

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**

MODUL VI

“STUCK & ARRAY”



Oleh:

Damara Galuh Pembayun

2311102110

IF 11 02

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

- Pemrogram dapat menciptakan tipe data baru dengan menggunakan tipe bentukan. Tipe bentukan terdiri dari dua jenis: Alias (Type) dan Struktur. Alias memungkinkan penggantian nama tipe data dengan nama yang lebih ringkas, sedangkan Struktur mengelompokkan beberapa data yang memiliki hubungan menjadi satu kesatuan. Dalam kebanyakan kasus, bahasa pemrograman memungkinkan pemrograman untuk mengubah nama tipe data menjadi nama baru yang lebih mudah diingat dan dikenal. Sebagai contoh, "integer" dapat diubah menjadi "bilangan". Caranya adalah dengan menggunakan istilah "tipe". Tipe bentukan terdiri dari dua jenis: Alias (Type) dan Struktur. Alias memungkinkan penggantian nama tipe data dengan nama yang lebih sederhana, sedangkan Struktur menggabungkan beberapa data yang memiliki hubungan menjadi satu kesatuan. Dalam kebanyakan kasus, bahasa pemrograman memungkinkan pemrograman untuk mengubah nama tipe data menjadi nama baru yang lebih mudah diingat dan dikenal. Sebagai contoh, "integer" dapat diubah menjadi "bilangan". Untuk melakukannya, gunakan kata kunci "type".
- Array adalah struktur data yang memiliki ukuran tetap selama program berjalan. Tipe data tertentu digunakan untuk mendeklarasikan elemen array, dan jumlah elemen termasuk dalam deklarasi variable. Beberapa jenis array termasuk Slice (Array Dinamik) dalam Go. Ukuran array juga dapat berubah. (Tidak digunakan dalam kelas Algoritma Pemrograman). Ini adalah peta tipe array lain, yang merupakan array dinamik. Di sini, indeks, atau kunci, tidak harus berbentuk integer. Tipe apa pun dapat menjadi sumber indeks. Struktur ini dikenal sebagai peta.

I. GUIDED

Guided 1

SourceCode

```
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

}

// Fungsi utama untuk mengelola dan mengurutkan data mahasiswa
berdasarkan nilai rata-rata

func main() {

    // Array untuk menampung data mahasiswa

    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},

    }

    // Menghitung rata-rata nilai tiap mahasiswa

    for i := range mahasiswa {

        hitungRataRata(&mahasiswa[i])

    }

    // Mengurutkan mahasiswa berdasarkan nilai rata-rata
    (descending)

    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",

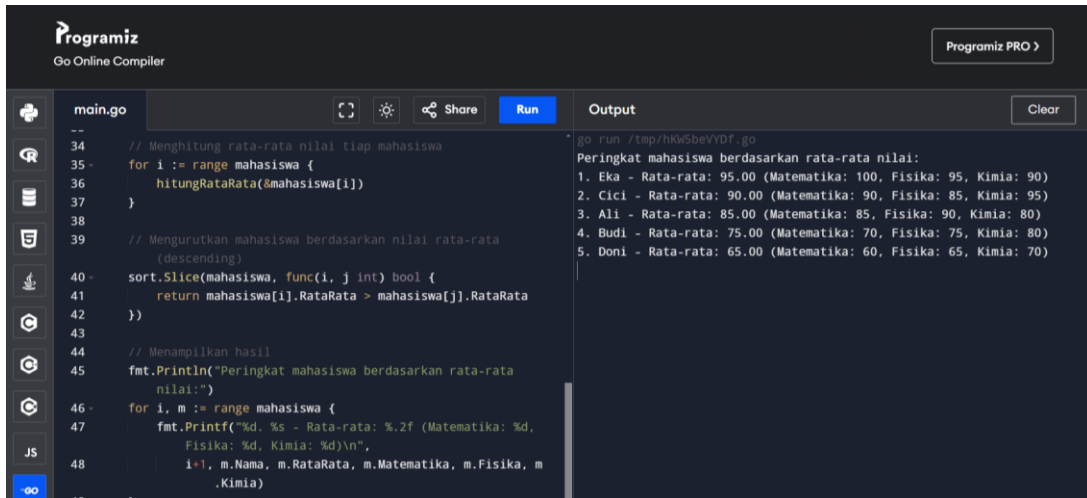
            i+1, m>Nama, m.RataRata, m.Matematika, m.Fisika,
            m.Kimia)

    }

}

