

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 7  
STRUCK DAN ARRAY**



Oleh:

ANANDA BASKORO PUTRA

2311102187

IF 11 02

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO 2024**

## I. DASAR TEORI

### a. 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.

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>	
3		type <nama alias> <tipe data>
4	algoritma	
5		func main(){
6		
7		}

### Struktur atau Record

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

	Notas1 A1gor1tma	Notas1 dalam bahasa Go
1	kamus	package main
2	type mama struct»«	type mama 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		

Berbeda dengan bahasa pemrograman lain. kesamaan tipe dari dua variabel berjenis stucture 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 stucture tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi stucture tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

## II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

### 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) / 30
}

// 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 Output

```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7\guided1.go"
Peringkat mahasiswa berdasarkan rata-rata nilai:
1. Eka - Rata-Rata: 9.500000 (Matematika: 100, Fisika: 95, Kimia: 90)
2. Cici - Rata-Rata: 9.000000 (Matematika: 90, Fisika: 85, Kimia: 95)
3. Ali - Rata-Rata: 8.500000 (Matematika: 85, Fisika: 90, Kimia: 80)
4. Budi - Rata-Rata: 7.500000 (Matematika: 70, Fisika: 75, Kimia: 80)
5. Doni - Rata-Rata: 6.500000 (Matematika: 60, Fisika: 65, Kimia: 70)
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7>
```

## Deskripsi Program

Program ini berguna untuk mengelola data mahasiswa dan mengurutkan mereka berdasarkan rata-rata nilai dari tiga mata pelajaran: Matematika, Fisika, dan Kimia. Setiap mahasiswa memiliki nama dan nilai pada ketiga mata pelajaran tersebut, yang disimpan dalam struktur data Mahasiswa. Program ini pertama-tama menghitung rata-rata nilai tiap mahasiswa menggunakan fungsi `hitungRataRata`, kemudian mengurutkan data mahasiswa berdasarkan nilai rata-rata dalam urutan menurun (descending) menggunakan fungsi `sort.Slice`. Akhirnya, program menampilkan daftar mahasiswa beserta nilai rata-rata dan nilai masing-masing mata pelajaran dalam format peringkat.

## Guided 2

```
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 {

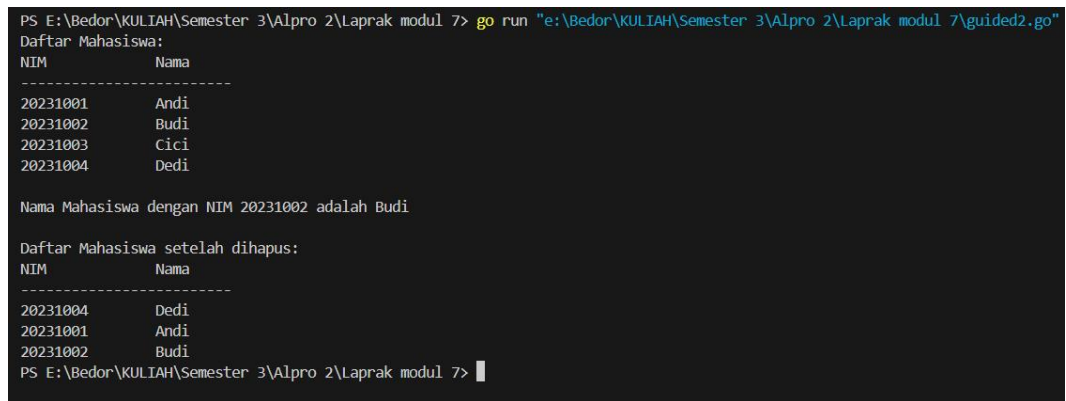
fmt.Printf("%s\t%s\n", nim, nama)

}

}

```

## Screenshoot Output



```

PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7\guided2.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
-----
20231004  Dedi
20231001  Andi
20231002  Budi
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7>

```

## Deskripsi Program

Program ini menggunakan struktur data map di Go untuk mengelola data mahasiswa, di mana NIM (Nomor Induk Mahasiswa) digunakan sebagai kunci dan nama mahasiswa sebagai nilai. Program dimulai dengan membuat map berisi tiga data mahasiswa, kemudian menambahkan satu data baru. Selanjutnya, program menampilkan daftar mahasiswa dalam format kolom dan baris. Program juga menampilkan nama mahasiswa berdasarkan NIM tertentu, menghapus data mahasiswa dengan NIM tertentu, dan akhirnya menampilkan kembali daftar mahasiswa setelah penghapusan data.



