

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN VII**

**MODUL 7
STRUCK DAN ARRAY**



**Disusun Oleh :
SHAFA ADILA SANTOSO / 2311102158
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

Tipe Bentukan

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 diubah dengan nama alias "bilangan". Caranya dengan menggunakan kata kunci "type".

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama alias> <tipe data>	type <nama alias> <tipe data>
3		
4	algoritma	func main(){
5
6		}
7		

Sebagai contoh perhatikan program Go berikut beserta hasil eksekusinya!

```
1 package main
2 import "fmt"
3 type bilangan int
4 type pecahan float64
5 func main(){
6     var a,b bilangan
7     var hasil pecahan
8     a = 9
9     b = 5
10    hasil = pecahan(a) / pecahan(b)
11    fmt.Println(hasil)
12 }
```

```
E:\DEV\GO>go build Demo.go
E:\DEV\GO> Demo.exe
1.8
```

2. Struct atau Record

Stucture memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari stucture tersebut.

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama struct> <	type <nama struct> struct {
3	<field 1> <tipe data>	<field 1> <tipe data>
4	<field 2> <tipe data>	<field 2> <tipe data>
5	<field 3> <tipe data>	<field 3> <tipe data>
6	>	}
7		

Berbeda dengan bahasa pemrograman lain, kesamaan tipe dari dua variabel berjenis stucture bukan karena namanya tetapi karena strukturnya. Dua variabel dengan nama-nama field dan tipe field yang sama (dan dalam urutan yang sama) dianggap mempunyai tipe yang sama. Tentunya akan lebih memudahkan jika stucture tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi stucture tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali,

```

1 package main
2 import "fmt"
3 type waktu struct {
4     jam, menit, detik int
5 }
6
7 func main(){
8     var wParkir, wPulang, durasi waktu
9     var dParkir, dPulang, lParkir int
10    fmt.Scan(&wParkir.jam, &wParkir.menit, &wParkir.detik)
11    fmt.Scan(&wPulang.jam, &wPulang.menit, &wPulang.detik)
12    dParkir = wParkir.detik + wParkir.menit*60 + wParkir.jam*3600
13    dPulang = wPulang.detik + wPulang.menit*60 + wPulang.jam*3600
14    lParkir = dPulang - dParkir
15    durasi.jam = lParkir / 3600
16    durasi.menit = lParkir % 3600 / 60
17    durasi.detik = lParkir % 3600 % 60
18    fmt.Printf("Lama parkir: %d jam %d menit %d detik",
19        durasi.jam, durasi.menit, durasi.detik)
20 }

```

```

E:\DEV\GO>go build Demo.go
E:\DEV\GO> Demo.exe
7 30 0
10 45 15
Lama parkir: 3 jam 15 menit 15 detik

```

Array

Array memiliki ukuran atau jumlah elemen yang tetap atau 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 diatas. Indeks array dimulai dari 0, sehingga indeks arr pada contoh adalah 0, 1,... `len(arr)-1`

Contoh:

```

1 // Mengganti isi elemen ke-0 dengan nilai dari elemen ke-7
2 arr[0] = arr[7]
3
4 // Mengambil data field x dari elemen ke-i
5 currX = arr[i].center.x
6
7 // Mengambil elemen terakhir
8 n := len(arr)
9 buf := arr[n-1]
```

Slice (Array dinamik)

Array dalam Go juga dapat mempunyai ukuran yang dinamik. (Tidak digunakan di kelas Algoritma Pemrograman). Deklarasinya mirip dengan deklarasi array, tetapi jumlah elemennya dikosongkan.

```

1 // declaring chop as an empty slice of float64
2 var chop []float64
3
4 // declaring sl01 as a slice
5 var sl01 = []int{ 11, 2, 3, 5, 7, 13 }
```

Sebuah slice dapat diprealokasi menggunakan fungsi built-in `make`

```

1 // Prealokasi 10 elemen untuk sl02 dan sejumlah tempat tambahan
2 var sl02 []int = make([]int, 10, 20)
3
4 // Prealokasi 7 elemen untuk sl03 tanpa tempat tambahan
5 var sl03 []CircType = make([]CircType, 7)
```

