**LAPORAN PRAKTIKUM   
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII**

**STRUCT & ARRAY**

**Sebuah gambar berisi logo, simbol, Grafis, Font

Deskripsi dibuat secara otomatis**

**Disusun Oleh :**

**Fariz Ilham / 2311102275**

**IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

1. **DASAR TEORI**

* Array merupakan struktur data yang menyimpan elemen-elemen yang memiliki tipe data yang sama dalam urutan yang terorganisir. Setiap elemen array diakses menggunakan indeks, yang memungkinkan pengambilan dan modifikasi elemen dengan cepat. Array sering digunakan dalam operasi seperti pencarian, pengurutan, dan manipulasi data lainnya. Kelebihan dari array adalah kemampuannya dalam menyediakan akses data yang efisien secara langsung (random access), tetapi ukuran array yang statis dapat menjadi keterbatasan, terutama saat data yang disimpan bersifat dinamis.
* Struct adalah kumpulan data yang dapat memiliki tipe data yang berbeda dalam satu entitas logis. Struct memungkinkan pengelompokan data yang beragam ke dalam satu unit, sehingga mempermudah pengelolaan data yang lebih kompleks, misalnya menyimpan data atribut suatu entitas seperti mahasiswa (nama, umur, nilai, dll). Struct sering digunakan dalam kombinasi dengan array untuk membuat array dari struct, memungkinkan penyimpanan data terstruktur dalam jumlah besar sekaligus menjaga keterkaitan antar data.

1. **GUIDED**

**Soal Studi Case**

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.

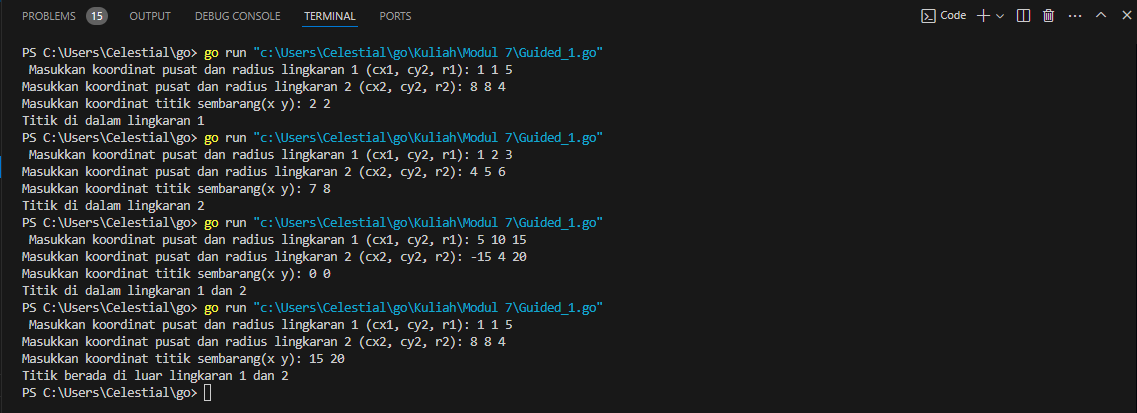
**Masukan** terdiri dari beberapa tiga baris. Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingkaran 2, sedangkan baris ketiga adalah koordinat titik sembarang. Asumsi sumbu x dan y dari semua titik dan juga radius direpresentasikan dengan bilangan bulat.

