

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7  
STRUCT AND ARRAY**



**Disusun Oleh :  
MAULANA GHANI ROLANDA | 2311102012  
IF-11-06**

**Dosen Pengampu :  
Abednego Dwi Septiadi S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS 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. Tipe bentuk ini dapat dibedakan atas dua jenis, yaitu Alias dan Struct.

#### 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 "bllangan ". Caranya dengan menggunakan kata kunci 'type'.

#### 2) Struct atau Record

Stucture memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari stucture tersebut. Berbeda dengan bahasa pemrograman lain. kesamaan tipe dari dua variabel berjenis structure 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 stucture tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi stucture tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

### 7.2 Array

Array mempunyai ukuran (jumlah elemen) yang tetap (statls) selama eksekusi program, sehingga jumlah elemen array menjadi bagian dari deklarasi variabel dengan tipe array. Jumlah elemen array dapat diminta dengan fungsi len yang tersedia. Sebagai contoh len(arr) akan menghasilkan 73 untuk contoh di atas.

#### Slice (Array dlnamIk)

Array dalam Go juga dapat mempunyai ukuran yang dinamik. (Tidak dIgunakan dI ketas Algoritma Pemrograman). Deklarasinya mirip dengan deklarasi array, tetapi jumlah elemennya dikosongkan. Sebuah slice dapat diprealokasi menggunakan fungsi built-in make. Fungsi built-in len dapat digunakan untuk mengetahui ukuran slice. Fungsi lain, cap, dapat digunakan untuk mengetahui total tempat yang disediakan untuk slice

tersebut. Fungsi built-in append dapat digunakan untuk menambahkan elemen ke suatu slice, dan bila perlu memperbesar tempat untuk slice tersebut. Sebuah slice baru juga dapat terbentuk dengan mengambil slice dari suatu array atau slice yang lain.

### 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

### 1. Sourcecode

```
// 2311102012
package main

import (
    "fmt"
    "math"
)

type titik struct {
    x int
    y int
}

type lingkaran struct {
    pusat titik
    radius int
}

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)))
}

func titikdalamlingkaran(t titik, l lingkaran) bool {
    jarak := hitungjarak(t, l.pusat)
    return jarak <= float64(l.radius)
}

func main() {
    var cx1, cy1, r1 int
    fmt.Print("Masukkan kordinat pusat dan radius lingkaran
1 (cx1 cy1 r1): ")
    fmt.Scanln(&cx1, &cy1, &r1)

    // Membuat objek lingkaran1
    lingkaran1 := lingkaran{pusat: titik{x: cx1, y: cy1},
radius: r1}

    var cx2, cy2, r2 int
    fmt.Print("Masukkan kordinat pusat dan radius lingkaran
2 (cx2 cy2 r2): ")
    fmt.Scanln(&cx2, &cy2, &r2)
```

```

    // Membuat objek lingkaran2
    lingkaran2 := lingkaran{pusat: titik{x: cx2, y: cy2},
radius: r2}

    var x, y int
    fmt.Print("Masukkan kordinat titik t (x y): ")
    fmt.Scanln(&x, &y)

    // Membuat objek titik yang akan diperiksa
    titik2 := titik{x: x, y: y}

    // Memeriksa apakah titik berada di dalam lingkaran1 dan
    lingkaran2
    didalam1 := titikdalamlingkaran(titik2, lingkaran1)
    didalam2 := titikdalamlingkaran(titik2, lingkaran2)

    if didalam1 && didalam2 {
        fmt.Println("Titik berada di dalam kedua
    lingkaran.")
    } else if didalam1 {
        fmt.Println("Titik berada di dalam lingkaran 1
    saja.")
    } else if didalam2 {
        fmt.Println("Titik berada di dalam lingkaran 2
    saja.")
    } else {
        fmt.Println("Titik berada di luar kedua lingkaran.")
    }
}

```

## Screenshoot Output

```

PS C:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6> go run "c:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6\guided1.go"
Masukkan kordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): 1 1 5
Masukkan kordinat pusat dan radius lingkaran 2 (cx2 cy2 r2): 8 8 4
Masukkan kordinat titik t (x y): 2 2
Titik berada di dalam lingkaran 1 saja.
PS C:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6>

```

## Deskripsi Program

Program ini memeriksa apakah sebuah titik berada di dalam satu atau dua lingkaran berdasarkan koordinat dan radiusnya. Dengan menggunakan

struct untuk merepresentasikan titik dan lingkaran, program menghitung jarak titik ke pusat lingkaran menggunakan rumus Euclidean. Kemudian, program membandingkan jarak tersebut dengan radius lingkaran untuk menentukan posisi titik. Setelah menerima input dari pengguna, program menghasilkan output berupa posisi titik: apakah di dalam kedua lingkaran, salah satu lingkaran, atau di luar keduanya.

