

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCT DAN ARRAY**



Disusun Oleh :

Liya Khoirunnisa / 2311102124

IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

a. Struct

Struct adalah kumpulan definisi variabel (atau property) dan atau fungsi (atau method), yang dibungkus dengan nama tertentu. Property dalam struct, tipe datanya bisa bervariasi. Mirip seperti map , hanya saja key-nya sudah didefinisikan di awal, dan tipe data tiap itemnya bisa berbeda. Dengan memanfaatkan struct, data akan terbungkus lebih rapi dan mudah di-maintain. Sebuah struct bisa dimanfaatkan untuk mencetak banyak objek. Keyword `type` digunakan untuk deklarasi struct. Cara membuat variabel objek sama seperti pembuatan variabel biasa. Tinggal tulis saja nama variabel diikuti nama struct, contoh: `var s1 student`. Semua property variabel objek pada awalnya memiliki nilai default sesuai tipe datanya. Property variabel objek bisa diakses nilainya menggunakan notasi titik, contohnya `s1.name` . Nilai property-nya juga bisa diubah. Cara inisialisasi variabel objek adalah dengan menambahkan kurung kurawal setelah nama struct. Nilai masing-masing property bisa diisi pada saat inisialisasi.

b. Array

Array adalah kumpulan data bertipe sama, yang disimpan dalam sebuah variabel. Array memiliki kapasitas yang nilainya ditentukan pada saat pembuatan, menjadikan elemen/data yang disimpan di array tersebut jumlahnya tidak boleh melebihi yang sudah dialokasikan. Default nilai tiap elemen array pada awalnya tergantung dari tipe datanya. Setiap elemen array memiliki indeks berupa angka yang merepresentasikan posisi urutan elemen tersebut. Indeks array dimulai dari 0. Cara mengisi slot elemen array bisa dilihat, yaitu dengan langsung mengakses elemen menggunakan indeks, lalu mengisinya. Pengisian elemen array bisa dilakukan pada saat deklarasi variabel. Caranya dengan menuliskan data elemen dalam kurung kurawal setelah tipe data, dengan pembatas antar elemen adalah tanda koma (,). Penggunaan fungsi `fmt.Println()` pada data array tanpa mengakses indeks tertentu, akan menghasilkan output dalam bentuk string dari semua array yang ada. Teknik ini biasa digunakan untuk debugging data array. Array multidimensi adalah array yang tiap elemennya juga berupa array (dan bisa seterusnya, tergantung jumlah dimensinya). Cara deklarasi array multidimensi secara umum sama dengan cara deklarasi array biasa. Khusus untuk array yang merupakan sub dimensi atau elemen, boleh tidak dituliskan jumlah datanya.

Insialisasi array

1. Inisialisasi horizontal dan vertikal

Elemen array bisa dituliskan dalam bentuk horizontal ataupun dalam bentuk vertikal. Perlu diperhatikan, khusus deklarasi menggunakan cara vertikal, perlu dituliskan tanda koma pada akhir elemen terakhir. Jika tidak ditulis akan muncul error.

2. Inisialisai nilai array tanpa jumlah elemen

Deklarasi array yang nilainya diset di awal, boleh tidak dituliskan jumlah lebar array-nya, cukup ganti dengan tanda 3 titik (...). Jumlah elemen akan dikalkulasi secara otomatis menyesuaikan data elemen yang diisikan.

Perulangan elemen array :

1. For

Keyword for dan array memiliki hubungan yang sangat erat. Dengan memanfaatkan variabel iterasi perulangan untuk mengakses elemen berdasarkan indeks-nya.

2. For -range

Ada cara yang lebih mudah yang bisa dimanfaatkan untuk me-looping sebuah data array, yaitu menggunakan keyword for – range.

3. Variabel underscore dalam For – range

Kadang kala ketika looping menggunakan for - range , ada kemungkinan dimana data yang dibutuhkan adalah elemen-nya saja, indeks-nya tidak. Range mengembalikan 2 data, yaitu indeks dan elemen. Seperti yang sudah diketahui, bahwa di Golang tidak memperbolehkan adanya variabel yang menaggur atau tidak dipakai. Jika dipaksakan, error akan muncul. variabel i diganti dengan _ , karena kebetulan variabel i tidak digunakan.