**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".

**Sourcecode**

|  |
| --- |
| package main  import (  "fmt"  "math"  )  // menyimpan titik dengan koordinat (x,y)  type titik struct {  x int  y int  }  // menyimpan lingkaran dengan pusat dan radius  type lingkaran struct {  pusat titik  radius int  }  // fungsi untuk menghitung jarak antara dua titik  func hitungjarak(a, b titik) float64 {  return math.Sqrt(float64((a.x-b.x)\*(a.x-b.x) + (a.y-b.y)\*(a.y-b.y)))  }  // fungsi untuk memeriksa apakah titik berada di dalam lingkaran  func titikDiDalamLingkaran(t titik, l lingkaran) bool {  jarak := hitungjarak(t, l.pusat)  return jarak <= float64(l.radius)  }  func main() {  // input untuk lingkaran 1  var cx1, cy1, r1 int  fmt.Print(" Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): ")  fmt.Scanln(&cx1, &cy1, &r1)  lingkaran1 := lingkaran{pusat: titik{x: cx1, y: cy1}, radius: r1}  //input untuk lingkaran 2  var cx2, cy2, r2 int  fmt.Print("Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): ")  fmt.Scanln(&cx2, &cy2, &r2)  lingkaran2 := lingkaran{pusat: titik{x: cx2, y: cy2}, radius: r2}  //input untuk titik sembarang  var x, y int  fmt.Print("Masukkan koordinat titik sembarang(x y): ")  fmt.Scanln(&x, &y)  titik := titik{x: x, y: y}  // pengecekan posisi titik  diDalam1 := titikDiDalamLingkaran(titik, lingkaran1)  diDalam2 := titikDiDalamLingkaran(titik, lingkaran2)  //menampilkan hasil sesuai kondisi  if diDalam1 && diDalam2 {  fmt.Println("Titik di dalam lingkaran 1 dan 2")  } else if diDalam1 {  fmt.Println("Titik di dalam lingkaran 1")  } else if diDalam2 {  fmt.Println("Titik di dalam lingkaran 2")  } else {  fmt.Println("Titik berada di luar lingkaran 1 dan 2")  }  } |

**Screenshoot Output**



**Deskripsi Program**

Program ini menentukan apakah suatu titik berada di dalam lingkaran 1, lingkaran 2, atau di luar keduanya. Lingkaran didefinisikan dengan pusat dan radius, dan pengguna diminta memasukkan koordinat serta radius untuk kedua lingkaran dan satu titik sembarang. Proses pemeriksaan dilakukan dengan menghitung jarak antara titik dan pusat lingkaran, kemudian membandingkan hasilnya dengan radius lingkaran.

* Algoritma:

1. Input (cx1,cy1,r1) sebagai koordinat pusat dan radius untuk lingkaran 1.

2. Input (cx2,cy2,r2) sebagai koordinat pusat dan radius untuk lingkaran 2.

3. Input (x, y) sebagai koordinat titik sembarang.

4. Hitung jarak antara titik sembarang dengan pusat lingkaran menggunakan fungsi `hitungjarak`:

- Rumus: Jarak = akar{(x1 - x2)^2 + (y1 - y2)^2}

5. Periksa apakah titik berada di dalam lingkaran dengan fungsi `titikDiDalamLingkaran`:

- Jika jarak antara titik dan pusat ≤ radius lingkaran, maka titik berada di dalam lingkaran.

- Jika titik di dalam kedua lingkaran, cetak "Titik di dalam lingkaran 1 dan 2".

- Jika titik di dalam salah satu lingkaran, cetak nama lingkaran tersebut.

- Jika titik di luar kedua lingkaran, cetak "Titik berada di luar lingkaran 1 dan 2".

* Cara Kerja:

1. Program dimulai dengan meminta pengguna memasukkan koordinat pusat dan radius dari dua lingkaran, serta koordinat titik sembarang.

2. Fungsi `hitungjarak` menghitung jarak antara titik sembarang dan pusat lingkaran.

3. Fungsi `titikDiDalamLingkaran` menentukan apakah jarak tersebut lebih kecil atau sama dengan radius lingkaran.

4. Berdasarkan hasil pemeriksaan, program mencetak posisi lingkaran sesuai dengan kondisi yang terpenuhi.

1. **UNGUIDED**

**Soal Studi Case**

1. 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 indek 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.

