

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL VII
STRUCK & ARRAY



Oleh:

NAMA : HAIKAL SATRIATAMA

NIM : 2311102066

KELAS : IF 11 02

S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

I. DASAR TEORI

Tipe Bentukan Tipe bentukan memungkinkan pembuatan tipe data baru dalam bahasa pemrograman. Tipe ini terbagi menjadi dua jenis:

1. **Alias (Type):** Pemrograman dapat mengubah nama tipe data menjadi nama baru yang lebih singkat dan familiar menggunakan kata kunci "type". Misalnya, "Integer" bisa diubah menjadi "bilangan".
2. **Struct atau Record:** Struct memungkinkan pengelompokan beberapa data yang memiliki keterkaitan menjadi satu kesatuan. Tipe kesamaan dua variabel struct ditentukan oleh strukturnya, bukan namanya, berdasarkan nama field dan tipe field yang sama serta urutan yang sama. Hal ini memudahkan dengan mendefinisikan struct sebagai tipe baru agar deklarasinya tidak perlu diulang.

Array: Array memiliki ukuran tetap (statis) selama eksekusi program, dan jumlah elemen menjadi bagian dari deklarasi variabel tipe array. Ukuran array dapat diketahui menggunakan fungsi `len`.

Slice (Array Dinamis): Di Go, slice memungkinkan ukuran dinamis. Deklarasinya mirip dengan array, tetapi tanpa jumlah elemen yang ditentukan. Fungsi `len` digunakan untuk mengetahui ukuran slice, `cap` untuk kapasitas totalnya, dan `append` untuk menambahkan elemen. Slice baru dapat dibuat dengan mengambil sebagian dari array atau slice lainnya.

Map: Map adalah struktur array dinamis di mana indeks (kunci) tidak harus berupa integer, tetapi dapat berasal dari tipe data lainnya.

II. GUIDED

NO 1.

Source code:

```
package main

import (
    "fmt"
    "sort"
)

type mahasiswa struct {
    nama      string
    matematika int
    fisika    int
    kimia     int
}
```

```

rata_rata float64
}

func HitungRata(n *mahasiswa) {
    n.rata_rata = float64(n.matematika+n.fisika+n.kimia) / float64(3)
}

func main() {
    mahasiswa := []mahasiswa{
        {"Ali", 85, 90, 80, 0},
        {"Budi", 70, 75, 80, 0},
        {"Cici", 90, 85, 95, 0},
        {"Doni", 60, 65, 70, 0},
        {"Eka", 100, 95, 90, 0},
    }

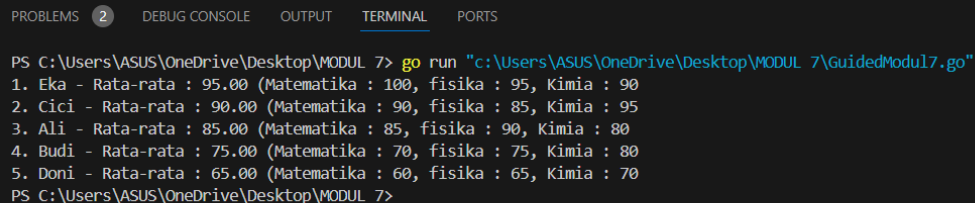
    for i := range mahasiswa {
        HitungRata(&mahasiswa[i])
    }

    sort.Slice(mahasiswa, func(i, j int) bool {
        return mahasiswa[i].rata_rata > mahasiswa[j].rata_rata
    })

    for i := range mahasiswa {
        fmt.Printf("%d. %s - Rata-rata : %.2f (Matematika : %d, fisika : %d, Kimia : %d)\n", i+1, mahasiswa[i].nama, mahasiswa[i].rata_rata, mahasiswa[i].matematika, mahasiswa[i].fisika, mahasiswa[i].kimia)
    }
}

