jia.m++
    }
}

func (jia *Ajinoto) PrintArray() {
    for i := 0; i < jia.m; i++ {
        fmt.Printf("%c", jia.tabel[i])
    }
    fmt.Println()
}

func (jia *Ajinoto) ReverseArray() Ajinoto {
    var terbalik Ajinoto
    terbalik.m = jia.m
    for i := 0; i < jia.m; i++ {
        terbalik.tabel[i] = jia.tabel[jia.m-1-i]
    }
    return terbalik
}

func (jia *Ajinoto) Palindrom(terbalik Ajinoto) bool {
    for i := 0; i < jia.m; i++ {
        if jia.tabel[i] != terbalik.tabel[i] {
```

```

        return false
    }
}
return true
}

func main() {
    var tab Ajinoto
    tab.InputArr()
    fmt.Print("Array: ")
    tab.PrintArray()
    tabKebalik := tab.ReverseArray()
    fmt.Print("Array yang sudah dibalik: ")
    tabKebalik.PrintArray()
    if tab.Palindrom(tabKebalik) {
        fmt.Println("TRUE")
    } else {
        fmt.Println("FALSE")
    }
}
}

```

Screenshoot Output

```

PS D:\Pratikum Alpro 2\Laparak 7> go run "d:\Pratikum Alpro 2\Laparak 7\Unguided 4\Unguided4.go"
Masukkan karakter (tuliskan '.' untuk berhenti):
S E N A N G .
Array: senang
Array yang sudah dibalik: gnanes
FALSE
PS D:\Pratikum Alpro 2\Laparak 7> go run "d:\Pratikum Alpro 2\Laparak 7\Unguided 4\Unguided4.go"
Masukkan karakter (tuliskan '.' untuk berhenti):
K A T A K .
Array: katak
Array yang sudah dibalik: katak
TRUE

```

Deskripsi Program

Program digunakan untuk memeriksa sebuah kata yang diinputkan oleh *user* tersebut sebuah kata palindrom. Program ini menggunakan Struct data bernama `type Ajinoto struct`, yang memiliki array berkapasitas maksimal (127) untuk menyimpan karakter dan variabel integer `m` untuk mencatat jumlah karakter yang diinputkan. Cara kerja program ini adalah

- *user* memasukkan karakter satu per satu, dengan karakter '.' sebagai penanda akhir input.
- Setelah *user* menginputkan, program mencetak array asli dan versi terbaliknya. Fungsi `func (jia *Ajinoto) ReverseArray()` `Ajinoto` membuat salinan array dalam urutan terbalik, sementara fungsi


```
func (jia *Ajinoto) Palindrom(terbalik Ajinoto) bool
```

membandingkan array asli dengan array terbalik untuk menentukan apakah kata tersebut membentuk palindrom.

- Jika array tersebut adalah palindrom, program mencetak "TRUE"; jika tidak, mencetak "FALSE".

IV. REFERENSI

- [1] N. Agung, *Dasar Pemrograman Golang*, "Array,"
<https://dasarpemrogramangolang.novalagung.com/A-array.html>.
Diakses pada 17 Nopember 2024
- [2] N. Agung, *Dasar Pemrograman Golang*, "Struct,".
<https://dasarpemrogramangolang.novalagung.com/A-struct.html>.
Diakses pada 17 Nopember 2024.