

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII**

**STRUCK & ARRAY**



**Disusun Oleh : Rory**

**Refaldo Sinulingga**

**(2311102323)**

**11-IF-06**

**Dosen Pengampu :**



**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## 1. DASAR TEORI

Dalam pemrograman, **struct** (struktur) adalah tipe data yang memungkinkan kita untuk mengelompokkan beberapa variabel yang berbeda tipe dalam satu kesatuan data. Penggunaan struct sangat berguna ketika kita ingin merepresentasikan suatu entitas atau objek yang memiliki beberapa atribut terkait. Sebagai contoh, jika kita ingin menyimpan data seorang mahasiswa, kita mungkin perlu menyimpan informasi seperti nama, nim (nomor induk mahasiswa), dan nilai. Dengan menggunakan struct, kita bisa mengelompokkan atribut-atribut ini dalam satu wadah, sehingga lebih mudah untuk dikelola dan diakses. Setiap bagian dalam struct disebut "field" atau "member".

**Array**, di sisi lain, adalah struktur data yang digunakan untuk menyimpan kumpulan elemen yang memiliki tipe data yang sama. Array menyimpan elemen secara berurutan, dan setiap elemen bisa diakses menggunakan indeks, yang dimulai dari 0 untuk elemen pertama. Array banyak digunakan saat kita perlu menyimpan data dalam jumlah besar yang memiliki tipe data seragam, seperti deretan angka atau daftar nama. Array dapat berbentuk satu dimensi atau lebih, seperti array dua dimensi yang dapat digunakan untuk merepresentasikan matriks atau tabel.

Kedua konsep ini, struct dan array, dapat digunakan bersama untuk menyimpan dan mengelola data yang lebih kompleks. Misalnya, jika kita ingin menyimpan data dari beberapa mahasiswa, kita bisa membuat array dari struct mahasiswa. Dengan cara ini, setiap elemen dalam array merepresentasikan data lengkap dari satu mahasiswa, sementara seluruh array merepresentasikan kumpulan data mahasiswa. Dengan kombinasi struct dan array, kita dapat menyusun data yang lebih kompleks namun tetap terorganisir, terutama dalam program yang membutuhkan pengelolaan data dalam jumlah besar.

Secara keseluruhan, **struct** dan **array** adalah dua konsep penting yang saling melengkapi dalam pemrograman. Struct memungkinkan kita menyimpan data yang saling terkait tetapi berbeda tipe dalam satu unit, sementara array memudahkan penyimpanan data bertipe sama dalam jumlah banyak secara terurut. Kombinasi keduanya memungkinkan kita mengelola data yang lebih kompleks secara efisien dan terstruktur.

## I. GUIDED

### 1. Soal Studi Case

#### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

type titik_aldo_323 struct {
    x_aldo_323 int
    y_aldo_323 int
}

type lingkaran_aldo_323 struct
{
    pusat_aldo_323
    titik_aldo_323
    radius_aldo_323 int
}

func
hitungjarak_aldo_323(a_aldo_3
23, b_aldo_323 titik_aldo_323)
```

```

float64 {
    return
    math.Sqrt(float64((a_aldo_323.
x_aldo_323-
b_aldo_323.x_aldo_323)*(a_ald
o_323.x_aldo_323-
b_aldo_323.x_aldo_323) +

(a_aldo_323.y_aldo_32
3-
b_aldo_323.y_aldo_323)*(a_ald
o_323.y_aldo_323-
b_aldo_323.y_aldo_323))))
}

func
titikdalamlingkaran_aldo_323(t
_aldo_323 titik_aldo_323,
l_aldo_323 lingkaran_aldo_323)
bool {
    jarak_aldo_323 :=
hitungjarak_aldo_323(t_aldo_32
3, l_aldo_323.pusat_aldo_323)

    return jarak_aldo_323
<=
float64(l_aldo_323.radius_aldo_
323)

```

```

}

func main() {

    var cx1_aldo_323,
cy1_aldo_323, r1_aldo_323 int

    fmt.Print("Masukkan
koordinator pusat dan radius
lingkaran 1 (cx1, cy1, r1): ")

