

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7
STRUCK DAN ARRAY**



Disusun Oleh :

HANIF REYHAN ZHAFRAN ADRYTONA / 2311102266

11 IF 06

Dosen Pengampu :

Abednego Dwi Septiadi, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dalam bahasa pemrograman Go, struct adalah tipe data yang digunakan untuk mengelompokkan beberapa nilai dengan tipe yang berbeda dalam satu kesatuan. Struct sering digunakan untuk merepresentasikan entitas dengan atribut-atributnya, seperti objek dalam paradigma pemrograman berorientasi objek. Struct didefinisikan menggunakan kata kunci ``type`` dan ``struct``. Setiap atribut dalam struct disebut field, yang dapat memiliki tipe data berbeda. Sebagai contoh, sebuah struct bernama ``Person`` dapat memiliki field seperti ``Name``, ``Age``, dan ``Email``, masing-masing dengan tipe string atau integer.

Sementara itu, array adalah struktur data yang menyimpan sejumlah elemen dengan tipe yang sama dan ukuran tetap. Elemen dalam array diakses menggunakan indeks yang dimulai dari nol. Deklarasi array di Go melibatkan penentuan ukuran dan tipe elemen yang akan disimpan. Misalnya, ``var arr [5]int`` akan mendeklarasikan sebuah array dengan lima elemen bertipe integer. Karena ukurannya tetap, array di Go bersifat efisien, tetapi kurang fleksibel dibandingkan tipe data lain seperti slice. Operasi seperti inisialisasi, pengubahan nilai, dan iterasi dilakukan dengan sintaks yang sederhana.

Struct dan array dapat digabungkan untuk merepresentasikan data yang lebih kompleks. Sebagai contoh, sebuah struct bernama ``Student`` dapat memiliki field berupa array untuk menyimpan nilai-nilai ujian. Hal ini memungkinkan data yang saling terkait dikelompokkan secara logis dalam satu entitas. Dengan penggabungan ini, pengelolaan data menjadi lebih terorganisir dan memudahkan dalam implementasi logika program, terutama dalam aplikasi yang membutuhkan representasi data yang terstruktur.

II. GUIDED

Guided 1.0

Sourcecode

```
// 2311102266_Hanif Reyhan Zhafran Arytona

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 titikDiDalamLingkaran(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 koordinat pusat dan radius lingkaran
1 (cx1 cy1 r1): ")
    fmt.Scanln(&cx1, &cy1, &r1)
    Lingkaran1 := Lingkaran{pusat: Titik{x: cx1, y: cy1},
radius: r1}

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

```

    fmt.Scanln(&cx2, &cy2, &r2)
    Lingkaran2 := Lingkaran{pusat: Titik{x: cx2, y: cy2},
radius: r2}

    var x, y int
    fmt.Print("Masukkan koordinat titik sembarang (x y): ")
    fmt.Scanln(&x, &y)
    titik := Titik{x: x, y: y}

    diDalam1 := titikDiDalamLingkaran(titik, Lingkaran1)
    diDalam2 := titikDiDalamLingkaran(titik, Lingkaran2)

    if diDalam1 && diDalam2 {
        fmt.Println("Titik berada di dalam lingkaran 1 dan
2")
    } else if diDalam1 {
        fmt.Println("Titik berada di dalam lingkaran 1")
    } else if diDalam2 {
        fmt.Println("Titik berada di dalam lingkaran 2")
    } else {
        fmt.Println("Titik berada di luar kedua lingkaran")
    }
}

```

Screenshoot Output

```

PROBLEMS 1 OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CONSOLE

PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 7\GUIDED.1.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): 2 3 2
Masukkan koordinat pusat dan radius lingkaran 2 (cx2 cy2 r2): 2 5 6
Masukkan koordinat titik sembarang (x y): 1 3
Titik berada di dalam lingkaran 1 dan 2
PS C:\Users\M S I>

```

Deskripsi Program

Program di atas adalah implementasi dalam bahasa Go yang memeriksa apakah suatu titik berada di dalam satu atau dua lingkaran, atau di luar keduanya. Program ini memanfaatkan dua struktur, yaitu `Titik` untuk merepresentasikan koordinat (x, y) dan `Lingkaran` untuk mendefinisikan lingkaran dengan pusat (sebagai `Titik`) dan radius. Program ini menggunakan fungsi `hitungJarak` untuk menghitung jarak antara dua titik menggunakan rumus jarak Euclidean, dan fungsi `titikDiDalamLingkaran` untuk

memeriksa apakah jarak antara suatu titik dengan pusat lingkaran lebih kecil atau sama dengan radius lingkaran tersebut.

Dalam eksekusi, program meminta pengguna untuk memasukkan koordinat pusat dan radius dari dua lingkaran, serta koordinat dari sebuah titik sembarang. Program kemudian mengevaluasi posisi titik terhadap kedua lingkaran dengan memanggil fungsi `titikDiDalamLingkaran` untuk masing-masing lingkaran. Berdasarkan hasil pengecekan, program menampilkan pesan apakah titik tersebut berada di dalam lingkaran pertama, lingkaran kedua, kedua lingkaran, atau di luar keduanya. Program ini memanfaatkan logika sederhana dengan operator logika untuk menentukan hasil akhir.

III. UNGUIDED

Sourcecode

