

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL VII
STRUCK&ARRAY**



Oleh:

NAMA : AHMAD TITANA NANDA PRAMUDYA

NIM : 2311102042

KELAS : IF 11 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

I. DASAR TEORI

- **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 "bilangan ". Caranya dengan menggunakan kata kunci "type".

- 2) **Struct atau Record**

Structure memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari structure 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 structure tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi structure tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

- **Array**

Array mempunyai ukuran (jumlah elemen) yang tetap (statis) 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.

Slice (Array dinamik)

Array dalam Go juga dapat mempunyai ukuran yang dinamik. (Tidak digunakan di kelas Algoritma Pemrograman). Deklarasinya mirip dengan deklarasi array, tetapi jumlah elemennya dikosongkan.

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

No 1.

Source code:

```
package main

import (
    "fmt"
    "sort"
)

// Struktur untuk menampung data mahasiswa
type Mahasiswa struct {
    Nama      string
    Matematika int
    Fisika    int
    Kimia     int
    RataRata  float64
}

// Fungsi untuk menghitung rata-rata nilai tiap mahasiswa
func hitungRataRata(m *Mahasiswa) {
    total := m.Matematika + m.Fisika + m.Kimia
    m.RataRata = float64(total) / 3.0
}

// Fungsi utama untuk mengelola dan mengurutkan data mahasiswa berdasarkan
// nilai rata-rata
func main() {
    // Array untuk menampung data mahasiswa
    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},
    }

    // Menghitung rata-rata nilai tiap mahasiswa
    for i := range mahasiswa {
        hitungRataRata(&mahasiswa[i])
    }

    // Mengurutkan mahasiswa berdasarkan nilai rata-rata (descending)
    sort.Slice(mahasiswa, func(i, j int) bool {
        return mahasiswa[i].RataRata > mahasiswa[j].RataRata
    })

    // Menampilkan hasil
    fmt.Println("Peringkat mahasiswa berdasarkan rata-rata nilai:")
}
```

```

    for i, m := range mahasiswa {
        fmt.Printf("%d. %s - Rata-rata: %.2f (Matematika: %d, Fisika: %d, Kimia: %d)\n",
            i+1, m>Nama, m.RataRata, m.Matematika, m.Fisika, m.Kimia)
    }
}
//2311102042_Ahmad Titana Nanda Pramudya

```

Output :

```

PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Guided> go run GuidedM7.1.go
Peringkat mahasiswa berdasarkan rata-rata nilai:
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 D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Guided>

```

Penjelasan :

Program ini menghitung rata-rata nilai Matematika, Fisika, dan Kimia tiap mahasiswa, lalu mengurutkan mereka berdasarkan rata-rata nilai secara menurun. Hasil akhirnya menampilkan peringkat mahasiswa berdasarkan rata-rata nilai tertinggi hingga terendah.

NO.2

Sourcode :

```

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\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)
}
}

//2311102042_Ahmad Titana Nanda Pramudya
```

Output :

```
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Guided> go run GuidedM7.2.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
-----
20231001     Andi
20231002     Budi
20231004     Dedi
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Guided>
```

Penjelasan :

Program ini menggunakan **map** untuk menyimpan data mahasiswa dengan NIM sebagai kunci dan nama sebagai nilai. Data dapat ditambahkan, diakses, dan dihapus dari map. Program juga menampilkan data sebelum dan sesudah penghapusan dalam format tabel.

III. UNGUIDE

NO.1

Source code :

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    pusat 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 didalamLingkaran(l Lingkaran, t Titik) bool {
    return jarak(l.pusat, t) <= float64(l.radius)
}

func posisiTitik(l1, l2 Lingkaran, t Titik) string {
    dalamL1 := didalamLingkaran(l1, t)
    dalamL2 := didalamLingkaran(l2, t)

    if dalamL1 && dalamL2 {
        return "Titik di dalam lingkaran 1 dan 2"
    } else if dalamL1 {
        return "Titik di dalam lingkaran 1"
    } else if dalamL2 {
        return "Titik di dalam lingkaran 2"
    } else {
        return "Titik di luar lingkaran 1 dan 2"
    }
}
```

```
func main() {  
  
    var l1, l2 Lingkaran  
    var t Titik  
  
    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 1:")  
    fmt.Scan(&l1.pusat.x, &l1.pusat.y, &l1.radius)  
    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 2:")  
    fmt.Scan(&l2.pusat.x, &l2.pusat.y, &l2.radius)  
    fmt.Println("Masukkan koordinat titik sembarang:")  
    fmt.Scan(&t.x, &t.y)  
  
    hasil := posisiTitik(l1, l2, t)  
    fmt.Println(hasil)  
}  
//2311102042_ Ahmad Titana Nanda Pramudya
```


Output :

