

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7
STRUCK & ARRAY**



Disusun Oleh :

Zahra Tsurroya Poetri / 231110217

IF 11 - 05

Dosen Pengampu :

Arif Amrulloh

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Tipe Bentukkan

Tipe bentukkan memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipe bentukkan 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!

```
package main
import "fmt"
type bilangan int
type pecahan float64
func main(){
    var a,b bilangan
    var hasil pecahan
    a = 9
    b = 5
    hasil pecahan (a) / pecahan (b)
    fmt.Println(hasil)
}
```

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> <tiipe data>	<field 1> <tiipe data>
4	<field 2> <tiipe data>	<field 2> <tiipe data>
5	<field 3> <tiipe data>	<field 3> <tiipe 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.

```
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, &Parkir.detik)
    fmt.Scan(&wPulang.jam, &wPulang.menit, &Pulang.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 menit %d detik",
durasi.jam, durasi.menit, durasi.detik)
}

```

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

```
// Mengganti isi elemen ke-0 dengan nilai dari elemen ke-7
arr[0] = arr[7]

// Mengambil data field x dari elemen ke-I
currX arr[i].center.x

// Mengambil elemen terakhir
n := len(arr)
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 terberituk dengan mengambil slice dari suatu array atau slice yang lain.

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

Map

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

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

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

II. GUIDED

Guided 1 : Program Menghitung Durasi Parkir Berdasarkan Waktu Masuk dan Waktu Keluar

Sourcecode

```
package main

import "fmt"

type waktu struct {
    jam, menit, detik int
}

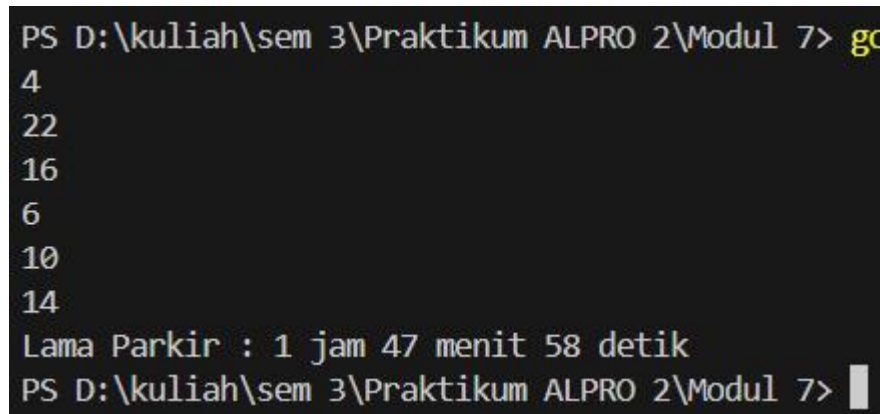
func main() {
    var wParkir, wPulang, durasi waktu
    var dParkir, dPulang, lParkir int

    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 // Konversi ke detik
    dPulang = wPulang.detik + wPulang.menit*60 +
wPulang.jam*3600 // detik
    lParkir = dPulang - dParkir
    //detik dari pulang-datang

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

```
        durasi.detik = lParkir % 3600 % 60 //17
        fmt.Printf("Lama Parkir : %d jam %d menit %d
detik", durasi.jam, durasi.menit, durasi.detik)
    }
```

Screenshoot Output



```
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> go
4
22
16
6
10
14
Lama Parkir : 1 jam 47 menit 58 detik
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> █
```

Deskripsi Program

Program di atas merupakan program menggunakan struct untuk menghitung durasi parkir berdasarkan waktu kedatangan dan kepulangan yang dimasukkan oleh pengguna. Pengguna diminta untuk memasukkan waktu kedatangan dan kepulangan dalam format jam, menit, dan detik, yang kemudian disimpan dalam variabel `wParkir` dan `wPulang`. Program mengkonversi kedua waktu tersebut ke dalam detik, lalu menghitung selisih antara waktu kepulangan dan kedatangan untuk mendapatkan durasi parkir dalam detik. Selanjutnya, durasi tersebut dikonversi kembali ke dalam format jam, menit, dan detik menggunakan variabel `durasi`, yang akhirnya ditampilkan sebagai output yang menunjukkan lama parkir pengguna.

