

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2  
MODUL 6 STRUCT & ARRAY**



Oleh:

BERTHA ADELA

103112400041 IF-12-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2025**

## I. DASAR TEORI

### 1. Type

Dalam bahasa pemrograman, kata kunci `type` digunakan untuk membuat nama alias dari tipe data yang sudah ada. Ini berguna untuk membuat penamaan menjadi lebih ringkas dan sesuai dengan konteks.

*Contoh:* tipe `int` bisa diberi nama alias seperti `bilangan`.

---

### 2. Struct (atau Record)

Struct digunakan untuk mengelompokkan beberapa data yang saling berhubungan dalam satu kesatuan. Setiap data dalam struct disimpan dalam *field*.

Yang unik di Go, dua struct dianggap memiliki tipe yang sama bukan berdasarkan nama, tapi berdasarkan kesamaan struktur (nama dan urutan field serta tipenya harus sama).

Agar lebih praktis, struct biasanya didefinisikan sebagai tipe baru, sehingga tidak perlu menulis ulang field-nya setiap kali digunakan.

---

### 3. Array

Array adalah kumpulan elemen dengan tipe yang sama dan jumlah elemen yang tetap (statis) selama program dijalankan. Ukuran array ditentukan saat deklarasi dan tidak bisa diubah setelahnya.

---

### 4. Slice

Slice adalah versi dinamis dari array di Go. Jumlah elemennya tidak ditentukan saat deklarasi, sehingga ukurannya bisa berubah-ubah.

Beberapa fungsi built-in yang digunakan pada slice:

- `len(slice)` – untuk mengetahui jumlah elemen saat ini.
- `cap(slice)` – untuk mengetahui kapasitas total tempat yang tersedia.
- `append(slice, elemen)` – untuk menambahkan elemen baru ke slice (otomatis menyesuaikan kapasitas bila diperlukan).

## II. GUIDED

- GUIDED 1 Code:

```
package main

import (
    "fmt"
    "time"
)

// Struct untuk barang dalam struk belanja
type Item struct {
    Name    string
    Price   float64
    Quantity int
}

// Struct untuk struk belanja
type Receipt struct {
    StoreInfo string
    Date      time.Time
    Items     []Item
    TotalAmount float64
}

// Method untuk menghitung total harga semua item
func (r *Receipt) CalculateTotal() {
    var total float64
    for _, item := range r.Items {
        total += item.Price * float64(item.Quantity)
    }
}
```

```

    }
    r.TotalAmount = total
}

// Method untuk mencetak struk belanja
func (r Receipt) PrintReceipt() {
    fmt.Println("=====")
    fmt.Println(r.StoreInfo)
    fmt.Println("Tanggal:", r.Date.Format("02-01-2006 15:04"))
    fmt.Println("=====")
    fmt.Printf("%-15s %-10s %-8s %-10s\n", "Item", "Harga", "Jumlah", "Total")
    fmt.Println("-----")

    for _, item := range r.Items {
        itemTotal := item.Price * float64(item.Quantity)
        fmt.Printf("%-15s Rp%-9.2f %-8d Rp%-9.2f\n", item.Name, item.Price,
item.Quantity, itemTotal)
    }

    fmt.Println("=====")
    fmt.Printf("%-35s Rp%-9.2f\n", "Total Belanja:", r.TotalAmount)
    fmt.Println("=====")
    fmt.Println("Terima kasih telah berbelanja!")
}

func main() {
    receipt := Receipt{
        StoreInfo: "Toko Sembako Makmur\nJl. Raya No. 123, Jakarta",
        Date:      time.Now(),
        Items: []Item{
            {Name: "Beras", Price: 12000, Quantity: 5},
            {Name: "Gula", Price: 15000, Quantity: 2},
            {Name: "Minyak", Price: 20000, Quantity: 1},
            {Name: "Telur", Price: 2000, Quantity: 10},
        },
    }

    receipt.CalculateTotal()
    receipt.PrintReceipt()
}

```

```

=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 29-04-2025 19:53
=====
Item          Harga      Jumlah  Total
-----
Beras         Rp12000.00   5      Rp60000.00
Gula          Rp15000.00   2      Rp30000.00
Minyak        Rp20000.00   1      Rp20000.00
Telur         Rp2000.00   10     Rp20000.00
=====
Total Belanja:                      Rp130000.00
=====
Terima kasih telah berbelanja!

