

LAPORAN PRAKTIKUM
Algoritma Pemrograman
MODUL VII
STRUCT & ARRAY



Oleh:

Ilhan Sahal Mansiz

2311102029

IF-11-02

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Pengertian Struct: Struct (struktur) adalah tipe data yang digunakan untuk mengelompokkan data yang berbeda menjadi satu kesatuan. Struct memungkinkan kita untuk membuat tipe data baru yang terdiri dari beberapa field (atribut) yang dapat memiliki tipe data yang berbeda.

Deklarasi Struct: Untuk mendeklarasikan struct, kita menggunakan kata kunci `type` diikuti dengan nama struct dan kata kunci `struct`. Berikut adalah contoh deklarasi struct:

```
type Mahasiswa struct {  
    Nama    string  
    NIM     string  
    Umur    int  
}
```

Menggunakan Struct: Setelah mendeklarasikan struct, kita dapat membuat instance dari struct tersebut dan mengakses field-nya:

```
func main() {  
    var m Mahasiswa  
    m.Nama = "Budi"  
    m.NIM = "123456"  
    m.Umur = 20  
  
    fmt.Println("Nama:", m.Nama)  
    fmt.Println("NIM:", m.NIM)  
    fmt.Println("Umur:", m.Umur)  
}
```

Keuntungan Menggunakan Struct:

1. **Organisasi Data:** Struct membantu dalam mengorganisir data yang kompleks dengan cara yang lebih terstruktur.

2. **Membuat Tipe Data Baru:** Kita dapat membuat tipe data baru yang sesuai dengan kebutuhan aplikasi kita.
3. **Mendukung Komposisi:** Struct dapat berisi field yang merupakan struct lain, memungkinkan pembuatan struktur data yang lebih kompleks.

Pengertian Array: Array adalah tipe data yang digunakan untuk menyimpan sekumpulan nilai dengan tipe data yang sama. Ukuran array ditentukan saat deklarasi dan tidak dapat diubah setelahnya.

Deklarasi Array: Untuk mendeklarasikan array, kita menggunakan sintaks berikut:

```
var angka [5]int // Array dengan 5 elemen bertipe int
```

Menggunakan Array: Setelah mendeklarasikan array, kita dapat mengisi dan mengakses elemen-elemen array tersebut:

```
func main() {  
    var angka [5]int  
    angka[0] = 10  
    angka[1] = 20  
    angka[2] = 30  
    angka[3] = 40  
    angka[4] = 50  
  
    for i := 0; i < len(angka); i++ {  
        fmt.Println(angka[i])  
    }  
}
```

Keuntungan Menggunakan Array:

1. **Akses Cepat:** Elemen dalam array dapat diakses dengan cepat menggunakan indeks.

2. **Memori Kontigu:** Array menyimpan elemen-elemen dalam memori secara kontigu, yang memungkinkan efisiensi dalam penggunaan memori.
3. **Tipe Data Homogen:** Semua elemen dalam array memiliki tipe data yang sama, yang membantu dalam menjaga konsistensi data.

II. GUIDED

1. Guided 1

Souce Code :

```
package main
import (
    "fmt"
    "sort"
)

// Struktur untuk menampung data mahasiswa
type mahasiswa struct {
    Nama string
    Matematika int
    Fisika int
    Kimia int
    RataRata float64
}

// fungsi untuk menghitung rata - rata nilai tiap mahasiswa
func hitungRataRata(m *mahasiswa) {
    total := m.Matematika +
m.Fisika + m.Kimia
    m.RataRata = float64(total) /
3.0
}

func main() {
    // array untuk menampung data
    mahasiswa := []mahasiswa{
        {"Ali", 85, 90, 80, 0},
        {"Budi", 70, 75, 80, 0},
        {"Cici", 90, 85, 95, 0},
    }
```

```

        {"Doni", 60, 65, 70, 0},

        {"Eka", 100, 95, 90, 0},
        }

        for i := range mahasiswa {

            hitungRataRata(&mahasiswa[i])
        }

        sort.Slice(mahasiswa, func(i, j int)

bool {

        return mahasiswa[i].RataRata > mahasiswa[j].RataRata
        })

        // menampilkan hasil
        fmt.Println("Peringkat    Mahasiswa
berdasarkan rata-rata nilai:")
        for i, m := range mahasiswa {

            fmt.Printf("%d. %s - Rata-rata: %.2f (Matematika:
%d, Fisika: %d, Kimia: %d)\n",

                i+1, m>Nama, m.RataRata, m.Matematika, m.Fisika,
                m.Kimia)
        }
    }
}