Fungsi built-in `len` dapat digunakan untuk mengetahui ukuran slice. Fungsi lain, `cap`, dapat digunakan untuk mengetahui total empat 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 apabila 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 berbentuk 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"
```

man 64 | Modul Praktikum Algoritma dan Pemrograman 2

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

II. GUIDED

1. Program Menghitung Lama Durasi Parkir

Sourcecode

```
package main

import "fmt"

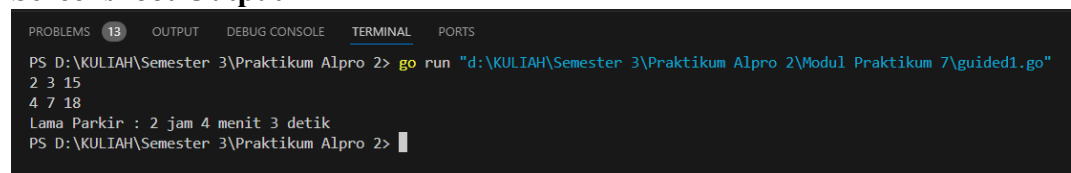
type waktu struct {
    jam, menit, detik int
}

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

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS (13), OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The command 'go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 7\guided1.go"' is entered. The output shows the input values '2 3 15' and '4 7 18', followed by the calculated result 'Lama Parkir : 2 jam 4 menit 3 detik'. The prompt 'PS D:\KULIAH\Semester 3\Praktikum Alpro 2>' is visible at the bottom.

```
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 7\guided1.go"
2 3 15
4 7 18
Lama Parkir : 2 jam 4 menit 3 detik
PS D:\KULIAH\Semester 3\Praktikum Alpro 2>
```

Deskripsi Program

Program diatas merupakan program Go yang dibuat untuk menghitung durasi waktu parkir antara waktu datang dan waktu pulang yang diberikan dalam format jam, menit, dan detik. Algoritma yang digunakan mengkonversi waktu parkir (wParkir) dan waktu pulang (wPulang) ke dalam satuan detik, dengan cara mengalikan jam dengan 3600 detik, menit dengan 60 detik, dan menambahkan detik. Selanjutnya, selisih antara

waktu pulang dan waktu parkir (dPulang - dParkir) dihitung dalam detik. Setelah itu, durasi parkir tersebut diubah kembali menjadi format jam, menit, dan detik menggunakan pembagian dan sisa bagi. Output yang dihasilkan adalah lama durasi parkir dalam format "x jam y menit z detik". Misalnya, jika waktu parkir dimulai pada jam 12:30:10 dan selesai pada jam 14:45:20, program akan menghitung durasi parkir tersebut dan menampilkan "Lama Parkir : 2 jam 15 menit 10 detik".

2. Program Slice

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)
```



```
}
```

Screenshot Output

Deskripsi Program

Program diatas merupakan program Go yang dibuat untuk mengelola daftar teman dengan memastikan tidak ada duplikasi nama dalam daftar tersebut. Algoritma yang digunakan dimulai dengan mendefinisikan sebuah slice yang berisi nama teman awal. Kemudian, program menerima nama-nama baru yang ingin ditambahkan ke dalam daftar. Sebelum menambahkan nama baru, program memeriksa apakah nama tersebut sudah ada dalam daftar menggunakan fungsi `sudahAda()`, yang akan mengembalikan `true` jika nama sudah ada, dan `false` jika nama belum ada. Jika nama belum ada, nama tersebut akan ditambahkan ke dalam daftar menggunakan fungsi `append()`, jika sudah ada, maka program akan menampilkan pesan yang memberitahukan bahwa nama tersebut sudah ada. Setelah semua nama baru diproses, program mencetak daftar teman akhir yang telah diperbarui. Misalnya, jika daftar awal berisi "Andi", "Budi", "Cici", dan nama baru yang diberikan adalah "Dewi", "Budi", "Eka", maka output yang dihasilkan adalah "Daftar Teman: [Andi Budi Cici Dewi Eka]", dengan pesan "Nama Budi sudah ada dalam daftar."

3. Program Map

Sourcecode