Guided 2 - Map : Program Menampilkan Harga Buah Menggunakan Map

Sourcode

```
//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:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> go run
Harga Buah:
Apel: Rp5000
Pisang: Rp3000
Mangga: Rp7000
Harga buah Mangga = 7000
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> █
```

Deskripsi Program

Program tersebut merupakan program menggunakan map untuk menyimpan data harga buah dengan nama buah sebagai kunci dan harga sebagai nilai. Map tersebut berisikan “Apel” dengan harga 5000, “Pisang” dengan harga 3000, dan “Mangga” dengan harga 7000. Program akan langsung menampilkan harga dari setiap buah dalam map dan menampilkan harga buah Mangga dengan memanggil fungsi `hargaBuah` melalui kunci “Mangga”. Jadi output yang dihasilkan adalah daftar harga buah serta harga khusus untuk Mangga.

Guided 2 - Slice: Program Mengecek Daftar Nama Teman

Menggunakan Slice

```
//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:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> go run
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7>
```

Deskripsi Program

Program tersebut merupakan program menggunakan slice untuk mengelola daftar teman apakah namanya sudah ada di dalamnya. Program ini memiliki `daftarTeman` sebagai slice yang berisi beberapa nama teman, dan `namaBaru` yang berisi nama-nama baru ketika ingin ditambahkan ke dalam daftar. Program akan melakukan pemeriksaan nama tersebut terlebih dahulu dengan fungsi `sudahAda`. Jika sudah ada, program akan menampilkan pesan bahwa nama tersebut sudah ada dalam daftar. Lalu program akan menampilkan output daftar nama teman yang sudah diperbarui.

III. UNGUIDED

Unguided 1

Soal Studi Case

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 duar 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 titik struct {
    x, y int
}

type lingkaran struct {
    titikPusat titik
    radius      int
}

// Fungsi untuk menghitung jarak antara dua titik
func jarak(p, q titik) float64 {
    return math.Sqrt(float64((p.x-q.x)*(p.x-q.x) +
        (p.y-q.y)*(p.y-q.y)))
}

// Fungsi untuk memeriksa apakah titik p berada di dalam
lingkaran c
func dalamL(c lingkaran, p titik) bool {
    return jarak(c.titikPusat, p) <= float64(c.radius)
}

func main() {
    var cx1, cy1, r1, cx2, cy2, r2, x, y int

    // Input lingkaran 1 (koordinat dan radius pada
    satu baris)
```

```

        fmt.Println("Masukkan koordinat x1, y1 radius 1
untuk lingkaran 1:")
        fmt.Scan(&cx1, &cy1, &r1)

        // Input lingkaran 2 (koordinat dan radius pada
satu baris)
        fmt.Println("Masukkan koordinat x2, y2 radius 2
untuk lingkaran 2:")
        fmt.Scan(&cx2, &cy2, &r2)

        // Input titik sembarang (koordinat pada satu
baris)
        fmt.Println("Masukkan koordinat x dan y untuk titik
sembarang:")
        fmt.Scan(&x, &y)

        // Deklarasi lingkaran dan titik sembarang
lingkaran1 := lingkaran{titik{cx1, cy1}, r1}
lingkaran2 := lingkaran{titik{cx2, cy2}, r2}
titikSembarang := titik{x, y}

        // Mengecek posisi titik terhadap lingkaran 1 dan
lingkaran 2
        dalamL1 := dalamL(lingkaran1, titikSembarang)
        dalamL2 := dalamL(lingkaran2, titikSembarang)

        // Menampilkan hasil sesuai kondisi
        if dalamL1 && dalamL2 {
            fmt.Println("Titik di dalam lingkaran 1 dan
2")
        } else if dalamL1 {
            fmt.Println("Titik di dalam lingkaran 1")
        } else if dalamL2 {
            fmt.Println("Titik di dalam lingkaran 2")
        } else {
            fmt.Println("Titik di luar lingkaran 1 dan
2")
        }
    }
}

```

Screenshoot Output