    fmt.Scanln(&cx1_aldo_
323, &cy1_aldo_323,
&r1_aldo_323)

    lingkaran1_aldo_323 :=
lingkaran_aldo_323{pusat_aldo
_323:
titik_aldo_323{x_aldo_323:
cx1_aldo_323, y_aldo_323:
cy1_aldo_323},
radius_aldo_323: r1_aldo_323}

    var cx2_aldo_323,
cy2_aldo_323, r2_aldo_323 int

    fmt.Print("Masukkan
koordinator pusat dan radius
lingkaran 2 (cx2, cy2, r2): ")

    fmt.Scanln(&cx2_aldo_
323, &cy2_aldo_323,

```

```

&r2_aldo_323)

    lingkaran2_aldo_323 :=
lingkaran_aldo_323{pusat_aldo
_323:
titik_aldo_323{x_aldo_323:
cx2_aldo_323, y_aldo_323:
cy2_aldo_323},
radius_aldo_323: r2_aldo_323}

    var x_aldo_323,
y_aldo_323 int

    fmt.Print("Masukkan
koordinat titik sembarang (x,
y): ")

    fmt.Scanln(&x_aldo_32
3, &y_aldo_323)

    titikSembarang_aldo_32
3 :=
titik_aldo_323{x_aldo_323:
x_aldo_323, y_aldo_323:
y_aldo_323}

    didalam1_aldo_323 :=
titikdalamlingkaran_aldo_323(ti
tikSembarang_aldo_323,
lingkaran1_aldo_323)

    didalam2_aldo_323 :=

```

```
titikdalamlingkaran_aldo_323(ti
tikSembarang_aldo_323,
lingkaran2_aldo_323)

    if didalam1_aldo_323
    && didalam2_aldo_323 {

        fmt.Println("Titik
berada di dalam lingkaran 1 dan
2")

    } else if
    didalam1_aldo_323 {

        fmt.Println("Titik
berada di dalam lingkaran 1")

    } else if
    didalam2_aldo_323 {

        fmt.Println("Titik
berada di dalam lingkaran 2")

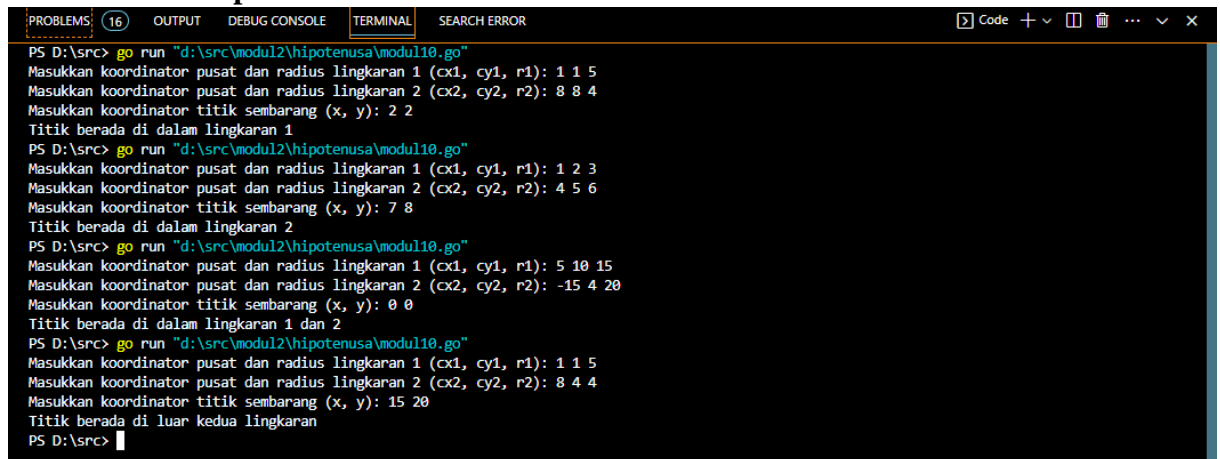
    } else {

        fmt.Println("Titik
berada di luar kedua lingkaran")