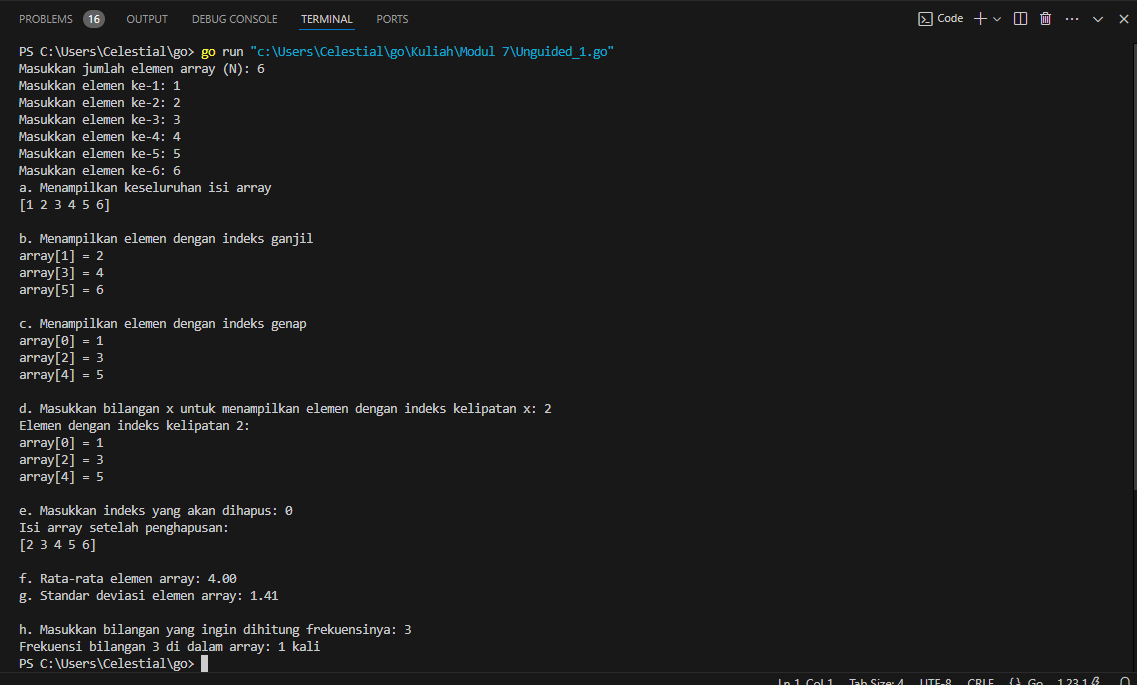
g. Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut.

h. Menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi tersebut.

**Sourcecode**

|  |
| --- |
| package main  import (  "fmt"  "math"  )  func mean(array []int) float64 {  sum := 0  for \_, v := range array {  sum += v  }  return float64(sum) / float64(len(array))  }  func stdDev(array []int, avg float64) float64 {  var sum float64  for \_, v := range array {  sum += math.Pow(float64(v)-avg, 2)  }  return math.Sqrt(sum / float64(len(array)))  }  func countFrequency(array []int, target int) int {  count := 0  for \_, v := range array {  if v == target {  count++  }  }  return count  }  func removeIndex(array []int, index int) []int {  return append(array[:index], array[index+1:]...)  }  func main() {  var n, x, target, index int  fmt.Print("Masukkan jumlah elemen array (N): ")  fmt.Scan(&n)  array := make([]int, n)  for i := 0; i < n; i++ {  fmt.Printf("Masukkan elemen ke-%d: ", i+1)  fmt.Scan(&array[i])  }  fmt.Println("a. Menampilkan keseluruhan isi array")  fmt.Println(array)  fmt.Println("\nb. Menampilkan elemen dengan indeks ganjil")  for i := 1; i < n; i += 2 {  fmt.Printf("array[%d] = %d\n", i, array[i])  }  fmt.Println("\nc. Menampilkan elemen dengan indeks genap")  for i := 0; i < n; i += 2 {  fmt.Printf("array[%d] = %d\n", i, array[i])  }  fmt.Print("\nd. Masukkan bilangan x untuk menampilkan elemen dengan indeks kelipatan x: ")  fmt.Scan(&x)  fmt.Printf("Elemen dengan indeks kelipatan %d:\n", x)  for i := 0; i < n; i++ {  if i%x == 0 {  fmt.Printf("array[%d] = %d\n", i, array[i])  }  }  fmt.Print("\ne. Masukkan indeks yang akan dihapus: ")  fmt.Scan(&index)  array = removeIndex(array, index)  fmt.Println("Isi array setelah penghapusan:")  fmt.Println(array)  avg := mean(array)  fmt.Printf("\nf. Rata-rata elemen array: %.2f\n", avg)  fmt.Printf("g. Standar deviasi elemen array: %.2f\n", stdDev(array, avg))  fmt.Print("\nh. Masukkan bilangan yang ingin dihitung frekuensinya: ")  fmt.Scan(&target)  fmt.Printf("Frekuensi bilangan %d di dalam array: %d kali\n", target, countFrequency(array, target))  } |

**Screenshoot Output**



**Deskripsi Program**

Program ini menerima input array dari pengguna dan memberikan beberapa operasi manipulasi dan analisis pada array, seperti menampilkan elemen-elemen dengan indeks tertentu, menghitung rata-rata, standar deviasi, menghapus indeks tertentu, serta menghitung frekuensi kemunculan suatu bilangan dalam array.