```
// 2311102266_Hanif Reyhan Zhafran Arytona

package main

import (
    "fmt"
    "math"
)

func main() {
    var n, x, delIndex, target int

    fmt.Print("Masukkan jumlah elemen array (N): ")
    fmt.Scan(&n)
    if n <= 0 {
        fmt.Println("Jumlah elemen array harus lebih besar dari 0.")
        return
    }

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

    fmt.Println("\nIsi array lengkap:", array)

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

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

    fmt.Print("\nMasukkan nilai x untuk kelipatan indeks: ")
    fmt.Scan(&x)
```

```

if x <= 0 {
    fmt.Println("Nilai x harus lebih besar dari 0.")
    return
}
fmt.Printf("\nElemen dengan indeks kelipatan %d:\n", x)
for i := 0; i < n; i++ {
    if i%x == 0 {
        fmt.Printf("Index %d: %d\n", i, array[i])
    }
}

fmt.Print("\nMasukkan indeks yang akan dihapus: ")
fmt.Scan(&delIndex)
if delIndex >= 0 && delIndex < len(array) {
    array = append(array[:delIndex],
array[delIndex+1:]...)
    fmt.Println("Isi array setelah penghapusan:", array)
} else {
    fmt.Println("Indeks tidak valid.")
    return
}

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

var varianceSum float64
for _, v := range array {
    varianceSum += math.Pow(float64(v)-rataRata, 2)
}
stdDeviasi := math.Sqrt(varianceSum /
float64(len(array)))
fmt.Printf("Standar deviasi elemen array: %.2f\n",
stdDeviasi)

fmt.Print("\nMasukkan bilangan untuk mengetahui
frekuensinya: ")
fmt.Scan(&target)
count := 0
for _, v := range array {
    if v == target {
        count++
    }
}

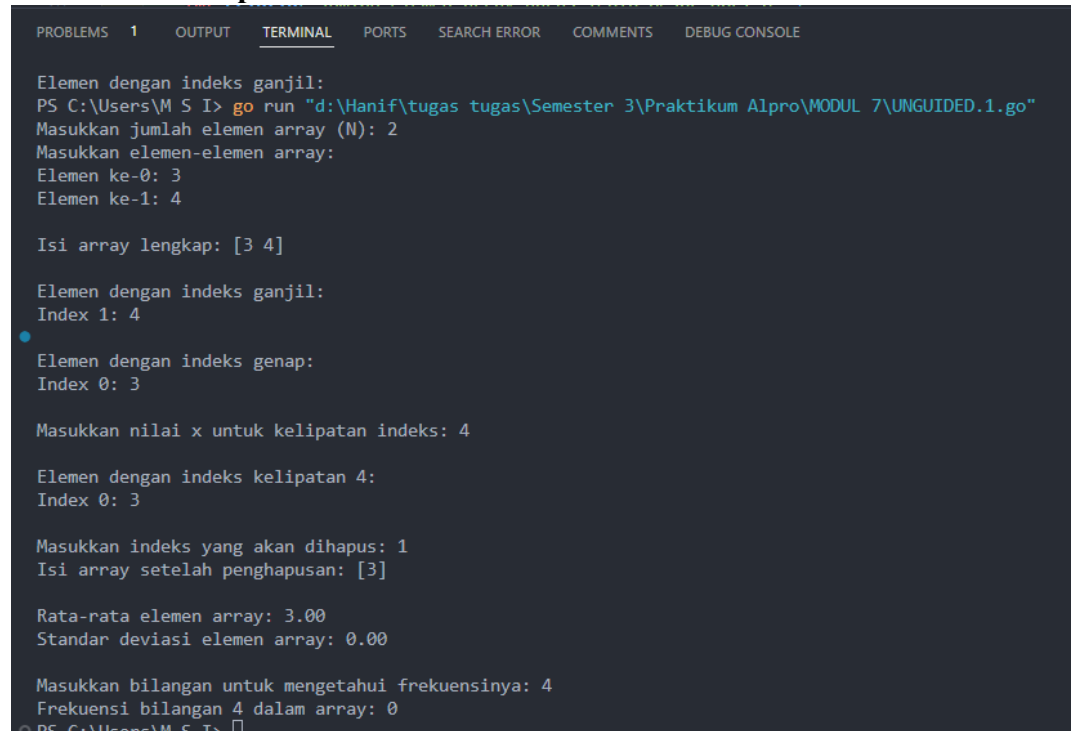
```

```

    }
}
fmt.Printf("Frekuensi bilangan %d dalam array: %d\n",
target, count)
}

```

Screenshoot Output