    }

}
```

## Screenshoot Output



```
PROBLEMS: 16 OUTPUT DEBUG CONSOLE TERMINAL SEARCH ERROR
PS D:\src> go run "d:\src\modul2\hipotenusa\modul10.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 berada di dalam lingkaran 1
PS D:\src> go run "d:\src\modul2\hipotenusa\modul10.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): 1 2 3
Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): 4 5 6
Masukkan koordinat titik sembarang (x, y): 7 8
Titik berada di dalam lingkaran 2
PS D:\src> go run "d:\src\modul2\hipotenusa\modul10.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): 5 10 15
Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): -15 4 20
Masukkan koordinat titik sembarang (x, y): 0 0
Titik berada di dalam lingkaran 1 dan 2
PS D:\src> go run "d:\src\modul2\hipotenusa\modul10.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 4 4
Masukkan koordinat titik sembarang (x, y): 15 20
Titik berada di luar kedua lingkaran
PS D:\src>
```

## Deskripsi Program

Program ini bertujuan untuk menentukan apakah suatu titik sembarang berada di dalam salah satu, kedua, atau di luar dua lingkaran yang didefinisikan. Program menggunakan konsep geometri dasar untuk menghitung jarak antara titik dan pusat lingkaran, serta membandingkan jarak tersebut dengan radius lingkaran.



## UNGUIDED

### 1. Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var N_aldo_323 int
    fmt.Println("Masukkan jumlah elemen array (N): ")
    fmt.Scanln(&N_aldo_323)

    array_aldo_323 := make([]int, N_aldo_323)
    fmt.Println("Masukkan elemen-elemen array:")
    for i_aldo_323 := 0; i_aldo_323 < N_aldo_323; i_aldo_323++ {
        fmt.Printf("Elemen %d: ", i_aldo_323)
        fmt.Scanln(&array_aldo_323[i_aldo_323])
    }

    fmt.Println("Isi array:", array_aldo_323)

    fmt.Print("Elemen pada indeks ganjil: ")
    for i_aldo_323 := 1; i_aldo_323 < N_aldo_323; i_aldo_323 += 2 {
        fmt.Print(array_aldo_323[i_aldo_323], " ")
    }
    fmt.Println()