II. GUIDED

1. Program menghitung durasi waktu parkir kendaraan menggunakan struct

Source code

```
package main

import "fmt"

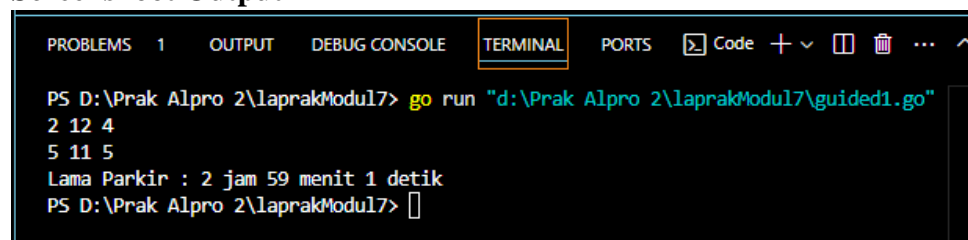
type waktu struct {
    jam, menit, detik int
}

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



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [ ] [ ] ... ^
PS D:\Prak Alpro 2\laprakModul7> go run "d:\Prak Alpro 2\laprakModul7\guided1.go"
2 12 4
5 11 5
Lama Parkir : 2 jam 59 menit 1 detik
PS D:\Prak Alpro 2\laprakModul7> [ ]
```

Deskripsi Program

Program di atas dibuat untuk menghitung durasi waktu parkir berdasarkan waktu masuk dan keluar kendaraan dalam format jam, menit, dan detik. Program menggunakan struct untuk menyimpan jam, menit, dan detik. Program meminta input waktu kedatangan (wParkir) dan waktu pulang (wPulang) dalam satuan jam, menit, dan detik. Lalu waktu yang telah diinputkan akan dikonversi ke detik dengan menyimpan total detik

kedatangan (dParkir) dan total detik pulang (dPulang). Untuk menentukan lama parkir yaitu dengan mengurangi detik pulang (dPulang) dengan detik kedatangan (dParkir). Lalu waktu yang didapatkan dikonversi ke jam, menit, dan detik. Kemudian durasi waktu parkir akan dicetak ke layar dengan format jam, menit, dan waktu.

***Note:** Pada terminal terdapat 1 problem dikarenakan package dan func main digunakan juga di file lain

2. Program untuk menambahkan nama baru ke daftar teman menggunakan slice

Source code

```
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"}

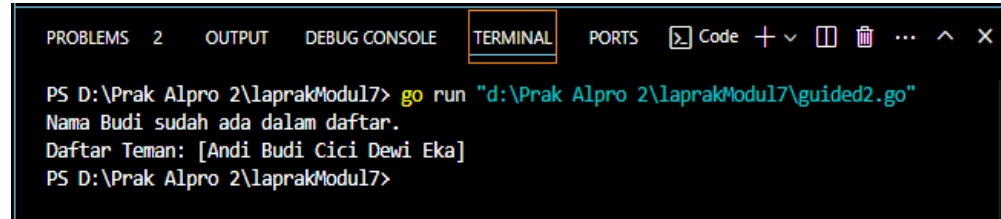
    // Nama-nama baru yang ingin ditambahkan
    namaBaru := []string{"Dewi", "Budi", "Eka"}

    // 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



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [ ] [ ] ... ^ X
PS D:\Prak Alpro 2\laprakModul7> go run "d:\Prak Alpro 2\laprakModul7\guided2.go"
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\Prak Alpro 2\laprakModul7>
```

Deskripsi Program

Program di atas dibuat untuk menambahkan nama baru ke daftar teman. Program menggunakan slice untuk menyimpan nama awal, yaitu "Andi, Budi, dan Cici". Terdapat fungsi sudahAda untuk mengecek nama yang sudah ada di dalam slice. Di dalam fungsi terdapat perulangan untuk memeriksa nama yang sudah ada dalam daftar. Jika nama ditemukan, maka akan mengembalikan nilai true, namun jika tidak maka akan mengembalikan nilai false. Pada fungsi main, terdapat slice namaBaru yang berisi daftar nama yang ditambahkan, yaitu "Dewi, Budi, Eka". Untuk menambahkan nama baru menggunakan perulangan. Jika nama yang ditambahkan belum ada, maka akan ditambahkan ke daftar teman. Namun jika nama yang ditambahkan sudah ada, maka akan menampilkan pesan "sudah ada dalam daftar". Kemudian jika proses sudah selesai, hasil akan dicetak ke layar.

