

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL VII
STRUCT DAN ARRAY



Bhara Yudha
Pamungkas
2311102186
IF-11-06

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITPURWOKERTO
2024

I. DASAR TEORI

Tipe Bentuk (Structs and Aliases)

1.1. Tipe Alias

Tipe alias memungkinkan programmer untuk mendefinisikan nama baru untuk tipe data yang sudah ada, sehingga lebih mudah dan lebih intuitif untuk digunakan. Misalnya, dalam bahasa pemrograman Go, kita dapat menggunakan kata kunci `type` untuk membuat alias. Contoh:

```
• type bilangan int
• type pecahan float64
```

- II. Dengan cara ini, kita dapat menggunakan bilangan dan pecahan sebagai tipe data baru yang lebih mudah diingat dan digunakan dalam kode.

1.2. Struct

Struct adalah tipe bentuk yang memungkinkan programmer untuk mengelompokkan beberapa data yang memiliki relasi menjadi satu kesatuan. Setiap data dalam struct disimpan dalam field yang berbeda. Contoh definisi struct dalam Go adalah sebagai berikut:

```
• type waktu struct {
•   jam int
•   menit int
•   detik int
5. }
```

- III. Dalam contoh ini, waktu adalah struct yang memiliki tiga field: jam, menit, dan detik, semuanya bertipe `int`.

1.3. Kesamaan Tipe

Dalam Go, dua variabel dengan struktur yang sama (nama field dan tipe field yang sama) dianggap memiliki tipe yang sama, meskipun nama struct-nya berbeda. Ini

memudahkan penggunaan struct dalam program.

2. Array

2.1. Definisi Array

Array adalah kumpulan elemen dengan tipe yang sama, di mana ukuran (jumlah elemen) ditentukan saat deklarasi dan bersifat tetap selama eksekusi program.

Contoh

deklarasi array dalam Go:

```
1. var arr [73]CircType
```

IV. Di sini, arr adalah array dengan 73 elemen yang bertipe CircType.

2.2. Mengakses Elemen Array

Indeks array dimulai dari 0, sehingga elemen pertama dapat diakses dengan arr[0], dan elemen terakhir dengan arr[len(arr)-1]. Fungsi len dapat digunakan untuk mendapatkan jumlah elemen dalam array.

2.3. Slice

Slice adalah array dinamik yang dapat berubah ukuran. Slice dapat dideklarasikan tanpa menentukan jumlah elemen, dan dapat diperluas menggunakan fungsi built-in append.

Contoh deklarasi slice:

```
1. var s []int
```

```
2. s = append(s, 1, 2, 3)
```

V. Fungsi len dan cap dapat digunakan untuk mengetahui ukuran dan kapasitas slice.

2.4. Map

Map adalah tipe data yang menyimpan pasangan kunci-nilai, di mana kunci tidak harus

berupa integer. Contoh deklarasi map dalam Go:

```
• var dct map[string]int  
• dct =
```

```
make(map[string]int)
```

```
3. dct["john"] = 1
```

VI. Map memungkinkan akses cepat ke nilai berdasarkan kunci yang

II. GUID

ED Guided 1

Source code

:

```
package main

import (
    "fmt"
    "sort"
)

type Mahasiswa
    struct { Nama
        string Matematika
        int Fisika int
        Kimia int RataRata
        float64
    }

func hitungRataRata(m *Mahasiswa) {
    total := m.Matematika + m.Fisika + m.Kimia
    m.RataRata = float64(total) / 3.0
}

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 {
            hitungRataRata(&mahasiswa[
                i])
        }

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

    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)
        }
    }
}

```

Screenshot output :

Deskripsi program :

```

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)

```

Program ini mendefinisikan program di Go yang menghitung skor rata-rata beberapa siswa dan mengurutkan siswa dalam urutan menurun berdasarkan skor rata-rata. Hasilnya adalah daftar siswa yang diurutkan dari nilai rata-rata tertinggi hingga terendah.