```
Masukkan koordinat x1, y1 radius 1 untuk lingkaran 1:
1 1 5
Masukkan koordinat x2, y2 radius 2 untuk lingkaran 2:
8 8 4
Masukkan koordinat x dan y untuk titik sembarang:
Titik di dalam lingkaran 1
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> go run "d:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7\unguided1.go"
Masukkan koordinat x1, y1 radius 1 untuk lingkaran 1:
1 2 3
Masukkan koordinat x2, y2 radius 2 untuk lingkaran 2:
4 5 6
Masukkan koordinat x dan y untuk titik sembarang:
7 8
Titik di dalam lingkaran 2
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> go run "d:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7\unguided1.go"
Masukkan koordinat x1, y1 radius 1 untuk lingkaran 1:
5 10 15
Masukkan koordinat x2, y2 radius 2 untuk lingkaran 2:
-15 4 20
Masukkan koordinat x dan y untuk titik sembarang:
0 0
Titik di dalam lingkaran 1 dan 2
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> go run "d:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7\unguided1.go"
Masukkan koordinat x1, y1 radius 1 untuk lingkaran 1:
1 1 5
Masukkan koordinat x2, y2 radius 2 untuk lingkaran 2:
8 8 4
Masukkan koordinat x dan y untuk titik sembarang:
15 20
Titik di luar lingkaran 1 dan 2
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> █
```

Deskripsi Program

Program tersebut merupakan program menggunakan struct untuk mengecek apakah sebuah titik berada di dalam dua lingkaran yang berbeda. Program meminta pengguna untuk menginputkan koordinat pusat (cx, cy) dan radius (x, y) dari dua lingkaran, serta koordinat titik sembarangnya. Kemudian program akan memproses dengan menghitung menggunakan rumus yang berada dalam fungsi jarak. Setelah dihitung, program akan

mengecek apakah titik sembarang berada di dalam lingkaran jika dibandingkan jaraknya dengan radius lingkaran. Hasil output merupakan hasil dari pemeriksaan dan menampilkan titik berada di dalam salah satu atau kedua lingkaran, atau bahkan di luar keduanya.

Unguided 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 indeks 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"
)

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

    // Membuat array dengan panjang n
    arr := make([]int, n)

    // Input elemen array
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i)
        fmt.Scan(&arr[i])
    }

    // Menu
    for {
        fmt.Println("\nMenu:")
        fmt.Println("a. Tampilkan keseluruhan isi array")
        fmt.Println("b. Tampilkan elemen dengan indeks ganjil")
        fmt.Println("c. Tampilkan elemen dengan indeks genap")
        fmt.Println("d. Tampilkan elemen dengan indeks kelipatan bilangan x")
        fmt.Println("e. Hapus elemen array pada indeks tertentu")
        fmt.Println("f. Tampilkan rata-rata dari elemen array")
        fmt.Println("g. Hitung standar deviasi elemen array")
        fmt.Println("h. Hitung frekuensi suatu

```

```

bilangan dalam array")

        fmt.Println("i. Keluar")

        var pilihan string
        fmt.Print("\nMasukkan pilihan (a-h): ")
        fmt.Scan(&pilihan)

        switch pilihan {
        case "a": // Menampilkan keseluruhan isi dari
array
                fmt.Println("\nKeseluruhan isi array:")
                fmt.Println(arr)

                case "b": // Menampilkan elemen-elemen array
dengan indeks ganjil saja
                        fmt.Println("\nElemen array dengan
indeks ganjil:")
                                for i := 1; i < n; i += 2 {
                                        fmt.Printf("arr[%d] = %d\n", i,
array[i])
                                                }

                                case "c": // Menampilkan elemen-elemen array
dengan indeks genap saja
                                        fmt.Println("\nElemen array dengan
indeks genap:")
                                                for i := 0; i < n; i += 2 {
                                                        fmt.Printf("arr[%d] = %d\n", i,
array[i])
                                                                }

                                                        case "d": // Menampilkan elemen-elemen array
dengan indeks kelipatan bilangan x
                                                                var x int
                                                                fmt.Print("\nMasukkan bilangan untuk
kelipatan indeks: ")

