

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCK & ARRAY**



Disusun Oleh :

Rendi Widya Anggita/2311102278

S1IF-11-06

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

Struck merupakan cara pengaturan dan penyimpanan data dalam memori komputer agar data tersebut dapat diakses dan digunakan secara lebih efisien. Dalam dunia pemrograman, struktur data adalah landasan utama untuk merancang algoritma yang efektif dalam menyelesaikan berbagai masalah. Contoh struktur data yang sering digunakan meliputi array, linked list, stack, queue, dan tree.

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama struct> <	type <nama struct> struct {
3	<field 1> <tiipe data>	<field 1> <tiipe data>
4	<field 2> <tiipe data>	<field 2> <tiipe data>
5	<field 3> <tiipe data>	<field 3> <tiipe data>
6	>	}
7		

Array adalah kumpulan elemen data yang memiliki tipe data sama dan disusun dalam memori secara berurutan. Setiap elemen array dapat diakses menggunakan indeks, yang menunjukkan posisi elemen tersebut dalam array. Array merupakan salah satu jenis struktur data yang paling dasar dan sering digunakan dalam pengembangan program.

	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)

II. GUIDED

1. Soal Studi Case

Membuat baris bilangan dari n hingga 1

Base case : bilangan == 1

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// Struktur untuk menyimpan titik dengan koordinat (x,y)
type Titik struct {
    x int
    y int
}

// Struktur untuk menyimpan lingkaran dengan pusat dan
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
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.Println("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(&cx1, &cy1, &r1)
    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.Print("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 di luar kedua lingkaran")
    }
}
}

```

Screenshot Output