```

Output:



```

PROBLEMS 2 DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7> go run "c:\Users\ASUS\OneDrive\Desktop\MODUL 7\GuidedModul7.go"
1. Eka - Rata-rata : 95.00 (Matematika : 100, fisika : 95, Kimia : 90)
2. Cici - Rata-rata : 90.00 (Matematika : 90, fisika : 85, Kimia : 95)
3. Ali - Rata-rata : 85.00 (Matematika : 85, fisika : 90, Kimia : 80)
4. Budi - Rata-rata : 75.00 (Matematika : 70, fisika : 75, Kimia : 80)
5. Doni - Rata-rata : 65.00 (Matematika : 60, fisika : 65, Kimia : 70)
PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7>

```

Penjelasan:

menghitung rata-rata nilai dari tiga mata pelajaran untuk beberapa mahasiswa dan mengurutkannya dari yang tertinggi hingga terendah. Program ini membuat daftar

mahasiswa dengan rata-rata nilai tertinggi ke terendah dalam tiga mata pelajaran (matematika, fisika, kimia) dan menampilkannya.

NO 2.

Source code:

```
package main

import "fmt"

func main() {
    // Membuat map dengan NIM sebagai kunci dan Nama sebagai nilai
    mahasiswa := map[string]string{
        "20231001": "Andi",
        "20231002": "Budi",
        "20231003": "Cici",
    }

    // Menambahkan data baru ke map
    mahasiswa["20231004"] = "Dedi"

    // Menampilkan seluruh isi map dalam format kolom dan baris
    fmt.Println("Daftar Mahasiswa:")
    fmt.Println("NIM\t\tNama")
    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }

    // Mengakses data berdasarkan NIM
    nim := "20231002"
    fmt.Println("\nNama Mahasiswa dengan NIM", nim, "adalah", mahasiswa[nim])

    // Menghapus data berdasarkan NIM
    delete(mahasiswa, "20231003")

    // Menampilkan isi map setelah data dihapus dalam format kolom dan baris
    fmt.Println("\nDaftar Mahasiswa setelah dihapus:")
    fmt.Println("NIM\t\tNama")
    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }
}
```

Output:

```
PROBLEMS 2 DEBUG CONSOLE OUTPUT TERMINAL PORTS

PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7> go run "c:\Users\ASUS\OneDrive\Desktop\MODUL 7\Guided Modul7.go"
Daftar Mahasiswa:
NIM      Nama
-----
20231001  Andi
20231002  Budi
20231003  Cici
20231004  Dedi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM      Nama
```

Penjelasan:

Program ini menggunakan **MAP** untuk menyimpan dan mengelola data mahasiswa dengan NIM sebagai kunci dan nama sebagai nilai. Inti program ini adalah pengelolaan data mahasiswa dengan menambah, mengambil, menghapus, dan menampilkan isi **MAP**.

III. UNGUIDED NO 1.

Source code:

```
package main

import (
    "fmt"
    "math"
)

type Point struct {
    x, y int
}

type Circle struct {
    center Point
    radius int
}
```

```

func distance(p1, p2 Point) float64 {
    return math.Sqrt(float64((p1.x-p2.x)*(p1.x-p2.x) + (p1.y-p2.y)*(p1.y-p2.y)))
}

func isInCircle(c Circle, p Point) bool {
    return distance(c.center, p) <= float64(c.radius)
}

func checkPointPosition(c1, c2 Circle, p Point) string {
    inCircle1 := isInCircle(c1, p)
    inCircle2 := isInCircle(c2, p)

    if inCircle1 && inCircle2 {
        return "Titik berada di dalam lingkaran 1 dan lingkaran 2"
    } else if inCircle1 {
        return "Titik berada di dalam lingkaran 1"
    } else if inCircle2 {
        return "Titik berada di dalam lingkaran 2"
    } else {
        return "Titik berada di luar lingkaran 1 dan lingkaran 2"
    }
}

func main() {
    var circle1, circle2 Circle
    var point Point

    fmt.Println("Masukkan koordinat pusat (x, y) dan radius untuk lingkaran 1:")
    fmt.Scan(&circle1.center.x, &circle1.center.y, &circle1.radius)

    fmt.Println("Masukkan koordinat pusat (x, y) dan radius untuk lingkaran 2:")
    fmt.Scan(&circle2.center.x, &circle2.center.y, &circle2.radius)

    fmt.Println("Masukkan koordinat titik (x, y):")
    fmt.Scan(&point.x, &point.y)

    result := checkPointPosition(circle1, circle2, point)
    fmt.Println(result)
}