```

Screenshot Program :

```

PS C:\Users\ACER\Downloads\Semester 3\Laprak Alpro smstr3\Ilhan Sahal Mansiz_2311102029_Modul7> g
2311102029_Modul7\guided1.go"
Peringkat Mahasiswa berdasarkan rata-rata nilai:
1. Eka - Rata-rata: 95.00 (Matematika: 100, Fisika: 95, Kimia: 90)
2. Cici - Rata-rata: 90.00 (Matematika: 90, Fisika: 85, Kimia: 95)
3. Ali - Rata-rata: 85.00 (Matematika: 85, Fisika: 90, Kimia: 80)
4. Budi - Rata-rata: 75.00 (Matematika: 70, Fisika: 75, Kimia: 80)
5. Doni - Rata-rata: 65.00 (Matematika: 60, Fisika: 65, Kimia: 70)
PS C:\Users\ACER\Downloads\Semester 3\Laprak Alpro smstr3\Ilhan Sahal Mansiz_2311102029_Modul7>

```

Deskripsi Program :

Program ini adalah aplikasi sederhana yang ditulis dalam bahasa Go untuk menghitung dan mengurutkan peringkat mahasiswa berdasarkan rata-rata nilai mereka dalam tiga mata pelajaran: Matematika, Fisika, dan Kimia. Struktur mahasiswa didefinisikan untuk menyimpan data mahasiswa, termasuk nama dan nilai masing-masing mata pelajaran, serta rata-rata nilai. Fungsi `hitungRataRata` digunakan untuk menghitung rata-rata nilai dari setiap mahasiswa, dan nilai-nilai tersebut disimpan dalam field `RataRata` dari struct. Setelah semua rata-rata dihitung, program mengurutkan mahasiswa berdasarkan nilai rata-rata tertinggi menggunakan `sort.Slice`, dan akhirnya menampilkan hasil peringkat mahasiswa beserta nilai masing-masing mata pelajaran dalam format yang terstruktur.

2. Guided 2

Source Code :

```
package main

import "fmt"

func main() {
    // Membuat map dengan NIM sebagai kunci dan Nama
    // sebagai nilai
    mahasiswa := map[string]string{
        "20231001": "Andi",
        "20231002": "Budi",
        "20231003": "Cici",
    }

    // Menambahkan data baru ke map
    mahasiswa["20231004"] = "Dedi"

    // Menampilkan seluruh isi map dalam format kolom dan
    // baris
    fmt.Println("Daftar Mahasiswa:")
    fmt.Println("NIM\t\tNama")
    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }

    // Mengakses data berdasarkan NIM
    nim := "20231002"
    fmt.Println("\nNama Mahasiswa dengan NIM", nim,
        "adalah", mahasiswa[nim])

    // Menghapus data berdasarkan NIM
    delete(mahasiswa, "20231003")

    // Menampilkan isi map setelah data dihapus dalam format
    // kolom dan baris
    fmt.Println("\nDaftar Mahasiswa setelah dihapus:")
}
```

