

**LAPORAN PRAKTIKUM  
ALGORITME DAN PEMROGRAMAN 2**

**MODUL 7  
STRUCT DAN ARRAY**



**Oleh:**

**MUHAMMAD AMIR SALEH**

**2311102233**

**IF - 11 - 06**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

## **I. Dasar Teori**

Struct adalah tipe data yang memungkinkan kita menggabungkan berbagai data berbeda menjadi satu kesatuan. Ini sangat berguna untuk merepresentasikan objek yang memiliki beberapa atribut, seperti data mahasiswa dengan nama, umur, dan jurusan. Kita bisa mengakses data di dalam struct dengan mudah menggunakan notasi titik, misalnya `mhs.Nama`, sehingga mempermudah pengelolaan data yang kompleks dan terstruktur.

Array adalah kumpulan elemen dengan tipe data yang sama dan ukuran tetap yang ditentukan sejak awal. Array cocok digunakan untuk menyimpan data dalam jumlah tetap, misalnya deretan angka atau daftar nama. Setiap elemen dalam array diakses dengan indeks, seperti `angka[0]`. Walaupun sederhana, array memiliki keterbatasan karena ukurannya tidak bisa diubah. Penggunaan struct dan array sering digabungkan untuk membuat struktur data yang lebih kompleks dan fleksibel dalam program.

## II. Guided Guided 1

```
package main

import (
    "fmt"
    "math"
)

type titik struct {
    x, y int
}

type lingkaran struct {
    tengah titik
    radius int
}

func jarak(p, q titik) float64 {
    dx := float64(p.x - q.x)
    dy := float64(p.y - q.y)
    return math.Sqrt(dx*dx + dy*dy)
}

func dalam(c lingkaran, p titik) bool {
    return jarak(c.tengah, p) <= float64(c.radius)
}

func posisi(c1, c2 lingkaran, p titik) string {
    inlingkaran1 := dalam(c1, p)
    inlingkaran2 := dalam(c2, p)

    if inlingkaran1 && inlingkaran2 {
        return "Titik di dalam lingkaran 1 dan 2"
    } else if inlingkaran1 {
        return "Titik di dalam lingkaran 1"
    } else if inlingkaran2 {
        return "Titik di dalam lingkaran 2"
    }
    return "Titik di luar lingkaran 1 dan 2"
}

func main() {
    var c1x, c1y, r1 int
    fmt.Scan(&c1x, &c1y, &r1)
    lingkaran1 := lingkaran{titik{c1x, c1y}, r1}
```

```

var c2x, c2y, r2 int
fmt.Scan(&c2x, &c2y, &r2)
lingkaran2 := lingkaran{titik{c2x, c2y}, r2}

var px, py int
fmt.Scan(&px, &py)
titik := titik{px, py}

result := posisi(lingkaran1, lingkaran2, titik)
fmt.Println(result)
}

```

## Screenshots Output

```

package main

import (
    "fmt"
    "math"
)

type titik struct {
    x, y int
}

type lingkaran struct {
    center titik
    radius int
}

func jarak(p, q titik) float64 {
    return math.Sqrt(float64((p.x - q.x)*(p.x - q.x) + (p.y - q.y)*(p.y - q.y)))
}

func posisi(l1, l2 lingkaran, t titik) string {
    r1 := l1.radius
    r2 := l2.radius
    d := jarak(l1.center, l2.center)

    if d < r1+r2 {
        if d > math.Abs(r1-r2) {
            return "Titik di dalam lingkaran 1"
        }
    }
    return "Titik di luar lingkaran"
}

func main() {
    var c1x, c1y, r1 int
    var c2x, c2y, r2 int
    var px, py int

    fmt.Scan(&c1x, &c1y, &r1)
    fmt.Scan(&c2x, &c2y, &r2)
    fmt.Scan(&px, &py)

    l1 := lingkaran{titik{c1x, c1y}, r1}
    l2 := lingkaran{titik{c2x, c2y}, r2}
    t := titik{px, py}

    result := posisi(l1, l2, t)
    fmt.Println(result)
}