***Note:** Pada terminal terdapat 2 problem dikarenakan package dan func main digunakan juga di file lain

3. Program untuk menampilkan daftar harga buah menggunakan map

Source code

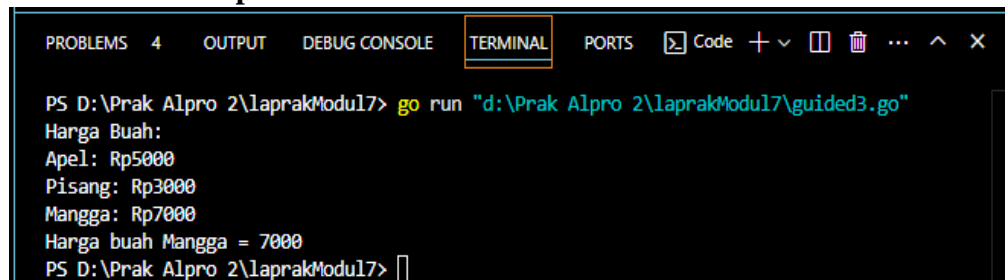
```
package main
import (
    "fmt"
)

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



```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [] [] ... ^ X

PS D:\Prak Alpro 2\laprakModul7> go run "d:\Prak Alpro 2\laprakModul7\guided3.go"
Harga Buah:
Apel: Rp5000
Pisang: Rp3000
Mangga: Rp7000
Harga buah Mangga = 7000
PS D:\Prak Alpro 2\laprakModul7> 
```

Deskripsi Program

Program di atas dibuat untuk menampilkan harga buah. Program menggunakan map untuk menyimpan nama buah dan harga dalam bentuk integer. Program akan mencetak semua nama buah dan harga yang terdapat pada map dengan menggunakan perulangan for untuk mengakses map. Kemudian hasil akan dicetak ke layar.

***Note:** Pada terminal terdapat 4 problem dikarenakan package dan func main digunakan juga di file lain.

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. Masukan terdiri dari beberapa tiga baris. Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingtran 2, sedangkan baris ketiga adalah koordinat titik sembarang. Asumsi sumbu x dan y dari semua titik dan juga radius direpresentasikan dengan bilangan bulat. Keluaran berupa string yang menyatakan posisi titik "Titik di dalam lingkaran 1 dan * 2 " Titik di dalam lingkaran 1", "Titik di dalam lingkaran 2", atau "Titik di luar lingkaran 1 dan 2". 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. Catatan: lihat paket math dalam lampiran untuk menggunakan fungsi math.sqrt() untuk menghitung akar kuadrat.

Source code

```
/* Liya Khoirunnisa - 2311102124 */
package main

import (
    "fmt"
    "math" // Untuk operasi matematika
)

// Struct untuk Lingkaran
type Lingkaran struct {
    cx, cy, radius int
}

// Fungsi untuk menghitung jarak antara dua titik
func jarak(x1, y1, x2, y2 int) float64 {
    return math.Sqrt(float64((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1)))
}

// Fungsi untuk menentukan apakah titik berada di dalam lingkaran
func cekDalamLingkaran(circle Lingkaran, x, y int) bool {
    return jarak(circle.cx, circle.cy, x, y) <=
float64(circle.radius)
}

func main() {
    // Deklarasi variabel untuk lingkaran 1 dan 2
    var lingkaran1, lingkaran2 Lingkaran
    var x, y int

    // Input lingkaran 1
    fmt.Println("Lingkaran 1")
    fmt.Print("Masukkan titik pusat (cx1, cy1) dan radius
```



```

r1: ")
    fmt.Scan(&lingkaran1.cx, &lingkaran1.cy,
&lingkaran1.radius)

    // Input lingkaran 2
    fmt.Println("\nLingkaran 2")
    fmt.Print("Masukkan titik pusat (cx2, cy2) dan radius
r2: ")
    fmt.Scan(&lingkaran2.cx, &lingkaran2.cy,
&lingkaran2.radius)

    // Input titik sembarang
    fmt.Print("\nMasukkan titik sembarang (x, y): ")
    fmt.Scan(&x, &y)

    // Mengecek letak titik di dalam lingkaran
    titikDalam1 := cekDalamLingkaran(lingkaran1, x, y)
    titikDalam2 := cekDalamLingkaran(lingkaran2, x, y)

    // Menentukan output berdasarkan posisi titik
    fmt.Print("\nPosisi titik: ")
    if titikDalam1 && titikDalam2 {
        // Jika titik berada di dalam kedua lingkaran
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if titikDalam1 {
        // Jika titik hanya berada di dalam lingkaran 1
        fmt.Println("Titik di dalam lingkaran 1")
    } else if titikDalam2 {
        // Jika titik hanya berada di dalam lingkaran 2
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        // Jika titik berada di luar kedua lingkaran
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

