

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2  
MODUL 7  
STRUCK & ARRAY**



Oleh:

MUHAMMAD AGHA ZULFADHLI

2311102015

S1-IF11-02

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

## I. DASAR TEORI

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 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".

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama alias> <tipe data>	type <nama alias> <tipe data>
3		
4	algoritma	func main(){
5	...	...
6		}
7		

### 2. Struct atau Record

Stucture memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari stucture tersebut.

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama struct> <	type <nama struct> struct {
3	<field 1> <tipe data>	<field 1> <tipe data>
4	<field 2> <tipe data>	<field 2> <tipe data>
5	<field 3> <tipe data>	<field 3> <tipe data>
6	>	}
7		

Array mempunyai ukuran (jumlah elemen) yang tetap (statls) selama eksekusi program, sehingga jumlah elemen array menjadi bagian dari deklarasi variabel dengan tipe array.

	Notasi dalam bahasa Go
1	var (
2	// array arr mempunyai 73 elemen, masing-masing bertipe CircType2
3	arr [73]CircType
4	
5	// array buf dengan 5 elemen, dengan nilai awal 7, 3, 5, 2, dan 11.
6	buf = [5]byte{7, 3, 5, 2, 11}
7	
8	// mhs adalah array dengan 2000 elemen bertipe NewType
9	mhs [2000]NewType
10	
11	// rec adalah array dari array, yaitu matriks, atau array berdimensi-2
12	rec [20][40]float64
13	)

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

```

1 // declaring chop as an empty slice of float64
2 var chop []float64
3
4 // declaring sl01 as a slice
5 var sl01 = []int{ 11, 2, 3, 5, 7, 13 }
```

Sebuah slice dapat diprealokasi menggunakan fungsi built-in `make()` 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.

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

### 1. Guided 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
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)
    }
}

```

### Screenshoot program

```

PS D:\Personal\KULIAH\ALPRO_2> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_7\GUIDED\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:\Personal\KULIAH\ALPRO_2>

```

### Deskripsi program

Program mengelola data mahasiswa dan menghitung rata-rata nilai mereka. Struktur `Mahasiswa` menyimpan nama dan nilai dari tiga mata pelajaran: Matematika, Fisika, dan Kimia, serta rata-rata nilainya. Dalam fungsi utama, array `mahasiswa` diinisialisasi dengan beberapa data mahasiswa. Setiap mahasiswa diproses melalui fungsi `hitungRataRata` untuk menghitung rata-rata nilai. Setelah itu, data mahasiswa diurutkan berdasarkan nilai rata-rata secara menurun (descending) menggunakan `sort.Slice`. Terakhir, daftar mahasiswa ditampilkan dalam urutan peringkat bersama dengan nilai dan rata-rata masing-masing.

## 2. Guided 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)
    }
}

```

## Screenshoot program

```

PS D:\Personal\KULIAH\ALPRO_2> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_7\GUIDED\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
-----
20231002  Budi
20231004  Dedi
20231001  Andi
PS D:\Personal\KULIAH\ALPRO_2>

```

## Deskripsi program

Kode di atas membuat sebuah map di Go dengan NIM (Nomor Induk Mahasiswa) sebagai kunci dan nama mahasiswa sebagai nilai. Pertama, beberapa data mahasiswa ditambahkan ke dalam map. Lalu, data baru dimasukkan ke map menggunakan NIM "20231004". Kode menampilkan semua data mahasiswa dalam format tabel menggunakan `fmt.Printf`. Selanjutnya, ia mengakses dan mencetak nama mahasiswa berdasarkan NIM tertentu. Setelah itu, data mahasiswa dengan NIM "20231003" dihapus dari map. Terakhir, kode kembali menampilkan isi map setelah penghapusan, masih dalam format tabel.