```

PROBLEMS 1 OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CONSOLE

Elemen dengan indeks ganjil:
PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 7\UNGUIDED.1.go"
Masukkan jumlah elemen array (N): 2
Masukkan elemen-elemen array:
Elemen ke-0: 3
Elemen ke-1: 4

Isi array lengkap: [3 4]

Elemen dengan indeks ganjil:
Index 1: 4

Elemen dengan indeks genap:
Index 0: 3

Masukkan nilai x untuk kelipatan indeks: 4

Elemen dengan indeks kelipatan 4:
Index 0: 3

Masukkan indeks yang akan dihapus: 1
Isi array setelah penghapusan: [3]

Rata-rata elemen array: 3.00
Standar deviasi elemen array: 0.00

Masukkan bilangan untuk mengetahui frekuensinya: 4
Frekuensi bilangan 4 dalam array: 0
PS C:\Users\M S I>

```

Deskripsi Program

Program di atas adalah sebuah program dalam bahasa Go yang berfungsi untuk melakukan berbagai operasi pada array integer, seperti menampilkan elemen berdasarkan kondisi tertentu, menghitung statistik, dan memodifikasi isi array. Program dimulai dengan meminta pengguna untuk menentukan jumlah elemen array dan memasukkan nilai-nilai array satu per satu. Selanjutnya, array ditampilkan secara keseluruhan, diikuti oleh elemen-elemen dengan indeks genap, ganjil, serta indeks yang merupakan kelipatan dari bilangan tertentu. Program juga mencakup validasi input untuk memastikan jumlah elemen array dan nilai kelipatan indeks adalah valid.

Program ini menawarkan fitur tambahan seperti menghapus elemen pada indeks tertentu, menghitung rata-rata, dan menghitung standar deviasi elemen array. Pengguna juga dapat memasukkan bilangan tertentu untuk mengetahui frekuensinya dalam array. Berkat validasi input dan format output yang terstruktur, program ini tidak hanya mudah digunakan tetapi juga mencegah kesalahan umum, seperti indeks yang di luar batas atau pembagian dengan nol. Fungsionalitas ini menunjukkan kemampuan dasar pengolahan array dan penerapan logika pemrograman yang efektif dalam bahasa Go.

Unguided 2

Sourcecode

```
// 2311102266_Hanif Reyhan Zhafran Arytona

package main

import (
    "fmt"
)

func main() {
    var klubA, klubB string
    var skorA, skorB int
    var pemenang []string

    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&klubA)
    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scanln(&klubB)

    for i := 1; i <= 3; i++ {
        fmt.Printf("Pertandingan %d - Masukkan skor %s: ", i, klubA)
        fmt.Scan(&skorA)
        fmt.Printf("Pertandingan %d - Masukkan skor %s: ", i, klubB)
        fmt.Scan(&skorB)

        if skorA < 0 || skorB < 0 {
            fmt.Println("Skor tidak valid. Pertandingan dibatalkan.")
            break
        }
    }
}
```

```

        if skorA > skorB {
            pemenang = append(pemenang, klubA)
            fmt.Printf("Hasil pertandingan %d: %s menang\n", i,
klubA)
        } else if skorB > skorA {
            pemenang = append(pemenang, klubB)
            fmt.Printf("Hasil pertandingan %d: %s menang\n", i,
klubB)
        } else {
            fmt.Printf("Hasil pertandingan %d: Seri (Draw)\n", i)
        }
    }

    if len(pemenang) > 0 {
        fmt.Println("\nDaftar klub yang memenangkan pertandingan:")
        for _, klub := range pemenang {
            fmt.Println(klub)
        }
    } else {
        fmt.Println("\nTidak ada pemenang dalam pertandingan.")
    }
}

```

Screenshoot Output