```

Hasil ScreenShoot



The screenshot shows the Programiz Go Online Compiler interface. The code in the editor calculates the average of three subjects (Matematika, Fisika, Kimia) for five students and sorts them by average in descending order. The output displays the sorted list of students with their names, average scores, and individual subject scores.

```
main.go
--
34 // Menghitung rata-rata nilai tiap mahasiswa
35 for i := range mahasiswa {
36     hitungRataRata(&mahasiswa[i])
37 }
38
39 // Mengurutkan mahasiswa berdasarkan nilai rata-rata
   (descending)
40 sort.Slice(mahasiswa, func(i, j int) bool {
41     return mahasiswa[i].RataRata > mahasiswa[j].RataRata
42 })
43
44 // Menampilkan hasil
45 fmt.Println("Peringkat mahasiswa berdasarkan rata-rata
   nilai:")
46 for i, m := range mahasiswa {
47     fmt.Printf("%d. %s - Rata-rata: %.2f (Matematika: %d,
   Fisika: %d, Kimia: %d)\n",
48         i+1, m>Nama, m.RataRata, m.Matematika, m.Fisika, m
   .Kimia)
49 }
```

Output

```
go run /tmp/hKwSbeVYDf.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)
```

Guided 2

SourchCode

```
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 {

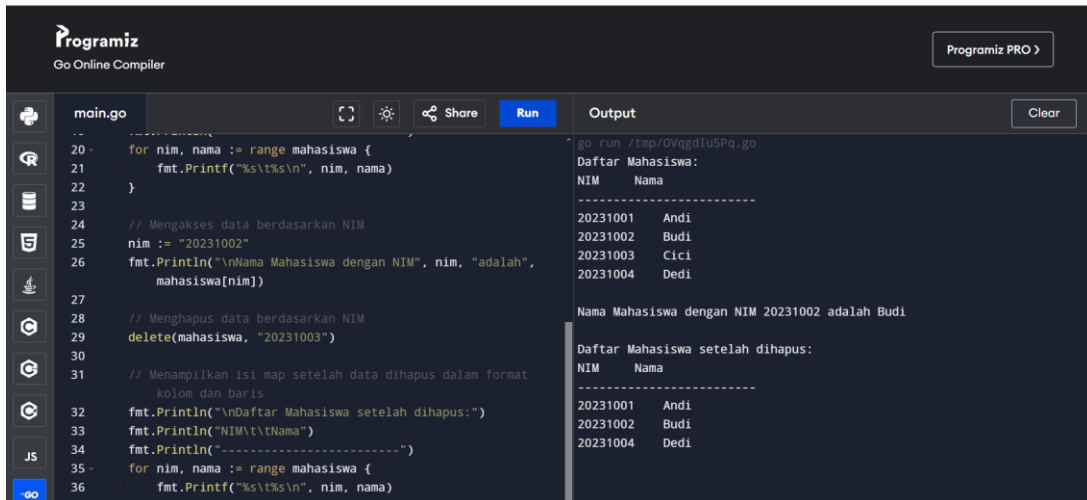
    fmt.Printf("%s\t%s\n", nim, nama)

}

}

```

Hasil ScreenShoot



The screenshot shows the Programiz Go Online Compiler interface. The editor on the left contains a Go program in `main.go` that defines a slice of student data, prints it, filters it by NIM, and prints the result. The output on the right shows the execution results, including the initial list of students, the filtered list for NIM 20231002, and the final list after deletion.

```
main.go
20 for nim, nama := range mahasiswa {
21     fmt.Printf("%s\t%s\n", nim, nama)
22 }
23
24 // Mengakses data berdasarkan NIM
25 nim := "20231002"
26 fmt.Println("\nNama Mahasiswa dengan NIM", nim, "adalah",
    mahasiswa[nim])
27
28 // Menghapus data berdasarkan NIM
29 delete(mahasiswa, "20231003")
30
31 // Menampilkan isi map setelah data dihapus dalam format
    kolom dan baris
32 fmt.Println("\nDaftar Mahasiswa setelah dihapus:")
33 fmt.Println("NIM\t\tNama")
34 fmt.Println("-----")
35 for nim, nama := range mahasiswa {
36     fmt.Printf("%s\t%s\n", nim, nama)
37 }
```

Output

```
go run /tmp/OVqgdIU5Pq.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
```

