

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCK & ARRAY**



Disusun Oleh :

Naya Putwi Setiasih / 2311102155

S1 11 IF - 05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

- Alias (Type)

Bahasa pemrograman biasanya 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”.

| | 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 | | |

- Struct atau Record

Structure memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing – masing nilai tersimpan dalam field dari structure 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 yang 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.

- 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.

- **Slice (Array dinamik)**

Array dalam Go juga dapat mempunyai ukuran dinamik. Deklarasi 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 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.

Soal Studi Case

Menghitung durasi parkir

Sourcecode

```
package main

import "fmt"

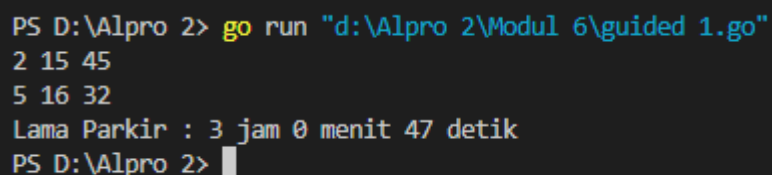
type waktu struct {
    jam, menit, detik int
}

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:\Alpro 2> go run "d:\Alpro 2\Modul 6\guided 1.go"
2 15 45
5 16 32
Lama Parkir : 3 jam 0 menit 47 detik
PS D:\Alpro 2> █
```

Deskripsi Program

Program ini ditulis dalam bahasa Go dan berfungsi untuk menghitung durasi parkir berdasarkan waktu masuk dan waktu keluar. Dalam program

ini, struktur ernama “waktu” didefinisikan untuk menyimpan informasi jam, menit, dan detik. Di dalam fungsi “main”, variable – variable seperti “wparkir” dan “wpulang” digunakan untuk menyimpan waktu parkir dan waktu pulang, sedangkan “durasi” menyimpan hasil perhitungan lama parkir. Program meminta pengguna untuk memasukkan waktu parkir dan waktu pulang dalam format jam, menit, dan detik menggunakan “fmt.Scan”. Selanjutnya, waktu tersebut dikonversikan ke detik untuk mempermudah penghitungan. Lama parkir dihitung dengan mengkurangkan waktu parkir dari waktu pulang. Setelah mendapatkan lama parkir dalam detik, program mengubahnya kembali ke format jam, menit, dan detik. Hasil akhir dicetak dalam format yang mudah dibaca, menunjukkan durasi parkir yang jelas.

2. Mengelola daftar teman

Sourcecode

```
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:\Alpro 2> go run "d:\Alpro 2\Modul 6\guided 2.go"
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\Alpro 2>

```

Deskripsi Program

Program ini ditulis dalam Bahasa Go dan bertujuan untuk mengelola daftar teman dengan menambahkan nama – anma baru hanya jika nama tersebut belum ada dalam daftar. Di dalam program, terdapat fungsi “sudahAda” yang digunakan untuk memeriksa apakah suatu nama sudah terdapat pada slice daftar teman. Fungsi ini menerima dua parameter : “daftarTeman”, yaitu slice yang berisi nama – nama teman, dan “nama”, yaitu nama yang ingin dicek. Jika nama ditemukan dalam daftar, fungsi akan mengembalikan nilai “true”; jika tidak, akan mengembalikan “false”. Dalam fungsi “main”, program dimulai dengan mendefinisikan slice “daftarTeman” yang berisi beberapa nama awal. Selanjutnya, program mendefinisikan slice “namaBaru” yang berisi nama – nama baru yang ingin ditambahkan dalam dafta. Program kemudian melakukan iterasi melalui “namaBaru”, dan untuk setiap nama, ia memeriksa apakah nama tersebut sudah ada di dalam “daftarYeman” menggunakan fungsi “sudahAda”. Jika nama belum ada, maka nama tersebut ditambahkan ke dalam daftar; jika sudah ada, program akan mencetak pesan bahwa nama tersebut sudah ada. Setelah semua nama baru diproses, program menampilkan daftar akhir teman yang telah diperbarui.