```

```

go run "/Users/raisaaw/Desktop/2311182233_Muhammad Amir Saleh_Modul 7/guided1.go"
raisaaw@Tokyo-Mac 2311182233_Muhammad Amir Saleh_Modul 7 % go run "/Users/raisaaw/Desktop/2311182233_Muhammad Amir Saleh_Modul 7/guided1.go"
1 1 5
8 8 4
Titik di dalam lingkaran 1
raisaaw@Tokyo-Mac 2311182233_Muhammad Amir Saleh_Modul 7 %

```

## Deskripsi:

Program ini mengevaluasi apakah sebuah titik berada di dalam, di luar, atau di antara dua lingkaran. Setiap lingkaran direpresentasikan oleh tipe lingkaran yang memiliki pusat berupa struktur titik dan jari-jari. Fungsi jarak digunakan untuk menghitung jarak antara dua titik menggunakan rumus jarak Euclidean. Fungsi dalam memeriksa apakah jarak antara titik tertentu dan pusat lingkaran lebih kecil atau sama dengan jari-jari lingkaran, untuk menentukan apakah titik tersebut berada di dalam lingkaran. Fungsi posisi mengembalikan status titik, apakah berada di dalam salah satu, kedua, atau di luar kedua lingkaran. Program menerima input berupa koordinat pusat dan jari-jari dari dua lingkaran serta koordinat titik yang ingin diuji. Setelah itu, hasil evaluasi dicetak ke layar, menunjukkan posisi titik relatif terhadap lingkaran-lingkaran tersebut.

### III. Unguided Unguided 1

```
package main

import (
    "fmt"
    "math"
)

type Statistik struct {
    data []float64
}

func BuatStatistik(ukuran int) *Statistik {
    angka := make([]float64, ukuran)
    fmt.Println("Masukkan", ukuran, "bilangan:")
    for i := 0; i < ukuran; i++ {
        fmt.Scan(&angka[i])
    }
    return &Statistik{data: angka}
}

func (s *Statistik) ProsesData() {
    fmt.Println("\na. Keseluruhan isi array:")
    fmt.Println(s.data)

    fmt.Print("\nb. Elemen dengan indeks ganjil: ")
    for i := 1; i < len(s.data); i += 2 {
        fmt.Printf("%.2f ", s.data[i])
    }
    fmt.Println()

    fmt.Print("\nc. Elemen dengan indeks genap: ")
    for i := 0; i < len(s.data); i += 2 {
        fmt.Printf("%.2f ", s.data[i])
    }
    fmt.Println()

    var kelipatan int
    fmt.Print("\nMasukkan nilai kelipatan untuk menampilkan elemen: ")
    fmt.Scan(&kelipatan)
    fmt.Printf("\nd. Elemen dengan indeks kelipatan %d: ",
        kelipatan)
    for i := 0; i < len(s.data); i += kelipatan {
        fmt.Printf("%.2f ", s.data[i])
    }
}
```

```

    }
    fmt.Println()

    var indeksHapus int
    fmt.Print("\nMasukkan indeks yang akan dihapus: ")
    fmt.Scan(&indeksHapus)
    if indeksHapus >= 0 && indeksHapus < len(s.data) {
        s.data = append(s.data[:indeksHapus],
s.data[indeksHapus+1:]...)
        fmt.Println("\ne. Array setelah penghapusan indeks",
indeksHapus, ":")
        fmt.Println(s.data)
    }

    total := 0.0
    for _, nilai := range s.data {
        total += nilai
    }
    average := total / float64(len(s.data))
    fmt.Printf("\nf. Rata-rata: %.2f\n", average)

    var totalKuadrat float64
    for _, nilai := range s.data {
        totalKuadrat += (nilai - average) * (nilai - average)
    }
    stdev := math.Sqrt(totalKuadrat / float64(len(s.data)))
    fmt.Printf("\ng. Simpangan baku: %.2f\n", stdev)

    var angkaFrekuensi float64
    fmt.Print("\nMasukkan bilangan yang ingin dihitung
frekuensinya: ")
    fmt.Scan(&angkaFrekuensi)
    frekuensi := 0
    for _, nilai := range s.data {
        if nilai == angkaFrekuensi {
            frekuensi++
        }
    }
    fmt.Printf("\nh. Frekuensi bilangan %.2f: %d kali\n",
angkaFrekuensi, frekuensi)
}

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