    fmt.Print("Elemen pada indeks genap: ")
    for i_aldo_323 := 0; i_aldo_323 < N_aldo_323; i_aldo_323 += 2 {
        fmt.Print(array_aldo_323[i_aldo_323], " ")
    }
    fmt.Println()

    var x_aldo_323 int
    fmt.Print("Masukkan bilangan untuk kelipatan indeks: ")
    fmt.Scanln(&x_aldo_323)
    fmt.Printf("Elemen pada indeks kelipatan %d: ", x_aldo_323)
    for i_aldo_323 := x_aldo_323; i_aldo_323 < N_aldo_323; i_aldo_323 += x_aldo_323 {
        fmt.Print(array_aldo_323[i_aldo_323], " ")
    }
    fmt.Println()

    var idx_aldo_323 int
    fmt.Print("Masukkan indeks elemen yang ingin dihapus: ")
    fmt.Scanln(&idx_aldo_323)
    if idx_aldo_323 >= 0 && idx_aldo_323 < N_aldo_323 {
        array_aldo_323 = append(array_aldo_323[:idx_aldo_323],
array_aldo_323[idx_aldo_323+1:]...)
        fmt.Println("Isi array setelah penghapusan:", array_aldo_323)
```

```

    } else {
        fmt.Println("Indeks tidak valid.")
    }

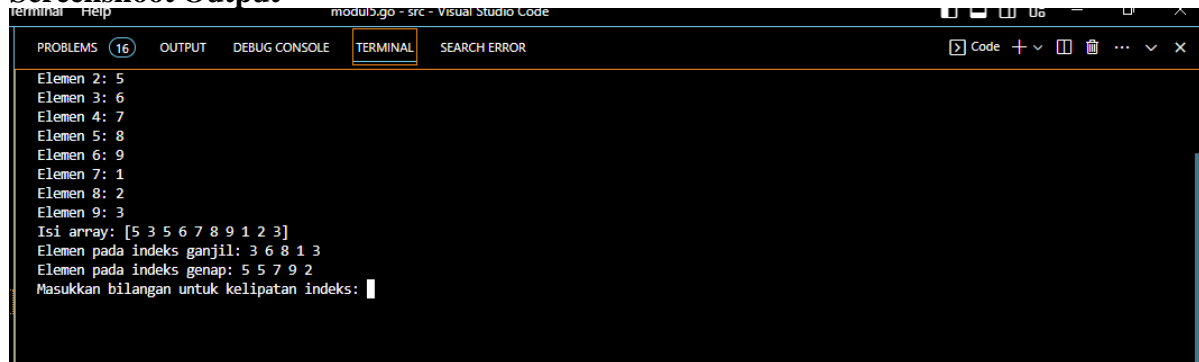
    sum_aldo_323 := 0
    for _, value_aldo_323 := range array_aldo_323 {
        sum_aldo_323 += value_aldo_323
    }
    rataRata_aldo_323 := float64(sum_aldo_323) / float64(len(array_aldo_323))
    fmt.Printf("Rata-rata elemen array: %.2f\n", rataRata_aldo_323)

    var varianceSum_aldo_323 float64
    for _, value_aldo_323 := range array_aldo_323 {
        varianceSum_aldo_323 += math.Pow(float64(value_aldo_323)-rataRata_aldo_323, 2)
    }
    stdDeviasi_aldo_323 := math.Sqrt(varianceSum_aldo_323 / float64(len(array_aldo_323)))
    fmt.Printf("Standar deviasi elemen array: %.2f\n", stdDeviasi_aldo_323)

    var cari_aldo_323 int
    fmt.Print("Masukkan bilangan yang ingin dicari frekuensinya: ")
    fmt.Scanln(&cari_aldo_323)
    frekuensi_aldo_323 := 0
    for _, value_aldo_323 := range array_aldo_323 {
        if value_aldo_323 == cari_aldo_323 {
            frekuensi_aldo_323++
        }
    }
    fmt.Printf("Frekuensi bilangan %d: %d\n", cari_aldo_323, frekuensi_aldo_323)
}