```

PS C:\Users\ASUS\OneDrive\Documents\KuliaH\Semester 3\IP, Alpro 2\modul 7> go run "c:\Users\ASUS\OneDrive\Documents\KuliaH\Semester 3\IP, Alpro 2\modul 7\guided\guided.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): 1 1 5
Masukkan koordinat pusat dan radius lingkaran 2 (cx2 cy2 r2): 8 8 4
Masukkan koordinat titik sembarang (x y): 2 2
Titik di dalam lingkaran 1
PS C:\Users\ASUS\OneDrive\Documents\KuliaH\Semester 3\IP, Alpro 2\modul 7>

```

Deskripsi Program

Program ini berfungsi untuk menentukan posisi sebuah titik sembarang terhadap dua lingkaran dalam bidang dua dimensi. Program ini menggunakan pendekatan geometris untuk mengecek apakah titik tersebut berada di dalam salah satu, kedua, atau di luar kedua lingkaran.

Pertama program mendefinisikan dua struktur data utama yaitu Titik dan Lingkaran. Struktur Titik menyimpan koordinat (x, y) dari suatu lokasi di bidang dua dimensi, struktur Lingkaran menyimpan pusat lingkaran, yang diwakili oleh sebuah titik, serta nilai radiusnya.

Program meminta pengguna untuk memasukkan data. Data pertama yang dimasukkan adalah koordinat pusat dan radius untuk lingkaran pertama. Selanjutnya, program meminta pengguna untuk menginput data yang sama untuk lingkaran kedua. Lalu, pengguna diminta memasukkan koordinat untuk titik sembarang yang akan dianalisis.

Setelah data dimasukkan, program memeriksa posisi titik terhadap kedua lingkaran. Jika titik berada di dalam kedua lingkaran, program akan mencetak pesan bahwa titik tersebut berada di dalam lingkaran 1 dan 2. Jika titik berada di luar kedua lingkaran, program akan menyatakan bahwa titik berada di luar.

III. UNGUIDED

1. Soal Studi Case

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.
-

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

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

    array := make([]int, n)

    // Mengisi array
    fmt.Println("Masukkan elemen array:")
    for i := 0; i < n; i++ {
        fmt.Printf("Elemen ke-%d: ", i)
        fmt.Scanln(&array[i])
    }

    // a. Menampilkan keseluruhan isi array
    fmt.Println("\nKeseluruhan isi array:")
    fmt.Println(array)

    // b. Menampilkan elemen array dengan indeks ganjil
    fmt.Println("\nElemen dengan indeks ganjil:")
    for i := 1; i < len(array); i += 2 {
        fmt.Printf("Indeks %d: %d\n", i, array[i])
    }

    // c. Menampilkan elemen array dengan indeks genap
    fmt.Println("\nElemen dengan indeks genap:")
    for i := 0; i < len(array); i += 2 {
        fmt.Printf("Indeks %d: %d\n", i, array[i])
    }

    // d. Menampilkan elemen dengan indeks kelipatan bilangan
    x
    var x int
```

```

    fmt.Print("\nMasukkan bilangan x untuk kelipatan indeks: ")
    fmt.Scanln(&x)
    fmt.Println("Elemen dengan indeks kelipatan", x, ":")
    for i := 0; i < len(array); i++ {
        if i%x == 0 {
            fmt.Printf("Indeks %d: %d\n", i, array[i])
        }
    }

    // e. Menghapus elemen array pada indeks tertentu
    var index int
    fmt.Print("\nMasukkan indeks elemen yang ingin dihapus: ")
    fmt.Scanln(&index)
    if index >= 0 && index < len(array) {
        array = append(array[:index], array[index+1:]...)
        fmt.Println("Array setelah elemen dihapus:")
        fmt.Println(array)
    } else {
        fmt.Println("Indeks tidak valid.")
    }

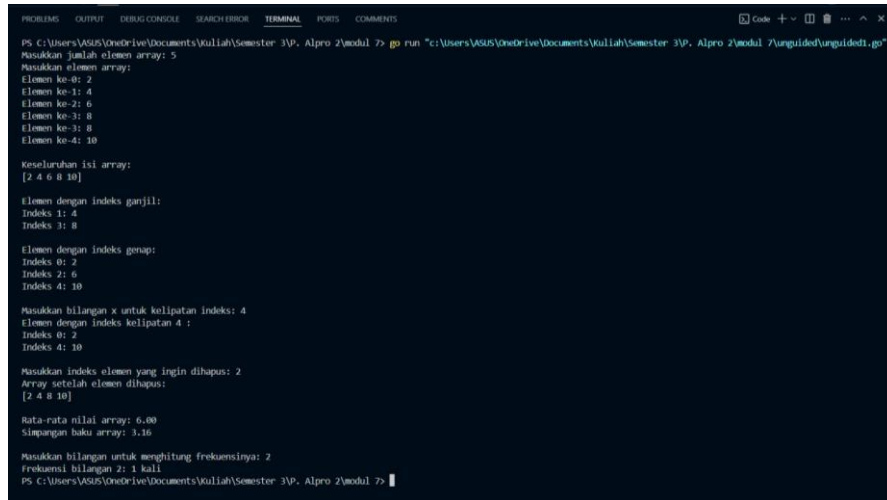
    // f. Menampilkan rata-rata bilangan dalam array
    sum := 0
    for _, value := range array {
        sum += value
    }
    rataRata := float64(sum) / float64(len(array))
    fmt.Printf("\nRata-rata nilai array: %.2f\n", rataRata)

    // g. Menampilkan standar deviasi (simpangan baku)
    var deviasiSum float64
    for _, value := range array {
        deviasiSum += math.Pow(float64(value)-rataRata, 2)
    }
    standarDeviasi := math.Sqrt(deviasiSum / float64(len(array)))
    fmt.Printf("Simpangan baku array: %.2f\n", standarDeviasi)

    // h. Menampilkan frekuensi suatu bilangan
    var target int
    fmt.Print("\nMasukkan bilangan untuk menghitung frekuensinya: ")
    fmt.Scanln(&target)
    frekuensi := 0
    for _, value := range array {
        if value == target {
            frekuensi++
        }
    }
    fmt.Printf("Frekuensi bilangan %d: %d kali\n", target, frekuensi)
}