### III. UNGUIDED

#### 1. Unguided 1

##### Source code

```
package main

import (
    "fmt"
    "math"
)

type titik struct {
    a, b float64
}

type lingkaran struct {
    x, y, r float64
}

func main() {
    var l1, l2 lingkaran
    var x titik
    fmt.Scan(&l1.x, &l1.y, &l1.r, &l2.x, &l2.y, &l2.r, &x.a,
    &x.b)
    dalam1 := didalam(l1, x)
    dalam2 := didalam(l2, x)

    if dalam1 && dalam2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dalam1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if dalam2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

func jarak(a lingkaran, c titik) float64 {
    return math.Sqrt(math.Pow(c.a-a.x, 2) + math.Pow(c.b-a.y, 2))
}
```



```
func didalam(l lingkaran, x titik) bool {
    return jarak(l, x) <= l.r
}
```

### Screenshoot program

```
PS D:\Personal\KULIAH\ALPRO_2> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_7\UNGUIDED\unguided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\Personal\KULIAH\ALPRO_2>
```

### Deskripsi program

Program mengecek posisi sebuah titik terhadap dua lingkaran, apakah titik tersebut berada di dalam salah satu, kedua, atau di luar keduanya. Dengan struktur titik untuk koordinat dan lingkaran untuk pusat dan jari-jari, program menerima input posisi lingkaran dan titik. Fungsi jarak menghitung jarak antara pusat lingkaran dan titik menggunakan rumus Euclidean, lalu fungsi didalam memverifikasi apakah titik berada di dalam lingkaran berdasarkan jarak ini. Program kemudian mencetak hasil apakah titik berada di dalam salah satu, kedua, atau di luar lingkaran.

## 2. Unguided 2

### Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    arr := []int{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

    fmt.Println("a. Tampilkan elemen-elemen array")
    fmt.Println("b. Tampilkan elemen-elemen array ganjil")
    fmt.Println("c. Tampilkan elemen-elemen array genap")
    fmt.Println("d. Tampilkan elemen-elemen array kelipatan")
    fmt.Println("x")
}
```

```

fmt.Println("e. Hapus indeks ke-\"x\" dari array")
fmt.Println("f. Tampilkan rata-rata elemen array")
fmt.Println("g. Standar deviasi dari array")
fmt.Println("h. Tampilkan frekuensi \"x\" dari array")
fmt.Println("_____")
    )
fmt.Print("Pilihan : ")

in := ""
fmt.Scan(&in)
fmt.Println()

switch in {
case "a": //? SHOW ELEMENTS
    print_arr(arr)
case "b":
    for i := 0; i < len(arr); i++ {
        if arr[i]%2 != 0 {
            fmt.Print(arr[i], " ")
        }
    }
case "c": //? SHOW GANJIL
    for i := 0; i < len(arr); i++ {
        if arr[i]%2 == 0 {
            fmt.Print(arr[i], " ")
        }
    }
case "d": //? SHOW GENAP
    num := 0
    fmt.Print("Nilai yang ingin dicari : ")
    fmt.Scan(&num)
    for i, value := range arr {
        if value == num {
            fmt.Print("Nilai yang dicari ada di index ke-",
i+1)
        }
    }
case "e": //? DELETE ELEMENT
    cut := 0
    fmt.Print("Index yang ingin dihapus : ")
    fmt.Scan(&cut)

```

```

        if cut < 0 {
            fmt.Println("Array Mulai dari 0")
        } else if cut > len(arr) {
            fmt.Println("Index diluar array")
            break
        }
        fmt.Print("Sebelum hapus : ")
        print_arr(arr)
        arr = append(arr[:cut], arr[cut+1:]...)
        fmt.Print("Selesai hapus : ")
        print_arr(arr)
    case "f": //? RATA RATA
        sum := 0
        for _, value := range arr {
            sum += value
        }
        fmt.Println(float64(sum) / float64(len(arr)))
    case "g": //? STANDARD DEVIATION
        fmt.Print("Array : ")
        print_arr(arr)
        fmt.Println(standardDeviation(arr))
    case "h": //? FREKUENSI
        var num, all int
        fmt.Print("Nilai yang ingin dicari : ")
        fmt.Scan(&num)
        for _, value := range arr {
            if value == num {
                all++
            }
        }
        fmt.Println(all)
    }
}

func print_arr(arr []int) {
    for _, value := range arr {
        fmt.Print(value, " ")
    }
    fmt.Println()
}

```