```

Output:

```
PROBLEMS 2 DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7> go run "c:\Users\ASUS\OneDrive\Desktop\MODUL 7\UNGUIDEDM7\Unguidedm7.go"
Masukkan koordinat pusat (x, y) dan radius untuk lingkaran 1:
2 2 8
Masukkan koordinat pusat (x, y) dan radius untuk lingkaran 2:
3 3 9
Masukkan koordinat titik (x, y):
1 1
Titik berada di dalam lingkaran 1 dan lingkaran 2
PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7>
```

Penjelasan:

Program ini menentukan apakah sebuah titik berada di dalam, di luar, atau pada salah satu dari dua lingkaran berdasarkan jarak titik dari pusat kedua lingkaran. Program ini menggunakan Go untuk menentukan posisi sebuah titik relatif terhadap dua lingkaran, yaitu apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, atau di luar keduanya.

NO 2.

```
package main

import (
    "fmt"
    "math"
)

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

func displayElementsByIndex(arr []int, start int, step int, description string) {
    fmt.Printf("Elemen dengan indeks %s: ", description)
    for i := start; i < len(arr); i += step {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}

func deleteElementAtIndex(arr *[]int, index int) {
    if index >= 0 && index < len(*arr) {
        *arr = append((*arr)[:index], (*arr)[index+1:]...)
        displayArray(*arr)
    } else {
        fmt.Println("Indeks tidak valid.")
    }
}
```

```

func calculateAverage(arr []int) float64 {
    if len(arr) == 0 {
        return 0
    }
    sum := 0
    for _, value := range arr {
        sum += value
    }
    return float64(sum) / float64(len(arr))
}

func calculateStandardDeviation(arr []int, average float64) float64 {
    if len(arr) == 0 {
        return 0
    }
    var varianceSum float64
    for _, value := range arr {
        varianceSum += math.Pow(float64(value)-average, 2)
    }
    return math.Sqrt(varianceSum / float64(len(arr)))
}

func countFrequency(arr []int, num int) int {
    count := 0
    for _, value := range arr {
        if value == num {
            count++
        }
    }
    return count
}

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

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

    displayArray(arr)

    displayElementsByIndex(arr, 1, 2, "ganjil")
}

```



```

displayElementsByIndex(arr, 0, 2, "genap")

var x int
fmt.Print("Masukkan bilangan x untuk kelipatan indeks: ")
fmt.Scan(&x)
displayElementsByIndex(arr, x, x, fmt.Sprintf("kelipatan %d", x))

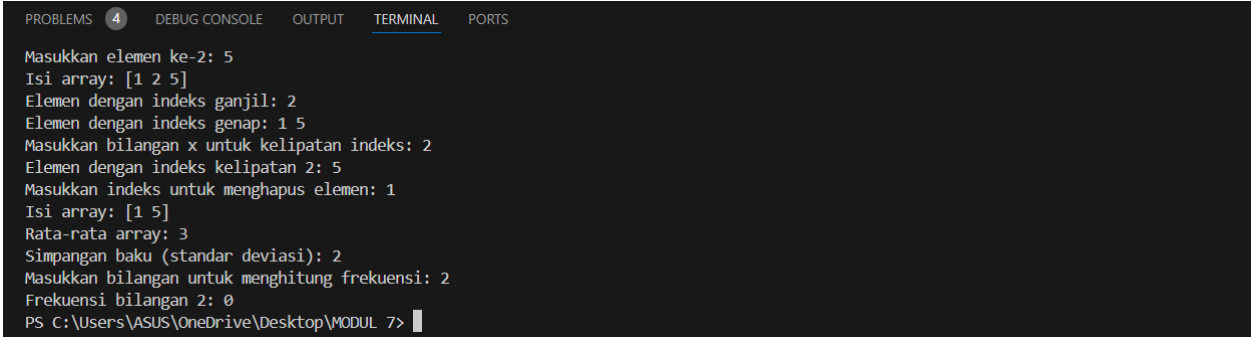
var index int
fmt.Print("Masukkan indeks untuk menghapus elemen: ")
fmt.Scan(&index)
deleteElementAtIndex(&arr, index)

average := calculateAverage(arr)
fmt.Println("Rata-rata array:", average)
stdDev := calculateStandardDeviation(arr, average)
fmt.Println("Simpangan baku (standar deviasi):", stdDev)

var num int
fmt.Print("Masukkan bilangan untuk menghitung frekuensi: ")
fmt.Scan(&num)
fmt.Printf("Frekuensi bilangan %d: %d\n", num, countFrequency(arr,
num))
}