```

## Screenshot Output

A screenshot of a Visual Studio Code terminal window. The terminal title bar shows 'terminal - Help' and 'modul5.go - src - Visual Studio Code'. The terminal interface includes tabs for 'PROBLEMS' (with 16 errors), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (selected), and 'SEARCH ERROR'. The output text in the terminal is as follows:

```
Elemen 2: 5  
Elemen 3: 6  
Elemen 4: 7  
Elemen 5: 8  
Elemen 6: 9  
Elemen 7: 1  
Elemen 8: 2  
Elemen 9: 3  
Isi array: [5 3 5 6 7 8 9 1 2 3]  
Elemen pada indeks ganjil: 3 6 8 1 3  
Elemen pada indeks genap: 5 5 7 9 2  
Masukkan bilangan untuk kelipatan indeks: █
```

## Deskripsi Program

Program ini adalah aplikasi berbasis CLI (Command Line Interface) yang memungkinkan pengguna untuk melakukan berbagai operasi pada array integer, seperti pengelompokan elemen berdasarkan indeks, penghapusan elemen, perhitungan statistik (rata-rata dan standar deviasi), serta pencarian frekuensi nilai tertentu dalam array.

2.

```
package main

import (
    "fmt"
)

func main() {

    var klubA_aldo_323, klubB_aldo_323 string
    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&klubA_aldo_323)
    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scanln(&klubB_aldo_323)

    var pemenang_aldo_323 []string
    pertandingan_aldo_323 := 1

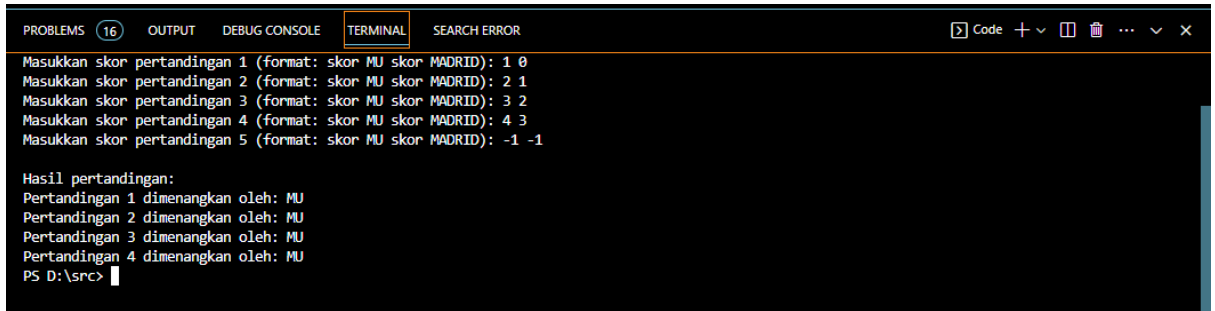
    for {
        var skorA_aldo_323, skorB_aldo_323 int
        fmt.Printf("Masukkan skor pertandingan %d (format: skor %s skor %s): ", pertandingan_aldo_323,
klubA_aldo_323, klubB_aldo_323)
        fmt.Scanf("%d %d\n", &skorA_aldo_323, &skorB_aldo_323)

        if skorA_aldo_323 < 0 || skorB_aldo_323 < 0 {
            break
        }

        if skorA_aldo_323 > skorB_aldo_323 {
            pemenang_aldo_323 = append(pemenang_aldo_323, klubA_aldo_323)
        } else if skorB_aldo_323 > skorA_aldo_323 {
            pemenang_aldo_323 = append(pemenang_aldo_323, klubB_aldo_323)
        } else {
            pemenang_aldo_323 = append(pemenang_aldo_323, "Seri") // Skor sama dianggap seri
        }
        pertandingan_aldo_323++
    }

    fmt.Println("\nHasil pertandingan:")
    for i_aldo_323, p_aldo_323 := range pemenang_aldo_323 {
        if p_aldo_323 == "Seri" {
            fmt.Printf("Pertandingan %d: Seri\n", i_aldo_323+1)
        } else {
            fmt.Printf("Pertandingan %d dimenangkan oleh: %s\n", i_aldo_323+1, p_aldo_323)
        }
    }
}
```

## Scrennshoot output

A screenshot of a Visual Studio Code terminal window. The terminal has a dark background with light-colored text. At the top, there is a tab labeled 'TERMINAL' which is highlighted with an orange border. To the left of the terminal, there are tabs for 'PROBLEMS' (with a count of 16), 'OUTPUT', 'DEBUG CONSOLE', and 'SEARCH ERROR'. The terminal content shows the execution of a program that prompts for five matches between MU and MADRID. The input for each match is '1 0', '2 1', '3 2', '4 3', and '-1 -1'. After the inputs, the program displays the results, stating that MU won the first four matches and that the fifth match was a draw. The prompt 'PS D:\src>' is visible at the bottom of the terminal.

```
PROBLEMS (16) OUTPUT DEBUG CONSOLE TERMINAL SEARCH ERROR
Masukkan skor pertandingan 1 (format: skor MU skor MADRID): 1 0
Masukkan skor pertandingan 2 (format: skor MU skor MADRID): 2 1
Masukkan skor pertandingan 3 (format: skor MU skor MADRID): 3 2
Masukkan skor pertandingan 4 (format: skor MU skor MADRID): 4 3
Masukkan skor pertandingan 5 (format: skor MU skor MADRID): -1 -1

Hasil pertandingan:
Pertandingan 1 dimenangkan oleh: MU
Pertandingan 2 dimenangkan oleh: MU
Pertandingan 3 dimenangkan oleh: MU
Pertandingan 4 dimenangkan oleh: MU
PS D:\src>
```