```

### Penjelasan:

Program di atas berfungsi untuk membuat struk belanja di Toko Sembako Makmur, dengan menghitung total harga dari barang-barang yang telah dibeli oleh pelanggan.

- **GUIDED 2 Code:**

```
package main

import (
    "fmt"
)

func main() {
    // Deklarasi dan inisialisasi array nilai mahasiswa
    nilaiMahasiswa := [5]int{85, 90, 78, 88, 95}

    fmt.Println("Data Nilai Mahasiswa:")
    fmt.Println("=====")

    // Menampilkan nilai per mahasiswa
    for i, nilai := range nilaiMahasiswa {
        fmt.Printf("Mahasiswa %d: %d\n", i+1, nilai)
    }

    // Menghitung rata-rata nilai
    var total int
    for _, nilai := range nilaiMahasiswa {
        total += nilai
    }
    rataRata := float64(total) / float64(len(nilaiMahasiswa))

    fmt.Println("=====")
    fmt.Printf("Rata-rata nilai: %.2f\n", rataRata)

    // Mencari nilai tertinggi dan terendah
    tertinggi := nilaiMahasiswa[0]
    terendah := nilaiMahasiswa[0]

    for _, nilai := range nilaiMahasiswa {
        if nilai > tertinggi {
            tertinggi = nilai
        }
        if nilai < terendah {
            terendah = nilai
        }
    }
}
```

```

        terendah = nilai
    }
}

fmt.Printf("Nilai tertinggi: %d\n", tertinggi)
fmt.Printf("Nilai terendah: %d\n", terendah)

// Contoh array 2 dimensi
fmt.Println("\nContoh Array 2 Dimensi:")
fmt.Println("=====")

// Nilai ujian mahasiswa dalam 2 mata kuliah (Matematika, Bahasa)
nilaiUjian := [3][2]int{
    {80, 85},
    {90, 75},
    {70, 95},
}

// Menampilkan nilai ujian per mahasiswa
fmt.Println("Nilai Ujian Mahasiswa (Matematika, Bahasa):")
for i, nilai := range nilaiUjian {
    fmt.Printf("Mahasiswa %d: Matematika = %d, Bahasa = %d\n", i+1,
nilai[0], nilai[1])
}
}

```

```

PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\S
2\ALPRO 2\MODUL6_103112400072\GUIDED2_103112400072\G2.
Data Nilai Mahasiswa:
=====
Mahasiswa 1: 85
Mahasiswa 2: 90
Mahasiswa 3: 78
Mahasiswa 4: 88
Mahasiswa 5: 95
=====
Rata-rata nilai: 87.20
Nilai tertinggi: 95
Nilai terendah: 78

Contoh Array 2 Dimensi:
=====
Nilai Ujian Mahasiswa (Matematika, Bahasa):
Mahasiswa 1: Matematika = 80, Bahasa = 85
Mahasiswa 2: Matematika = 90, Bahasa = 75
Mahasiswa 3: Matematika = 70, Bahasa = 95
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\S

```

### Penjelasan:

Program di atas merupakan contoh penggunaan array dalam Go. Program ini digunakan untuk menghitung rata-rata nilai mahasiswa, menentukan nilai tertinggi dan terendah, serta menampilkan data dalam bentuk array dua dimensi berdasarkan nilai ujian mahasiswa.



### **III. UNGUIDED**

- **UNGUIDED 1**

**Code:**

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y float64
}

type Lingkaran struct {
    titikPusat Titik
    radius    float64
}

func jarak(p, q Titik) float64 {
    return math.Sqrt(math.Pow(p.x-q.x, 2) + math.Pow(p.y-q.y, 2))
}

func didalam(p Titik, l Lingkaran) bool {
    return jarak(p, l.titikPusat) <= l.radius
}

func main() {
```

```

var x1, y1, r1 float64
var x2, y2, r2 float64
var xp, yp float64

fmt.Print("Masukkan x1 y1 r1: ")
fmt.Scan(&x1, &y1, &r1)
fmt.Print("Masukkan x2 y2 r2: ")
fmt.Scan(&x2, &y2, &r2)
fmt.Print("Masukkan xp yp: ")
fmt.Scan(&xp, &yp)