Screenshoot Output

```

PROBLEMS 23  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Code + v  [ ]  [ ]  ...
PS D:\Prak Alpro 2> go run "d:\Prak Alpro 2\laprakModul7\unguided1.go"
Lingkaran 1
Masukkan titik pusat (cx1, cy1) dan radius r1: 1 1 5

Lingkaran 2
Masukkan titik pusat (cx2, cy2) dan radius r2: 8 8 4

Masukkan titik sembarang (x, y): 2 2

Posisi titik: Titik di dalam lingkaran 1
PS D:\Prak Alpro 2> 

```

Deskripsi Program

Program di atas dibuat untuk menentukan apakah sebuah titik sembarang berada di dalam lingkaran satu, dua, atau tidak ada dari kedua lingkaran. Terdapat struct untuk mendefinisikan atribut lingkaran, yaitu cx, cy dan radius. Terdapat juga dua fungsi yaitu jarak untuk menghitung jarak dan cekDalam lingkaran untuk mengecek jarak titik ke pusat lingkaran. Pengguna diminta untuk memasukkan koordinat pusat dan radius dari dua lingkaran, serta titik sembarang. Selanjutnya program menghitung jarak antara titik sembarang dengan pusat masing-masing lingkaran dengan rumus $jarak = \sqrt{(a - c)^2 + (b - d)^2}$. Berdasarkan hasil perhitungan, program akan memberikan output yang menyatakan apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, atau berada di luar keduanya.

***Note:** Pada terminal terdapat 23 problem dikarenakan package dan func main digunakan juga di file lain

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

Source code

```
/* Liya Khoirunnisa - 2311102124 */
package main

import (
    "fmt"
    "math" // Untuk operasi matematika
)

// Fungsi untuk menampilkan isi array
```

```

func tampilkanArray(data []int) {
    fmt.Println("Isi array:", data)
}

// Fungsi untuk menampilkan elemen dengan indeks ganjil
func tampilkanIndeksGanjil(data []int) {
    fmt.Print("Elemen berindeks ganjil: ")
    for i := 1; i < len(data); i += 2 {
        fmt.Print(data[i], " ")
    }
    fmt.Println()
}

// Fungsi untuk menampilkan elemen dengan indeks genap
func tampilkanIndeksGenap(data []int) {
    fmt.Print("Elemen berindeks genap: ")
    for i := 0; i < len(data); i += 2 {
        fmt.Print(data[i], " ")
    }
    fmt.Println()
}

// Fungsi untuk menampilkan elemen dengan indeks kelipatan x
func tampilkanKelipatan(data []int, x int) {
    fmt.Print("Elemen dengan indeks kelipatan ", x, ": ")
    for i := 0; i < len(data); i++ {
        if i%x == 0 {
            fmt.Print(data[i], " ")
        }
    }
    fmt.Println()
}

// Fungsi untuk menghapus elemen pada indeks tertentu
func hapusIndeks(data *[]int, indeks int) {
    if indeks < 0 || indeks >= len(*data) {
        fmt.Println("Indeks tidak valid.")
        return
    }
    *data = append((*data)[:indeks], (*data)[indeks+1:]...)
    fmt.Println("Elemen pada indeks ke-", indeks, "telah dihapus.")
    tampilkanArray(*data)
}

// Fungsi untuk menghitung rata-rata
func hitungRataRata(data []int) float64 {
    if len(data) == 0 {
        return 0
    }
    jumlah := 0
    for _, nilai := range data {
        jumlah += nilai
    }
    return float64(jumlah) / float64(len(data))
}

```

```

// Fungsi untuk menghitung simpangan baku
func hitungSimpanganBaku(data []int) float64 {
    if len(data) == 0 {
        return 0
    }
    rataRata := hitungRataRata(data)
    jumlah := 0.0
    for _, nilai := range data {
        jumlah += math.Pow(float64(nilai)-rataRata, 2)
    }
    return math.Sqrt(jumlah / float64(len(data)))
}

// Fungsi untuk menghitung frekuensi dari suatu bilangan
func frekuensi(data []int, nilai int) int {
    hitung := 0
    for _, v := range data {
        if v == nilai {
            hitung++
        }
    }
    return hitung
}

func main() {
    // Deklarasi variabel
    var n, pilihan int
    fmt.Print("Masukkan jumlah elemen array: ")
    fmt.Scan(&n) // Meminta input jumlah elemen

    // Inisialisai array dengan ukuran n
    data := make([]int, n)
    fmt.Println("Masukkan elemen array:")

    // Perulangan meminta input elemen array
    for i := 0; i < n; i++ {
        fmt.Printf("Elemen ke-%d: ", i)
        fmt.Scan(&data[i])
    }

    // Menu
    for {
        fmt.Println("\nMenu:")
        fmt.Println("1. Tampilkan 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 x")
        fmt.Println("5. Hapus elemen pada indeks tertentu")
        fmt.Println("6. Tampilkan rata-rata")
        fmt.Println("7. Tampilkan simpangan baku")
        fmt.Println("8. Tampilkan frekuensi dari suatu bilangan")
    }
}