```

```

statistik := BuatStatistik(jumlahData)
statistik.ProsesData()
}

```

## Screenshots Output

The screenshot shows a Go IDE with the following code in `guided1.go`:

```

1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 type Statistik struct {
9     data []float64
10 }
11
12 func BuatStatistik(ukuran int) *Statistik {
13     angka := make([]float64, ukuran)
14     fmt.Println("Masukkan", ukuran, "bilangan:")
15     for i := 0; i < ukuran; i++ {
16         fmt.Scan(&angka[i])
17     }
18 }

```

The terminal output shows the program's execution:

```

Masukkan jumlah elemen array: 5
Masukkan 5 bilangan:
2 2 4 3 6
a. Keseluruhan isi array:
[2 2 4 3 6]
b. Elemen dengan indeks ganjil: 2.00 3.00
c. Elemen dengan indeks genap: 2.00 4.00 6.00
Masukkan nilai kelipatan untuk menampilkan elemen: 2
d. Elemen dengan indeks kelipatan 2: 2.00 4.00 6.00
Masukkan indeks yang akan dihapus: 1
e. Array setelah penghapusan indeks 1 :
[2 4 3 6]
f. Rata-rata: 3.75
g. Simpangan baku: 1.48
Masukkan bilangan yang ingin dihitung frekuensinya: 3
h. Frekuensi bilangan 3.00: 1 kali

```

## Deskripsi:

Program ini mengelola data statistik menggunakan array dalam bentuk tipe Statistik yang berisi data bertipe float64. Program dimulai dengan meminta pengguna untuk memasukkan jumlah elemen array, kemudian mengisi elemen-elemen tersebut melalui fungsi BuatStatistik, yang menginisialisasi array dan menerima input untuk setiap elemen. Setelah data dimasukkan, fungsi ProsesData menjalankan beberapa analisis terhadap data tersebut, seperti menampilkan seluruh isi array, elemen dengan indeks ganjil, dan elemen dengan indeks genap. Fungsi ini juga meminta pengguna untuk memasukkan kelipatan tertentu dan menampilkan elemen dengan indeks kelipatan tersebut, serta meminta indeks elemen yang ingin dihapus dan menampilkan array setelah penghapusan. Selain itu, fungsi ini menghitung dan menampilkan rata-rata dan simpangan baku dari elemen-elemen array. Terakhir, program meminta pengguna untuk memasukkan angka dan menghitung frekuensi kemunculannya dalam array. Program ini memanfaatkan fitur slicing dan manipulasi array di Go untuk melakukan berbagai operasi statistik secara interaktif.