3. Program map untuk menyimpan data

Sourcecode

```

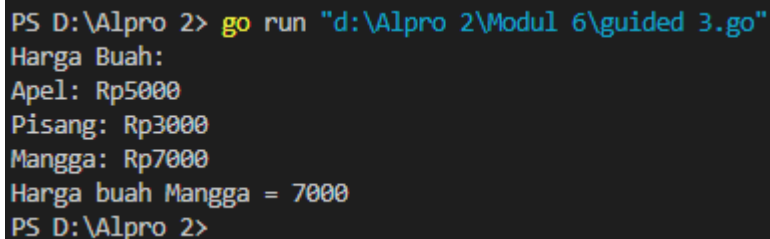
package main
import (
    "fmt"
)

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:\Alpro 2> go run "d:\Alpro 2\Modul 6\guided 3.go"  
Harga Buah:  
Apel: Rp5000  
Pisang: Rp3000  
Mangga: Rp7000  
Harga buah Mangga = 7000  
PS D:\Alpro 2>
```

Deskripsi Program

Program ini ditulis dalam Bahasa Go dan berfungsi untuk menyimpan serta menampilkan harga berbagai jenis buah menggunakan struktur data map. Di dalam program, sebuah map bernama “hargaBuah” didefinisikan dengan kunci berupa nama buah (tipe string) dan nilai berupa harga buah tersebut (tipe int). Pada awalnya, map ini diisi dengan beberapa entri, yaitu “Apel” dengan harga 5000, “Pisang” dengan harga 3000, dan “Mangga” dengan harga 7000. Program kemudian menampilkan daftar harga buah dengan melakukan iterasi melalui map menggunakan loop “for range”. Dalam setiap iterasi, nama buah dan harganya dicetak ke layar dalam format yang mudah dibaca. Selain itu, program juga mencetak harga khusus untuk buah mangga secara langsung dengan mengakses nilai dalam map menggunakan kunci “Mangga”. Program ini tidak hanya menunjukkan cara penggunaan map untuk menyimpan data tetapi juga memberikan contoh bagaimana cara mengakses dan menampilkan informasi dari map tersebut.

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 dipresentasikan dengan bilangan bulat.

Keluaran : berupa string yang menyatakan posisi titik "titik didalam lingkaran 1 dan 2", "titik di dalam lingkaran 1", "titik di dalam lingkaran 2", atau "titik di luar lingkaran 1 dan 2".

| 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 {
    center 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 menentukan apakah titik berada di dalam
lingkaran
func didalam(c Lingkaran, p Titik) bool {
    return jarak(c.center, p) <= float64(c.radius)
}

// Fungsi utama untuk mengecek posisi titik terhadap dua
lingkaran
func cekPosisiTitik(lingkaran1, lingkaran2 Lingkaran,
titik Titik) string {
    dalamLingkaran1 := didalam(lingkaran1, titik)
    dalamLingkaran2 := didalam(lingkaran2, titik)

    if dalamLingkaran1 && dalamLingkaran2 {

```

```

        return "Titik di dalam lingkaran 1 dan 2"
    } else if dalamLingkaran1 {
        return "Titik di dalam lingkaran 1"
    } else if dalamLingkaran2 {
        return "Titik di dalam lingkaran 2"
    } else {
        return "Titik di luar lingkaran 1 dan 2"
    }
}

func main() {
    var x1, y1, r1 int
    var x2, y2, r2 int
    var xt, yt int

    // Meminta input dari pengguna
    fmt.Println("Masukkan koordinat pusat dan radius
untuk Lingkaran 1 (x y r):")
    fmt.Scanln(&x1, &y1, &r1)

    fmt.Println("Masukkan koordinat pusat dan radius
untuk Lingkaran 2 (x y r):")
    fmt.Scanln(&x2, &y2, &r2)

    fmt.Println("Masukkan koordinat titik sembarang (x
y):")
    fmt.Scanln(&xt, &yt)

    // Membuat objek lingkaran dan titik berdasarkan
input
    lingkaran1 := Lingkaran{Titik{x1, y1}, r1}
    lingkaran2 := Lingkaran{Titik{x2, y2}, r2}
    titik := Titik{xt, yt}

    // Menampilkan hasil
    fmt.Println(cekPosisiTitik(lingkaran1, lingkaran2,
titik))
}

```

Screenshoot Output