```
func standardDeviation(arr []int) float64 {
    var sum, mean, variance float64
    n := float64(len(arr))

    for _, value := range arr {
        sum += float64(value)
    }
    mean = sum / n

    for _, value := range arr {
        variance += math.Pow(float64(value)-mean, 2)
    }
    return math.Sqrt(variance / n)
}
```

### Screenshoot program

```
PROBLEMS 68 TERMINAL OUTPUT DEBUG CONSOLE PORTS
• PS D:\Personal\KULIAH\ALPRO_2> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_7\UNGUIDED\unguided2.go"
a. Tampilkan elemen-elemen array
b. Tampilkan elemen-elemen array ganjil
c. Tampilkan elemen-elemen array genap
d. Tampilkan elemen-elemen array kelipatan "x"
e. Hapus indeks ke-"x" dari array
f. Tampilkan rata-rata elemen array
g. Standar deviasi dari array
h. Tampilkan frekuensi "x" dari array

Pilihan : a

0 1 2 3 4 5 6 7 8 9 10
PS D:\Personal\KULIAH\ALPRO_2>
```

### Deskripsi program

Program menyediakan berbagai operasi pada array integer [0, 1, 2, ..., 10] melalui menu interaktif. Pengguna dapat memilih operasi untuk menampilkan elemen array (a), menampilkan elemen ganjil (b), menampilkan elemen genap (c), menampilkan elemen kelipatan nilai tertentu (d), menghapus elemen pada indeks tertentu (e), menghitung rata-rata nilai elemen (f), menghitung standar deviasi (g), dan menghitung frekuensi kemunculan nilai tertentu dalam array (h). Fungsi `print_arr` digunakan untuk menampilkan seluruh elemen array, sedangkan `standardDeviation` menghitung standar deviasi dengan cara menghitung variansi dari rata-rata. Program ini menggunakan `switch` untuk mengarahkan operasi sesuai pilihan pengguna dan menyertakan validasi sederhana untuk mengelola input, seperti indeks yang valid saat menghapus elemen dari array.

## 3. Unguided 3

## Source code

```
package main

import "fmt"

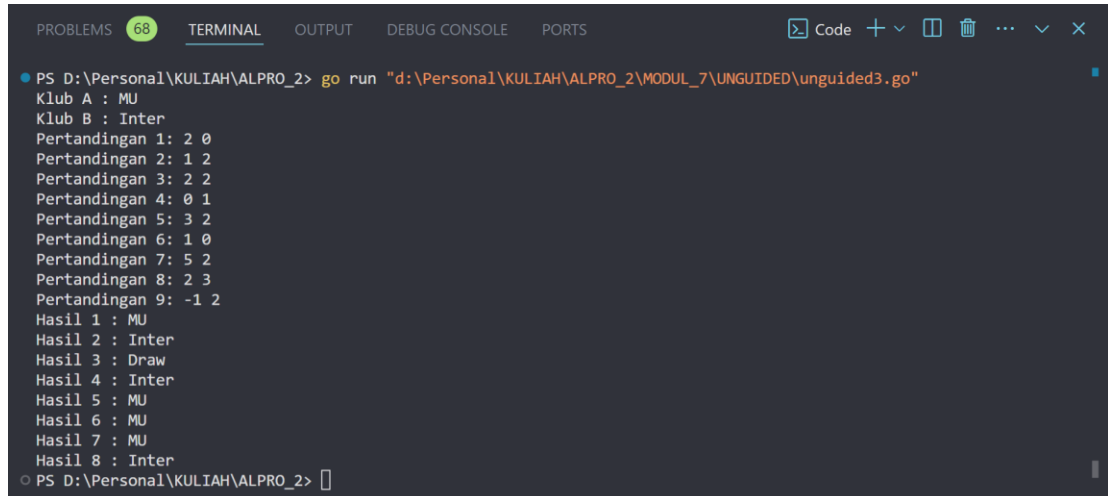
func main() {
    var nama_a, nama_b string
    var a, b int
    var arr []string