## Unguided 2

```
package main

import (
    "fmt"
)

type HasilPertandingan string

const (
    MENANG HasilPertandingan = "MENANG"
    SERI    HasilPertandingan = "SERI"
    KALAH   HasilPertandingan = "KALAH"
)

type Klub struct {
    nama string
    skor int
}

type RekapPertandingan struct {
    klubA      Klub
    klubB      Klub
    pemenang   []string
    jumlahData int
}

func BuatRekapPertandingan() *RekapPertandingan {
    var namaKlubA, namaKlubB string

    fmt.Print("Klub A : ")
    fmt.Scan(&namaKlubA)
    fmt.Print("Klub B : ")
    fmt.Scan(&namaKlubB)

    return &RekapPertandingan{
        klubA:      Klub{nama: namaKlubA},
        klubB:      Klub{nama: namaKlubB},
        pemenang:   make([]string, 0),
        jumlahData: 0,
    }
}

func (r *RekapPertandingan) tentukanPemenang(skorA, skorB int)
string {
    if skorA > skorB {
```



```

        return r.klubA.nama
    } else if skorB > skorA {
        return r.klubB.nama
    }
    return "Draw"
}

func (r *RekapPertandingan) prosesInputSkor() {
    nomorPertandingan := 1

    for {
        fmt.Printf("Pertandingan %d : ", nomorPertandingan)
        fmt.Scan(&r.klubA.skor, &r.klubB.skor)

        if r.klubA.skor < 0 || r.klubB.skor < 0 {
            break
        }

        hasil := r.tentukanPemenang(r.klubA.skor, r.klubB.skor)
        if hasil != "Draw" {
            r.pemenang = append(r.pemenang, hasil)
        } else {
            r.pemenang = append(r.pemenang, "Draw")
        }

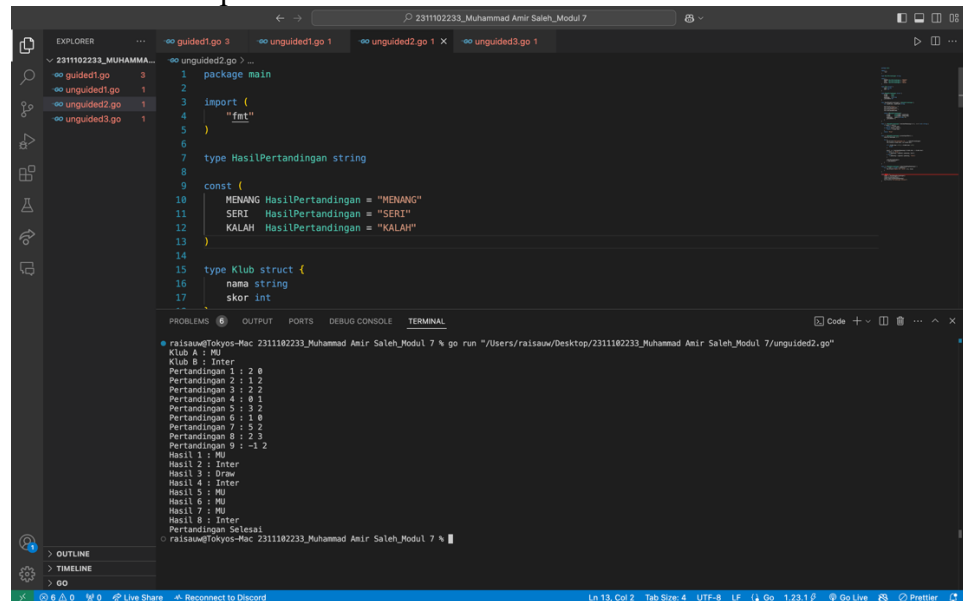
        nomorPertandingan++
        r.jumlahData++
    }
}

func (r *RekapPertandingan) tampilkanRekapPemenang() {
    for i, klub := range r.pemenang {
        fmt.Printf("Hasil %d : %s\n", i+1, klub)
    }
}

func main() {
    rekap := BuatRekapPertandingan()
    rekap.prosesInputSkor()
    rekap.tampilkanRekapPemenang()
    fmt.Println("Pertandingan Selesai")
}

```

## Screenshots Output



The screenshot shows a Go program in VS Code. The code defines a package, imports 'fmt', and defines a 'HasilPertandingan' string type and a 'Klub' struct with 'nama' (string) and 'skor' (int) fields. It also defines a 'const' block with 'MENANG', 'SERI', and 'KALAH' results. The terminal output shows the program running and displaying the results of 9 matches.

```
package main

import (
    "fmt"
)

type HasilPertandingan string

const (
    MENANG HasilPertandingan = "MENANG"
    SERI HasilPertandingan = "SERI"
    KALAH HasilPertandingan = "KALAH"
)

type Klub struct {
    nama string
    skor int
}
```

```
raisauf@Tokyo-Mac 2311182233_Muhammad Amir Saleh_Modul 7 % go run "/Users/raisauf/Desktop/2311182233_Muhammad Amir Saleh_Modul 7/unguided2.go"
Klub A : MU
Klub B : Inter
Pertandingan 1 : 2 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
raisauf@Tokyo-Mac 2311182233_Muhammad Amir Saleh_Modul 7 %
```