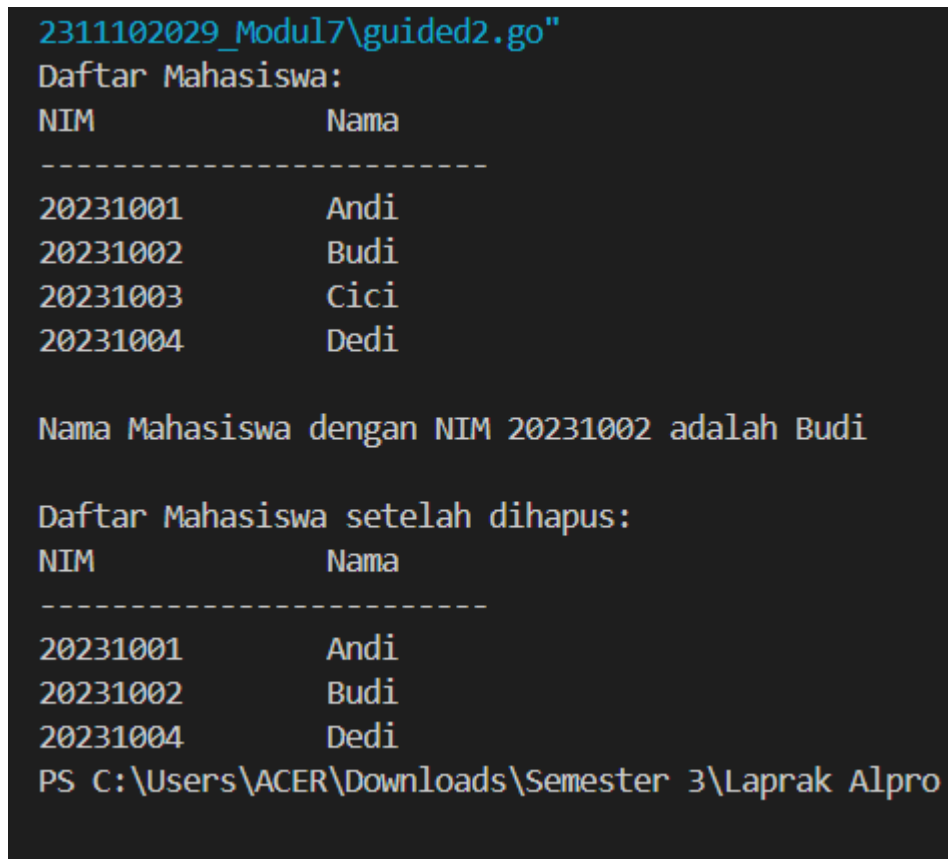


```

        fmt.Println("NIM\t\tNama")
        fmt.Println("-----")
        for nim, nama := range mahasiswa {
            fmt.Printf("%s\t%s\n", nim, nama)
        }
    }
}

```

Screenshot Program :



```

2311102029_Modul7\guided2.go"
Daftar Mahasiswa:
NIM          Nama
-----
20231001     Andi
20231002     Budi
20231003     Cici
20231004     Dedi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM          Nama
-----
20231001     Andi
20231002     Budi
20231004     Dedi
PS C:\Users\ACER\Downloads\Semester 3\Laprak Alpro

```

Deskripsi Program :

Program ini merupakan aplikasi sederhana yang menggunakan struktur data map dalam bahasa Go untuk menyimpan dan mengelola data mahasiswa, di mana NIM (Nomor Induk Mahasiswa) berfungsi sebagai kunci dan Nama sebagai nilai. Program dimulai dengan mendeklarasikan dan menginisialisasi map dengan beberapa entri mahasiswa, kemudian menambahkan data baru ke dalam map. Seluruh isi map ditampilkan dalam format kolom dan baris, memperlihatkan NIM dan Nama mahasiswa. Selanjutnya, program mengakses dan menampilkan nama

mahasiswa berdasarkan NIM tertentu, serta menghapus entri mahasiswa dengan NIM yang ditentukan. Setelah penghapusan, program kembali menampilkan isi map untuk menunjukkan perubahan yang terjadi, sehingga memberikan gambaran yang jelas tentang bagaimana data mahasiswa dapat dikelola menggunakan map.

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 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"**.

Source Code :

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    pusat Titik
    radius int
}

func jarak(p, q Titik) float64 {
    return math.Sqrt(math.Pow(float64(p.x-q.x), 2) +
        math.Pow(float64(p.y-q.y), 2))
}
```

```

    }

    func didalam(c Lingkaran, p Titik) bool {
        return jarak(c.pusat, p) <= float64(c.radius)
    }

    func main() {
        var cx1, cy1, r1, cx2, cy2, r2, px, py int

        fmt.Scan(&cx1, &cy1, &r1, &cx2, &cy2, &r2, &px, &py)

        c1 := Lingkaran{Titik{cx1, cy1}, r1}
        c2 := Lingkaran{Titik{cx2, cy2}, r2}
        p := Titik{px, py}

        inC1 := didalam(c1, p)
        inC2 := didalam(c2, p)

        if inC1 && inC2 {
            fmt.Println("Titik berada di dalam lingkaran 1 dan 2")
        } else if inC1 {
            fmt.Println("Titik berada di dalam lingkaran 1")
        } else if inC2 {
            fmt.Println("Titik berada di dalam lingkaran 2")
        } else {
            fmt.Println("Titik berada di luar lingkaran 1 dan 2")
        }
    }
}

```

Screenshot Program :

