

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6**

**MATERI**



Oleh:

ABISAR FATHIR

103112400068

12-IF-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2025**

## **I. DASAR TEORI**

Modul ini membahas konsep tipe data bentukan dan struktur data array dalam pemrograman Go. Tipe bentukan terdiri dari alias untuk menyederhanakan penamaan tipe data yang sudah ada dan struct untuk mengelompokkan data terkait menjadi satu unit. Array diperkenalkan sebagai struktur data statis dengan ukuran tetap, sedangkan slice dan map ditampilkan sebagai alternatif yang lebih fleksibel untuk menangani data dinamis dan asosiatif.

Contoh aplikasi meliputi analisis geometri untuk menentukan posisi titik terhadap lingkaran, pemeriksaan palindrom pada teks, serta pengelolaan data seperti statistik pertandingan olahraga atau nilai siswa. Modul ini menekankan pentingnya pemilihan struktur data yang sesuai untuk menyelesaikan masalah secara efisien dan terorganisir.

Dengan memahami materi ini, pembaca dapat mengimplementasikan tipe bentukan dan struktur data array, slice, serta map dalam pengembangan program menggunakan bahasa Go.

## II. GUIDED

### 1. Program ke 1

Source Code:

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("Masukkan jumlah tanaman: ")

    fmt.Scan(&n)

    var tinggitanaman [500]float64

    for i := 0; i < n; i++ {

        fmt.Printf("Masukkan tinggi tanaman ke-%d(cm):", i+1)

        fmt.Scan(&tinggitanaman[i])

    }

    min, max := tinggitanaman[0], tinggitanaman[0]

    for i := 0; i < n; i++ {

        if tinggitanaman[i] < min {

            min = tinggitanaman[i]
```

```

}

if tinggitanaman[i] > max {
    max = tinggitanaman[i]
}
}

fmt.Printf("\nTinggi tanaman tertinggi: %.2f cm\n", max)

fmt.Printf("\nTinggi tanaman terendah: %.2f cm\n", min)

}

```

Output Program:

```

Masukkan jumlah tanaman: 2
Masukkan tinggi tanaman ke-1(cm):2
Masukkan tinggi tanaman ke-2(cm):3

Tinggi tanaman tertinggi: 3.00 cm
Tinggi tanaman terendah: 2.00 cm

```

Deskripsi Program: Program digunakan untuk menghitung tanaman yang paling tinggi dan terendah berdasarkan inputan yang dimasukkan ke indeks array tinggi tanaman

## 2. Program ke 2

Source Code:

```
package main

import "fmt"

func main() {
    var x, y int

    fmt.Print("Masukkan jumlah buku dan jumlah buku per rak : ")

    fmt.Scan(&x, &y)

    var hargabuku [500]float64

    fmt.Println("\nMasukkan harga setiap buku (dalam ribuan rp):")

    for i := 0; i < x; i++ {

        fmt.Scan(&hargabuku[i])

    }

    var hargaratarata []float64

    for i := 0; i < x; i += y {

        total := 0.0

        for j := i; j < i+y && j < x; j++ {

            total += hargabuku[j]

        }

        hargaratarata = append(hargaratarata, total/float64(y))

    }
```

```
min, max := hargaratarata[0], hargaratarata[0]

for _, harga := range hargabuku[:x] {
    if harga < min {
        min = harga
    }
    if harga > max {
        max = harga
    }
}

fmt.Printf("\nRata-rata harga per rak")

for _, avg := range hargaratarata {
    fmt.Printf("%.2f ", avg)

}

fmt.Printf("\nHarga termahal: %.2f Rp\n", max)
fmt.Printf("\nHarga termurah: %.2f Rp\n", min)
}
```

Output Program:

```

Masukkan jumlah buku dan jumlah buku per rak : 4 2

Masukkan harga setiap buku (dalam ribuan rp):
10000
100
100
100

Rata-rata harga per rak5050.00 100.00
Harga termahal: 10000.00 Rp

Harga termurah: 100.00 Rp

```

Deskripsi Program:Program ini dirancang untuk mengelola dan menganalisis data harga buku yang disimpan dalam rak-rak buku. Aplikasi ini memulai dengan meminta pengguna untuk memasukkan dua data utama: jumlah total buku yang akan diolah (dengan batas maksimal 500 buku) dan kapasitas setiap rak buku (berapa banyak buku yang dapat dimasukkan dalam satu rak).

### 3. Program ke 3

Source Code:

```

package main

import "fmt"

func main() {
    var n int

    fmt.Println("Masukkan jumlah siswa: ")
    fmt.Scan(&n)

    var nilaiSiswa [200]float64

    var totalNilai float64 = 0

```

```
fmt.Println("\nMasukkan nilai ujian masing-masing siswa:")
```

```
for i := 0; i < n; i++ {
```

```
    fmt.Scan(&nilaiSiswa[i])
```

```
    totalNilai += nilaiSiswa[i]
```

```
}
```

```
rataRata := totalNilai / float64(n)
```

```
min, max := nilaiSiswa[0], nilaiSiswa[0]
```

```
var diAtasRataRata int = 0
```

```
for _, nilai := range nilaiSiswa[:n] {
```

```
    if nilai < min {
```

```
        min = nilai
```

```
    }
```

```
    if nilai > max {
```

```
        max = nilai
```

```
    }
```

```
    if nilai > rataRata {
```

```
        diAtasRataRata++
```

```
    }
```

```
}
```

```
fmt.Printf("\nNilai terendah: %.0f\n", min)
```

```
fmt.Printf("Nilai tertinggi: %.0f\n", max)
```



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

fmt.Printf("Jumlah siswa di atas rata-rata: %d\n", diAtasRataRata)

}
```

Output Program:

```
Masukkan jumlah siswa: 2

Masukkan nilai ujian masing-masing siswa:
10
100

Nilai terendah: 10
Nilai tertinggi: 100
Rata-rata kelas: 55.00
Jumlah siswa di atas rata-rata: 1
```

Deskripsi Program: Program ini merupakan sebuah aplikasi sederhana untuk menganalisis nilai ujian siswa. Program memulai dengan meminta pengguna memasukkan jumlah siswa yang akan dianalisis, dengan batas maksimal 200 siswa. Setelah itu, program akan meminta input nilai ujian untuk masing-masing siswa satu per satu.

### III. UNGUIDED

#### 1. Program ke 1

Source Code:

```
package main

import "fmt"

func main() {

var x1, y1, r1, x2, y2, r2, x, y float64

fmt.Scan(&x1, &y1, &r1)

fmt.Scan(&x2, &y2, &r2)

fmt.Scan(&x, &y)

jarak1 := (x-x1)*(x-x1) + (y-y1)*(y-y1)

jarak2 := (x-x2)*(x-x2) + (y-y2)*(y-y2)

diDalamLingkaran1 := jarak1 <= r1*r1
```

```

diDalamLingkaran2 := jarak2 <= r2*r2

if diDalamLingkaran1 && diDalamLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if diDalamLingkaran1 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if diDalamLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

Output Program:

```

1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1

```

Deskripsi Program: bertujuan untuk menentukan posisi sebuah titik terhadap dua lingkaran berbeda dalam bidang dua dimensi.

## 2. Program ke 2

Source Code:

```

package main

```

```
import (  
    "fmt"  
    "math"  
)  
  
func main() {  
    var n int  
  
    fmt.Print("Masukkan jumlah elemen: ")  
    fmt.Scan(&n)  
  
    arr := make([]int, n)  
  
    fmt.Println("Masukkan nilai-nilai array:")  
    for i := 0; i < n; i++ {  
        fmt.Scan(&arr[i])  
    }  
  
    fmt.Println("\nSeluruh isi array:")  
    for i := 0; i < n; i++ {  
        fmt.Printf("%d ", arr[i])  
    }  
  
    fmt.Println("\n\nElemen dengan indeks ganjil:")
```

```
for i := 1; i < n; i += 2 {  
    fmt.Printf("%d ", arr[i])  
}  
  
fmt.Println("\n\nElemen dengan indeks genap:")  
  
for i := 0; i < n; i += 2 {  
    fmt.Printf("%d ", arr[i])  
}  
  
sum := 0  
  
for i := 0; i < n; i++ {  
    sum += arr[i]  
}  
  
rata := float64(sum) / float64(n)  
  
fmt.Printf("\n\nRata-rata: %.2f\n", rata)  
  
var sumSquaredDiff float64  
  
for i := 0; i < n; i++ {  
    diff := float64(arr[i]) - rata  
    sumSquaredDiff += diff * diff  
}  
  
stdDev := math.Sqrt(sumSquaredDiff / float64(n))  
  
fmt.Printf("Standar deviasi: %.2f\n", stdDev)  
}
```

Output Program:

```
Masukkan jumlah elemen: 2
Masukkan nilai-nilai array:
20
2

Seluruh isi array:
20 2

Elemen dengan indeks ganjil:
2

Elemen dengan indeks genap:
20

Rata-rata: 11.00
Standar deviasi: 9.00
```

Deskripsi Program: Program ini merupakan alat untuk menganalisis data dalam bentuk array (kumpulan data terurut) dengan berbagai perhitungan statistik. Aplikasi ini dirancang untuk membantu pengguna memahami karakteristik data numerik yang dimasukkan.

### 3. Program ke 3

Source Code:

```
package main

import "fmt"

func main() {

    var klubA, klubB string

    fmt.Print("Klub A : ")
```

```
fmt.Scan(&klubA)
```

```
fmt.Print("Klub B : ")
```

```
fmt.Scan(&klubB)
```

```
var skorA, skorB int
```

```
var pertandingan int = 0
```

```
var menangA, menangB, seri int = 0, 0, 0
```

```
for {
```

```
    pertandingan++
```

```
    fmt.Printf("Pertandingan %d : ", pertandingan)
```

```
    fmt.Scan(&skorA, &skorB)
```

```
    if skorA < 0 || skorB < 0 {
```

```
        pertandingan--
```

```
        break
```

```
    }
```

```
    if skorA > skorB {
```

```
        fmt.Printf("Hasil %d : %s\n", pertandingan, klubA)
```

```
        menangA++
```

```
    } else if skorB > skorA {
```

```
        fmt.Printf("Hasil %d : %s\n", pertandingan, klubB)
```

```

        menangB++
    } else {
        fmt.Printf("Hasil %d : Draw\n", pertandingan)
        seri++
    }
}

fmt.Println("Pertandingan selesai")
}

```

Output Program:

```

Klub A : MU
Klub B : inter
Pertandingan 1 : 2 0
Hasil 1 : MU
Pertandingan 2 : 1 2
Hasil 2 : inter
Pertandingan 3 : 2 2
Hasil 3 : Draw
Pertandingan 4 : 0 1
Hasil 4 : inter
Pertandingan 5 : 3 2
Hasil 5 : MU
Pertandingan 6 : 1 0
Hasil 6 : MU
Pertandingan 7 : 5 2
Hasil 7 : MU
Pertandingan 8 : 2 3
Hasil 8 : inter
Pertandingan 9 : -1 2
Pertandingan selesai

```

Deskripsi Program: mensimulasikan serangkaian pertandingan sepak bola antara dua klub. Aplikasi ini dirancang untuk mencatat hasil pertandingan dan memberikan statistik sederhana tentang performa kedua tim.

## 4. Program ke 4

Source Code:



```
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

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

    var c rune

    *n = 0

    fmt.Print("Teks: ")

    for {

        fmt.Scanf("%c", &c)

        if c == '.' || *n >= NMAX {

            break

        }

        t[*n] = c

        *n++

    }

}
```

```
func cetakArray(t tabel, n int) {  
  
    for i := 0; i < n; i++ {  
  
        fmt.Printf("%c", t[i])  
  
    }  
  
    fmt.Println()  
  
}  
  
func balikArray(t *tabel, n int) {  
  
    var temp tabel  
  
    for i := 0; i < n; i++ {  
  
        temp[i] = t[n-1-i]  
  
    }  
  
    for i := 0; i < n; i++ {  
  
        t[i] = temp[i]  
  
    }  
  
}  
  
func palindrom(t tabel, n int) bool {  
  
    var temp tabel  
  
    for i := 0; i < n; i++ {  
  
        temp[i] = t[i]
```

```
}

balikArray(&temp, n)

for i := 0; i < n; i++ {
    if t[i] != temp[i] {
        return false
    }
}

return true
}

func main() {
    var tab tabel
    var m int
    isiArray(&tab, &m)

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

    balikArray(&tab, m)
    fmt.Print("Reverse teks: ")
    cetakArray(tab, m)
```

```

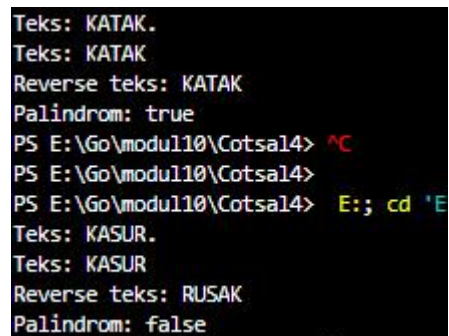
balikArray(&tab, m)

fmt.Print("Palindrom: ")

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

```

Output Program:



```

Teks: KATAK.
Teks: KATAK
Reverse teks: KATAK
Palindrom: true
PS E:\Go\modul10\Cotsa14> ^C
PS E:\Go\modul10\Cotsa14>
PS E:\Go\modul10\Cotsa14> E;; cd 'E
Teks: KASUR.
Teks: KASUR
Reverse teks: RUSAK
Palindrom: false

```

Deskripsi Program: Program ini dirancang untuk menganalisis apakah sebuah teks merupakan palindrom atau bukan. Palindrom adalah kata, frasa, atau susunan karakter yang sama jika dibaca dari depan maupun belakang (contoh: "katak", "kasur rusak").

#### IV. KESIMPULAN

Modul ini menjelaskan konsep fundamental dalam pemrograman Go tentang tipe data bentukan dan struktur data array. Tipe bentukan mencakup alias

untuk menyederhanakan nama tipe data yang ada dan struct untuk mengelompokkan data terkait menjadi satu kesatuan. Array dibahas sebagai struktur data statis dengan ukuran tetap, sementara slice diperkenalkan sebagai alternatif dinamis yang lebih fleksibel. Map dijelaskan sebagai struktur data asosiatif dengan kunci non-integer.

## **REFERENSI**