```

Output:



```

PROBLEMS 4 DEBUG CONSOLE OUTPUT TERMINAL PORTS
Masukkan elemen ke-2: 5
Isi array: [1 2 5]
Elemen dengan indeks ganjil: 2
Elemen dengan indeks genap: 1 5
Masukkan bilangan x untuk kelipatan indeks: 2
Elemen dengan indeks kelipatan 2: 5
Masukkan indeks untuk menghapus elemen: 1
Isi array: [1 5]
Rata-rata array: 3
Simpangan baku (standar deviasi): 2
Masukkan bilangan untuk menghitung frekuensi: 2
Frekuensi bilangan 2: 0
PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7>

```

Penjelasan:

Program ini memanfaatkan konsep dasar pemrograman Go, seperti penggunaan slice, fungsi, dan interaksi dengan pengguna melalui input/output. Secara keseluruhan, program ini menunjukkan penggunaan dasar dari array, fungsi, dan interaksi dengan pengguna dalam pemrograman.

NO 3.

```
package main

import (
    "fmt"
)

func main() {
    var klubA, klubB string
    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&klubA)
    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scanln(&klubB)

    var pemenang []string
    var pertandingan int

    for i := 1; ; i++ {
        var skorA, skorB int
        fmt.Printf("Masukkan skor pertandingan %d (%s vs %s): ", i, klubA, klubB)
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            fmt.Println("Skor tidak valid, menghentikan input...")
            pertandingan = i - 1
            break
        }

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

    fmt.Println("Hasil pertandingan:")
    for i := 0; i < pertandingan; i++ {
        fmt.Println(pemenang[i])
    }
    fmt.Println("Pertandingan selesai")
}
```

Output:

```
PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7> go run "c:\Users\ASUS\OneDrive\Desktop\MODUL 7\UNGUIDEDM7\Unguidedm7MO3.go"
Masukkan nama Klub A: MADRID
Masukkan nama Klub B: BARCELONA
Masukkan skor pertandingan 1 (MADRID vs BARCELONA): 2 0
Masukkan skor pertandingan 2 (MADRID vs BARCELONA): 1 2
Masukkan skor pertandingan 3 (MADRID vs BARCELONA): 2 2
Masukkan skor pertandingan 4 (MADRID vs BARCELONA): 0 1
Masukkan skor pertandingan 5 (MADRID vs BARCELONA): 3 2
Masukkan skor pertandingan 6 (MADRID vs BARCELONA): 1 0
Masukkan skor pertandingan 7 (MADRID vs BARCELONA): 5 2
Masukkan skor pertandingan 8 (MADRID vs BARCELONA): 2 3
Masukkan skor pertandingan 9 (MADRID vs BARCELONA): -1 2
Skor tidak valid, menghentikan input...
Hasil pertandingan:
Hasil 1: MADRID menang
Hasil 2: BARCELONA menang
Hasil 3: Draw
Hasil 4: BARCELONA menang
Hasil 5: MADRID menang
Hasil 6: MADRID menang
Hasil 7: MADRID menang
Hasil 8: BARCELONA menang
Pertandingan selesai
PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7> █
```