```

2311102029_Modul7\unguided1.go"
1 1 5
8 8 4
2 2
Titik berada di dalam lingkaran 1
PS C:\Users\ACER\Downloads\Semester 3\Laprak A
2311102029_Modul7\unguided1.go"
1 2 3
4 5 6
7 8
Titik berada di dalam lingkaran 2
PS C:\Users\ACER\Downloads\Semester 3\Laprak A
2311102029_Modul7\unguided1.go"
5 10 15
-15 4 20
0 0
Titik berada di dalam lingkaran 1 dan 2
PS C:\Users\ACER\Downloads\Semester 3\Laprak A
2311102029_Modul7\unguided1.go"
1 1 5
8 8 4
15 20
Titik berada di luar lingkaran 1 dan 2
PS C:\Users\ACER\Downloads\Semester 3\Laprak A

```

Deskripsi Program :

Program ini adalah aplikasi dalam bahasa Go yang digunakan untuk menentukan posisi suatu titik relatif terhadap dua lingkaran yang didefinisikan oleh pusat dan radiusnya. Program dimulai dengan mendefinisikan dua struktur, yaitu Titik untuk merepresentasikan koordinat titik dalam bentuk x dan y, serta Lingkaran yang menyimpan pusat lingkaran (dalam bentuk Titik) dan radiusnya. Fungsi jarak menghitung jarak antara dua titik menggunakan rumus Pythagoras, sedangkan fungsi didalam mengecek apakah titik tersebut berada di dalam lingkaran dengan membandingkan jarak titik ke pusat lingkaran dengan radius lingkaran. Dalam fungsi main, program meminta input dari pengguna untuk dua lingkaran dan satu titik, kemudian memeriksa apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, dan mencetak hasilnya. Dengan demikian, program ini

memberikan cara yang efisien untuk mengevaluasi posisi titik terhadap lingkaran dalam bidang dua dimensi.

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:

Source Code :

```
package main

import (
    "fmt"
    "math"
    "sort"
)

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

func tampilkanIndeksGanjil(arr []int) {
    fmt.Print("Elemen dengan indeks ganjil: ")
    for i := 0; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}

func tampilkanIndeksGenap(arr []int) {
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 1; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}
```

```

    }

    func tampilkanKelipatanIndeks(arr []int, x int) {
        fmt.Printf("Elemen dengan indeks kelipatan %d: ", x)
        for i := x; i < len(arr); i += x {
            fmt.Print(arr[i], " ")
        }
        fmt.Println()
    }

    func hapusElemen(arr []int, index int) []int {
        if index < 0 || index >= len(arr) {
            fmt.Println("Indeks tidak valid!")
            return arr
        }
        return append(arr[:index], arr[index+1:]...)
    }

    func hitungRataRata(arr []int) float64 {
        total := 0
        for _, nilai := range arr {
            total += nilai
        }
        return float64(total) / float64(len(arr))
    }

    func hitungStandarDeviasi(arr []int) float64 {
        rata := hitungRataRata(arr)
        var total float64
        for _, nilai := range arr {
            total += math.Pow(float64(nilai)-rata, 2)
        }
        return math.Sqrt(total / float64(len(arr)))
    }

    func tampilkanFrekuensi(arr []int) {
        frekuensi := make(map[int]int)
        for _, nilai := range arr {
            frekuensi[nilai]++
        }
    }

```

```

    }

    keys := make([]int, 0, len(frekuensi))
    for k := range frekuensi {
        keys = append(keys, k)
    }
    sort.Ints(keys)

    fmt.Println("Frekuensi elemen dalam array:")
    for _, k := range keys {
        fmt.Printf("%d: %d kali\n", k, frekuensi[k])
    }
}

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

    arr := make([]int, n)
    fmt.Println("Masukkan elemen-elemen array:")
    for i := 0; i < n; i++ {
        fmt.Scan(&arr[i])
    }

    tampilkanArray(arr)
    tampilkanIndeksGanjil(arr)
    tampilkanIndeksGenap(arr)