```

PS D:\Alpro 2> go run "d:\Alpro 2\Modul 6\unguided 1.go"
Masukkan koordinat pusat dan radius untuk Lingkaran 1 (x y r):
1 5 8
Masukkan koordinat pusat dan radius untuk Lingkaran 2 (x y r):
2 3 5
Masukkan koordinat titik sembarang (x y):
3 5
Titik di dalam lingkaran 1 dan 2
PS D:\Alpro 2> 

```

Deskripsi Program

Program ini ditulis dalam Bahasa Go dan dirancang untuk menentukan posisi sebuah titik relative terhadap dua lingkaran yang didefinisikan oleh pusat dan radiusnya. Program dimulai dengan mendefinisikan dua struktur, yaitu “titik” yang menyimpan koordinat x dan y, serta “lingkaran”, yang menyimpan pusat lingkaran (sebagai objek “titik”) dan radiusnya. Fungsi “jarak” digunakan untuk menghitung jarak antara dua titik menggunakan rumus Pythagoras, sementara fungsi “didalam” menentukan apakah suatu titik berada di dalam lingkaran dengan membandingkan jarak titik tersebut dari pusat lingkaran dengan radius lingkaran. Dalam fungsi “cekPosisiTitik”, program memeriksa posisi titik terhadap kedua lingkaran dan mengembalikan string yang menjelaskan apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, atau diluar keduanya. Pada bagian “main”, program meminta pengguna untuk memasukkan koordinat pusat dan radius untuk kedua lingkaran, serta koordinat titik yang ingin diperiksa. Setelah input diterima, objek lingkaran dan titik dibuat berdasarkan data tersebut. Akhirnya, program memanggil fungsi “cekPosisiTitik” untuk menentukan posisi titik dan menampilkan hasilnya kepada pengguna. Program ini memberikan cara yang efektif untuk memvisualisasikan hubungan geometris antara titik dan lingkaran dalam ruang dua dimensi.

2. Sebuah array digunakan untuk menampung sekumpulan bilangan bulat. Buatlah program yang digunakan untuk mengisis 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 (asumsikan indeks ke – 0 adalah genap)
 - d. Menampilkan elemen – elemen array dengan indeks kelipatan bilangan x. X bisa diperoleh dari masukkan 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.Scanln(&n)

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

    for {
        fmt.Println("\nMenu:")
        fmt.Println("1. Tampilkan keseluruhan isi 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 bilangan tertentu")
        fmt.Println("5. Hapus elemen pada indeks tertentu")
        fmt.Println("6. Tampilkan rata-rata elemen")
        fmt.Println("7. Tampilkan standar deviasi elemen")
        fmt.Println("8. Tampilkan frekuensi bilangan tertentu")
        fmt.Println("9. Keluar")

        var choice int
        fmt.Print("Pilih menu: ")
        fmt.Scanln(&choice)

        switch choice {
        case 1:
            // a. Menampilkan keseluruhan isi dari array
            fmt.Println("1. Keseluruhan isi array:",
                arr)

        case 2:
```

```

        // b. Menampilkan elemen-elemen array dengan
        indeks ganjil saja
        fmt.Print("2. Elemen pada indeks ganjil: ")
        for i := 1; i < n; i += 2 {
            fmt.Print(arr[i], " ")
        }
        fmt.Println()

        case 3:
            // c. Menampilkan elemen-elemen array dengan
            indeks genap saja
            fmt.Print("3. Elemen pada indeks genap: ")
            for i := 0; i < n; i += 2 {
                fmt.Print(arr[i], " ")
            }
            fmt.Println()

        case 4:
            // d. Menampilkan elemen-elemen array dengan
            indeks kelipatan bilangan x
            var x int
            fmt.Print("Masukkan bilangan x untuk
            kelipatan indeks: ")
            fmt.Scanln(&x)

            fmt.Print("4. Elemen pada indeks kelipatan
            ", x, ": ")
            for i := 0; i < n; i += x {
                fmt.Print(arr[i], " ")
            }
            fmt.Println()

        case 5:
            // e. Menghapus elemen array pada indeks
            tertentu
            var delIndex int
            fmt.Print("Masukkan indeks yang ingin
            dihapus: ")
            fmt.Scanln(&delIndex)

            if delIndex >= 0 && delIndex < len(arr) {
                arr = append(arr[:delIndex],
                arr[delIndex+1:]...)
                fmt.Println("5. Array setelah
                penghapusan:", arr)
            } else {
                fmt.Println("5. Indeks tidak valid untuk
                penghapusan")
            }

        case 6:

```

```

        // f. Menampilkan rata-rata dari bilangan
yang ada di dalam array
        sum := 0
        for _, value := range arr {
            sum += value
        }
        rataRata := float64(sum) / float64(len(arr))
        fmt.Printf("6. Rata-rata elemen array:
%.2f\n", rataRata)

        case 7:
            // g. Menampilkan standar deviasi dari
bilangan yang ada di dalam array
            sum := 0
            for _, value := range arr {
                sum += value
            }
            rataRata := float64(sum) / float64(len(arr))

            sumSquares := 0.0
            for _, value := range arr {
                sumSquares += math.Pow(float64(value)-
rataRata, 2)
            }
            standarDeviasi := math.Sqrt(sumSquares /
float64(len(arr)))
            fmt.Printf("7. Standar deviasi elemen array:
%.2f\n", standarDeviasi)

            case 8:
                // h. Menampilkan frekuensi dari suatu
bilangan tertentu
                var cari int
                fmt.Print("Masukkan bilangan yang ingin
dicari frekuensinya: ")
                fmt.Scanln(&cari)

                frekuensi := 0
                for _, value := range arr {
                    if value == cari {
                        frekuensi++
                    }
                }
                fmt.Printf("8. Frekuensi bilangan %d di
dalam array: %d\n", cari, frekuensi)

            case 9:
                fmt.Println("Keluar dari program.")
                return

            default:

```

```
        fmt.Println("Pilihan tidak valid, silakan  
coba lagi.")  
    }  
}  
}
```

Screenshoot Output

```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 6\unguided 2.go"
```

```
Masukkan jumlah elemen array: 5
```

```
Masukkan elemen ke-0: 3
```

```
Masukkan elemen ke-1: 2
```

```
Masukkan elemen ke-2: 5
```

```
Masukkan elemen ke-3: 4
```

```
Masukkan elemen ke-4: 6
```

```
Menu:
```

1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan tertentu
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi bilangan tertentu
9. Keluar

```
Pilih menu: █
```

```
Menu:
```

1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan tertentu
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi bilangan tertentu
9. Keluar

```
Pilih menu: 1
```

```
1. Keseluruhan isi array: [3 2 5 4 6]
```

```
Menu:
```

1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan tertentu
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi bilangan tertentu
9. Keluar

```
Pilih menu: 2
```

```
2. Elemen pada indeks ganjil: 2 4
```


Menu:

1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan tertentu
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi bilangan tertentu
9. Keluar

Pilih menu: 3

3. Elemen pada indeks genap: 3 5 6

Menu:

1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan tertentu
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi bilangan tertentu
9. Keluar

Pilih menu: 4

Masukkan bilangan x untuk kelipatan indeks: 3

4. Elemen pada indeks kelipatan 3: 3 4

Menu:

1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan tertentu
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi bilangan tertentu
9. Keluar

Pilih menu: 5

Masukkan indeks yang ingin dihapus:

5. Array setelah penghapusan: [2 5 4 6]

Menu:

1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan tertentu
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi bilangan tertentu
9. Keluar

Pilih menu: 6

6. Rata-rata elemen array: 4.25

```
Menu:
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan tertentu
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi bilangan tertentu
9. Keluar
Pilih menu: 7
7. Standar deviasi elemen array: 1.48
```

```
Menu:
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan tertentu
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi bilangan tertentu
9. Keluar
Pilih menu: 8
Masukkan bilangan yang ingin dicari frekuensinya: 2
8. Frekuensi bilangan 2 di dalam array: 1
```

```
Menu:
1. Tampilkan keseluruhan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan bilangan tertentu
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata elemen
7. Tampilkan standar deviasi elemen
8. Tampilkan frekuensi bilangan tertentu
9. Keluar
Pilih menu: 9
Keluar dari program.
PS D:\Alpro 2>
```

Deskripsi Program

Program yang ditulis dalam Bahasa Go ini adalah aplikasi sederhana untuk mengelola dan menganalisis array bilangan bulat. Pertama, pengguna diminta untuk memasukkan jumlah elemen yang akan disimpan dalam array, diikuti dengan input masing – masing elemen. Setelah array terisi, program menampilkan menu interaktif yang memungkinkan pengguna untuk melakukan berbagai operasi pada array tersebut.