```


Screenshoot Output



```
PS C:\Users\ASUS\OneDrive\Documents\KuliaH\Semester 3\Pro. Alpro 2\modul 7> go run "c:\Users\ASUS\OneDrive\Documents\KuliaH\Semester 3\Pro. Alpro 2\modul 7\unguided\unguided.go"
Masukkan jumlah elemen array: 5
Masukkan elemen array:
Elemen ke-0: 2
Elemen ke-1: 4
Elemen ke-2: 6
Elemen ke-3: 8
Elemen ke-4: 10

Keseluruhan isi array:
[2 4 6 8 10]

Elemen dengan indeks ganjil:
Indeks 1: 4
Indeks 3: 8

Elemen dengan indeks genap:
Indeks 0: 2
Indeks 2: 6
Indeks 4: 10

Masukkan bilangan x untuk kelipatan indeks: 4
Elemen dengan indeks kelipatan 4 :
Indeks 0: 2
Indeks 4: 10

Masukkan indeks elemen yang ingin dihapus: 2
Array setelah elemen dihapus:
[2 4 8 10]

Rata-rata nilai array: 6.80
Simpangan baku array: 3.16

Masukkan bilangan untuk menghitung frekuensinya: 2
frekuensi bilangan 2: 1 kali
PS C:\Users\ASUS\OneDrive\Documents\KuliaH\Semester 3\Pro. Alpro 2\modul 7> |
```

Deskripsi Program

Program ini berfungsi untuk melakukan berbagai operasi pada array berdasarkan masukan dari pengguna. Operasi yang dilakukan meliputi penampilan elemen array dengan kriteria tertentu, perhitungan statistik sederhana, hingga modifikasi isi array. Program ini memberikan contoh penerapan array secara dinamis, di mana pengguna dapat menentukan jumlah elemen dan mengatur isi array sesuai kebutuhan.

Pertama pengguna diminta untuk menentukan jumlah elemen array yang ingin dibuat, kemudian pengguna menginputkan isi dari elemen array secara berurutan. Kemudian program menampilkan seluruh isi array sekaligus isinya.

Selanjutnya program meminta pengguna untuk memasukkan nilai x untuk menjadi indeks kelipatan, indeks elemen yang ingin dihapus, dan bilangan untuk menghitung nilai frekuensinya.

2. Soal Studi Case

Sebuah program digunakan untuk menyimpan dan menampilkan nama-nama dari 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 masukkan nama-nama klub yang bertanding, kemudian program meminta masukkan 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 bola.

```
Klub A : MU
Klub B : Inter
Pertandingan 1 : 2 0 // MU = 2 sedangkan Inter = 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 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
```

Sourcecode

```
package main

import (
    "fmt"
)

// Struct untuk menyimpan informasi klub dan skor
type Klub struct {
    nama string
    skor int
}

func main() {
    // Input nama klub
    var klub1, klub2 Klub
    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&klub1.nama)
    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scanln(&klub2.nama)

    // Array untuk menyimpan daftar pemenang
    var hasilPertandingan []string

    // Perulangan untuk memasukkan skor tiap pertandingan
    pertandingan := 1
    for {
        fmt.Printf("\nPertandingan %d\n", pertandingan)
        fmt.Printf("Masukkan skor untuk %s: ", klub1.nama)
        fmt.Scanln(&klub1.skor)
        fmt.Printf("Masukkan skor untuk %s: ", klub2.nama)
        fmt.Scanln(&klub2.skor)

        // Periksa validitas skor
        if klub1.skor < 0 || klub2.skor < 0 {
            fmt.Println("\nSkor tidak valid! Proses dihentikan.")
            break
        }

        // Tentukan hasil pertandingan
        if klub1.skor > klub2.skor {
            hasilPertandingan = append(hasilPertandingan, klub1.nama)
            fmt.Printf("Hasil pertandingan %d: %s\n",
```

```

pertandingan, klub1.nama)
    } else if klub2.skor > klub1.skor {
        hasilPertandingan = append(hasilPertandingan,
klub2.nama)
        fmt.Printf("Hasil pertandingan %d: %s\n",
pertandingan, klub2.nama)
    } else {
        hasilPertandingan = append(hasilPertandingan,
"Draw")
        fmt.Printf("Hasil pertandingan %d: Draw\n",
pertandingan)
    }

    pertandingan++
}

// Tampilkan hasil semua pertandingan
fmt.Println("\nHasil seluruh pertandingan:")
for i, hasil := range hasilPertandingan {
    if hasil == "Draw" {
        fmt.Printf("Pertandingan %d: Draw\n", i+1)
    } else {
        fmt.Printf("Pertandingan %d dimenangkan oleh:
%s\n", i+1, hasil)
    }
}
}