    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)
    tampilkanKelipatanIndeks(arr, x)

    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&index)
    arr = hapusElemen(arr, index)
    tampilkanArray(arr)

    rataRata := hitungRataRata(arr)

```



```
fmt.Printf("Rata-rata: %.2f\n", rataRata)

standarDeviasi := hitungStandarDeviasi(arr)
fmt.Printf("Standar deviasi: %.2f\n", standarDeviasi)

tampilkanFrekuensi(arr)
}
```

- a. Menampilkan keseluruhan isi dari array

```
Masukkan jumlah elemen: 8
Masukkan elemen-elemen array:
1 2 3 4 5 6 7 8
Isi array: [1 2 3 4 5 6 7 8]
```

- b. Menampilkan elemen-elemen array dengan indeks ganjil

```
Elemen dengan indeks ganjil: 1 3 5 7
```

- c. Menampilkan elemen-elemen array dengan indeks genap

```
Elemen dengan indeks genap: 2 4 6 8
```

- d. Menampilkan elemen-elemen array dengan indeks kelipatan x

```
Masukkan nilai x: 2
Elemen dengan indeks kelipatan 2: 3 5 7
```

- e. Menghapus elemen array pada indeks tertentu, asumsi indeks yang dihapus selalu valid

```
Masukkan indeks yang ingin dihapus: 3
Isi array: [1 2 3 5 6 7 8]
```

- f. Menampilkan Rata-rata dari bilangan yang ada didalam array

```
Rata-rata: 4.57
```

- g. Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada didalam array

```
Standar deviasi: 2.44
```

- h. Menampilkan Frekuensi dari suatu bilangan tertentu didalam array

```
Frekuensi elemen dalam array:  
1: 1 kali  
2: 1 kali  
3: 1 kali  
5: 1 kali  
6: 1 kali  
7: 1 kali  
8: 1 kali
```

Deskripsi Program :

Program ini adalah aplikasi berbasis bahasa Go yang dirancang untuk mengelola dan menganalisis array bilangan bulat. Pengguna diminta untuk memasukkan jumlah elemen dan nilai-nilai array, setelah itu program menampilkan isi array, elemen dengan indeks ganjil dan genap, serta elemen yang berada pada indeks kelipatan tertentu yang ditentukan oleh pengguna. Program juga memungkinkan pengguna untuk menghapus elemen pada indeks yang diinginkan dan menghitung rata-rata serta standar deviasi dari elemen yang tersisa dalam array. Selain itu, program menghitung dan menampilkan frekuensi kemunculan setiap elemen dalam array. Dengan berbagai fungsi analisis ini, program memberikan alat yang komprehensif untuk manipulasi dan evaluasi data dalam bentuk array.

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 :

```
package main

import (
    "fmt"
)

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

    fmt.Print("Masukkan nama klub A: ")
    fmt.Scan(&klubA)
    fmt.Print("Masukkan nama klub B: ")
```

```

fmt.Scan(&klubB)

fmt.Println("Masukkan skor untuk pertandingan, masukkan skor
negatif untuk mengakhiri:")
for {
    fmt.Scan(&skorA, &skorB)

    if skorA < 0 || skorB < 0 {
        fmt.Println("Pertandingan selesai")
        break
    }

    hasilPertandingan = append(hasilPertandingan, [2]int{skorA,
skorB})
}

for _, hasil := range hasilPertandingan {
    skorA, skorB = hasil[0], hasil[1]

    if skorA > skorB {
        pemenang = append(pemenang, klubA)
    } else if skorB > skorA {
        pemenang = append(pemenang, klubB)
    } else {
        pemenang = append(pemenang, "Draw")
    }
}

fmt.Println("\nDaftar hasil pertandingan:")
for i, hasil := range hasilPertandingan {
    fmt.Printf("Pertandingan %d: %s %d - %d %s\n", i+1, klubA,
hasil[0], hasil[1], klubB)
}

fmt.Println("\nDaftar klub yang memenangkan pertandingan:")
for i, klub := range pemenang {
    if klub == "Draw" {
        fmt.Printf("Pertandingan %d: Draw\n", i+1)
    } else {

```

