

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

MODUL VII

STRUCK & ARRAY



**Telkom
University
PURWOKERTO**

Oleh:

Destia Ananda Putra

2311102176

IF-11-02

S1 TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

I. DASAR TEORI

Tipe Bentukan

Tipe bentukan memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipe bentukan ini dapat dibedakan atas dua jenis, yaitu Alias dan Struct.

1) Alias (Type)

Bahasa pemrograman pada umumnya mengizinkan pemrogram untuk mengubah nama suatu tipe data dengan nama baru yang lebih ringkas dan familiar. Sebagai contoh "integer" dapat diubah 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 adalah struktur data yang digunakan untuk menyimpan sekumpulan elemen dengan tipe data yang sama dalam urutan memori yang berkelanjutan. Setiap elemen dalam array memiliki posisi unik yang disebut indeks, yang dimulai dari angka 0. Dengan menggunakan indeks ini, kita dapat mengakses elemen-elemen dalam array secara cepat dan efisien. Array memiliki beberapa karakteristik utama yang membedakannya dari struktur data lainnya, yaitu:

1. **Homogenitas:** Semua elemen dalam array harus memiliki tipe data yang sama, sehingga menjaga konsistensi dan keutuhan data.

2. **Ukuran Tetap:** Setelah array dibuat, jumlah elemen atau kapasitas array tersebut tidak dapat diubah selama program berjalan.
3. **Akses Acak:** Array memungkinkan akses langsung ke setiap elemen melalui indeksnya, sehingga proses pengambilan atau pengubahan data dapat dilakukan dengan cepat.
4. **Banyak Dimensi:** Array dapat memiliki lebih dari satu dimensi, seperti array satu dimensi (vektor) dan array dua dimensi (matriks), yang memungkinkan penyimpanan data dalam bentuk yang lebih kompleks.

Slice (Array Dinamik)

struktur data yang memungkinkan penyimpanan elemen-elemen yang memiliki tipe data yang sama, namun dengan ukuran yang fleksibel, tidak seperti array statis yang memiliki ukuran tetap. Slice array dinamik memudahkan penambahan atau pengurangan elemen secara dinamis, sehingga tidak perlu mendefinisikan ukuran awal yang pasti.

II. Guided

Guided 1

```
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)

    }

}

```

Output

```
PS D:\Destia Ananda Putra.Go\src>
go run "c:\Users\Lenovo\Documents\File Tugas Semester 3\Pratikum Alpro 2\Modul 7\De
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:\Destia Ananda Putra.Go\src>
```

Penjelasan

Program menghitung dan menampilkan peringkat mahasiswa berdasarkan rata-rata nilai tiga mata pelajaran. Hal ini berguna untuk mengetahui siapa mahasiswa dengan performa akademik terbaik dan siapa yang membutuhkan perhatian lebih dalam peningkatan akademis. sederhana namun efektif dalam menyajikan informasi performa akademik dengan jelas, serta dapat diadaptasi untuk menambahkan lebih banyak mahasiswa atau mata pelajaran. mendefinisikan daftar mahasiswa beserta nilai mereka pada tiga mata pelajaran. Data ini disimpan dalam struktur tertentu (misalnya, array atau slice) yang memungkinkan setiap mahasiswa memiliki nama dan tiga nilai mata pelajaran.

Guided II

```
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

```

PS C:\Users\Lenovo> go run "c:\Users\Lenovo\Documents\go"
Daftar Mahasiswa:
NIM          Nama
-----
20231003     Cici
20231004     Dedi
20231001     Andi
20231002     Budi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM          Nama
-----
20231001     Andi
20231002     Budi
20231004     Dedi
PS C:\Users\Lenovo>

```

Penjelasan

Program ini mengelola data mahasiswa dengan fitur utama seperti menambah, menghapus, mengakses data berdasarkan NIM, dan menampilkan daftar secara rapi. Dimana Semua fungsi berjalan dengan baik, program dapat dikembangkan lebih lanjut dengan menambahkan validasi data agar NIM unik, pengurutan data berdasarkan NIM, penanganan kesalahan, serta memungkinkan input data secara dinamis dari pengguna. Dengan penambahan ini, program akan menjadi lebih fleksibel dan fungsional.

III. Unguided

Unguided I

```

package main

// Destia Ananda Putra
// 2311102176

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

```

```

type Lingkaran struct {
    pusat Titik
    radius int
}

func hitungJarak(t Titik, l Lingkaran) float64 {
    return math.Sqrt(math.Pow(float64(t.x-l.pusat.x), 2)
    + math.Pow(float64(t.y-l.pusat.y), 2))
}

func apakahDiDalamLingkaran(jarak float64, radius
int) bool {
    return jarak <= float64(radius)
}

func main() {
    var x1, y1, r1 int
    fmt.Print("Masukkan koordinat x1, y1 dan radius r1
lingkaran pertama: ")
    fmt.Scan(&x1, &y1, &r1)
    lingkaran1 := Lingkaran{Titik{x1, y1}, r1}

    var x2, y2, r2 int
    fmt.Print("Masukkan koordinat x2, y2 dan radius r2
lingkaran kedua: ")
    fmt.Scan(&x2, &y2, &r2)
    lingkaran2 := Lingkaran{Titik{x2, y2}, r2}