## Deskripsi:

Program ini mencatat hasil pertandingan antara dua klub sepak bola, dengan menggunakan tipe data Klub, RekapPertandingan, dan HasilPertandingan untuk menyimpan informasi tentang klub, skor, serta pemenang dari setiap pertandingan. Program dimulai dengan meminta input nama kedua klub, lalu menyimpan informasi ini dalam objek RekapPertandingan. Selanjutnya, program meminta input skor untuk setiap pertandingan secara berulang, dan menentukan pemenang berdasarkan perbandingan skor. Jika skor kedua klub sama, maka hasil pertandingan akan dicatat sebagai "Draw". Hasil dari setiap pertandingan akan disimpan dalam slice pemenang untuk kemudian ditampilkan setelah seluruh pertandingan selesai. Program akan terus meminta input hingga skor negatif dimasukkan, yang menandakan akhir dari input pertandingan. Setelah itu, program akan menampilkan rekap hasil pertandingan yang mencakup pemenang atau hasil imbang untuk setiap pertandingan yang telah berlangsung.

### Unguided 3

```
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

var tab tabel
var m int

func isiArray(t *tabel, n *int) {
    var input string
    fmt.Scan(&input)
    *n = len(input)

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

func cetakArray(t tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Printf("%c", t[i])
    }
    fmt.Println()
}

func balikanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        t[i], t[n-i-1] = t[n-i-1], t[i]
    }
}

func palindrom(t tabel, n int) bool {
    var tbalikan tabel
    copy(tbalikan[:], t[:])

    balikanArray(&tbalikan, n)

    for i := 0; i < n; i++ {
        if t[i] != tbalikan[i] {
            return false
        }
    }
}
```

```

    return true
}

func main() {
    fmt.Print("Teks: ")
    isiArray(&tab, &m)

    fmt.Print("Reverse teks: ")
    balikanArray(&tab, m)
    cetakArray(tab, m)

    fmt.Print("Palindrom? ")
    fmt.Println(palindrom(tab, m))
}

```

## Screenshots Output

```

1 package main
2
3 import "fmt"
4
5 const NMAX int = 127
6
7 type tabel [NMAX]rune
8
9 var tab tabel
10 var m int
11
12 func isiArray(t *tabel, n *int) {
13     var input string
14     fmt.Scan(&input)
15     *n = len(input)
16
17     for i := 0; i < *n; i++ {

```

```

raisauf@Tokyo-Mac 2311182233_Muhammad Amir Saleh_Modul 7 % go run "/Users/raisauf/Desktop/2311182233_Muhammad Amir Saleh_Modul 7/unguided3.go"
Teks: QWNE
Reverse teks: QWNE
Palindrom? false
raisauf@Tokyo-Mac 2311182233_Muhammad Amir Saleh_Modul 7 % go run "/Users/raisauf/Desktop/2311182233_Muhammad Amir Saleh_Modul 7/unguided3.go"
Teks: KATAK
Reverse teks: KATAK
Palindrom? true
raisauf@Tokyo-Mac 2311182233_Muhammad Amir Saleh_Modul 7 %

```

## Deskripsi:

Program ini digunakan untuk memeriksa apakah sebuah string adalah palindrom dan menampilkan teks yang dibalik. Fungsi `isiArray` membaca input berupa string dan menyimpannya dalam array `tab`, serta menghitung panjang string tersebut. Fungsi `cetakArray` digunakan untuk menampilkan array rune dalam bentuk string tanpa pemisah antar karakter. Fungsi `balikanArray` membalikkan urutan elemen dalam array hingga mencapai setengah panjang array. Fungsi `palindrom` memeriksa apakah string yang diberikan merupakan palindrom dengan membandingkan array asli dengan array yang sudah dibalik.