### III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

#### Unguided 1

```
package main

import (
    "fmt"
    "math"
)

type titik struct {
    x, y int
}

type lingkaran struct {
    cx, cy, r int
}

func jarak(p titik, c lingkaran) float64 {
    return math.Sqrt(float64((p.x-c.cx)*(p.x-c.cx) + (p.y-c.cy)*(p.y-c.cy)))
}

func posisiTitik(p titik, l1, l2 lingkaran) string {
    jarak1 := jarak(p, l1)
    jarak2 := jarak(p, l2)
    dalam1 := jarak1 < float64(l1.r)
    dalam2 := jarak2 < float64(l2.r)
    if dalam1 && dalam2 {
        return "Titik di dalam lingkaran 1 dan 2"
    }
}
```

```

    } else if dalam1 {
        return "Titik di dalam lingkaran 1"
    } else if dalam2 {
        return "Titik di dalam lingkaran 2"
    } else {
        return "Titik di luar lingkaran 1 dan 2"
    }
}

func main() {
    var l1, l2 lingkaran
    var p titik

    fmt.Print("Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r): ")
    fmt.Scan(&l1.cx, &l1.cy, &l1.r)

    fmt.Print("Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r): ")
    fmt.Scan(&l2.cx, &l2.cy, &l2.r)

    fmt.Print("Masukkan koordinat titik sembarang (x y): ")
    fmt.Scan(&p.x, &p.y)

    fmt.Print("\n")

    hasil := posisiTitik(p, l1, l2)

    fmt.Println(hasil)
}

```

## Screenshoot Output

```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7\Unguided1.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r): 1 1 5
Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r): 8 8 4
Masukkan koordinat titik sembarang (x y): 2 2

Titik di dalam lingkaran 1
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> |
```

## Deskripsi Program

Program ini mengecek posisi sebuah titik terhadap dua lingkaran dengan menentukan apakah titik tersebut berada di dalam atau di luar masing-masing lingkaran. Program dimulai dengan meminta pengguna untuk memasukkan koordinat pusat dan jari-jari untuk dua lingkaran serta koordinat titik yang akan diperiksa. Fungsi jarak digunakan untuk menghitung jarak antara titik dengan pusat lingkaran menggunakan rumus jarak Euclidean. Berdasarkan jarak ini, fungsi posisiTitik memeriksa apakah titik tersebut berada di dalam satu atau kedua lingkaran, atau di luar keduanya. Program kemudian menampilkan hasil posisi titik terhadap lingkaran-lingkaran tersebut.

## Unguided 2

```
package main

import (
    "fmt"
    "math"
)

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

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

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

func tampilkanIndeksKelipatan(arr []int, x int) {
```

```

    fmt.Printf("Elemen dengan indeks kelipatan %d: ", x)

    for i := x; i < len(arr); i += x {
        fmt.Print(arr[i], " ")
    }

    fmt.Println()
}

func hapusElemen(arr []int, index int) []int {
    if index < 0 || index >= len(arr) {
        fmt.Println("Indeks tidak valid.")
        return arr
    }

    return append(arr[:index], arr[index+1:]...)
}

func hitungRataRata(arr []int) float64 {
    sum := 0

    for _, v := range arr {
        sum += v
    }

    return float64(sum) / float64(len(arr))
}

func hitungStandarDeviasi(arr []int) float64 {
    rataRata := hitungRataRata(arr)

    var sum float64

    for _, v := range arr {