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

    jarakKeLingkaran1 := hitungJarak(titik,
lingkaran1)

```



```

    jarakKeLingkaran2 := hitungJarak(titik,
    lingkaran2)

    dalamLingkaran1 :=
    apakahDiDalamLingkaran(jarakKeLingkaran1,
    lingkaran1.radius)
    dalamLingkaran2 :=
    apakahDiDalamLingkaran(jarakKeLingkaran2,
    lingkaran2.radius)

    if dalamLingkaran1 && dalamLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dalamLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if dalamLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

Output

```

PS C:\Users\Lenovo> go run "c:\Users\Lenovo\Documents\File Tugas S
Masukkan koordinat x1, y1 dan radius r1 lingkaran pertama: 1 1 5
Masukkan koordinat x2, y2 dan radius r2 lingkaran kedua: 8 8 4
Masukkan koordinat x dan y titik sembarang: 2 2
Titik di dalam lingkaran 1
PS C:\Users\Lenovo> go run "c:\Users\Lenovo\Documents\File Tugas S
Masukkan koordinat x1, y1 dan radius r1 lingkaran pertama: 1 2 3
Masukkan koordinat x2, y2 dan radius r2 lingkaran kedua: 4 5 6
Masukkan koordinat x dan y titik sembarang: 7 8
Titik di dalam lingkaran 2

```

Penjelasan

Program ini bertujuan untuk mengecek apakah suatu titik berada di dalam satu atau kedua lingkaran dari dua lingkaran yang diberikan. Program menerima input berupa koordinat pusat dan radius dari dua lingkaran, serta koordinat titik sembarang. Selanjutnya, program menghitung jarak antara titik sembarang dengan pusat lingkaran menggunakan formula jarak dan mengecek apakah jarak tersebut lebih kecil atau sama dengan radius lingkaran. Program ini menentukan apakah sebuah titik berada di dalam salah satu atau kedua lingkaran berdasarkan jarak titik

ke pusat lingkaran. Program menerima input koordinat pusat dan radius dua lingkaran serta koordinat titik sembarang. program menghitung jarak titik ke masing-masing lingkaran dan memeriksa apakah jarak tersebut lebih kecil atau sama dengan radius lingkaran.

Unguided II

```
package main

// Destia Ananda Putra
// 2311102176

import (
    "fmt"
    "math"
)

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

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

    fmt.Println("a. Isi seluruh array:", arr)

    fmt.Print("b. Elemen dengan indeks ganjil: ")
    for i := 1; i < n; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    fmt.Print("c. Elemen dengan indeks genap: ")
    for i := 0; i < n; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    var x int
    fmt.Print("d. Masukkan angka x untuk melihat elemen dengan indeks kelipatan x: ")
    fmt.Scan(&x)
    fmt.Printf("Elemen dengan indeks kelipatan %d: ", x)
    for i := x; i < n; i += x {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    var index int
```

```

    fmt.Print("e. Masukkan indeks elemen yang ingin dihapus: ")
    fmt.Scan(&index)
    if index >= 0 && index < n {
        arr = append(arr[:index], arr[index+1:]...)
        fmt.Println("Array setelah elemen dihapus:", arr)
    } else {
        fmt.Println("Indeks tidak valid.")
    }

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

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

    frekuensi := make(map[int]int)
    for _, value := range arr {
        frekuensi[value]++
    }
    fmt.Println("h. Frekuensi kemunculan setiap elemen:")
    for key, freq := range frekuensi {
        fmt.Printf("Angka %d muncul %d kali\n", key, freq)
    }
}

```

Output

```
PS C:\Users\Lenovo> go run "c:\Users\Lenovo\Documents\File Tugas Semester 3\
e.go"
Masukkan jumlah elemen array (N): 10
Masukkan elemen-elemen array:
Elemen ke-0: 1
Elemen ke-1: 2
Elemen ke-2: 3
Elemen ke-3: 4
Elemen ke-4: 5
Elemen ke-5: 6
Elemen ke-6: 7
Elemen ke-7: 8
Elemen ke-8: 9
Elemen ke-9: 10
a. Isi seluruh array: [1 2 3 4 5 6 7 8 9 10]
b. Elemen dengan indeks ganjil: 2 4 6 8 10
c. Elemen dengan indeks genap: 1 3 5 7 9
d. Masukkan angka x untuk melihat elemen dengan indeks kelipatan x: 3
Elemen dengan indeks kelipatan 3: 4 7 10
e. Masukkan indeks elemen yang ingin dihapus: 2
Array setelah elemen dihapus: [1 2 4 5 6 7 8 9 10]
f. Rata-rata elemen array: 5.78
g. Standar deviasi elemen array: 2.90
h. Frekuensi kemunculan setiap elemen:
Angka 7 muncul 1 kali
Angka 2 muncul 1 kali
Angka 4 muncul 1 kali
Angka 5 muncul 1 kali
Angka 6 muncul 1 kali
Angka 8 muncul 1 kali
Angka 9 muncul 1 kali
Angka 10 muncul 1 kali
Angka 1 muncul 1 kali
PS C:\Users\Lenovo>
```