### III. UNGUIDED

#### 1. Sourcecode

```
//2311102012
package main

import (
    "fmt"
    "math"
)

func tampilSemua(array []int) {
    fmt.Println("Isi array:", array)
}

func tampilGanjil(array []int) {
    fmt.Print("Elemen dengan indeks ganjil: ")
    for i := 1; i < len(array); i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}

func tampilGenap(array []int) {
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 0; i < len(array); i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}

func tampilKelipatan(array []int, x int) {
    fmt.Printf("Elemen dengan indeks kelipatan %d: ", x)
    for i := x; i < len(array); i += x {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}

func hapusIndeks(array []int, indeks int) []int {
    fmt.Printf("Menghapus elemen pada indeks %d\n", indeks)
    return append(array[:indeks], array[indeks+1:]...)
}

func rataRata(array []int) float64 {
```

```

    total := 0
    for _, val := range array {
        total += val
    }
    return float64(total) / float64(len(array))
}

func standarDeviasi(array []int) float64 {
    mean := rataRata(array)
    var sum float64
    for _, val := range array {
        sum += math.Pow(float64(val)-mean, 2)
    }
    return math.Sqrt(sum / float64(len(array)))
}

func frekuensi(array []int, nilai int) int {
    count := 0
    for _, val := range array {
        if val == nilai {
            count++
        }
    }
    return count
}

func main() {
    var N, x, hapusIdx, cariFrekuensi int

    fmt.Print("Masukkan jumlah elemen array (N): ")
    fmt.Scan(&N)

    array := make([]int, N)

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

    tampilSemua(array)
    tampilGanjil(array)
    tampilGenap(array)
}

```



```

    fmt.Print("Masukkan nilai x untuk menampilkan elemen
dengan indeks kelipatan x: ")
    fmt.Scan(&x)
    tampilKelipatan(array, x)

    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&hapusIdx)
    if hapusIdx >= 0 && hapusIdx < len(array) {
        array = hapusIndeks(array, hapusIdx)
        tampilSemua(array)
    } else {
        fmt.Println("Indeks yang dimasukkan tidak valid!")
    }

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

    fmt.Printf("Standar deviasi elemen array: %.2f\n",
standarDeviasi(array))

    fmt.Print("Masukkan bilangan yang ingin dicari
frekuensinya: ")
    fmt.Scan(&cariFrekuensi)
    fmt.Printf("Frekuensi %d di dalam array: %d\n",
cariFrekuensi, frekuensi(array, cariFrekuensi))
}

```

## Screenshoot Output

```

PS C:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6> go run "c:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6\unguided1.go"
Masukkan jumlah elemen array (N): 4
Masukkan elemen array:
Elemen ke-0: 1
Elemen ke-1: 2
Elemen ke-2: 3
Elemen ke-3: 4
Isi array: [1 2 3 4]
Elemen dengan indeks ganjil: 2 4
Elemen dengan indeks genap: 1 3
Masukkan nilai x untuk menampilkan elemen dengan indeks kelipatan x: 5
Elemen dengan indeks kelipatan 5:
Masukkan indeks yang ingin dihapus: 3
Menghapus elemen pada indeks 3
Isi array: [1 2 3]
Rata-rata elemen array: 2.00
Standar deviasi elemen array: 0.82
Masukkan bilangan yang ingin dicari frekuensinya: 4
Frekuensi 4 di dalam array: 0
PS C:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6>

```

## Deskripsi Program

Program ini memberikan fungsionalitas dasar untuk manipulasi array, seperti menampilkan elemen berdasarkan kriteria tertentu (genap, ganjil,

kelipatan), menghitung statistik dasar (rata-rata dan standar deviasi), serta memungkinkan pengguna untuk menghapus elemen dari array dan menghitung frekuensinya.

Program ini juga cukup interaktif, dengan meminta input dari pengguna untuk berbagai operasi yang dilakukan pada array.