```

```

        fmt.Scan(&x)

        if x <= 0 || x >= n { // Memastikan x
valid untuk digunakan sebagai kelipatan indeks
            continue
        }

        // Menampilkan elemen-elemen array
dengan indeks kelipatan x
        fmt.Printf("Elemen array dengan indeks
kelipatan %d:\n", x)

        kelipatan := false // Untuk mengecek
apakah ada indeks kelipatan x yang valid
        for i := x; i < n; i += x {
            fmt.Printf("arr[%d] = %d\n", i,
arr[i])

            kelipatan = true
        }

        // Jika tidak ada elemen dengan indeks
kelipatan x
        if !kelipatan {
            fmt.Printf("Tidak ada elemen pada
indeks kelipatan %d\n", x)
        }

        case "e": // Menghapus elemen array pada
indeks tertentu
            var hapusIndeks int // Mendeklarasikan
variabel hapusIndeks dengan tipe data integer untuk
menyimpan indeks yang akan dihapus
            fmt.Print("\nMasukkan indeks elemen yang
akan dihapus: ")

            fmt.Scan(&hapusIndeks)
            if hapusIndeks >= 0 && hapusIndeks < n
{ // Memastikan indeks yang dimasukkan valid

```

```

        arr = append(arr[:hapusIndeks],
arr[hapusIndeks+1:]...) // Menghapus elemen pada indeks
hapusIndeks dengan menggabungkan bagian array sebelumnya
dan sesudah dihapus

        fmt.Println("Array setelah
dihapus:")

        fmt.Println(arr)
    } else {
        fmt.Println("Indeks tidak valid.")
// Mencetak pesan jika indeks yang dimasukkan tidak valid
    }

    case "f": // Menampilkan rata-rata dari
bilangan yang ada di dalam array
        sum := 0
        for _, nilai := range arr {
            sum += nilai
        }
        rataRata := float64(sum) /
float64(len(arr))
        fmt.Printf("\nRata-rata elemen
array: %.2f\n", rataRata)

    case "g": // Menghitung standar deviasi
        sum := 0
        for _, nilai := range arr {
            sum += nilai
        }
        rataRata := float64(sum) /
float64(len(arr))
        variansiSum := 0.0
        for _, nilai := range arr {
            variansiSum += (float64(nilai) -
rataRata) * (float64(nilai) - rataRata)
        }
        standarDeviasi := variansiSum /

```

```

float64(len(arr)) // Variansi
        standarDeviasi =
math.Sqrt(standarDeviasi)
        fmt.Printf("Standar deviasi elemen
array: %.2f\n", standarDeviasi)

        case "h": // Menampilkan frekuensi dari suatu
bilangan tertentu di dalam array
            var cari int
            fmt.Print("\nMasukkan bilangan untuk
dihitung frekuensinya: ")
            fmt.Scan(&cari)
            frekuensi := 0
            for _, nilai := range arr {
                if nilai == cari {
                    frekuensi++
                }
            }
            fmt.Printf("Frekuensi %d di dalam
array: %d kali\n", cari, frekuensi)

        case "i": // Keluar dari program
            fmt.Println("Keluar dari program.")
            return

        default:
            fmt.Println("Pilihan tidak valid.
Silakan coba lagi.")
    }
}

```

Screenshoot Output

Memasukkan jumlah elemen array dan elemen :

```
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> go run "unguided2.go"
Masukkan jumlah elemen array: 10
Masukkan elemen ke-0: 22
Masukkan elemen ke-1: 12
Masukkan elemen ke-2: 21
Masukkan elemen ke-3: 22
Masukkan elemen ke-4: 42
Masukkan elemen ke-5: 2
Masukkan elemen ke-6: 2
Masukkan elemen ke-7: 12
Masukkan elemen ke-8: 32
Masukkan elemen ke-9: 42
```

