

**LAPORAN PRAKTIKUM**  
**ALGORITMA PEMROGRAMAN 2**  
**MODUL 7**  
**STRUCK & ARRAY**



Oleh :

Dimas Bagus Firmansyah

2311102002

S1 IF 11 02

**S1 TEKNIK INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

# I. DASAR TEORI

## 7.1 Tipe Bentuk

Tipe bentuk memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman.

### 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 dirubah 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> |                               |
| 3 |                               | type <nama alias> <tipe data> |
| 4 | algoritma                     | func main(){                  |
| 5 | ...                           | ...                           |
| 6 |                               | }                             |
| 7 |                               |                               |

### 2) Struct atau Record

Structure memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari structure 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 struktur 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 struktur tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi struktur tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

## 7.2 Array

Array mempunyai ukuran (jumlah elemen) yang tetap (statis) selama eksekusi program, sehingga jumlah elemen array menjadi bagian dari deklarasi variable 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 | )  |

### Slice Array

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

### 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.

## II. GUIDED

### GUIDED 1

```
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)
    }
}
```

## SCREENSHOOT PROGRAM

```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7> go run "c:\KULIAH
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:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7>
```

## DESKRIPSI PROGRAM

Program ini mengelola data mahasiswa dengan nilai untuk tiga mata pelajaran: Matematika, Fisika, dan Kimia. Setiap mahasiswa punya nilai rata-rata yang dihitung dari ketiga mata pelajaran tersebut menggunakan fungsi `hitungRataRata`. Setelah rata-rata dihitung, program mengurutkan mahasiswa dari yang rata-ratanya paling tinggi hingga paling rendah. Hasil akhirnya adalah daftar peringkat mahasiswa berdasarkan rata-rata nilai mereka, lengkap dengan nilai setiap mata pelajaran.

## UNGUIDED 2

```
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\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)
}
}

```

## SCREENSHOOT PROGRAM

```

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:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7>

```

## DESKRIPSI PROGRAM

Program ini membuat daftar mahasiswa menggunakan map dengan NIM sebagai kunci dan nama mahasiswa sebagai nilai. Awalnya, ada beberapa data mahasiswa yang sudah dimasukkan, dan program juga menambahkan satu data mahasiswa baru. Semua data mahasiswa ditampilkan dalam bentuk tabel. Program ini juga bisa menampilkan nama mahasiswa berdasarkan NIM yang dicari, lalu menghapus data mahasiswa tertentu dengan menggunakan NIM sebagai acuan. Setelah data dihapus, daftar mahasiswa ditampilkan lagi untuk menunjukkan hasil penghapusannya.

### III. UNGUIDED

#### UNGUIDED 1

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    pusat Titik
    radius int
}

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

// Fungsi untuk menentukan apakah titik berada di dalam lingkaran
func dalamLingkaran(t Titik, l Lingkaran) bool {
    return jarak(t, l.pusat) <= float64(l.radius)
}

func main() {
    // Input data lingkaran 1
    var cx1, cy1, r1 int
    fmt.Scan(&cx1, &cy1, &r1)
    lingkaran1 := Lingkaran{Titik{cx1, cy1}, r1}

    // Input data lingkaran 2
    var cx2, cy2, r2 int
    fmt.Scan(&cx2, &cy2, &r2)
    lingkaran2 := Lingkaran{Titik{cx2, cy2}, r2}

    // Input data titik sembarang
    var x, y int
    fmt.Scan(&x, &y)
    titik := Titik{x, y}

    // Cek posisi titik terhadap kedua lingkaran
    dalamL1 := dalamLingkaran(titik, lingkaran1)
    dalamL2 := dalamLingkaran(titik, lingkaran2)

    // Output berdasarkan posisi titik
    if dalamL1 && dalamL2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dalamL1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if dalamL2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

```
}  
}
```

## SCREENSHOOT PROGRAM

```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7> go run "c:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7\unguided1.go"  
1 1 5  
8 8 4  
2 2  
Titik di dalam lingkaran 1  
PS C:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7> █
```

## DESKRIPSI PROGRAM

Program ini bertujuan untuk menentukan posisi sebuah titik terhadap dua lingkaran berdasarkan data yang dimasukkan. Pertama, kita memasukkan koordinat pusat dan radius untuk kedua lingkaran, serta koordinat untuk titik yang ingin kita cek. Program lalu menghitung jarak titik tersebut dari pusat masing-masing lingkaran. Dengan fungsi dalamLingkaran, program memeriksa apakah titik itu berada di dalam lingkaran atau tidak (jaraknya lebih kecil atau sama dengan radius lingkaran). Terakhir, program akan menampilkan apakah titik tersebut berada di dalam satu lingkaran, kedua lingkaran, atau di luar keduanya.