Guided 2

Sourch 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\tNama") fmt.Println("
    ") for nim, nama := range
    mahasiswa {
fmt.Printf("%s\t%s\n", nim, nama)
}
}

```

Screenshot output :

Deskripsi program :

```

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

```

Program ini memungkinkan pengguna untuk, Menambahkan data mahasiswa (NIM dan Nama), Menampilkan seluruh data mahasiswa dalam format tabel, Mengakses data mahasiswa berdasarkan NIM, Menghapus data mahasiswa dari map, Menampilkan data mahasiswa yang tersisa setelah penghapusan,. Map di sini berfungsi sebagai struktur penyimpanan data yang efisien, memungkinkan pencarian, penambahan, dan penghapusan data dengan cepat.

III. UNGUIDE

D Unguided 1

Source code :

```
// Fajar Farizqi Azmi  
// 2311102192  
  
package main  
    import (  
        "fmt"  
        "math"  
    )  
  
// Struct untuk merepresentasikan titik type  
    Titik struct {  
        x, y int  
    }
```

```

// Struct untuk merepresentasikan
lingkaran type Lingkaran struct {
    titikPusat
    Titik
    jariJari int

// Fungsi untuk mengecek apakah suatu titik berada di dalam
lingkaran
    jarak := math.Sqrt(math.Pow(float64(t.x-l.titikPusat.x), 2) +
math.Pow(float64(t.y-l.titikPusat.y), 2))
    return jarak <= float64(l.jariJari)
}

func main() {
    // Masukan baris pertama untuk
    lingkaran 1 var x1, y1, r1 int
    fmt.Printf("Masukkan untuk lingkaran
1: ") fmt.Scan(&x1, &y1, &r1)
    lingkaran1 := Lingkaran{Titik{x1, y1},
    // Masukan baris kedua untuk
    lingkaran 2 var x2, y2, r2 int
    fmt.Printf("Masukkan untuk lingkaran
2: ") fmt.Scan(&x2, &y2, &r2)
    lingkaran2 := Lingkaran{Titik{x2, y2},
    // Masukan baris ketiga untuk titik
    sembarang var x, y int
    fmt.Printf("Masukkan untuk titik
sembarang: ") fmt.Scan(&x, &y)
    titik := Titik{x, y}

```

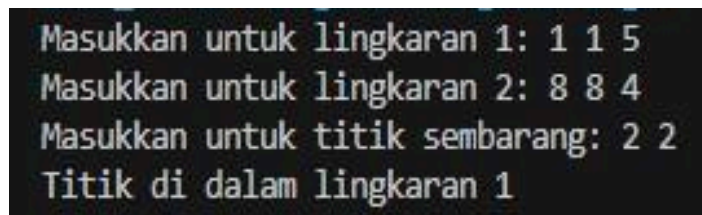
```

        // Mengecek apakah titik berada di dalam lingkaran 1, lingkaran
        2, atau keduanya
diLingkaran1 := apakahTitikDiDalamLingkaran(titik, lingkaran1)
        diLingkaran2 := apakahTitikDiDalamLingkaran(titik,
        lingkaran2)

// Menentukan keluaran berdasarkan hasil pengecekan if
        diLingkaran1 && diLingkaran2 {
fmt.Println("Titik di dalam lingkaran 1 dan 2")
        } else if diLingkaran1 {
fmt.Println("Titik di dalam lingkaran 1")
        } else if diLingkaran2 {
fmt.Println("Titik di dalam lingkaran 2")
        } else {
fmt.Println("Titik di luar lingkaran 1 dan 2")
        }
    }
}

```

Screenshot output :



```

Masukkan untuk lingkaran 1: 1 1 5
Masukkan untuk lingkaran 2: 8 8 4
Masukkan untuk titik sembarang: 2 2
Titik di dalam lingkaran 1

```

Deskripsi program :

Program ini dipakai buat memilih apakah sebuah titik berada pada pada keliru satu, kedua, atau pada luar 2 bundar yg diberikan dari input koordinat & jari-jari bundar. Dengan memakai konsep jeda Euclidean, acara ini melakukan pengecekan posisi nisbi titik terhadap bundar secara efisien & menaruh output sinkron menggunakan syarat geometris yg diuji.

Unguided 2

Sourch code :

```

// Fajar Farizqi Azmi
// 2311102192
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.")
        }
    }

```

```

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++
}
}
}

```

```
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)
```

```
}
```

Screenshot output :

```
PS D:\semester3\Alpro 2 materi dan praktikum\Fajar Farizqi Azmi_2311102192_modul 7> go run "d:\semester3\Alpro 2 materi dan praktikum\Fajar Farizqi Azmi_2311102192_modul 7\Unguided\unguided2.go"
Masukkan jumlah elemen array: 5
Masukkan elemen ke-0: 23
Masukkan elemen ke-1: 4
Masukkan elemen ke-2: 6
Masukkan elemen ke-3: 8
Masukkan elemen ke-4: 9
Isi array: [23 4 6 8 9]
Elemen dengan indeks ganjil: 4 8
Elemen dengan indeks genap: 23 6 9
Masukkan bilangan x untuk kelipatan indeks: 
```

Deskripsi program :

Program ini melakukan operasi pada elemen, Input Array, Menampilkan Isi Array, Kelipatan Indeks, Hapus Elemen, Statistik, Rata-rata, Standar Deviasi dan Frekuensi. Program ini mencetak hasil setiap operasi secara langsung.

Source code :