Menu :

```
Menu:
a. Tampilkan keseluruhan isi array
b. Tampilkan elemen dengan indeks ganjil
c. Tampilkan elemen dengan indeks genap
d. Tampilkan elemen dengan indeks kelipatan bilangan x
e. Hapus elemen array pada indeks tertentu
f. Tampilkan rata-rata dari elemen array
g. Hitung standar deviasi elemen array
h. Hitung frekuensi suatu bilangan dalam array
i. Keluar
```

Pilihan a:

```
Masukkan pilihan (a-h): a

Keseluruhan isi array:
[22 12 21 22 42 2 2 12 32 42]
```

Pilihan b:

```
Masukkan pilihan (a-h): b

Elemen array dengan indeks ganjil:
arr[1] = 12
arr[3] = 22
arr[5] = 2
arr[7] = 12
arr[9] = 42
```

Pilihan c:

```
Masukkan pilihan (a-h): c

Elemen array dengan indeks genap:
arr[0] = 22
arr[2] = 21
arr[4] = 42
arr[6] = 2
arr[8] = 32
```

Pilihan d:

```
Masukkan pilihan (a-h): d

Masukkan bilangan untuk kelipatan indeks: 2
Elemen array dengan indeks kelipatan 2:
arr[2] = 21
arr[4] = 42
arr[6] = 2
arr[8] = 32
```


Pilihan e:

```
Masukkan pilihan (a-h): e
Masukkan indeks elemen yang akan dihapus: 8
Array setelah dihapus:
[22 12 21 22 42 2 2 12 42]
```

Pilihan f:

```
Masukkan pilihan (a-h): f
Rata-rata elemen array: 19.67
```

Pilihan g:

```
Masukkan pilihan (a-h): g
Standar deviasi elemen array: 13.95
```

Pilihan h:

```
Masukkan bilangan untuk dihitung frekuensinya: 1
Frekuensi 1 di dalam array: 0 kali
```

Pilihan i:

```
Masukkan pilihan (a-h): i
Keluar dari program.
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> █
```

Deskripsi Program

Program tersebut merupakan program untuk mengelola array yang dimasukkan pengguna dan menampilkan beberapa pilihan pada menu. Program meminta inputan oleh pengguna untuk memberikan jumlah elemen array. Lalu program akan menampilkan menu yang sesuai dengan ketentuan soal sebagai 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 indeks 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.

Program meminta pengguna untuk bebas memilih pilihan dalam menu tersebut sesuai dengan keinginan pengguna. Pada menu tersebut memiliki tambahan pilihan sebagai i. Keluar, yakni untuk pengguna dapat menyelesaikan atau keluar dari perulangan pada program tersebut.

Unguided 2

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 9:12

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"
)

func main() {
    var klubA, klubB string
    var skorA, skorB int
    var hasilPertandingan []string

    // Meminta input nama klub
    fmt.Print("Klub A: ")
    fmt.Scanln(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scanln(&klubB)

    // Proses input pertandingan
    for i := 1; i <= 9; i++ {
        // Input skor pertandingan
        fmt.Printf("Pertandingan %d: ", i)
        _, err := fmt.Scanln(&skorA, &skorB)

        // Cek jika input tidak valid atau skor
negatif
        if err != nil || skorA < 0 || skorB < 0 {
            // Hanya berhenti jika input tidak valid
            break
        }

        // Menentukan pemenang
        if skorA > skorB {
```

```
        hasilPertandingan =  
append(hasilPertandingan, klubA)  
        } else if skorB > skorA {  
            hasilPertandingan =  
append(hasilPertandingan, klubB)  
        } else {  
            hasilPertandingan =  
append(hasilPertandingan, "Draw")  
        }  
    }  
  
    // Menampilkan hasil pertandingan yang valid  
    for i, hasil := range hasilPertandingan {  
        fmt.Printf("Hasil %d: %s\n", i+1, hasil)  
    }  
  
    // Menampilkan pesan bahwa pertandingan telah  
    selesai  
    fmt.Println("Pertandingan selesai")  
}
```

Screenshoot Output