II. UNGUIDED

Unguided 1

SourchCode

```
package main

import (
    "fmt"
    "math"
)

// Titik represents a point in a 2D coordinate system
type Titik struct {
    x int
    y int
}

// Lingkaran represents a circle with a center and radius
type Lingkaran struct {
    pusat Titik
    radius int
}

// jarak calculates the distance between two points
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))
}

// didalam checks if a point is inside a circle
func didalam(c Lingkaran, p Titik) bool {
    return jarak(c.pusat, p) <= float64(c.radius)
}

func main() {
```

```

var t1, t2, p Titik
var r1, r2 int

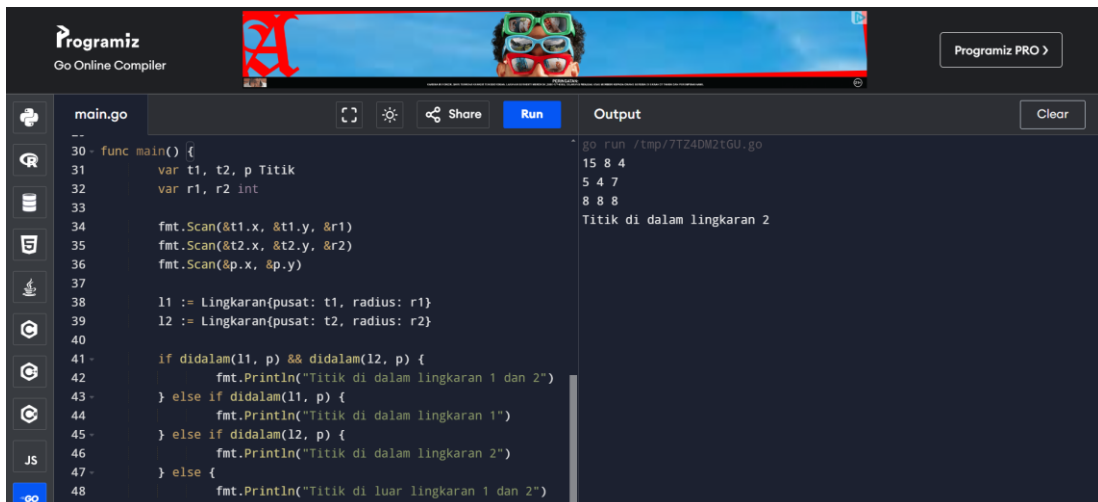
fmt.Scan(&t1.x, &t1.y, &r1)
fmt.Scan(&t2.x, &t2.y, &r2)
fmt.Scan(&p.x, &p.y)

l1 := Lingkaran{pusat: t1, radius: r1}
l2 := Lingkaran{pusat: t2, radius: r2}

if didalam(l1, p) && didalam(l2, p) {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if didalam(l1, p) {
    fmt.Println("Titik di dalam lingkaran 1")
} else if didalam(l2, p) {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

Hasil ScreenShoot



The screenshot shows the Programiz Go Online Compiler interface. The code editor on the left contains the Go code from the previous block. The output window on the right shows the results of running the program with the input values 15 8 4, 5 4 7, and 8 8 8. The output is:

```

go run /tmp/7TZ4DM2tGU.go
15 8 4
5 4 7
8 8 8
Titik di dalam lingkaran 2

```

Deskripsi

Program dalam bahasa Go ini berfungsi untuk mengetahui apakah suatu titik berada di dalam, di luar, atau tepat di tepi dua lingkaran tertentu. Setiap lingkaran memiliki titik pusat dan jari-jari. Program meminta pengguna untuk memberikan koordinat titik pusat dan radius dari dua lingkaran, serta koordinat titik yang akan diperiksa. Kemudian, menggunakan rumus jarak geometris untuk menghitung jarak antara titik dan pusat lingkaran, program ini menentukan apakah titik tersebut berada di dalam, di tepi, atau di luar masing-masing lingkaran.

Unguided 2

SourceCode

```
package main

import (
    "fmt"
    "math"
)

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

    // Membuat array dengan ukuran N
    arr := make([]int, N)

    // Meminta input nilai untuk setiap elemen array
    fmt.Println("Masukkan nilai-nilai array:")
    for i := 0; i < N; i++ {
        fmt.Scan(&arr[i])
    }

    // Menampilkan seluruh isi array
    fmt.Println("Seluruh isi array:")
    for _, v := range arr {
        fmt.Print(v, " ")
    }
    fmt.Println()