```

```

        fmt.Println("9. Keluar")

        fmt.Print("Pilih menu: ")
        fmt.Scan(&pilihan) // Input pilihan menu

        // Switch case pilihan menu
        switch pilihan {
        case 1:
            // Menampilkan isi array
            tampilkanArray(data)
        case 2:
            // Menampilkan elemen berindeks ganjil
            tampilkanIndeksGanjil(data)
        case 3:
            // Menampilkan elemen berindeks genap
            tampilkanIndeksGenap(data)
        case 4:
            var x int
            fmt.Print("Masukkan bilangan x: ")
            fmt.Scan(&x) // Input kelipatan x

            // Menampilkan elemen dengan indeks kelipatan
            tampilkanKelipatan(data, x)
        case 5:
            var indeks int
            fmt.Print("Masukkan indeks yang ingin dihapus: ")
            fmt.Scan(&indeks) // Input indeks yang ingin dihapus

            // Menghapus elemen pada indeks yang diinputkan
            hapusIndeks(&data, indeks)
        case 6:
            // Menghitung rata-rata
            rataRata := hitungRataRata(data)
            fmt.Printf("Rata-rata: %.2f\n", rataRata)
        case 7:
            // Menghitung simpangan baku
            simpanganBaku := hitungSimpanganBaku(data)
            fmt.Printf("Simpangan baku: %.2f\n", simpanganBaku)
        case 8:
            // Menghitung frekuensi nilai
            var nilai int
            fmt.Print("Masukkan bilangan untuk frekuensi: ")
            fmt.Scan(&nilai)
            frekuensiVal := frekuensi(data, nilai)
            fmt.Printf("Frekuensi dari %d: %d\n", nilai, frekuensiVal)
        case 9:
            // Keluar
            fmt.Println("Keluar dari program.")
            return
        default:
            fmt.Println("Pilihan tidak tersedia")
        }
    }