## 2. Source Code

```
// 2311102012
package main

import (
    "fmt"
)

func main() {
    var clubA, clubB string
    var scoreA, scoreB int
    var winners []string

    // Meminta nama klub
    fmt.Print("Klub A: ")
    fmt.Scanln(&clubA)
    fmt.Print("Klub B: ")
    fmt.Scanln(&clubB)

    // Memasukkan skor pertandingan
    for i := 1; ; i++ {
        fmt.Printf("Pertandingan %d (skor %s): ", i, clubA)
        fmt.Scanln(&scoreA)
        fmt.Printf("Pertandingan %d (skor %s): ", i, clubB)
        fmt.Scanln(&scoreB)

        // Memeriksa apakah skor valid
        if scoreA < 0 || scoreB < 0 {
            fmt.Println("Input skor tidak valid. Program selesai.")
            break
        }

        // Menentukan pemenang
        if scoreA > scoreB {
```

```

        winners = append(winners, clubA)
    } else if scoreA < scoreB {
        winners = append(winners, clubB)
    } else {
        winners = append(winners, "Draw")
    }
}

// Menampilkan hasil
fmt.Println("\nHasil pertandingan:")
for i, winner := range winners {
    fmt.Printf("Pertandingan %d: %s\n", i+1, winner)
}
}

```

### Screenshot Output

```

PS C:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6> go run "c:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6\unguided2.go"
Klub A: MU
Klub B: PSG
Pertandingan 1 (skor MU): 0
Pertandingan 1 (skor PSG): 10
Pertandingan 2 (skor MU): 0
Pertandingan 2 (skor PSG): 5
Pertandingan 3 (skor MU): 1
Pertandingan 3 (skor PSG): 15
Pertandingan 4 (skor MU): -1
Pertandingan 4 (skor PSG): -1
Input skor tidak valid. Program selesai.

Hasil pertandingan:
Pertandingan 1: PSG
Pertandingan 2: PSG
Pertandingan 3: PSG
PS C:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6>

```

### Deskripsi Program

Program ini bertujuan untuk mencatat hasil pertandingan sepak bola antara dua klub dan menampilkan daftar pemenang dari setiap pertandingan yang valid. Pada awalnya, program meminta pengguna untuk memasukkan nama dua klub yang bertanding, yaitu Klub A dan Klub B. Nama-nama ini digunakan untuk mencatat hasil pertandingan

### 3. Source Code

```
//2311102012
package main

import (
    "fmt"
)

const NMAX = 127

type tabel [NMAX]rune

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

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

// Prosedur untuk mencetak isi array
func cetakArray(t tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Print(string(t[i]))
    }
    fmt.Println()
}

// Prosedur untuk membalik isi array
func balikArray(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 membentuk palindrome
func palindrom(t tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t[i] != t[n-i-1] {
            return false
        }
    }
    return true
}
```

```

    }
}
return true
}

func main() {
    var tab tabel
    var n int

    // Memanggil prosedur untuk mengisi array
    isiArray(&tab, &n)

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

    // Membalik isi array
    balikArray(&tab, n)

    // Menampilkan isi array yang sudah dibalik
    fmt.Print("Reverse: ")
    cetakArray(tab, n)

    // Memeriksa apakah array membentuk palindrome
    if palindrom(tab, n) {
        fmt.Println("Palindrom")
    } else {
        fmt.Println("Bukan Palindrom")
    }
}

```

### Screenshot Program

```

PS C:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6> go run "c:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6\unguided3.go"
Teks: ketek
Teks: ketek
Reverse: ketek
Palindrom
PS C:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6> go run "c:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6\unguided3.go"
Teks: agus
Teks: agus
Reverse: suga
Bukan Palindrom
PS C:\olan\KULYEAH\LaPrak Alpro 2\2311102012-MaulanaGhaniRolanda-M6>

```

### Deskripsi Program

Program ini sederhana tetapi efektif untuk memproses string dan memeriksa sifat palindrome. Dengan memanfaatkan array dan beberapa prosedur sederhana, program mampu membalik string dan mengevaluasi karakter-karakter dalam array untuk menentukan apakah string memenuhi

syarat sebagai palindrome. Program ini cocok digunakan untuk mempelajari algoritma dasar manipulasi string dan validasi palindrome.