Pengguna dapat memilih untuk menampilkan seluruh isi array, elemen dengan indeks ganjil atau genap, serta elemen pada indeks kelipatan bilangan tertentu yang ditentukan oleh pengguna. Program juga menyediakan opsi untuk menghapus elemen pada indeks tertentu dan

menghitung rata – rata serta standar deviasi dari elemen – elemen dalam array. Selain itu, pengguna dapat mencari frekuensi kemunculan bilangan tertentu dalam array.

Setiap kali pengguna memilih opsi dari menu, program akan mengeksekusi fungsi yang sesuai dan menampilkan hasilnya. Program ini terus berjalan dalam loop hingga pengguna memilih untuk keluar dengan memasukkan pilihan yang sesuai. Program ini memberikan cara yang mudah dan interaktif untuk bekerja dengan array dan melakukan analisis dasar terhadap data numerik yang dimasukkan.

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 (negative). 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"
)

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

    var pemenang []string
    pertandingan := 1

    for {
        var skorA, skorB int
        fmt.Printf("Pertandingan %d: ", pertandingan)
        _, err := fmt.Scanf("%d %d\n", &skorA, &skorB)
```

```
        if err != nil || skorA < 0 || skorB < 0 {
            break
        }

        if skorA > skorB {
            fmt.Printf("Hasil %d: %s\n", pertandingan,
klubA)
            pemenang = append(pemenang, klubA)
        } else if skorA < skorB {
            fmt.Printf("Hasil %d: %s\n", pertandingan,
klubB)
            pemenang = append(pemenang, klubB)
        } else {
            fmt.Printf("Hasil %d: Draw\n", pertandingan)
        }

        pertandingan++
    }

    fmt.Println("Pertandingan selesai")
    fmt.Println("Daftar klub yang menang:")
    for _, klub := range pemenang {
        fmt.Println(klub)
    }
}
```

Screenshot Output