```

Screenshoot Output

```

Pertandingan 4
P: C:\Users\AFB\Documents\kullan\Gomester P.P. Alpro Ziahid 71 go run "%User'sAFB\Documents\kullan\Gomester P.P. Alpro Ziahid 71\ProgramFungsional.go"
Masukkan nama Klub A: RU
Masukkan nama Klub B: DPTD

Pertandingan 1
Masukkan skor untuk Ru: 2
Masukkan skor untuk DPTD: 0
Hasil pertandingan 1: RU

Pertandingan 2
Masukkan skor untuk Ru: 1
Masukkan skor untuk DPTD: 2
Hasil pertandingan 2: DPTD

Pertandingan 3
Masukkan skor untuk Ru: 2
Masukkan skor untuk DPTD: 2
Hasil pertandingan 3: Draw

Pertandingan 4
Masukkan skor untuk Ru: 0
Masukkan skor untuk DPTD: 1
Hasil pertandingan 4: DPTD

Pertandingan 5
Masukkan skor untuk Ru: 0
Masukkan skor untuk DPTD: 2
Hasil pertandingan 5: RU

Pertandingan 6
Masukkan skor untuk Ru: 1
Masukkan skor untuk DPTD: 0
Hasil pertandingan 6: RU

Pertandingan 7
Masukkan skor untuk Ru: 5
Masukkan skor untuk DPTD: 2
Hasil pertandingan 7: RU

Pertandingan 8
Masukkan skor untuk Ru: 2
Masukkan skor untuk DPTD: 3
Hasil pertandingan 8: DPTD

Pertandingan 9
Masukkan skor untuk Ru: 4
Masukkan skor untuk DPTD: 1
Hasil pertandingan 9: DPTD

Hasil seluruh pertandingan:
Pertandingan 1 dimenangkan oleh: RU
Pertandingan 2 dimenangkan oleh: DPTD
Pertandingan 3: Draw
Pertandingan 4 dimenangkan oleh: DPTD
Pertandingan 5 dimenangkan oleh: RU
Pertandingan 6 dimenangkan oleh: RU
Pertandingan 7 dimenangkan oleh: RU
Pertandingan 8 dimenangkan oleh: DPTD
Pertandingan 9 dimenangkan oleh: DPTD
P: C:\Users\AFB\Documents\kullan\Gomester P.P. Alpro Ziahid 71

```

Deskripsi Program

Pertama program meminta pengguna untuk memasukkan nama dari dua klub yang akan bertanding, kemudian dalam program terdapat perulangan untuk skor masing-masing club untuk menentukan pemenang. Jika skor salah satu atau kedua klub negatif, program akan berhenti menerima input. Jika skor salah satu klub lebih besar, nama klub tersebut ditambahkan ke daftar pemenang. Jika skor kedua klub sama, pertandingan dianggap seri, dan tidak ada pemenang yang dicatat.

Nama klub pemenang disimpan dalam array pemenang. Array ini hanya berisi nama klub yang menang, tanpa mencatat klub yang kalah atau pertandingan yang seri. Kemudian program akan menampilkan daftar pemenang dari semua pertandingan yang telah berlangsung ditampilkan secara berurutan, lengkap dengan urutan pertandingan.