Penjelasan

Program pada array yang berisi bilangan bulat. Berdasarkan output yang dihasilkan, program mampu menampilkan elemen array sesuai kebutuhan pengguna, seperti elemen dengan indeks ganjil, genap, atau kelipatan bilangan tertentu. Program juga memungkinkan penghapusan elemen pada indeks tertentu dengan hasil yang akurat, serta menghitung rata-rata dan standar deviasi untuk analisis statistik sederhana. Selain itu, program juga dapat menampilkan frekuensi kemunculan setiap elemen, yang bermanfaat untuk memahami distribusi data. Menampilkan elemen array dengan pola tertentu (ganjil, genap, kelipatan x), Menghapus elemen secara dinamis dan memberikan array hasil penghapusan, Program ini sangat berguna untuk analisis data berbasis array, dengan fleksibilitas tinggi untuk kebutuhan data sederhana hingga menengah.

Unguided III

```
package main

// Destia Ananda Putra
// 2311102176

import (
    "fmt"
)

func main() {

    var klubA, klubB string
    fmt.Print("Klub A : ")
    fmt.Scanln(&klubA)
    fmt.Print("Klub B : ")
    fmt.Scanln(&klubB)

    var hasil []string

    var skorA, skorB int
    pertandingan := 1

    for {
        fmt.Printf("Pertandingan %d : ", pertandingan)
        fmt.Scanln(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            fmt.Println("Pertandingan selesai")
            break
        }

        if skorA > skorB {
            hasil = append(hasil, klubA)
        } else if skorA < skorB {
            hasil = append(hasil, klubB)
        }
    }
}
```

```

    } else {
        hasil = append(hasil, "Draw")
    }

    pertandingan++
}

for i, result := range hasil {
    fmt.Printf("Hasil %d : %s\n", i+1, result)
}

fmt.Println("Pertandingan selesai")
}

```

Output

```

PS C:\Users\Lenovo> go run "c:\Users\Lenovo\Doc
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
Pertandingan selesai
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
PS C:\Users\Lenovo> █

```

Penjelasan

Program ini mencatat hasil pertandingan sepak bola antara dua klub berdasarkan skor yang dimasukkan oleh pengguna. Pengguna diminta memasukkan nama Klub A dan Klub B, lalu skor pertandingan secara bertahap. Setiap skor dianalisis untuk menentukan pemenang atau hasil seri. Program berhenti saat skor negatif dimasukkan, lalu menampilkan hasil semua pertandingan. Program ini membantu mencatat dan menampilkan hasil pertandingan dengan format yang jelas dan terstruktur. Setiap input skor dianalisis, dan hasilnya ditampilkan sesuai urutan. Program berhenti secara otomatis saat skor negatif dimasukkan, memastikan pengelolaan data yang efisien.

Unguided IV

```
package main

// Destia Ananda Putra
// 2311102176

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var char rune
    fmt.Println("Masukkan karakter satu per satu (akhiri dengan titik '.'): ")
    for *n = 0; *n < NMAX; *n++ {
        fmt.Scanf("%c", &char)
        if char == '.' {
            break
        }
        t[*n] = char
    }
}

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 {
    for i := 0; i < n/2; i++ {
        if t[i] != t[n-i-1] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var m int

    isiArray(&tab, &m)

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

```
    balikArray(&tab, m)

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

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

OUTPUT

```
PS C:\Users\Lenovo> go run "c:\Users\Lenovo\Documents\File
Masukkan karakter satu per satu (akhiri dengan titik '.'):
S E N A N G.
Teks asli: S E N A N G
Teks setelah dibalik: G N A N E S
Apakah palindrom? false
PS C:\Users\Lenovo> go run "c:\Users\Lenovo\Documents\File
Masukkan karakter satu per satu (akhiri dengan titik '.'):
K A T A K.
Teks asli: K A T A K
Teks setelah dibalik: K A T A K
Apakah palindrom? true
PS C:\Users\Lenovo> █
```

Penjelasan

Program ini berfungsi untuk menentukan apakah teks yang dimasukkan merupakan palindrom atau bukan, dengan menampilkan hasil yang jelas dalam satu kesatuan proses. Pengguna diminta untuk memasukkan teks satu per satu karakter dan mengakhiri input dengan tanda titik (.). Setelah itu, program akan mencetak teks asli yang dimasukkan, mencetak versi teks yang telah dibalik, lalu menyimpulkan apakah teks tersebut adalah palindrom. Palindrom adalah teks yang tetap sama jika dibaca dari depan maupun belakang, misalnya "SENANG". program akan mencetak teks asli SENANG, membaliknya menjadi GNANES, dan menyimpulkan bahwa teks tersebut bukan palindrom karena berbeda jika dibaca dari depan dan belakang. Program ini menampilkan semua informasi secara terintegrasi, mulai dari input, proses pembalikan, hingga kesimpulan akhir, sehingga memudahkan pengguna untuk memahami hasil evaluasi.