```

PROBLEMS 1 OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CONSOLE
PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 7\UNGUIDED.2.go"
Masukkan nama Klub A: inter
Masukkan nama Klub B: mu
Pertandingan 1 - Masukkan skor inter: 1
Pertandingan 1 - Masukkan skor mu: 9
Hasil pertandingan 1: mu menang
Pertandingan 2 - Masukkan skor inter: 2
Pertandingan 2 - Masukkan skor mu: 3
Hasil pertandingan 2: mu menang
Pertandingan 3 - Masukkan skor inter: 2
Pertandingan 3 - Masukkan skor mu: 2
Hasil pertandingan 3: Seri (Draw)

Daftar klub yang memenangkan pertandingan:
mu
mu
PS C:\Users\M S I>

```

Deskripsi Program

Program di atas merupakan program dengan bahasa go yang digunakan untuk mencatat dan menilai hasil dari tiga pertandingan antara dua klub sepak bola.

Pengguna diminta untuk memasukkan nama kedua klub terlebih dahulu, diikuti dengan skor masing-masing klub untuk setiap pertandingan. Program ini dirancang untuk menerima skor yang valid (angka positif atau nol) dan memberikan hasil pertandingan berdasarkan skor yang lebih tinggi, baik itu kemenangan Klub A, kemenangan Klub B, atau hasil seri.

Setelah masing-masing pertandingan, program mencatat pemenang (jika ada) ke dalam daftar untuk ditampilkan di akhir. Jika skor negatif dimasukkan, program akan membatalkan pertandingan tersebut dan menghentikan proses penilaian dengan pesan yang jelas. Pembatasan pada tiga pertandingan dilakukan melalui perulangan dengan batas tetap (`for i := 1; i <= 3; i++`), sehingga program berjalan dalam jumlah iterasi yang pasti dan terstruktur.

Pada akhir program, daftar pemenang dari ketiga pertandingan akan ditampilkan, atau pesan bahwa tidak ada pemenang jika semua pertandingan seri atau dibatalkan. Dengan struktur kode yang sederhana, program ini cocok untuk mendemonstrasikan pengelolaan input pengguna, logika pemrosesan hasil pertandingan, serta validasi dasar untuk memastikan skor yang dimasukkan sesuai dengan aturan.

UNGUIDED 3

Sourcecode

```
// 2311102266_Hanif Reyhan Zhafran Arytona

package main

import (
    "fmt"
)

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var ch rune
    fmt.Println("Masukkan teks (akhiri dengan TITIK):")
    *n = 0
    for {
        fmt.Scanf("%c", &ch)
        if ch == '.' || *n >= NMAX {
            break
        }
    }
}
```

```

    }
    t[*n] = ch
    *n++
}
}

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 reversed tabel
    copy(reversed[:n], t[:n])
    balikanArray(&reversed, n)
    for i := 0; i < n; i++ {
        if t[i] != reversed[i] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var m int

    isiArray(&tab, &m)

    fmt.Print("Teks: ")
    cetakArray(tab, m)

    var reversed tabel
    copy(reversed[:m], tab[:m])
    balikanArray(&reversed, m)
    fmt.Print("Reverse teks: ")
    cetakArray(reversed, m)
}

```

```

    if palindrom(tab, m) {
        fmt.Println("Palindrom? true")
    } else {
        fmt.Println("Palindrom? false")
    }
}

```

Screenshoot Output

```

PROBLEMS 1 OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CONSOLE
PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 7\UNGUIDED.3.go"
Masukkan teks (akhiri dengan TITIK):
s e n a n g .
Teks: s e n a n g
Reverse teks: g n a n e s
Palindrom? false
PS C:\Users\M S I>

```

Deskripsi Program

Program ini adalah sebuah program dalam bahasa go yang memungkinkan pengguna untuk memasukkan teks, membalikkan urutannya, dan memeriksa apakah teks tersebut membentuk sebuah palindrom (teks yang dibaca sama dari depan dan belakang). Pengguna diminta untuk memasukkan teks yang diakhiri dengan titik (.), dan teks tersebut kemudian disalin ke dalam array `tabel` dengan ukuran maksimum 127 karakter. Setelah itu, program menampilkan teks asli, membalikkan urutannya, dan mencetak hasilnya. Proses pembalikan dilakukan dengan algoritma penukaran elemen dalam array, yang kemudian dicetak kembali setelah dibalik.

Program ini juga memeriksa apakah teks yang dimasukkan adalah palindrom. Setelah membalikkan array, program membandingkan teks asli dengan teks yang dibalik. Jika keduanya sama, maka teks tersebut dianggap sebagai palindrom dan program akan menampilkan "Palindrom? true". Jika tidak, maka hasilnya adalah "Palindrom? false". Program ini mengilustrasikan penggunaan array, manipulasi string, dan algoritma pembalikan dalam bahasa Go, memberikan pemahaman dasar tentang cara mengolah teks dalam array dan memeriksa properti seperti palindrom.