* Algoritma:

1. Input (n) dari pengguna untuk menentukan jumlah elemen array.

2. Input elemen-elemen array.

3. Menampilkan seluruh elemen array.

4. Menampilkan elemen dengan indeks ganjil.

5. Menampilkan elemen dengan indeks genap.

6. Input (x) dari pengguna, lalu menampilkan elemen-elemen array dengan indeks kelipatan (x).

7. Input (index) dari pengguna untuk menentukan elemen yang akan dihapus dari array.

8. Menghitung rata-rata elemen array.

9. Menghitung standar deviasi elemen array.

10. Input (target) dari pengguna untuk menghitung frekuensi kemunculan (target) di dalam array.

* Cara Kerja:

1. Program dimulai dengan meminta pengguna memasukkan jumlah elemen pada array (n).

2. Program meminta pengguna memasukkan elemen-elemen array satu per satu.

3. Menampilkan seluruh isi array.

4. Menggunakan loop untuk menampilkan elemen-elemen pada indeks ganjil (indeks mulai dari 1) dan indeks genap (indeks mulai dari 0).

5. Meminta input (x) dari pengguna dan menampilkan elemen-elemen yang indeksnya merupakan kelipatan (x).

6. Meminta input (index) dari pengguna untuk menentukan elemen array yang akan dihapus, lalu elemen pada indeks tersebut dihapus dari array menggunakan fungsi `removeIndex`.

7. Menggunakan fungsi `mean` untuk menghitung rata-rata elemen-elemen dalam array.

8. Menggunakan fungsi `stdDev` untuk menghitung standar deviasi elemen-elemen dalam array berdasarkan rata-rata.

9. Meminta input (target) dari pengguna dan menghitung frekuensi kemunculan (target) dalam array menggunakan fungsi `countFrequency`.

1. 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.

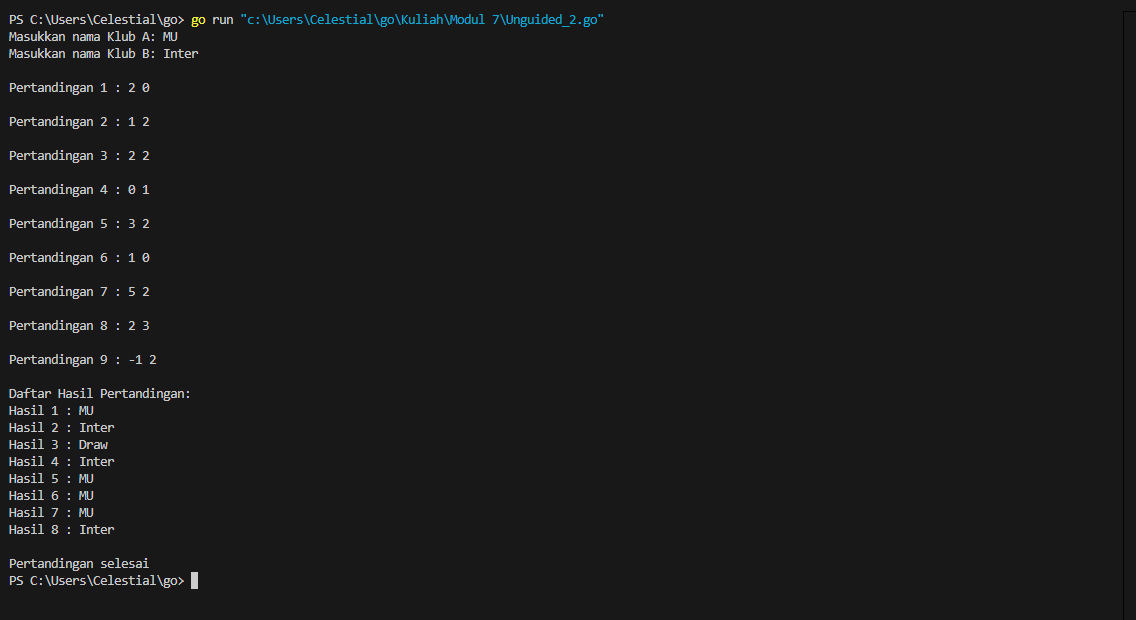
Pertama-tama program meminta masukan nama-nama klub yang bertanding, kemudian program meminta masukan skor hasil pertandingan kedua klub tersebut. Yang disimpan dalam array adalah nama-nama klub yang menang saja.