    fmt.Print("Klub A : ")
    fmt.Scan(&nama_a)
    fmt.Print("Klub B : ")
    fmt.Scan(&nama_b)

    for i := 0; ; i++ {
        fmt.Print("Pertandingan ", i+1, ": ")
        fmt.Scan(&a, &b)
        if a < 0 || b < 0 {
            break
        }
        if a > b {
            arr = append(arr, nama_a)
        } else if a < b {
            arr = append(arr, nama_b)
        } else {
            arr = append(arr, "Draw")
        }
    }

    for i := 0; i < len(arr); i++ {
        fmt.Println("Hasil", i+1, ":", arr[i])
    }
}
```

## Screenshoot program



```
PS D:\Personal\KULIAH\ALPRO_2> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_7\UNGUIDED\unguided3.go"
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
Hasil 1 : MU
Hasil 2 : Inter
Hasil 3 : Draw
Hasil 4 : Inter
Hasil 5 : MU
Hasil 6 : MU
Hasil 7 : MU
Hasil 8 : Inter
PS D:\Personal\KULIAH\ALPRO_2>
```

## Deskripsi program

Program mencatat hasil pertandingan antara dua klub, yaitu "Klub A" dan "Klub B". Pertama, program meminta nama kedua klub sebagai input dari pengguna. Selanjutnya, program meminta skor dari setiap pertandingan berturut-turut, di mana pengguna memasukkan skor "Klub A" dan "Klub B" untuk setiap pertandingan. Jika skor "Klub A" lebih tinggi, nama "Klub A" disimpan dalam array hasil (arr); jika skor "Klub B" lebih tinggi, nama "Klub B" disimpan; jika seri, "Draw" yang disimpan. Program akan berhenti menerima skor jika pengguna memasukkan skor negatif untuk salah satu klub. Akhirnya, program mencetak hasil pertandingan satu per satu sesuai dengan urutan pertandingan yang diinputkan.

## 4. Unguided 4

### Source code

```
package main

import (
    "fmt"
)

const NMAX int = 127

type tabel [NMAX] rune
```

```

func isiArray(t *tabel, n *int) {
    var input rune
    *n = 0
    for *n < NMAX {
        fmt.Scanf("%c\n", &input)
        if input == '.' {
            break
        }
        t[*n] = input
        *n++
    }
}

func cetakArray(t tabel, n int) {
    fmt.Print("Teks\t\t: ")
    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

```

```

// balikanArray(&tab, m)
// fmt.Println("Isi array setelah dibalik:")
// cetakArray(tab, m)
isiArray(&tab, &m)
cetakArray(tab, m)
if palindrom(tab, m) {
    fmt.Println("Palindrom\t? true")
} else {
    fmt.Println("Palindrom\t? false")
}
}

```

### Screenshoot program

```

PS D:\Personal\KULIAH\ALPRO_2> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_7\UNGUIDED\unguided4.go"
K
A
T
A
K
.
Teks      : K A T A K
Palindrom ? true
PS D:\Personal\KULIAH\ALPRO_2>

PS D:\Personal\KULIAH\ALPRO_2> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_7\UNGUIDED\unguided4.go"
S
E
N
A
N
G
.
Teks      : S E N A N G
Palindrom ? false
PS D:\Personal\KULIAH\ALPRO_2>

```

### Deskripsi program

Program membaca serangkaian karakter yang dimasukkan pengguna hingga karakter titik (.) dimasukkan, lalu menyimpannya dalam array tabel berukuran maksimal 127 karakter. Fungsi `isiArray` digunakan untuk mengisi array dengan karakter input, sementara `cetakArray` menampilkan karakter yang tersimpan di array. Program juga memiliki fungsi `balikanArray` yang membalik urutan karakter dalam array dan `palindrom` yang memeriksa apakah urutan karakter membentuk palindrom (urutan sama jika dibaca dari depan dan belakang). Di dalam fungsi main, program memanggil `isiArray` untuk mengisi array, lalu menampilkan array, dan mengecek apakah array tersebut palindrom dengan mencetak hasil `true` atau `false` sesuai dengan hasil pengecekan.