```
//Guided 2 - Map
package main

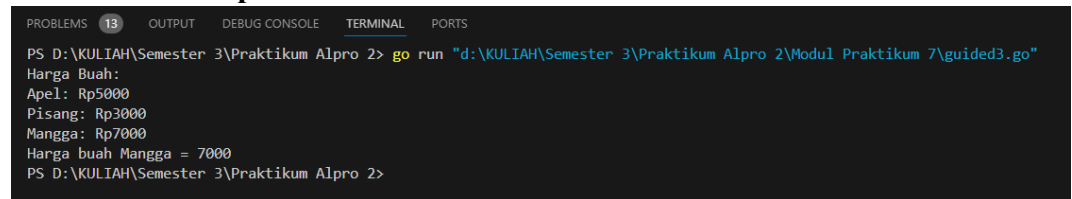
import (
    "fmt"
)

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"])  
}
```

Screenshot Output

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS (13), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), and PORTS. The command executed is 'go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 7\guided3.go"'. The output shows a list of fruit prices: 'Harga Buah:', 'Apel: Rp5000', 'Pisang: Rp3000', 'Mangga: Rp7000', and 'Harga buah Mangga = 7000'. The prompt 'PS D:\KULIAH\Semester 3\Praktikum Alpro 2>' is visible at the bottom.

```
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum 7\guided3.go"  
Harga Buah:  
Apel: Rp5000  
Pisang: Rp3000  
Mangga: Rp7000  
Harga buah Mangga = 7000  
PS D:\KULIAH\Semester 3\Praktikum Alpro 2>
```

Deskripsi Program

Program diatas merupakan program Go yang dibuat untuk menyimpan dan menampilkan informasi harga buah menggunakan struktur data map di Go. Dalam program, sebuah map bernama hargaBuah dibuat dengan nama buah sebagai kunci (key) dan harga buah sebagai nilai (value). Program mengisi map tersebut dengan tiga entri: "Apel" dengan harga 5000, "Pisang" dengan harga 3000, dan "Mangga" dengan harga 7000. Algoritma program melibatkan iterasi map menggunakan `range` untuk mencetak setiap buah beserta harganya. Setelah itu, program secara khusus menampilkan harga buah "Mangga" menggunakan kunci "Mangga" untuk mengakses nilainya di dalam map. Output program menampilkan daftar harga semua buah dan harga buah "Mangga" secara spesifik.

III. UNGUIDED

1. Suatu lingkaran didefinisikan dengan koordinat titik pusat (cx, cy) dengan radius r. Apabila diberikan dua buah lingkaran, amaka 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 dna lingkaran 2, sednagkan baris ketiga adalah koordinat titik sembarang. Asumsi sumbu x dan y dari semua titik dan juga radius direpresentasikan dengan bulat. Keluaran berupa string yang menyatakan posisi titik "Titik di dalam lingkaran 1 dan 1", "Titik didalam lingkaran 1", "Titik didalam lingkaran 2", atau "Titik diluar 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 didalam suatu lingkaran atau tidak.

Sourcecode

```
package main

import (
    "fmt"
    "math"
```

```

)

// Struct untuk menyimpan data lingkaran
type Rectangle_158 struct {
    xc, yc, radius float64
}

// Struct untuk menyimpan data titik
type Point_158 struct {
    x, y float64
}

// Fungsi untuk menghitung jarak antara dua titik
func jarak(xc, yc, x, y float64) float64 {
    return math.Sqrt(math.Pow(x-xc, 2) + math.Pow(y-yc,
2))
}