Penjelasan:

Program ini berfungsi sebagai alat sederhana untuk mencatat dan menampilkan hasil pertandingan antara dua klub. Ini meminta input dari pengguna, melakukan validasi, menyimpan hasil, dan akhirnya menampilkan ringkasan dari semua pertandingan yang telah dicatat. Kode ini cukup efisien untuk tujuan yang dimaksud dan mudah untuk dimodifikasi jika diperlukan.

4o mini.

NO 4.

Source code:

```
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune
```

```

func isiArray(t *tabel, n *int) {
    var input string
    fmt.Scanln(&input)
    *n = 0
    for _, ch := range input {
        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 balikkanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        t[i], t[n-i-1] = t[n-i-1], t[i]
    }
}

func main() {
    var tab tabel
    var n int
    fmt.Print("Teks: ")
    isiArray(&tab, &n)
    balikkanArray(&tab, n)
    fmt.Print("Reverse Teks: ")
    cetakArray(tab, n)
}

```

Output:

```
PROBLEMS 8 DEBUG CONSOLE OUTPUT TERMINAL PORTS

PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7> go run "c:\Users\ASUS\OneDrive\Desktop\MODUL 7\UNGUIDEDM7\Unguidedm7MO4.go"
Teks: REMOVE
Reverse Teks: EVOMER
PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7> |
```

Penjelasan:

Program ini berfungsi untuk menerima teks dari pengguna, membalikkan urutannya, dan menampilkannya. Proses input dihentikan jika pengguna memasukkan karakter titik (.). Kode ini menggunakan array dengan ukuran tetap dan menyusun fungsionalitas ke dalam beberapa fungsi terpisah, menjadikannya lebih teratur dan mudah dimengerti.

MODIFIKASI PROGRAM

Source code:

```
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var input string
    fmt.Print("Masukkan teks (akhiri dengan '.'): ")
    fmt.Scanln(&input)
    *n = 0
    for _, ch := range input {
        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 balikkanArray(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 {
    for i := 0; i < n/2; i++ {
        if t[i] != t[n-i-1] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var n int
    isiArray(&tab, &n)
    fmt.Print("Teks yang dimasukkan: ")
    cetakArray(tab, n)

    if palindrom(tab, n) {
        fmt.Println("Apakah teks ini palindrom? Ya")
    } else {
        fmt.Println("Apakah teks ini palindrom? Tidak")
    }
}

```

Output:

```

PROBLEMS 20 DEBUG CONSOLE OUTPUT TERMINAL PORTS

PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7> go run "c:\Users\ASUS\OneDrive\Desktop\MODUL 7\UNGUIDEDM7\ModUnguidedm7N04.go"
Masukkan teks (akhiri dengan '.'): KATAK
Teks yang dimasukkan: KATAK
Apakah teks ini palindrom? Ya
PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7> go run "c:\Users\ASUS\OneDrive\Desktop\MODUL 7\UNGUIDEDM7\ModUnguidedm7N04.go"
Masukkan teks (akhiri dengan '.'): SENANG
Teks yang dimasukkan: SENANG
Apakah teks ini palindrom? Tidak
PS C:\Users\ASUS\OneDrive\Desktop\MODUL 7> █

```

Penjelasan:

Program ini memungkinkan pengguna untuk memasukkan teks, memeriksa apakah teks tersebut palindrom (sama jika dibaca dari depan atau belakang), dan mencetak hasilnya. Input dihentikan jika pengguna memasukkan karakter titik (.). Struktur program ini modular dengan penggunaan fungsi-fungsi terpisah untuk berbagai tugas, menjadikannya lebih teratur dan mudah dipahami.