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

    // Menampilkan elemen dengan indeks genap
    fmt.Println("Elemen dengan indeks genap:")
    for i := 0; i < N; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    // Menampilkan elemen dengan indeks kelipatan x
    var x int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)
    fmt.Println("Elemen dengan indeks kelipatan", x)
    for i := 0; i < N; i++ {
        if i%x == 0 {
            fmt.Print(arr[i], " ")
        }
    }
}
```

```

    }
    fmt.Println()

    // Menghapus elemen pada indeks tertentu
    var index int
    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&index)
    arr = append(arr[:index], arr[index+1:]...)
    fmt.Println("Array setelah penghapusan:")
    for _, v := range arr {
        fmt.Print(v, " ")
    }
    fmt.Println()

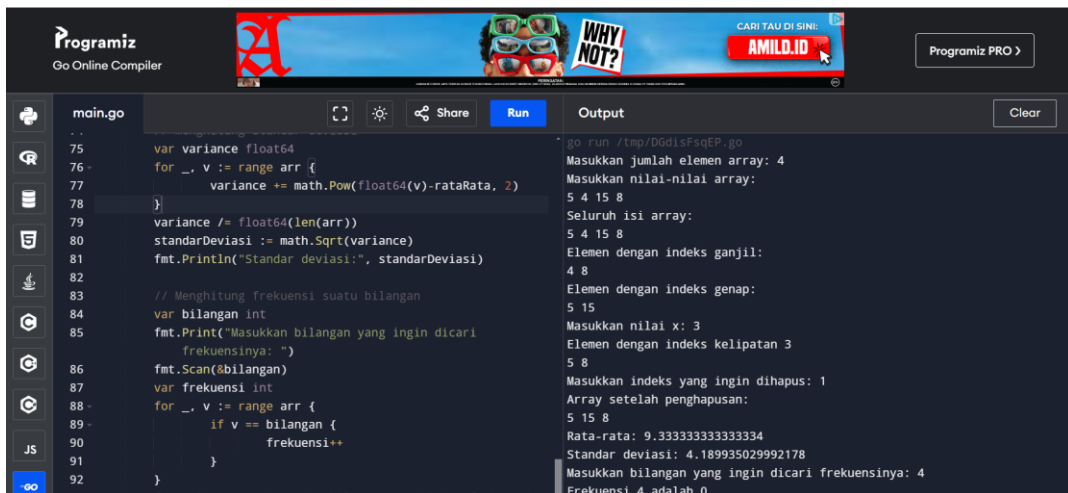
    // Menghitung rata-rata
    var sum int
    for _, v := range arr {
        sum += v
    }
    rataRata := float64(sum) / float64(len(arr))
    fmt.Println("Rata-rata:", rataRata)

    // Menghitung standar deviasi
    var variance float64
    for _, v := range arr {
        variance += math.Pow(float64(v)-rataRata, 2)
    }
    variance /= float64(len(arr))
    standarDeviasi := math.Sqrt(variance)
    fmt.Println("Standar deviasi:", standarDeviasi)

    // Menghitung frekuensi suatu bilangan
    var bilangan int
    fmt.Print("Masukkan bilangan yang ingin dicari
frekuensinya: ")
    fmt.Scan(&bilangan)
    var frekuensi int
    for _, v := range arr {
        if v == bilangan {
            frekuensi++
        }
    }
    fmt.Println("Frekuensi", bilangan, "adalah", frekuensi)
}

```

Hasil ScreenShoot



Deskripsi

Program ini dibuat dalam bahasa pemrograman Go dan dimaksudkan untuk melakukan berbagai operasi pada array yang dimasukkan oleh user. Program ini meminta user untuk memasukkan sejumlah elemen, kemudian melakukan berbagai operasi, seperti menampilkan elemen dengan indeks, menghapus elemen, menghitung rata-rata, standar deviasi, dan menghitung frekuensi bilangan dalam array.

Unguided 3

SourceCode

```

package main

import (
    "fmt"
)