```

```
}  
}
```

Screenshoot Output

```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS [x] Code + - [ ] [ ] ...  
  
PS D:\Prak Alpro 2> go run "d:\Prak Alpro 2\laprakModul7\unguided2.go"  
Masukkan jumlah elemen array: 5  
Masukkan elemen array:  
Elemen ke-0: 0  
Elemen ke-1: 2  
Elemen ke-2: 1  
Elemen ke-3: 2  
Elemen ke-4: 4  
  
Menu:  
1. Tampilkan isi array  
2. Tampilkan elemen dengan indeks ganjil  
3. Tampilkan elemen dengan indeks genap  
4. Tampilkan elemen dengan indeks kelipatan x  
5. Hapus elemen pada indeks tertentu  
6. Tampilkan rata-rata  
7. Tampilkan simpangan baku  
8. Tampilkan frekuensi dari suatu bilangan  
9. Keluar  
Pilih menu: 1  
Isi array: [0 2 1 2 4]  
  
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS [x] Code + - [ ] [ ] ...  
  
Menu:  
1. Tampilkan isi array  
2. Tampilkan elemen dengan indeks ganjil  
3. Tampilkan elemen dengan indeks genap  
4. Tampilkan elemen dengan indeks kelipatan x  
5. Hapus elemen pada indeks tertentu  
6. Tampilkan rata-rata  
7. Tampilkan simpangan baku  
8. Tampilkan frekuensi dari suatu bilangan  
9. Keluar  
Pilih menu: 2  
Elemen berindeks ganjil: 2 2  
  
Menu:  
1. Tampilkan isi array  
2. Tampilkan elemen dengan indeks ganjil  
3. Tampilkan elemen dengan indeks genap  
4. Tampilkan elemen dengan indeks kelipatan x  
5. Hapus elemen pada indeks tertentu  
6. Tampilkan rata-rata  
7. Tampilkan simpangan baku  
8. Tampilkan frekuensi dari suatu bilangan  
9. Keluar  
Pilih menu: 3  
Elemen berindeks genap: 0 1 4
```

```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + v [] [] ...

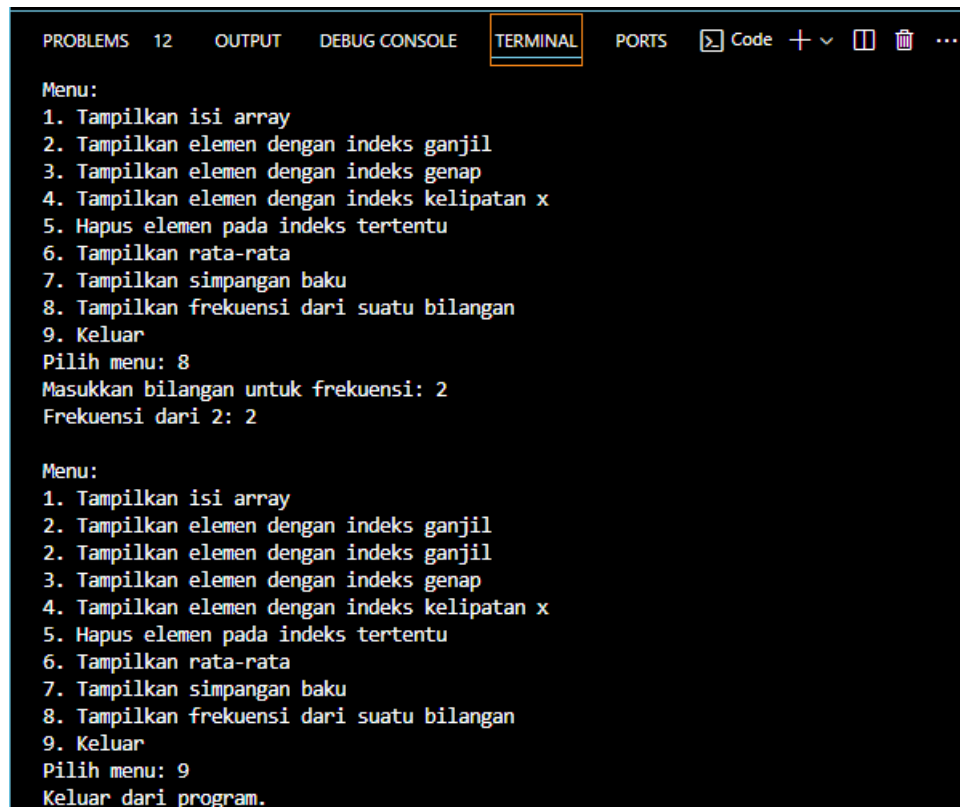
Menu:
1. Tampilkan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan simpangan baku
8. Tampilkan frekuensi dari suatu bilangan
9. Keluar
Pilih menu: 4
Masukkan bilangan x: 2
Elemen dengan indeks kelipatan 2: 0 1 4

Menu:
1. Tampilkan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan simpangan baku
8. Tampilkan frekuensi dari suatu bilangan
9. Keluar
Pilih menu: 5
Masukkan indeks yang ingin dihapus: 0
Elemen pada indeks ke- 0 telah dihapus.
Isi array: [2 1 2 4]

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + v [] [] ...

Menu:
1. Tampilkan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan simpangan baku
8. Tampilkan frekuensi dari suatu bilangan
9. Keluar
Pilih menu: 6
Rata-rata: 2.25

Menu:
1. Tampilkan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan simpangan baku
8. Tampilkan frekuensi dari suatu bilangan
9. Keluar
Pilih menu: 7
Simpangan baku: 1.09
```



```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [] ...

Menu:
1. Tampilkan isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan simpangan baku
8. Tampilkan frekuensi dari suatu bilangan
9. Keluar
Pilih menu: 8
Masukkan bilangan untuk frekuensi: 2
Frekuensi dari 2: 2

Menu:
1. Tampilkan isi array
2. Tampilkan elemen dengan indeks ganjil
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan simpangan baku
8. Tampilkan frekuensi dari suatu bilangan
9. Keluar
Pilih menu: 9
Keluar dari program.
```

Deskripsi Program

Program di atas dibuat untuk melakukan berbagai operasi array dengan memiliki 9 menu. Program memiliki 8 fungsi untuk melakukan operasi array. Program menggunakan array dinamis (slice) untuk membuat array fleksibel. Program diawali dengan meminta input jumlah elemen dari pengguna dan menginputkan elemen array sebanyak jumlah elemen. Setelah itu pengguna dapat memilih operasi yang akan dilakukan menggunakan switch case. Kemudian hasil akan dicetak ke layar.