// Fungsi untuk menentukan posisi titik relatif terhadap
dua lingkaran
func posisiTitik(lingkaran1, lingkaran2 Rectangle_158,
titik Point_158) string {
    jarak1 := jarak(lingkaran1.xc, lingkaran1.yc,
titik.x, titik.y)
    jarak2 := jarak(lingkaran2.xc, lingkaran2.yc,
titik.x, titik.y)

    if jarak1 <= lingkaran1.radius && jarak2 <=
lingkaran2.radius {
        return "Titik di dalam lingkaran 1 dan 2"
    } else if jarak1 <= lingkaran1.radius {
        return "Titik di dalam lingkaran 1"
    } else if jarak2 <= lingkaran2.radius {
        return "Titik di dalam lingkaran 2"
    } else {
        return "Titik di luar lingkaran 1 dan 2"
    }
}

func main() {
    var xc1, yc1, r1, xc2, yc2, r2, x, y float64
    fmt.Scan(&xc1, &yc1, &r1)
    fmt.Scan(&xc2, &yc2, &r2)
    fmt.Scan(&x, &y)

    lingkaran1 := Rectangle_158{xc: xc1, yc: yc1,
radius: r1}
    lingkaran2 := Rectangle_158{xc: xc2, yc: yc2,
radius: r2}
    titik := Point_158{x: x, y: y}

    hasil := posisiTitik(lingkaran1, lingkaran2, titik)

```

```
    fmt.Println(hasil)
}
```

Screenshot Output

```
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 7\Unguided1.go"
1 1 5
8 8 4
2 2
Point_158 di dalam lingkaran 1
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> █
```

Deskripsi Program

Program diatas merupakan program yang mengimplementasikan penggunaan struct untuk menentukan posisi sebuah titik terhadap dua lingkaran, apakah titik tersebut berada di dalam salah satu lingkaran, di dalam keduanya, atau di luar kedua lingkaran. Algoritma dimulai dengan membaca koordinat pusat dan radius dari dua lingkaran serta koordinat titik yang akan diperiksa. Program menggunakan fungsi jarak() untuk menghitung jarak antara titik dan pusat lingkaran dengan rumus jarak Euclidean. Selanjutnya, program membandingkan jarak tersebut dengan radius masing-masing lingkaran untuk menentukan posisi relatif titik. Jika jarak titik ke pusat lingkaran kurang dari atau sama dengan radius, titik dianggap berada di dalam lingkaran tersebut. Program memberikan output berupa salah satu dari empat kemungkinan: "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." Sebagai contoh, jika lingkaran 1 berpusat di (0,0) dengan radius 5, lingkaran 2 di (3,3) dengan radius 4, dan titik berada di (2,2), maka outputnya adalah "Titik di dalam lingkaran 1 dan 2."

2. Sebuah array digunakan untuk menampung sekumpulan bilangan bulat. Buatlah program yang digunakan untuk mengisi array tersebut sebanyak N elemen nilai. Asumsikan array memiliki kapasitas penyimpanan data sejumlah elemen tertentu. Program dapat menampilkan beberapa informasi berikut:
 - a. Menampilkan keseluruhan isi dari array.
 - b. Menampilkan elemen-elemen array dengan indeks ganjil saja.
 - c. Menampilkan elemen-elemen array dengan indeks genap saja (asumsi 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_158, index, x_158, target int
    fmt.Print("Masukkan N: ")
    fmt.Scan(&n_158)

    array := make([]int, n_158)

    // Mengisi array
    for i := 0; i < n_158; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i)
        fmt.Scan(&array[i])
    }

    for {
        fmt.Println("\nPilih operasi:")
        fmt.Println("1. Tampilkan seluruh 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. Hitung rata-rata")
        fmt.Println("7. Hitung standar deviasi")
        fmt.Println("8. Hitung frekuensi bilangan tertentu")
        fmt.Println("9. Keluar")
        fmt.Print("Pilihan: ")
        var pilihan int
        fmt.Scan(&pilihan)

        switch pilihan {
            case 1:
                fmt.Println("Isi array:", array)
            case 2:
                fmt.Println("Elemen dengan indeks ganjil:")
            case 3:
                fmt.Println("Elemen dengan indeks genap:")
            case 4:
                fmt.Println("Elemen dengan indeks kelipatan:")
            case 5:
                fmt.Println("Hapus elemen:")
            case 6:
                fmt.Println("Rata-rata:")
            case 7:
                fmt.Println("Standar deviasi:")
            case 8:
                fmt.Println("Frekuensi:")
            case 9:
                return
        }
    }
}
```

```

        for i := 1; i < len(array); i += 2 {
            fmt.Printf("Indeks %d: %d\n_158", i,
array[i])
        }
    case 3:
        fmt.Println("Elemen dengan indeks genap:")
        for i := 0; i < len(array); i += 2 {
            fmt.Printf("Indeks %d: %d\n_158", i,
array[i])
        }
    case 4:
        fmt.Print("Masukkan bilangan untuk kelipatan
indeks (x): ")
        fmt.Scan(&x_158)
        fmt.Println("Elemen dengan indeks
kelipatan", x_158, ":")
        for i := 0; i < len(array); i++ {
            if i%x_158 == 0 {
                fmt.Printf("Indeks %d: %d\n_158", i,
array[i])
            }
        }
    case 5:
        fmt.Print("Masukkan indeks yang ingin
dihapus: ")
        fmt.Scan(&index)
        if index >= 0 && index < len(array) {
            array = append(array[:index],
array[index+1:]...)
            fmt.Println("Isi array setelah
dihapus:", array)
        } else {
            fmt.Println("Indeks tidak valid!")
        }
    case 6:
        sum := 0
        for _, val := range array {
            sum += val
        }
        average := float64(sum) /
float64(len(array))
        fmt.Printf("Rata-rata: %.2f\n_158", average)
    case 7:
        sum := 0
        for _, val := range array {
            sum += val
        }
        average := float64(sum) /
float64(len(array))

        var varianceSum float64
        for _, val := range array {