```
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Unguided> go run unguided1.7.go
Masukkan koordinat pusat dan radius lingkaran 1:
1 1 5
Masukkan koordinat pusat dan radius lingkaran 2:
8 8 4
Masukkan koordinat titik sembarang:
2 2
Titik di dalam lingkaran 1
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Unguided> █
```

Penjelasan :

Program ini menentukan posisi suatu titik terhadap dua lingkaran. Kode ini menghitung jarak titik ke pusat lingkaran dan memeriksa apakah titik tersebut berada di dalam radius lingkaran. Berdasarkan hasil pengecekan, kode mengeluarkan salah satu dari empat hasil: "Titik di dalam lingkaran 1 dan 2", "Titik di dalam lingkaran 1", "Titik di dalam lingkaran 2", dan, "Titik di luar lingkaran 1 dan 2". Ini mempermudah pengecekan posisi titik terhadap dua lingkaran.

NO.2

Source code :

```
package main

import (
    "fmt"
    "math"
)

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

func displayElementsByIndex(arr []int, start int, step int, desc string) {
    fmt.Printf("Elemen dengan indeks %s: ", desc)
    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 _, v := range arr {
        sum += v
    }
    return float64(sum) / float64(len(arr))
}

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

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

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

    arr := make([]int, N)
    for i := 0; i < N; 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)

avg := calculateAverage(arr)
fmt.Println("Rata-rata array:", avg)

stdDev := calculateStandardDeviation(arr, avg)
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))
}
//2311102042_ Ahmad Titana Nanda Pramudya

```

Output :

```

PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Unguided> go run unguided2.7.go
Masukkan jumlah elemen array: 3
Masukkan elemen ke-0: 1
Masukkan elemen ke-1: 2
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

```

Penjelasan :

Program ini mengelola array bilangan bulat dengan fitur berikut: Menampilkan seluruh elemen array, Menampilkan elemen di indeks ganjil, genap, atau kelipatan bilangan x, Menghapus elemen pada indeks tertentu, Menghitung rata-rata dan simpangan baku array, Dan Menghitung frekuensi kemunculan suatu bilangan dalam array.

NO.3

Source code :

```
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")
}
//2311102042_ Ahmad Titana Nanda Pramudya
```

Output :

```
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Unguided> go run unguided3.7.go
Masukkan nama Klub A: MU
Masukkan nama Klub B: INTER
Masukkan skor pertandingan 1 (MU vs INTER): 2 0
Masukkan skor pertandingan 2 (MU vs INTER): 1 2
Masukkan skor pertandingan 3 (MU vs INTER): 2 2
Masukkan skor pertandingan 4 (MU vs INTER): 0 1
Masukkan skor pertandingan 5 (MU vs INTER): 3 2
Masukkan skor pertandingan 6 (MU vs INTER): 1 0
Masukkan skor pertandingan 7 (MU vs INTER): 5 2
Masukkan skor pertandingan 8 (MU vs INTER): 2 3
Masukkan skor pertandingan 9 (MU vs INTER): -1 2
Skor tidak valid, menghentikan input...
Hasil pertandingan:
Hasil 1: MU menang
Hasil 2: INTER menang
Hasil 3: Draw
Hasil 4: INTER menang
Hasil 5: MU menang
Hasil 6: MU menang
Hasil 7: MU menang
Hasil 8: INTER menang
Pertandingan selesai
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Unguided> █
```

Penjelasan :

hasil pertandingan antara dua klub sepak bola dengan memungkinkan pengguna untuk memasukkan skor. Input akan berhenti jika ada skor negatif. Program menentukan pemenang berdasarkan skor: Klub A menang jika skornya lebih tinggi, Klub B menang jika skornya lebih tinggi, dan hasilnya "Draw" jika skor sama. Hasil setiap pertandingan disimpan dalam array, dan setelah input selesai, program menampilkan rekap hasil pertandingan dan pesan "Pertandingan selesai". Program ini efektif dalam mencatat dan menampilkan hasil pertandingan dengan kontrol input yang baik.

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)
}

// 2311102042_Ahmad Titana Nanda Pramudya
```

Output :

```
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Unguided> go run unguided4.7.go
Teks: SENENG
Reverse Teks: GNEENS
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Unguided> █
```

Penjelasan:

Program ini pengolahan string dan penggunaan array. Meskipun memiliki keterbatasan dalam hal fleksibilitas input, program ini berhasil mencapai tujuannya untuk membalikkan urutan karakter dan menampilkan hasilnya. Dengan beberapa perbaikan, seperti memperluas kemampuan input dan memproses berbagai jenis karakter, program ini dapat menjadi alat yang lebih berguna dan fungsional.

MODIFIKASI PROGRAM

Sourcode:

```
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 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

fmt.Print("Teks: ")

isiArray(&tab, &n)

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

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

//2311102042_Ahmad Titana Nanda Pramudya

```

Output :

```

PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Unguided> go run modifikasi4.7.go
Teks: KATAK
Teks: KATAK
Palindrome? true
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Unguided> go run modifikasi4.7.go
Teks: SENENG
Teks: SENENG
Palindrome? false
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_ MODUL7\Unguided>

```

Penjelasan :

Modifikasi dari program awal menjadi lebih fungsional dengan menambahkan pemeriksaan palindrom, memberikan nilai lebih pada pengguna dalam hal validasi teks. Program yang dimodifikasi mencerminkan peningkatan dalam logika pemrograman dan interaksi pengguna, menjadikannya alat yang lebih berguna untuk analisis string.