```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 6\unguided 3.go"
Klub A: MU
Klub B: Inter
Pertandingan 1: 2 0
Hasil 1: MU
Pertandingan 2: 3 1
Hasil 2: MU
Pertandingan 3: 4 6
Hasil 3: Inter
Pertandingan 4: 1 0
Hasil 4: MU
Pertandingan 5: 2 3
Hasil 5: Inter
Pertandingan 6: 2 1
Hasil 6: MU
Pertandingan 7: 2 5
Hasil 7: Inter
Pertandingan 8: 2 0
Hasil 8: MU
Pertandingan 9: -1 3
Pertandingan selesai
Daftar klub yang menang:
MU
MU
Inter
MU
Inter
MU
Inter
MU
PS D:\Alpro 2> █
```

Deskripsi Program

Program ini ditulis dalam Bahasa Go bertujuan untuk menghitung hasil pertandingan antara dua tim sepak bola. Pengguna diminta untuk memasukkan nama kedua tim, setelah itu program akan menerima skor dari setiap pertandingan secara berulang hingga pengguna memasukkan skor yang tidak valid atau negative. Program akan menentukan pemenang berdasarkan skor yang diinput; jika skor sama, hasilnya dinyatakan sebagai "Draw". Setiasp hasil pertandingan dicetak bersama nama tim yang menang, dan setelah semua pertandingan selesai, program menampilkan daftar klub yang telah menang.

4. Sebuah array digunakan untuk menampung sekumpulan karakter, anda diminta 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 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 polindrom. 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 (
    "bufio"
    "fmt"
    "os"
    "strings"
)

const NMAX int = 127

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

// Prosedur untuk mengisi array
func isiArray(t *tabel, n *int) {
    reader := bufio.NewReader(os.Stdin)
    fmt.Println("Masukkan karakter satu per satu (akhiri dengan menekan Enter pada baris kosong):")

    *n = 0
    for {
        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)

        // Jika input kosong, hentikan proses pengisian
        if input == "" || *n >= NMAX {
            break
        }

        // Hanya karakter pertama dari input yang dimasukkan
        t.tab[*n] = rune(input[0])
    }
}
```



```

        *n++
    }
}

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

// Prosedur untuk membalikkan isi array
func balikanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        t.tab[i], t.tab[n-i-1] = t.tab[n-i-1], t.tab[i]
    }
}

// Fungsi untuk memeriksa apakah array membentuk
palindrome
func palindrom(t tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t.tab[i] != t.tab[n-i-1] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var n int

    // Memanggil prosedur isiArray
    isiArray(&tab, &n)

    // Mencetak array sebelum dibalik
    fmt.Println("Teks asli:")
    cetakArray(tab, n)

    // Membalik isi array
    balikanArray(&tab, n)

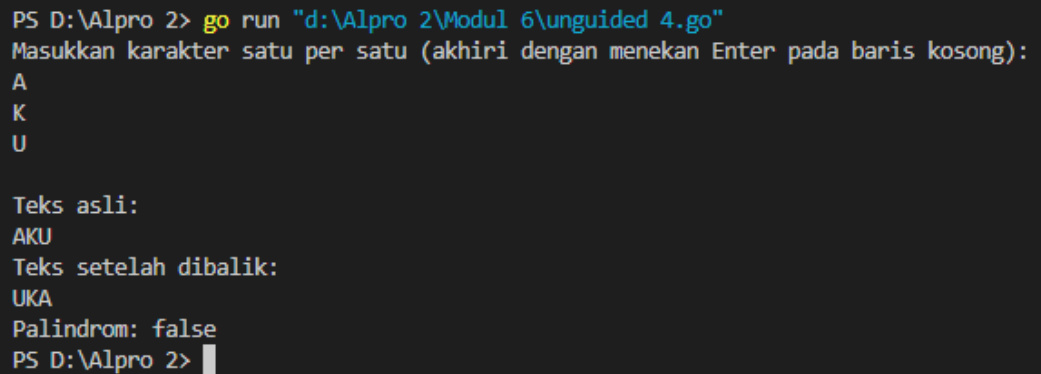
    // Mencetak array setelah dibalik
    fmt.Println("Teks setelah dibalik:")
    cetakArray(tab, n)

    // Memeriksa apakah teks merupakan palindrome
    if palindrom(tab, n) {
        fmt.Println("Palindrom: true")
    } else {

```

```
        fmt.Println("Palindrom: false")
    }
}
```

Screenshoot Output



```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 6\unguided 4.go"
Masukkan karakter satu per satu (akhiri dengan menekan Enter pada baris kosong):
A
K
U

Teks asli:
AKU
Teks setelah dibalik:
UKA
Palindrom: false
PS D:\Alpro 2> █
```

Deskripsi Program

Program ini ditulis dalam Bahasa Go bertujuan untuk mengelola dan memeriksa karakter yang dimasukkan oleh pengguna, dengan fokus pada kemampuan untuk membalik urutan karakter dan menentukan apakah urutan tersebut membentuk sebuah palindrome. Program dimulai dengan mendeklarasikan struktur data bernama "table", yang menyimpan array karakter dan jumlah karakter yang dimasukkan. Pengguna diminta untuk memasukkan karakter satu per satu melalui input, dan proses ini berlanjut hingga pengguna menekan enter pada baris kosong atau jumlah karakter mencapai batas maksimum. Setelah karakter diinput, program mencetak urutan karakter asli, kemudian membalik urutan tersebut dan mencetak hasilnya. Selanjutnya, program memeriksa apakah urutan karakter yang diinput merupakan palindrome, yaitu sebuah kata atau frasa yang dibaca sama dari depan maupun belakang. Hasil pemeriksaan ini ditampilkan kepada pengguna sebagai output terakhir.

DAFTAR PUSTAKA

Asisten Praktikum, “Modul 7 Struck & Array”, Learning Management System, 2024.