***Note:** Pada terminal terdapat 12 problem dikarenakan package dan func main digunakan juga di file lain

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.

Source code

```
/* Liya Khoirunnisa - 2311102124 */
package main

import (
    "fmt"
)

// Struct untuk menyimpan informasi pertandingan
type pertandingan struct {
    klubA string // Nama klub A
    klubB string // Nama klub B
    skorA int    // Skor klub A
    skorB int    // Skor klub B
    hasil string // Hasil pertandingan
}

func main() {
    // Deklarasi variabel
    var klubA, klubB string
    var listPertandingan []pertandingan // Array untuk
    menyimpan hasil pertandingan

    // Input nama klub A
    fmt.Print("Klub A: ")
    fmt.Scanln(&klubA)

    // Input nama klub B
    fmt.Print("Klub B: ")
    fmt.Scanln(&klubB)

    fmt.Println()

    // Proses input skor hingga skor negatif
    pertandinganKe := 1 // Variabel untuk menghitung
    pertandingan ke berapa
    for {
        var skorA, skorB int
        fmt.Printf("Pertandingan %d: ", pertandinganKe)
        fmt.Scan(&skorA, &skorB)

        // Cek validitas skor
        if skorA < 0 || skorB < 0 {
            break
        }

        // Menentukan pemenang
        var hasil string
        if skorA > skorB {
            hasil = klubA
        } else if skorB > skorA {
```

```

        hasil = klubB
    } else {
        hasil = "Draw"
    }

    // Simpan hasil pertandingan dalam array
    listPertandingan = append(listPertandingan,
pertandingan{
        klubA: klubA,
        klubB: klubB,
        skorA: skorA,
        skorB: skorB,
        hasil: hasil,
    })

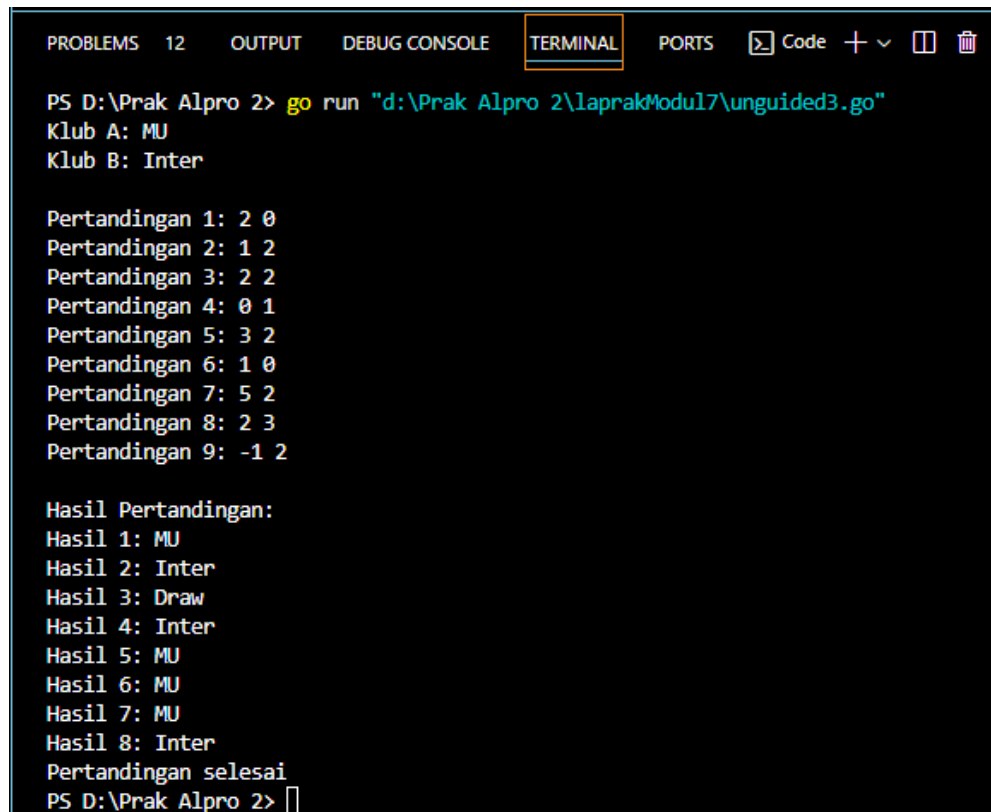
    pertandinganKe++ // Increment untuk pertandingan
    berikutnya
}

// Menampilkan hasil pertandingan
fmt.Println("\nHasil Pertandingan:")
for i, p := range listPertandingan {
    fmt.Printf("Hasil %d: %s\n", i+1, p.hasil)
}

// Menampilkan pesan bahwa pertandingan selesai
fmt.Println("Pertandingan selesai")
}