```

```

        sum += math.Pow(float64(v)-rataRata, 2)

    }

    varian := sum / float64(len(arr))

    return math.Sqrt(varian)
}

func hitungFrekuensi(arr []int) map[int]int {
    frekuensi := make(map[int]int)

    for _, v := range arr {
        frekuensi[v]++
    }

    return frekuensi
}

func main() {
    // Input jumlah elemen dan nilai-nilai elemen array
    var n int

    fmt.Print("Masukkan jumlah elemen array: ")

    fmt.Scan(&n)

    arr := make([]int, n)

    fmt.Println("Masukkan elemen-elemen array:")

    for i := 0; i < n; i++ {
        fmt.Scan(&arr[i])
    }

    tampilkanSeluruhArray(arr)
}

```

```

    tampilkanIndeksGanjil(arr)

    tampilkanIndeksGenap(arr)

    var x int

    fmt.Print("Masukkan nilai x untuk indeks kelipatan: ")

    fmt.Scan(&x)

    tampilkanIndeksKelipatan(arr, x)

    var index int

    fmt.Print("Masukkan indeks elemen yang ingin dihapus: ")

    fmt.Scan(&index)

    arr = hapusElemen(arr, index)

    fmt.Println("Array setelah elemen dihapus:")

    tampilkanSeluruhArray(arr)

    rataRata := hitungRataRata(arr)

    fmt.Printf("Rata-rata elemen array: %.2f\n", rataRata)

    standarDeviasi := hitungStandarDeviasi(arr)

    fmt.Printf("Standar deviasi elemen array: %.2f\n", standarDeviasi)

    frekuensi := hitungFrekuensi(arr)

    fmt.Println("Frekuensi dari setiap elemen:")

    for key, value := range frekuensi {

        fmt.Printf("Elemen %d muncul %d kali\n", key, value)

    }

}

```

## Screenshoot Output

```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7\Unguided2.go"
Masukkan jumlah elemen array: 10
Masukkan elemen-elemen array:
1
2
3
4
5
6
7
8
9
10
Isi array: [1 2 3 4 5 6 7 8 9 10]
Elemen dengan indeks ganjil: 2 4 6 8 10
Elemen dengan indeks genap: 1 3 5 7 9
Masukkan nilai x untuk indeks kelipatan: 2
Elemen dengan indeks kelipatan 2: 3 5 7 9
Masukkan indeks elemen yang ingin dihapus: 1
Array setelah elemen dihapus:
Isi array: [1 3 4 5 6 7 8 9 10]
Rata-rata elemen array: 5.89
Standar deviasi elemen array: 2.77
Frekuensi dari setiap elemen:
Elemen 3 muncul 1 kali
Elemen 4 muncul 1 kali
Elemen 5 muncul 1 kali
Elemen 7 muncul 1 kali
Elemen 9 muncul 1 kali
Elemen 1 muncul 1 kali
Elemen 6 muncul 1 kali
Elemen 8 muncul 1 kali
Elemen 10 muncul 1 kali
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> █
```

## Deskripsi Program

Program ini melakukan berbagai operasi pada sebuah array integer, seperti menampilkan isi array, elemen pada indeks ganjil dan genap, serta elemen pada indeks yang merupakan kelipatan dari bilangan yang ditentukan pengguna. Program juga memungkinkan pengguna untuk menghapus elemen pada indeks tertentu, menghitung rata-rata, dan standar deviasi elemen-elemen dalam array.



### Unguided 3

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var klub1, klub2 string
    var skor1, skor2 int
    var pemenang []string

    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scan(&klub1)

    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scan(&klub2)

    for i := 1; ; i++ {
        fmt.Printf("Pertandingan %d (skor %s vs %s): ", i, klub1, klub2)

        n, err := fmt.Scan(&skor1, &skor2)

        if err != nil || n != 2 || skor1 < 0 || skor2 < 0 {
            fmt.Println("Input skor tidak valid. Pertandingan selesai.")
            break
        }

        if skor1 > skor2 {
            fmt.Printf("Hasil %d: %s menang\n", i, klub1)
```

```
pemenang = append(pemenang, klub1)

} else if skor2 > skor1 {

fmt.Printf("Hasil %d: %s menang\n", i, klub2)

pemenang = append(pemenang, klub2)

} else {

fmt.Printf("Hasil %d: Draw\n", i)

}

}

fmt.Println("\nDaftar klub yang memenangkan pertandingan:")

if len(pemenang) > 0 {

fmt.Println(strings.Join(pemenang, ", "))

} else {

fmt.Println("Tidak ada klub yang menang.")