3. Soal Studi Case

Sourcecode

```
package main

import (
    "fmt"
)

// Konstanta maksimum jumlah elemen array
const NMAX int = 127

// Struct untuk merepresentasikan array dan jumlah elemen
type Tabel struct {
    tab [NMAX]rune // Array untuk menyimpan karakter
    n    int       // Jumlah elemen yang valid dalam array
}

// Fungsi untuk mengisi array dengan karakter dari input
// pengguna
func isiArray(t *Tabel) {
    fmt.Println("Masukkan karakter untuk kata (diakhiri
dengan TITIK '.'):")

    for {
        var input rune
        fmt.Scanf("%c", &input)

        if input == '.' || t.n >= NMAX { // Berhenti jika
            ditemukan titik atau jumlah elemen maksimal tercapai
            break
        }
        t.tab[t.n] = input
        t.n++
    }
}

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

```

// Fungsi untuk membalikkan urutan elemen dalam array
func balikkanArray(t Tabel) Tabel {
    var reversed Tabel
    reversed.n = t.n
    for i := 0; i < t.n; i++ {
        reversed.tab[i] = t.tab[t.n-1-i]
    }
    return reversed
}

// Fungsi untuk memeriksa apakah array membentuk palindrom
func isPalindrom(t Tabel) bool {
    for i, j := 0, t.n-1; i < j; i, j = i+1, j-1 {
        if t.tab[i] != t.tab[j] {
            return false
        }
    }
    return true
}

func main() {
    var tab Tabel

    // Meminta input dari pengguna
    isiArray(&tab)

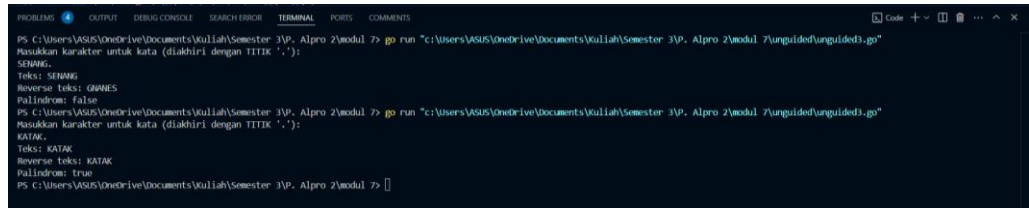
    // Menampilkan teks asli
    fmt.Print("Teks: ")
    cetakArray(tab)

    // Membalikkan array dan menampilkan hasilnya
    reversedTab := balikkanArray(tab)
    fmt.Print("Reverse teks: ")
    cetakArray(reversedTab)

    // Menentukan apakah teks merupakan palindrom
    if isPalindrom(tab) {
        fmt.Println("Palindrom: true")
    } else {
        fmt.Println("Palindrom: false")
    }
}

```

Screenshoot Output



```
PS C:\Users\ASUS\OneDrive\Documents\kuliah\Semester 3\Pro Alpro 2\modul 7\unguided\unguided3.go> go run "c:\Users\ASUS\OneDrive\Documents\kuliah\Semester 3\Pro Alpro 2\modul 7\unguided\unguided3.go"
Masukkan karakter untuk kata (diakhiri dengan titik '.'):
SENANG.
Teks: SENANG
Reverse teks: GNANGS
Palindrome: false
PS C:\Users\ASUS\OneDrive\Documents\kuliah\Semester 3\Pro Alpro 2\modul 7\unguided\unguided3.go> go run "c:\Users\ASUS\OneDrive\Documents\kuliah\Semester 3\Pro Alpro 2\modul 7\unguided\unguided3.go"
Masukkan karakter untuk kata (diakhiri dengan titik '.'):
KATAK.
Teks: KATAK
Reverse teks: KATAK
Palindrome: true
PS C:\Users\ASUS\OneDrive\Documents\kuliah\Semester 3\Pro Alpro 2\modul 7\unguided\unguided3.go>
```

Deskripsi Program

Program ini berfungsi untuk mengecek apakah kata yang dimasukkan oleh pengguna merupakan palindrom atau bukan, pertama program meminta pengguna memasukkan sebuah kata yang diakhiri dengan tanda titik (.). Kata yang dimasukkan kemudian disimpan dalam sebuah array karakter, setelah input selesai, program akan menampilkan kata tersebut dan kebalikannya, yang dihasilkan melalui fungsi pembalik array. Selanjutnya, program akan memeriksa apakah kata tersebut merupakan palindrom, yaitu apakah kata tersebut tetap sama jika dibaca dari depan maupun belakang. Hasil pengecekan akan ditampilkan dalam bentuk true jika kata merupakan palindrom, atau false jika bukan.