```
// Fajar Farizqi Azmi
// 2311102192

package main import

    "fmt"

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

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

pertandingan := 1 for {
fmt.Printf("Pertandingan %d: ", pertandingan)
    fmt.Scanln(&skorA, &skorB)

    if skorA < 0 || skorB < 0 {
        break
    }

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



```

}

pertandingan++
}

fmt.Println("Pertandingan selesai") for i, p
:= range pemenang {
fmt.Printf("Hasil %d: %s\n", i+1, p)
}
}

```

Screenshot output :

Deskripsi program :

Program ini memungkinkan pengguna buat mencatat & melihat output

```

Klub A: MU
Klub B: Inter
Pertandingan 1: 2 4
Pertandingan 2: 2 7
Pertandingan 3: 3 7
Pertandingan 4: 9 8
Pertandingan 5: 1 1
Pertandingan 6: 9 6
Pertandingan 7: 3 5
Pertandingan 8: 3 9
Pertandingan 9: -1 8
Pertandingan selesai
Hasil 1: Inter
Hasil 2: Inter
Hasil 3: Inter
Hasil 4: MU
Hasil 5: Draw
Hasil 6: MU

```

pertandingan antar 2 klub sepak bola. Input terus diterima sampai masih ada skor negatif, & output pertandingan dicetak secara berurutan sehabis pertandingan selesai.

Unguided 4

Sourch code :

```
// FAJAR FARIZQI AZMI
```

```
// 2311102192
```

```
package main
```

```

import "fmt"
const NMAX int = 127
type
    tabel struct {
        tab [NMAX]rune
        m int
    }

func isiArray(t *tabel, n *int) {

    fmt.Print("Masukkan karakter (akhiri dengan titik): ")
    var
        input rune
        for *n = 0; *n < NMAX;
            *n++ {
                fmt.Scanf("%c",
                    &input)
                if input ==
                    '.' { break
                }
                t.tab[*n] = input
            }
        t.m = *n
    }

    func cetakArray(t tabel, n
        int) {
        for i := 0; i < n;

            i++ {

```

```

for i := 0; i < n/2; i++ {
    t.tab[i], t.tab[n-i-1] = t.tab[n-i-1], t.tab[i]
}
}

```

```

func palindrom(t tabel, n int)

```

```

    bool { for i := 0; i < n/2; i++

```

```

    {

```

```

        if t.tab[i] !=

```

```

            t.tab[n-i-1] { return

```

```

                false

```

```

    }

```

```

}

```

```

return true

```

```

}

```

```

func main()

```

```

    { var tab

```

```

        tabel var

```

```

        m int

```

```

    isiArray(&tab, &m)

```

```

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

```

```
fmt.Println("Palindrom: false")  
  
}  
  
}
```

Screenshot output :

```
PS D:\semester3\Alpro 2 materi dan praktikum\Fajar Farizqi Azmi_2311102192_modul 7> go run "d:\semester3\Alpro 2 materi dan praktikum\Fajar Farizqi Azmi_2311102192_modul 7\Unguided\unguided4.go"  
Masukkan karakter (akhiri dengan titik): senang.  
Teks: senang  
Reverse teks: gnanes  
Palindrom: false  
PS D:\semester3\Alpro 2 materi dan praktikum\Fajar Farizqi Azmi_2311102192_modul 7> |
```

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS D:\semester3\Alpro 2 materi dan praktikum\Fajar Farizqi Azmi_2311102192_modul 7> go run "d:\semester3\Alpro 2 materi dan praktikum\Fajar Farizqi Azmi_2311102192_modul 7\Unguided\unguided4.go"  
Masukkan karakter (akhiri dengan titik): katak.  
Teks: katak  
Reverse teks: katak  
Palindrom: true  
PS D:\semester3\Alpro 2 materi dan praktikum\Fajar Farizqi Azmi_2311102192_modul 7> |
```

Deskripsi program :

Program untuk memasukkan teks berupa karakter, membalikkan urutan karakter, dan memeriksa apakah teks tersebut adalah palindrom. Program ini menggunakan array dan operasi pertukaran elemen untuk membalikkan urutan teks, serta memanfaatkan perbandingan elemen array untuk memeriksa palindrom.