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{}  pertandingan := 1  fmt.Print("Masukkan nama Klub A: ")  fmt.Scan(&klubA)  fmt.Print("Masukkan nama Klub B: ")  fmt.Scan(&klubB)  for {  fmt.Printf("\nPertandingan %d : ", pertandingan)  fmt.Scan(&skorA, &skorB)  if skorA < 0 || skorB < 0 {  break  }  if skorA > skorB {  klubMenang = append(klubMenang, klubA)  } else if skorB > skorA {  klubMenang = append(klubMenang, klubB)  } else {  klubMenang = append(klubMenang, "Draw")  }  pertandingan++  }  fmt.Println("\nDaftar Hasil Pertandingan:")  for i, pemenang := range klubMenang {  fmt.Printf("Hasil %d : %s\n", i+1, pemenang)  }  fmt.Println("\nPertandingan selesai")  } |

**Screenshoot Output**



**Deskripsi Program**

Program ini menerima input skor pertandingan antara dua klub dan menyimpan hasil dari setiap pertandingan dalam sebuah array. Pertandingan berakhir jika pengguna memasukkan skor negatif untuk salah satu klub. Program kemudian menampilkan daftar hasil semua pertandingan, menunjukkan pemenang setiap pertandingan atau hasil imbang ("Draw") jika skor kedua klub sama.

* Algoritma

1. Input (klubA) dan (klubB) sebagai nama kedua klub yang akan bertanding.

2. Inisialisasi variabel `pertandingan` sebagai penghitung pertandingan dan array `klubMenang` untuk menyimpan hasil pertandingan.

3. Lakukan iterasi hingga skor salah satu klub negatif:

- Input (skorA) dan (skorB) dari pengguna sebagai skor masing-masing klub dalam satu pertandingan.

- Jika (skorA) atau (skorB) negatif, hentikan proses input skor.

- Bandingkan (skorA) dengan (skorB):

- Jika (skorA) > (skorB), tambahkan `klubA` ke array `klubMenang`.

- Jika (skorB) > (skorA), tambahkan `klubB` ke array `klubMenang`.

- Jika (skorA) = (skorB), tambahkan "Draw" ke array `klubMenang`.

- Tambahkan 1 ke penghitung `pertandingan`.

4. Tampilkan hasil semua pertandingan.

* Cara Kerja:

1. Program dimulai dengan meminta pengguna memasukkan nama Klub A dan Klub B.

2. Iterasi dilakukan selama skor yang dimasukkan oleh pengguna bukan nilai negatif. Skor setiap pertandingan dibandingkan untuk menentukan pemenang:

- Jika skor Klub A lebih tinggi dari Klub B, nama Klub A dimasukkan ke dalam array hasil.

- Jika skor Klub B lebih tinggi dari Klub A, nama Klub B dimasukkan ke dalam array hasil.

- Jika skor sama, "Draw" dimasukkan ke dalam array hasil.

3. Proses berhenti jika skor yang dimasukkan negatif untuk salah satu klub.

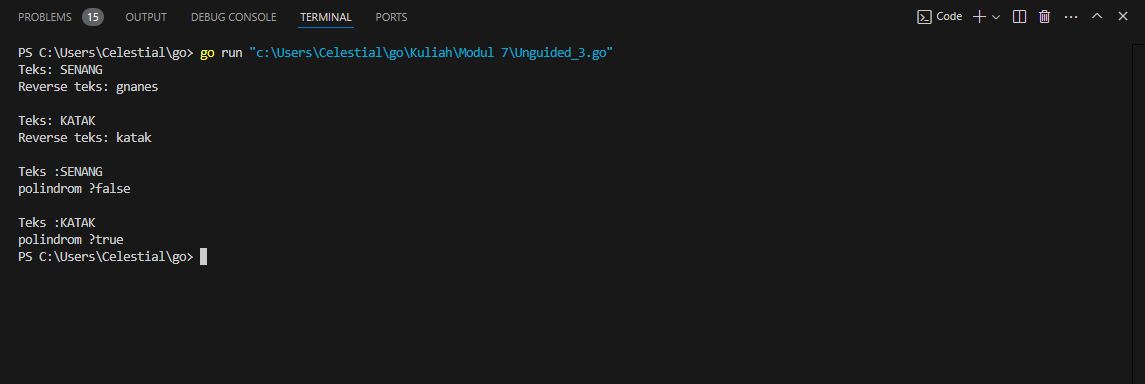
4. Program kemudian mencetak daftar hasil pertandingan, termasuk pemenang atau hasil imbang ("Draw") untuk setiap pertandingan.