## UNGUIDED 2

```
package main  
  
import (  
    "fmt"  
    "math"  
)  
  
// Fungsi untuk menghapus elemen pada indeks tertentu  
func hapusElemen(arr []int, index int) []int {  
    return append(arr[:index], arr[index+1:]...)  
}  
  
// Fungsi untuk menghitung rata-rata  
func rataRata(arr []int) float64 {  
    sum := 0  
    for _, val := range arr {  
        sum += val  
    }  
    return float64(sum) / float64(len(arr))  
}  
  
// Fungsi untuk menghitung standar deviasi  
func standarDeviasi(arr []int) float64 {  
    mean := rataRata(arr)  
    var variance float64  
    for _, val := range arr {  
        variance += math.Pow(float64(val)-mean, 2)  
    }  
    return math.Sqrt(variance / float64(len(arr)))  
}  
  
// Fungsi untuk menghitung frekuensi elemen tertentu dalam array  
func frekuensi(arr []int, x int) int {  
    count := 0
```



```

        for _, val := range arr {
            if val == x {
                count++
            }
        }
        return count
    }

func main() {
    // Memasukkan jumlah elemen dan isi dari array
    var n int
    fmt.Print("Masukkan jumlah elemen array: ")
    fmt.Scan(&n)

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

    // a. Menampilkan keseluruhan isi dari array
    fmt.Println("Keseluruhan isi array:", arr)

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

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

    // d. Menampilkan elemen array dengan indeks kelipatan bilangan x
    var x int
    fmt.Print("Masukkan nilai x untuk indeks kelipatan: ")
    fmt.Scan(&x)

    fmt.Print("Elemen pada indeks kelipatan ", x, ": ")
    for i := x; i < len(arr); i += x {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    // e. Menghapus elemen array pada indeks tertentu
    var index int
    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&index)
    if index >= 0 && index < len(arr) {
        arr = hapusElemen(arr, index)
        fmt.Println("Isi array setelah penghapusan:", arr)
    } else {
        fmt.Println("Indeks tidak valid!")
    }

    // f. Menampilkan rata-rata dari bilangan yang ada di dalam array

```

```

fmt.Printf("Rata-rata bilangan di array: %.2f\n", rataRata(arr))

// g. Menampilkan standar deviasi dari bilangan yang ada di dalam array
fmt.Printf("Standar deviasi bilangan di array: %.2f\n", standarDeviasi(arr))

// h. Menampilkan frekuensi dari suatu bilangan tertentu di dalam array
var cari int
fmt.Print("Masukkan bilangan yang ingin dihitung frekuensinya: ")
fmt.Scan(&cari)
fmt.Printf("Frekuensi bilangan %d dalam array: %d\n", cari, frekuensi(arr, cari))
}

```

## SCREENSHOOT PROGRAM

```

Masukkan jumlah elemen array: 10
Masukkan elemen-elemen array:
0 1 2 3 4 5 6 7 8 9
Keseluruhan isi array: [0 1 2 3 4 5 6 7 8 9]
Elemen dengan indeks ganjil: 1 3 5 7 9
Elemen dengan indeks genap: 0 2 4 6 8
Masukkan nilai x untuk indeks kelipatan: 8
Elemen pada indeks kelipatan 8: 8
Masukkan indeks yang ingin dihapus: 3
Isi array setelah penghapusan: [0 1 2 4 5 6 7 8 9]
Rata-rata bilangan di array: 4.67
Standar deviasi bilangan di array: 2.98
Masukkan bilangan yang ingin dihitung frekuensinya: 7
Frekuensi bilangan 7 dalam array: 1
PS C:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7>

```

## DESKRIPSI PROGRAM

Program ini memproses daftar angka dengan berbagai operasi yang bisa dipilih pengguna. Pertama, pengguna memasukkan angka-angka yang akan disimpan dalam daftar. Program bisa menampilkan seluruh isi daftar, elemen dengan posisi ganjil atau genap, dan elemen pada posisi yang merupakan kelipatan angka tertentu. Program juga memungkinkan pengguna untuk menghapus angka di posisi tertentu, menghitung rata-rata dan standar deviasi dari angka-angka dalam daftar, serta mencari seberapa sering angka tertentu muncul. Semua hasil dari operasi ini ditampilkan ke layar agar mudah dibaca pengguna.

## UNGUIDED 3

```

package main

import (
    "fmt"
)

func main() {
    // Memasukkan nama kedua klub
    var klubA, klubB string
    fmt.Print("Klub A: ")
    fmt.Scan(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scan(&klubB)

    // Array untuk menyimpan hasil klub yang menang
    var pemenang []string
}

```

```

// Input skor pertandingan
var skorA, skorB int
pertandingan := 1

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

    // Hentikan input jika skor tidak valid (negatif)
    if skorA < 0 || skorB < 0 {
        break
    }

    // Menentukan pemenang berdasarkan skor
    if skorA > skorB {
        pemenang = append(pemenang, klubA)
    } else if skorA < skorB {
        pemenang = append(pemenang, klubB)
    } else {
        pemenang = append(pemenang, "Draw")
    }

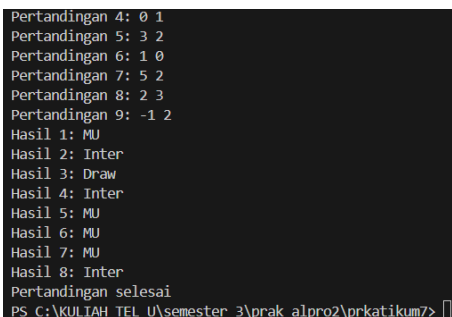
    pertandingan++
}

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

fmt.Println("Pertandingan selesai")
}