```

```

                                varianceSum += math.Pow(float64(val) -
average, 2)
                                }
                                standardDeviation := math.Sqrt(varianceSum /
float64(len(array)))
                                fmt.Printf("Standar deviasi: %.2f\n_158",
standardDeviation)
                                case 8:
                                    fmt.Print("Masukkan bilangan yang ingin
dicari frekuensinya: ")
                                    fmt.Scan(&target)
                                    count := 0
                                    for _, val := range array {
                                        if val == target {
                                            count++
                                        }
                                    }
                                    fmt.Printf("Frekuensi %d: %d kali\n_158",
target, count)
                                case 9:
                                    fmt.Println("Keluar dari program.")
                                    return
                                default:
                                    fmt.Println("Pilihan tidak valid.")
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Screenshoot Output

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 7\Unguided2.go"
Masukkan jumlah elemen array (N): 5
Masukkan elemen ke-0: 1
Masukkan elemen ke-1: 2
Masukkan elemen ke-2: 3
Masukkan elemen ke-3: 4
Masukkan elemen ke-4: 5

Pilih operasi:
1. Tampilkan seluruh 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. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi bilangan tertentu
9. Keluar
Pilihan: 1
Isi array: [1 2 3 4 5]

```

Deskripsi Program

Program diatas merupakan program Go yang dibuat untuk mengelola dan menganalisis elemen dalam sebuah array integer. Pengguna pertama-tama diminta untuk menentukan ukuran array (N) dan mengisi elemen-elemennya. Setelah array terisi, pengguna diberikan berbagai pilihan operasi, seperti menampilkan seluruh isi array, menampilkan elemen

dengan indeks tertentu (ganjil, genap, atau kelipatan angka tertentu), menghapus elemen pada indeks tertentu, menghitung rata-rata dan standar deviasi elemen, serta menghitung frekuensi kemunculan suatu bilangan tertentu. Algoritma untuk setiap operasi melibatkan iterasi array, operasi matematis seperti pembagian dan akar kuadrat (untuk standar deviasi), serta manipulasi array menggunakan metode append untuk menghapus elemen. Pengguna dapat terus memilih operasi hingga mereka memilih opsi untuk keluar dari program. Program ini memberikan fleksibilitas untuk memanipulasi array dengan cara yang terstruktur dan mudah diikuti.