}

}
```

## Screenshoot Output

```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7\Unguided3.go"
Masukkan nama Klub A: Barcelona
Masukkan nama Klub B: Vardrid
Pertandingan 1 (skor Barcelona vs Vardrid): 3 0
Hasil 1: Barcelona menang
Pertandingan 2 (skor Barcelona vs Vardrid): 2 1
Hasil 2: Barcelona menang
Pertandingan 3 (skor Barcelona vs Vardrid): 2 3
Hasil 3: Vardrid menang
Pertandingan 4 (skor Barcelona vs Vardrid): 3 3
Hasil 4: Draw
Pertandingan 5 (skor Barcelona vs Vardrid): 4 5
Hasil 5: Vardrid menang
Pertandingan 6 (skor Barcelona vs Vardrid): 9 0
Hasil 6: Barcelona menang
Pertandingan 7 (skor Barcelona vs Vardrid): 40 0
Hasil 7: Barcelona menang
Pertandingan 8 (skor Barcelona vs Vardrid): 1 2
Hasil 8: Vardrid menang
Pertandingan 9 (skor Barcelona vs Vardrid): -1 2
Input skor tidak valid. Pertandingan selesai.

Daftar klub yang memenangkan pertandingan:
Barcelona, Barcelona, Vardrid, Vardrid, Barcelona, Barcelona, Vardrid
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> █
```

## Deskripsi Program

Program ini mencatat hasil pertandingan antara dua klub sepak bola dengan mencatat nama klub, skor tiap pertandingan, dan menentukan pemenang tiap pertandingan. Program dimulai dengan meminta pengguna memasukkan nama kedua klub. Kemudian, program meminta input skor untuk tiap pertandingan secara berulang. Jika skor dari salah satu klub lebih tinggi, klub tersebut dinyatakan sebagai pemenang pertandingan dan ditambahkan ke daftar pemenang. Jika skor sama, pertandingan dinyatakan sebagai seri. Program berhenti saat pengguna memasukkan input yang tidak valid (seperti skor negatif atau bukan angka). Di akhir, program menampilkan daftar klub yang memenangkan pertandingan. Jika tidak ada klub yang menang, program menampilkan pesan bahwa tidak ada klub yang menang.

## Unguided 4

```
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++ {
        fmt.Printf("%c", t.tab[i])
    }
}
```

```

}

fmt.Println()

}

func balikanArray(t *tabel, n int) {
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, m)

fmt.Print("Reverse teks: ")

```

```
balikanArray(&tab, m)

cetakArray(tab, m)

isPalindrom := palindrom(tab, m)

if isPalindrom {

    fmt.Println("Palindrom: true")

} else {

    fmt.Println("Palindrom: false")

}

}
```

## Screenshoot Output

```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7\Unguided4.go"
Masukkan karakter (akhiri dengan titik): KATAK.
Teks: KATAK
Reverse teks: KATAK
Palindrom: true
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> █
```

```
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> go run "e:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7\Unguided4.go"
Masukkan karakter (akhiri dengan titik): SENANG.
Teks: SENANG
Reverse teks: GNANES
Palindrom: false
PS E:\Bedor\KULIAH\Semester 3\Alpro 2\Laprak modul 7> █
```

## **DESKRIPSI PROGRAM**

Program ini mengelola array berisi karakter-karakter yang diinput oleh pengguna, menyajikan beberapa fungsi untuk memproses teks tersebut. Pengguna memasukkan karakter satu per satu hingga mengetik tanda titik ('.') sebagai tanda berhenti. Fungsi `isiArray` mengisi array tab dengan karakter yang dimasukkan pengguna hingga mencapai batas maksimal (127 karakter). Program kemudian mencetak teks asli menggunakan fungsi `cetakArray`. Setelah itu, fungsi `balikanArray` digunakan untuk membalikkan urutan karakter dalam array, dan hasilnya juga ditampilkan. Terakhir, program mengecek apakah teks tersebut adalah palindrom (teks yang sama jika dibaca dari depan maupun belakang) menggunakan fungsi `palindrom`, dan hasilnya ditampilkan sebagai "true" atau "false".