```

## SCREENSHOOT PROGRAM



```

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 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 C:\KULIAH TEL U\semester 3\prak a\pro2\prkatikum7>

```

## DESKRIPSI PROGRAM

Program ini meminta pengguna untuk memasukkan nama dua klub sepak bola, lalu mencatat hasil dari setiap pertandingan antara kedua klub tersebut. Setiap kali pertandingan berlangsung, program meminta skor masing-masing klub. Jika skor klub A lebih tinggi, maka program mencatat klub A sebagai pemenang. Jika skor klub B lebih tinggi, klub B yang dicatat sebagai pemenang. Kalau skornyaimbang, hasilnya dicatat sebagai "Draw". Proses ini berulang hingga pengguna memasukkan skor negatif, yang menandakan bahwa pertandingan sudah selesai. Setelah itu, program menampilkan hasil dari setiap pertandingan yang sudah dimainkan, lalu menutup dengan pesan "Pertandingan selesai".

## UNGUIDED 4

```
package main

import (
    "fmt"
)

const NMAX int = 127

type Tabel [NMAX]rune

// Fungsi untuk mengisi array dengan karakter dari input user
func isiArray(t *Tabel, n *int) {
    var input string
    fmt.Print("Teks: ")
    fmt.Scanln(&input)

    *n = len(input)
    for i := 0; i < *n && i < NMAX; i++ {
        t[i] = rune(input[i])
    }
}

// Fungsi untuk mencetak array karakter
func cetakArray(t Tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Printf("%c", t[i])
    }
    fmt.Println()
}

// Fungsi untuk membalikkan urutan karakter dalam array
func balikkanArray(t *Tabel, n int) {
    for i := 0; i < n/2; i++ {
        t[i], t[n-i-1] = t[n-i-1], t[i]
    }
}

func main() {
    var tab Tabel
    var n int

    // Mengisi array dengan karakter dari input
    isiArray(&tab, &n)

    // Menampilkan array asli
    fmt.Print("Teks: ")
    cetakArray(tab, n)

    // Membalikkan array dan menampilkannya
    balikkanArray(&tab, n)
    fmt.Print("Reverse teks: ")
    cetakArray(tab, n)
}
```

## SCREENSHOOT PROGRAM

```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7> go run "c:\KULIAH TEL U\semester 3\prak_
Teks: SENANG
Teks: SENANG
Reverse teks: GNANES
PS C:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7> █
```

## DESKRIPSI PROGRAM

Program ini adalah sebuah aplikasi sederhana yang ditulis dalam bahasa Go untuk mengolah karakter dari sebuah kalimat. Pengguna diminta untuk memasukkan sebuah kalimat, yang akan disimpan dalam array dengan maksimal 127 karakter. Setelah kalimat diinput, program akan menampilkan kalimat asli dan kemudian membalik urutan karakter dalam kalimat tersebut. Hasilnya, program akan mencetak kalimat yang sudah dibalik. Aplikasi ini memiliki beberapa fungsi utama, yaitu untuk mengisi array dengan kalimat, mencetak kalimat, dan membalikkan urutan karakter dalam kalimat.

## UNGUIDED 4 YANG DIEDIT

```
package main

import (
    "fmt"
)

const NMAX int = 127

type Tabel [NMAX]rune

// Fungsi untuk mengisi array dengan karakter dari input user
func isiArray(t *Tabel, n *int) {
    var input string
    fmt.Print("Teks: ")
    fmt.Scanln(&input)

    *n = len(input)
    for i := 0; i < *n && i < NMAX; i++ {
        t[i] = rune(input[i])
    }
}

// Fungsi untuk mencetak array karakter
func cetakArray(t Tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Printf("%c", t[i])
    }
    fmt.Println()
}

// Fungsi untuk membalikkan urutan karakter dalam array
func balikkanArray(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 adalah palindrom
```

```

func isPalindrome(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() {
    var tab Tabel
    var n int

    // Mengisi array dengan karakter dari input
    isiArray(&tab, &n)

    // Menampilkan array asli
    fmt.Print("Teks: ")
    cetakArray(tab, n)

    // Membalikkan array dan menampilkannya
    balikkanArray(&tab, n)
    fmt.Print("Reverse teks: ")
    cetakArray(tab, n)

    // Mengecek apakah array adalah palindrom
    if isPalindrome(tab, n) {
        fmt.Println("Teks merupakan palindrom")
    } else {
        fmt.Println("Teks bukan palindrom")
    }
}

```

## SCREENSHOOT PROGRAM

```

PS C:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7> go run "c:\KULIAH TEL U\semester 3
Teks: KATAK
Teks: KATAK
Reverse teks: KATAK
Teks merupakan palindrom
PS C:\KULIAH TEL U\semester 3\prak_alpro2\prkatikum7>

```