3. Sebuah program digunakan untuk menyimpan dan menampilkan nama-nama klub yang memenangkan pertandingan bola pada suatu grup pertandingan. Buatlah program yang digunakan untuk merekap skor pertandingan bola 2 buah klub bola yang berlaga. Pertama-tama program meminta masukan nama-nama klub yang bertanding, kemudian program meminta masukan skor hasil pertandingan kedua klub tersebut. Yang disimpan dalam array adalah nama-nama klub yang menang saja. Proses input skor berhenti ketika skor salah satu atau kedua klub tidak valid (negatif). Di akhir program, tampilkan daftar klub yang memenangkan pertandingan.

Sourcecode

```
package main

import "fmt"

func main() {
    var klubA, klubB string
    var skorA, skorB int
    klubMenang := []string{}

    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&klubA)

    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scanln(&klubB)

    for pertandingan := 1; pertandingan <= 9;
    pertandingan++ {
        fmt.Printf("Masukkan skor pertandingan %d: ",
        pertandingan)
        fmt.Scanln(&skorA, &skorB)

        // Cek validitas skor
        if skorA < 0 || skorB < 0 {
            fmt.Println("Skor tidak valid. Program
            berhenti.")
        }
    }
}
```

```

        break
    }

    // Tentukan pemenang
    if skorA > skorB {
        klubMenang = append(klubMenang,
fmt.Sprintf("Hasil %d: %s", pertandingan, klubA))
    } else if skorB > skorA {
        klubMenang = append(klubMenang,
fmt.Sprintf("Hasil %d: %s", pertandingan, klubB))
    } else {
        klubMenang = append(klubMenang,
fmt.Sprintf("Hasil %d: Draw", pertandingan))
    }
}

fmt.Println("\nDaftar klub yang memenangkan
pertandingan:")
for _, klub := range klubMenang {
    fmt.Println(klub)
}
}

```

Screenshoot Output

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 7\Unguided3.go"
Masukkan nama Klub A: MU
Masukkan nama Klub B: Inter
Masukkan skor pertandingan 1: 2 0
Masukkan skor pertandingan 2: 1 2
Masukkan skor pertandingan 3: 2 2
Masukkan skor pertandingan 4: 0 1
Masukkan skor pertandingan 5: 3 2
Masukkan skor pertandingan 6: 1 0
Masukkan skor pertandingan 7: 5 2
Masukkan skor pertandingan 8: 2 3
Masukkan skor pertandingan 9: -1 2
Skor tidak valid. Program berhenti.