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

    fmt.Print("Klub A: ")
    fmt.Scan(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scan(&klubB)

    var pemenang []string

    for {
        fmt.Printf("Pertandingan: ")
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            break
        }

        if skorA > skorB {

```

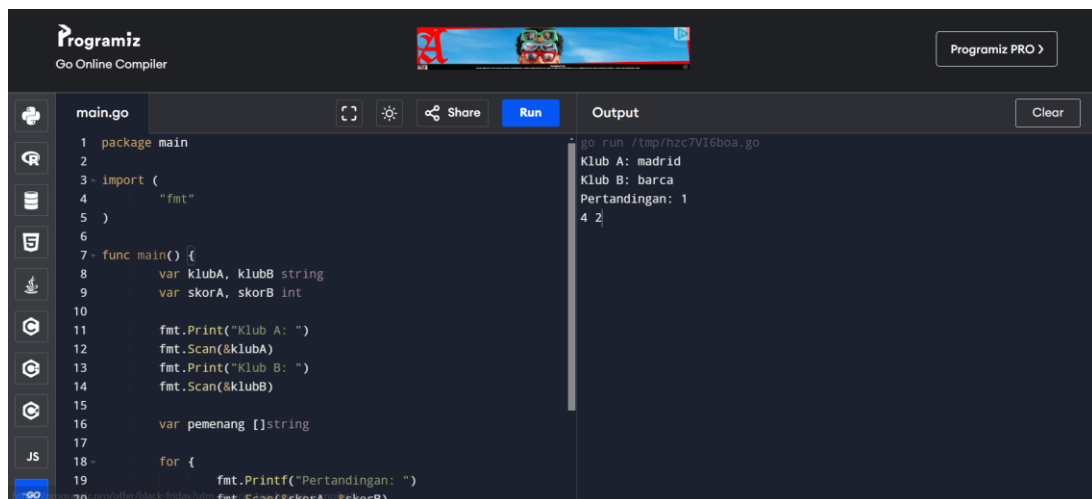
```

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

fmt.Println("Hasil pertandingan:")
for i, p := range pemenang {
    fmt.Printf("Hasil %d: %s\n", i+1, p)
}
}

```

Hasil ScreenShoot



Deskripsi

Ini adalah game Go yang memungkinkan Anda mencatat dan menampilkan hasil pertandingan antara dua klub sepak bola. Setelah pengguna memasukkan nama klub, program akan meminta skor dari setiap pertandingan sampai ada skor negatif, yang akan menghentikan proses input. Setelah skor negatif dimasukkan, program akan menentukan pemenang dari setiap pertandingan berdasarkan skor tersebut. Di akhir program, hasil dari setiap pertandingan akan disimpan dalam sebuah slice dan ditampilkan.

Unguided 4

SourchCode

```

package main

import "fmt"

const NMAX int = 127
type tabel [NMAX] rune

```

```

func isiArray(t tabel, n int) {
    // ... (asumsikan sudah diimplementasikan)
}

func cetakArray(t tabel, n int) {
    // ... (asumsikan sudah diimplementasikan)
}

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

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

func main() {
    var tab tabel
    var m int

    // Isi array tab dengan memanggil prosedur isiArray
    isiArray(&tab, &m)


    // Balikan isi array tab dengan memanggil balikanArray
    balikanArray(tab, m)

    // Cetak isi array tab
    cetakArray(tab, m)

    // Periksa apakah tab adalah palindrom
    if palindrom(tab, m) {
        fmt.Println("Palindrom")
    } else {
        fmt.Println("Bukan Palindrom")
    }
}

```

Hasil ScreenShoot



```
main.go
32
33 func main() {
34     var tab tabel
35     var m int
36
37     // Isi array tab dengan memanggil prosedur isiArray
38     isiArray(&tab, &m)
39
40     // Balikan isi array tab dengan memanggil balikanArray
41     balikanArray(tab, m)
42
43     // Cetak isi array tab
44     cetakArray(tab, m)
45
46     // Periksa apakah tab adalah palindrom
47     if palindrom(tab, m) {
48         fmt.Println("Palindrom")
49     } else {
50         fmt.Println("Bukan Palindrom")
51     }
52 }
```

Output

```
go run /tmp/MyqHfEzX5a.go
kataPalindrom
```

Deskripsi

Dalam bahasa Go, program ini berfungsi untuk mengolah teks yang dimasukkan oleh pengguna. Program ini memungkinkan pengguna memasukkan teks dan kemudian menyimpannya dalam array bertipe rune yang dapat menangani karakter Unicode. Program dapat melakukan beberapa hal, seperti mengisi array dengan input pengguna, mencetak isi array, membalikkan isi array, dan memeriksa apakah teks tersebut adalah palindrom, yaitu teks yang dibaca sama dari depan dan belakang.