1. 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"  "strings"  )  func balikArray(arr []rune) []rune {  n := len(arr)  for i := 0; i < n/2; i++ {  arr[i], arr[n-1-i] = arr[n-1-i], arr[i]  }  return arr  }  func cekPalindrom(arr []rune) bool {  n := len(arr)  for i := 0; i < n/2; i++ {  if arr[i] != arr[n-1-i] {  return false  }  }  return true  }  func main() {  var input1, input2 string  fmt.Print("Teks: ")  fmt.Scanln(&input1)  txtLower1 := strings.ToLower(input1)  array1 := []rune(txtLower1)  balik1 := balikArray(array1)  fmt.Printf("Reverse teks: %s\n", strings.ToLower(string(balik1)))  fmt.Println()  fmt.Print("Teks: ")  fmt.Scanln(&input2)  txtLower2 := strings.ToLower(input2)  array2 := []rune(txtLower2)  balik2 := balikArray(array2)  fmt.Printf("Reverse teks: %s\n", strings.ToLower(string(balik2)))  fmt.Println()  isPalindrom1 := cekPalindrom(array1)  fmt.Printf("Teks :%s\n", input1)  fmt.Printf("polindrom ?%t\n", isPalindrom1)  fmt.Println()  isPalindrom2 := cekPalindrom(array2)  fmt.Printf("Teks :%s\n", input2)  fmt.Printf("polindrom ?%t\n", isPalindrom2)  } |

**Screenshoot Output**



**]**

**Deskripsi Program**

Program ini menerima dua input teks dari pengguna, kemudian memeriksa apakah teks tersebut merupakan sebuah palindrome (kata yang terbaca sama dari depan dan belakang). Program juga membalikkan teks yang dimasukkan dan menampilkan hasilnya.

* Algoritma:

1. Input teks pertama dan kedua dari pengguna.

2. Mengkonversi kedua teks menjadi huruf kecil menggunakan `strings.ToLower`.

3. Mengubah kedua teks yang sudah dikonversi menjadi array `rune` untuk mempermudah manipulasi karakter.

4. Membalikkan teks yang telah diubah menjadi array `rune` menggunakan fungsi `balikArray`.

5. Menampilkan hasil teks yang dibalik.

6. Memeriksa apakah teks yang dimasukkan adalah palindrome menggunakan fungsi `cekPalindrom`:

- Jika karakter pertama sama dengan karakter terakhir, teruskan pengecekan.

- Jika ditemukan ketidaksamaan karakter, hasilkan `false`.

1. Menampilkan hasil apakah teks pertama dan kedua merupakan palindrome atau tidak.

* Cara Kerja:

1. Program dimulai dengan meminta pengguna memasukkan dua teks secara berurutan.

2. Setiap teks yang dimasukkan kemudian diubah menjadi huruf kecil dengan fungsi `strings.ToLower`.

3. Teks yang sudah diubah menjadi huruf kecil selanjutnya dikonversi menjadi array `rune` untuk memudahkan manipulasi karakter dalam bentuk array.

4. Fungsi `balikArray` membalikkan urutan karakter dalam array `rune`.

5. Program mencetak teks yang telah dibalik dengan menggunakan `strings.ToLower` untuk memastikan output konsisten dalam huruf kecil.

6. Fungsi `cekPalindrom` digunakan untuk memeriksa apakah teks yang dimasukkan adalah palindrome:

- Jika teks sama ketika dibaca dari depan dan belakang, hasilnya adalah `true`, jika tidak maka `false`.

7. Hasil pemeriksaan palindrome untuk kedua teks ditampilkan.