```

Screenshoot Output



```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [ ] [X]
PS D:\Prak Alpro 2> go run "d:\Prak Alpro 2\laprakModul7\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 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
Pertandingan selesai
PS D:\Prak Alpro 2> [ ]
```

Deskripsi Program

Program di atas dibuat untuk mencatat hasil pertandingan sepak bola antara dua klub. Program menerima input skor dari beberapa pertandingan. Skor negatif yang diinputkan sebagai penanda program berhenti lalu program menampilkan hasil seluruh pertandingan sebelumnya. Program menyimpan nama kedua klub dalam array untuk mencatat hasil pertandingan. Informasi akan disimpan dalam array yang menggunakan struct pertandingan.

***Note:** Pada terminal terdapat 12 problem dikarenakan package dan func main digunakan juga di file lain

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.

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
}

```

Modifikasi program dengan menambahkan fungsi polindrom. Tambahkan instruksi untuk memanggil fungsi tersebut dan menampilkan hasilnya pada program utama.

```

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 */

```

Source code

```

/* Liya Khoirunnisa - 2311102124 */
package main

import "fmt"

// Konstanta maksimum array
const NMAX int = 127

// Tipe data array menyimpan karakter
type tabel [NMAX]rune

// Fungsi untuk mengisi array
func isiArray(t *tabel) int {
    // Deklarasi variabel
    var input string

```

```

// Input teks dari pengguna
fmt.Print("Teks: ")
fmt.Scanln(&input)

// Jika input adalah titik
if input == "." {
    return 0
}

// Panjang input
n := len(input)

// Mengisi array dengan karakter inputan pengguna
for i := 0; i < n && i < NMAX; i++ {
    (*t)[i] = rune(input[i])
}
return n
}

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

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

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

func main() {
    // Deklarasi variabel
    var tab tabel
    var m int

    // Mengisi array
    m = isiArray(&tab)

    // Jika input adalah titik
    if m == 0 {
        return
    }
}

```

```

    }

    // Membalikkan isi array
    balikanArray(&tab, m)

    // Menampilkan teks yang sudah dibalik
    fmt.Print("Reverse teks: ")
    cetakArray(tab, m)

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

```

Screenshoot Output

```

PROBLEMS 13  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Code + - 
PS D:\Prak Alpro 2> go run "d:\Prak Alpro 2\laprakModul7\unguided4.go"
Teks: KATAK
Reverse teks: KATAK
Palindrom: true
PS D:\Prak Alpro 2> go run "d:\Prak Alpro 2\laprakModul7\unguided4.go"
Teks: SENANG
Reverse teks: GNANES
Palindrom: false
PS D:\Prak Alpro 2> 

```

Deskripsi Program

Program di atas dibuat untuk memeriksa kata palindrom. Program menggunakan algoritma manipulasi array yang berisi karakter (tipe data rune) untuk mengolah input. Pengguna diminta memasukkan teks yang ingin dicek. Lalu program membalikkan dan memeriksa apakah teks yang dimasukkan adalah palindrom atau bukan. Teks yang diinputkan dipecah menjadi karakter dan disimpan ke dalam array bertipe rune dengan batas 127 karakter. Program membalikkan isi array menggunakan dua pointer, yang mana elemen awal ditukar dengan elemen akhir secara berpasangan hingga mencapai tengah array. Lalu teks memeriksa apakah urutan karakter dari awal ke akhir sama dengan urutan akhir ke awal. Kemudian hasilnya akan dicetak ke layar dengan output true/false.

***Note:** Pada terminal terdapat 13 problem dikarenakan package dan func main digunakan juga di file lain

DAFTAR PUSTAKA

- [1] Agung, N. (2017). *Dasar Pemrograman Golang*. Dinkes Provinsi Jatim. Tersedia dari <https://dinkes.jatimprov.go.id/userfile/dokumen/Dasar%20Pemrograman%20Golang.pdf>