```
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> go 7\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
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> |
```

Deskripsi Program

Program tersebut merupakan program menggunakan slice untuk menyimpan hasil pertandingan yang diinputkan oleh pengguna. Pengguna dapat memasukkan dua nama klub yang berbeda dan meminta memasukkan skor pertandingan. Program akan terus berjalan hingga pengguna memasukkan skor yang tidak valid dan program akan berhenti. Kemudian program menampilkan hasil pertandingan dari skor yang telah dimasukkan pengguna. Hasilnya menyesuaikan nama dua klub, ketika klub A lebih tinggi poinnya akan menampilkan klub A, begitupun

sebaliknya. Dan ketika skornya sama, hasil akan menampilkan “Draw”. Setelah itu program menampilkan pesan bahwa pertandingan telah selesai.

Unguided 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
// 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	: <u>S E N A N G</u> ,
Reverse teks	: G N A N E S
Teks	: <u>K A T A K</u> .
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	: <u>K A T A K</u>
Palindrom	? true

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

Sourcecode

```
package main

import "fmt"

const NMAX int = 127
type tabel [NMAX] rune

var tab tabel
var m int

// Prosedur untuk mengisi array dengan karakter yang
dimasukkan user
func isiArray(t *tabel, n *int) {
    var input string
    fmt.Scanln(&input) // Membaca input dalam bentuk
string
    *n = len(input)    // Menghitung panjang string
    for i := 0; i < *n; i++ {
        (*t)[i] = rune(input[i]) // Menyimpan karakter
ke dalam array
    }
}

// Prosedur untuk mencetak array
func cetakArray(t tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Print(string(t[i])) // Mencetak karakter
satu per satu
    }
    fmt.Println()
}

// Prosedur untuk membalikkan urutan array
func balikanArray(t *tabel, n int) {
    for i, j := 0, n-1; i < j; i, j = i+1, j-1 {
        (*t)[i], (*t)[j] = (*t)[j], (*t)[i]
    }
}

// Fungsi untuk memeriksa apakah array membentuk
palindrom
func palindrom(t tabel, n int) bool {
```

```

    // Membalikkan array sementara untuk pemeriksaan
    palindrom
        balikanArray(&t, n)

    // Memeriksa apakah array asli sama dengan array yang
    dibalik
    for i := 0; i < n; i++ {
        if t[i] != tab[i] {
            return false
        }
    }
    return true
}

func main() {
    // Mengisi array dengan memanggil prosedur isiArray
    isiArray(&tab, &m)

    // Menampilkan teks yang dimasukkan
    fmt.Print("Teks : ")
    cetakArray(tab, m)

    // Membalikkan isi array dan menampilkan hasilnya
    balikanArray(&tab, m)
    fmt.Print("Reverse teks : ")
    cetakArray(tab, m)

    // Memeriksa apakah teks tersebut palindrom
    if palindrom(tab, m) {
        fmt.Println("Teks Palindrom : true")
    } else {
        fmt.Println("Teks Palindrom : false")
    }
}

```

Screenshoot Output

```

PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> go run
Teks : SENANG
Reverse teks : GNANES
Teks Palindrom : false
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7>

```

```
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> go run  
Teks : KATAK  
Reverse teks : KATAK  
Teks Palindrom : true  
PS D:\kuliah\sem 3\Praktikum ALPRO 2\Modul 7> █
```

Deskripsi Program

Program tersebut merupakan program menggunakan array untuk mereverse teks dan palindrom. Pengguna diminta untuk menginputkan sebuah teks oleh program, kemudian program akan menyimpan teks tersebut dalam array bentuk tipe data rune. Setelah itu, program akan mencetak teks dan langsung menampilkan hasil output reverse teks atau kebalikannya dari teks yang telah diinputkan pengguna. Dan program akan memeriksa jika teks adalah palindrom (bisa dibaca dari depan atau belakang) maka hasil true , jika bukan maka hasil false.

Daftar Pustaka

- [1] Susilowati, A. Zahara, A. (2024). *Modul 7 Praktikum Algoritma Pemrograman 2*. Program Studi Teknik Informatika, Telkom University Purwokerto.