Daftar klub yang memenangkan pertandingan:
Hasil 1: MU
Hasil 2: Inter
Hasil 3: Draw
Hasil 4: Inter
Hasil 5: MU
Hasil 6: MU
Hasil 7: MU
Hasil 8: Inter
Hasil 9: Inter
PS D:\KULIAH\Semester 3\Praktikum Alpro 2>

```

Deskripsi Program

Program diatas merupakan program Go yang dibuat untuk mencatat hasil pertandingan antara dua klub sepak bola dan menampilkan daftar pemenang dari setiap pertandingan. Pengguna diminta memasukkan nama dua klub, yaitu Klub A dan Klub B. Selanjutnya, program menerima skor dari masing-masing klub untuk maksimal 9 pertandingan. Algoritma program mengecek validitas skor (tidak boleh negatif) dan menentukan

pemenang berdasarkan skor tertinggi. Jika skor Klub A lebih besar, maka Klub A dianggap sebagai pemenang; sebaliknya, jika skor Klub B lebih besar, Klub B yang menang. Jika skornya sama, pertandingan dianggap seri (draw). Hasil dari setiap pertandingan dicatat dalam daftar klubMenang, yang mencakup informasi pertandingan keberapa dan hasilnya. Setelah semua skor dimasukkan atau jika terjadi kesalahan (skor negatif), program mencetak daftar hasil pertandingan. Misalnya, jika Klub A menang pada pertandingan 1, Klub B menang pada pertandingan 2, dan pertandingan 3 berakhir seri, hasil akhirnya adalah daftar yang menunjukkan "Hasil 1: Klub A", "Hasil 2: Klub B", dan "Hasil 3: Draw".

4. Sebuah array digunakan untuk menampung sekumpulan karakter, Anda diminta untuk membuat sebuah subprogram untuk melakukan membalikkan urutan isi array dan memeriksa apakah membentuk palindrom.

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk membalikkan array
func reverseTeks_158(tab []rune, n int) {
    for i := 0; i < n/2; i++ {
        temp := tab[i]
        tab[i] = tab[n-i-1]
        tab[n-i-1] = temp
    }
}

// Fungsi untuk memeriksa apakah array membentuk
palindrome
func Palindrome_158(tab []rune, n int) bool {
    for i := 0; i < n/2; i++ {
        if tab[i] != tab[n-i-1] {
            return false
        }
    }
    return true
}

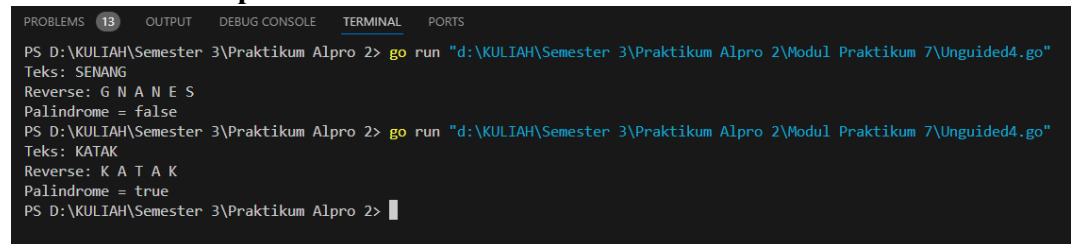
func main() {
    var input string
    fmt.Print("Teks: ")
    fmt.Scanln(&input)
```

```
// Menghilangkan spasi dari input dan mengubah
menjadi array rune
var tab []rune
for _, char := range input {
    if char != ' ' {
        tab = append(tab, char)
    }
}
n := len(tab)

reverseTeks_158(tab, n)
fmt.Print("Reverse: ")
for _, char := range tab {
    fmt.Printf("%c ", char)
}
fmt.Println()

palindrome := Palindrome_158(tab, n)
fmt.Printf("Palindrome = %v\n", palindrome)
}
```

Screenshoot Output



```
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 7\Unguided4.go"
Teks: SENANG
Reverse: G N A N E S
Palindrome = false
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 7\Unguided4.go"
Teks: KATAK
Reverse: K A T A K
Palindrome = true
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> |
```

Deskripsi Program

Program diatas merupakan program Go yang dibuat untuk memeriksa apakah sebuah teks adalah palindrome, yaitu teks yang sama jika dibaca dari depan maupun dari belakang, dengan mengabaikan spasi. Algoritma dimulai dengan membaca input berupa teks, lalu menghapus semua spasi dari teks tersebut untuk menghasilkan array karakter (rune). Program memiliki dua fungsi utama: `reverseTeks_158`, yang membalikkan urutan elemen dalam array, dan `Palindrome_158`, yang memeriksa apakah elemen array simetris dari awal hingga akhir. Pertama, program membalikkan teks menggunakan `reverseTeks_158` dan mencetak hasilnya. Kemudian, `Palindrome_158` memeriksa kesamaan karakter pada posisi simetris dalam array untuk menentukan apakah teks tersebut adalah palindrome. Hasil akhir program mencakup teks yang sudah dibalik dan informasi apakah teks tersebut merupakan palindrome (true atau false). Sebagai contoh, jika input adalah "kasur ini rusak", output akan menunjukkan teks yang dibalik sebagai "k a s u r i n i r u s a k" dan menyatakan bahwa teks tersebut adalah palindrome ('true').

DAFTAR PUSTAKA

[1] Asisten Praktikum, “Modul 7 Struct dan Array”, Modul Praktikum, 2024