## Deskripsi program

Program ini adalah aplikasi berbasis CLI (Command Line Interface) yang digunakan untuk mencatat hasil serangkaian pertandingan antara dua klub sepak bola. Program memungkinkan pengguna untuk mencatat skor, menentukan pemenang setiap pertandingan, dan menampilkan hasil keseluruhan dari semua pertandingan.

3.

```
package main

import (
    "fmt"
    "strings"
)

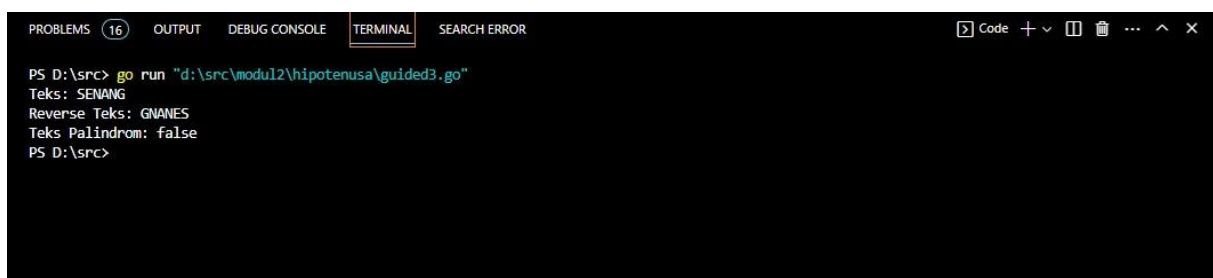
func main() {
    teks_aldo_323 := "SENANG"
    reverseTeks_aldo_323 := reverseString_aldo_323(teks_aldo_323)
    fmt.Println("Teks:", teks_aldo_323)
    fmt.Println("Reverse Teks:", reverseTeks_aldo_323)

    isPalindrom_aldo_323 := isPalindrome_aldo_323(teks_aldo_323)
    fmt.Println("Teks Palindrom:", isPalindrom_aldo_323)
}

func reverseString_aldo_323(s_aldo_323 string) string {
    runes_aldo_323 := []rune(s_aldo_323)
    for i_aldo_323, j_aldo_323 := 0, len(runes_aldo_323)-1; i_aldo_323 < j_aldo_323; i_aldo_323, j_aldo_323 = i_aldo_323+1, j_aldo_323-1 {
        runes_aldo_323[i_aldo_323], runes_aldo_323[j_aldo_323] =
        runes_aldo_323[j_aldo_323], runes_aldo_323[i_aldo_323]
    }
    return string(runes_aldo_323)
}

func isPalindrome_aldo_323(s_aldo_323 string) bool {
    s_aldo_323 = strings.ToLower(s_aldo_323)
    return s_aldo_323 == reverseString_aldo_323(s_aldo_323)
}
```

Screenshoot program

A screenshot of a code editor's terminal window. The terminal shows the execution of a Go program. The command entered is 'go run "d:\src\modul2\hipotenusa\guided3.go"'. The output shows 'Teks: SENANG', 'Reverse Teks: GNANES', and 'Teks Palindrom: false'. The terminal window has a dark background with light-colored text. The editor's interface includes tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'SEARCH ERROR'. The 'TERMINAL' tab is active. The terminal window also shows the prompt 'PS D:\src>'.

## Deskripsi program

Program ini adalah aplikasi berbasis CLI (Command Line Interface) yang berfungsi untuk membalikkan teks (reverse string) dan memeriksa apakah teks tersebut adalah **palindrom**. Sebuah palindrom adalah teks yang terbaca sama dari depan maupun dari belakang.