```
        fmt.Printf("Pertandingan %d: %s\n", i+1, klub)
    }
}
```

Screenshot Program :

```
Masukkan nama klub A: MU
Masukkan nama klub B: Inter
Masukkan skor untuk pertandingan, masukkan skor negatif untuk mengakhiri:
2 0
1 2
2 2
0 1
3 2
1 0
5 2
2 3
-1 2
Pertandingan selesai

Daftar hasil pertandingan:
Pertandingan 1: MU 2 - 0 Inter
Pertandingan 2: MU 1 - 2 Inter
Pertandingan 3: MU 2 - 2 Inter
Pertandingan 4: MU 0 - 1 Inter
Pertandingan 5: MU 3 - 2 Inter
Pertandingan 6: MU 1 - 0 Inter
Pertandingan 7: MU 5 - 2 Inter
Pertandingan 8: MU 2 - 3 Inter

Daftar klub yang memenangkan pertandingan:
Pertandingan 1: MU
Pertandingan 2: Inter
Pertandingan 3: Draw
Pertandingan 4: Inter
Pertandingan 5: MU
Pertandingan 6: MU
Pertandingan 7: MU
Pertandingan 8: Inter
```

Deskripsi Program :

Program ini adalah aplikasi sederhana dalam bahasa Go yang digunakan untuk mencatat dan menganalisis hasil pertandingan antara dua klub sepak bola. Pengguna diminta untuk memasukkan nama dua klub dan kemudian memasukkan skor pertandingan secara berulang hingga skor negatif dimasukkan, yang menandakan akhir dari input. Program menyimpan hasil setiap pertandingan dalam sebuah array dan menentukan pemenang berdasarkan skor yang dimasukkan; jika skor klub A lebih tinggi, klub A dinyatakan sebagai pemenang, jika skor klub B lebih tinggi, klub B yang menang, dan jika kedua skor sama, hasilnya dianggap "Draw". Setelah semua skor dicatat, program menampilkan daftar hasil pertandingan dan pemenang masing-masing pertandingan. Dengan demikian, program ini memberikan cara yang mudah untuk melacak hasil pertandingan dan menentukan pemenang antara dua klub.

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.

Source Code :

```
package main

import (
    "fmt"
    "bufio"
    "os"
)

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    reader := bufio.NewReader(os.Stdin)
    input, _ := reader.ReadString('.')
    for i, char := range input {
        if char == '.' {
            break
        }
        (*t)[i] = char
        *n++
    }
}

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

```

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

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

func main() {
    var tab tabel
    var m int

    fmt.Print("Teks: ")
    isiArray(&tab, &m)

    fmt.Print("Reverse Teks: ")
    balikanArray(&tab, m)
    cetakArray(tab, m)

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

```


Screenshot Program :

```
Teks: KATAK.  
Reverse Teks: KATAK  
Palindrom: true  
PS C:\Users\ACER\Downloads\Semes  
2311102029_Modul7\unguided4.go"  
Teks: KODOK.  
Reverse Teks: KODOK  
Palindrom: true  
PS C:\Users\ACER\Downloads\Semes  
2311102029_Modul7\unguided4.go"  
Teks: AMBADIL.  
Reverse Teks: LIDABMA  
Palindrom: false  
PS C:\Users\ACER\Downloads\Semes
```

Deskripsi Program :

Program ini adalah aplikasi dalam bahasa Go yang dirancang untuk menerima dan memproses input teks dari pengguna, dengan fokus pada pengujian palindrom dan pembalikan teks. Program menggunakan tipe data khusus tabel yang merupakan array dari rune dengan kapasitas maksimum 127 karakter. Pengguna diminta untuk memasukkan teks yang diakhiri dengan tanda titik ('.'), dan program akan menyimpan karakter-karakter tersebut dalam array. Setelah itu, program membalikkan urutan karakter dalam array dan mencetak hasilnya. Selain itu, program juga memeriksa apakah teks yang dimasukkan adalah palindrom—yaitu, teks yang sama jika dibaca dari depan maupun belakang—dan mencetak hasil pemeriksaan tersebut. Dengan demikian, program ini memberikan alat yang sederhana namun efektif untuk manipulasi string dan analisis palindrom.