ling1 := Lingkaran{Titik{x1, y1}, r1}
ling2 := Lingkaran{Titik{x2, y2}, r2}
titik := Titik{xp, yp}

in1 := didalam(titik, ling1)
in2 := didalam(titik, ling2)

if in1 && in2 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if in1 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if in2 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

```

Masukkan x1 y1 r1: 1 1 5
Masukkan x2 y2 r2: 8 8 4
Masukkan xp yp: 2 2
Titik di dalam lingkaran 1
PS C:\Users\fazaf\OneDrive\ドキュメント\2\ALPRO 2\MODUL6_103112400072\UG1_103112
Masukkan x1 y1 r1: 1 2 3
Masukkan x2 y2 r2: 4 5 6
Masukkan xp yp: 7 8
Titik di dalam lingkaran 2
PS C:\Users\fazaf\OneDrive\ドキュメント\2\ALPRO 2\MODUL6_103112400072\UG1_103112
Masukkan x1 y1 r1: 5 10 15
Masukkan x2 y2 r2: -15 4 20
Masukkan xp yp: 0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Users\fazaf\OneDrive\ドキュメント\2\ALPRO 2\MODUL6_103112400072\UG1_103112
Masukkan x1 y1 r1: 1 1 5
Masukkan x2 y2 r2: 8 8 4
Masukkan xp yp: 15 20
Titik di luar lingkaran 1 dan 2
PS C:\Users\fazaf\OneDrive\ドキュメント\

```

### Penjelasan:

Program ini berfungsi untuk menentukan apakah suatu titik koordinat berada di dalam lingkaran pertama, lingkaran kedua, keduanya, atau berada di luar kedua lingkaran tersebut.

- **UNGUIDED 2**

**Code:**

```
package main

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("Data ke-%d: ", i)
        fmt.Scan(&arr[i])
    }

    originalArr := make([]int, len(arr))
    copy(originalArr, arr)

    fmt.Println("\n===== HASIL SEMUA OPERASI =====")

    fmt.Println("\na. Semua elemen array:")
    fmt.Println("Isi array:", originalArr)
```

```

fmt.Println("\nb. Elemen dengan indeks ganjil:")
for i := 1; i < len(originalArr); i += 2 {
    fmt.Printf("Index %d: %d\n", i, originalArr[i])
}

fmt.Println("\nc. Elemen dengan indeks genap:")
for i := 0; i < len(originalArr); i += 2 {
    fmt.Printf("Index %d: %d\n", i, originalArr[i])
}

x := 3
fmt.Printf("\nd. Elemen dengan indeks kelipatan %d:\n", x)
for i := 0; i < len(originalArr); i++ {
    if i%x == 0 {
        fmt.Printf("Index %d: %d\n", i, originalArr[i])
    }
}

indexToDelete := 1
fmt.Printf("\ne. Hapus elemen pada indeks %d:\n", indexToDelete)
fmt.Println("Array sebelum dihapus:", originalArr)

arrAfterDelete := make([]int, len(originalArr))
copy(arrAfterDelete, originalArr)

if indexToDelete >= 0 && indexToDelete < len(arrAfterDelete) {
    arrAfterDelete = append(arrAfterDelete[:indexToDelete],
arrAfterDelete[indexToDelete+1:]...)
    fmt.Println("Array setelah dihapus:", arrAfterDelete)
} else {
    fmt.Println("Indeks tidak valid.")
}

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

mean := 0.0

```

```

for _, v := range originalArr {
    mean += float64(v)
}
mean /= float64(len(originalArr))

variance := 0.0
for _, v := range originalArr {
    variance += math.Pow(float64(v)-mean, 2)
}
stddev := math.Sqrt(variance / float64(len(originalArr)))
fmt.Printf("\ng. Standar deviasi elemen: %.2f\n", stddev)

fmt.Println("\nh. Frekuensi setiap bilangan dalam array:")

freqMap := make(map[int]int)
for _, v := range originalArr {
    freqMap[v]++
}

for num, count := range freqMap {
    fmt.Printf("Bilangan %d muncul %d kali\n", num, count)
}

fmt.Println("\nProgram selesai.")
}

```

**Output:**

Masukkan jumlah elemen array (N): 5

Masukkan elemen-elemen array:

Data ke-0: 1

Data ke-1: 2

Data ke-2: 3

Data ke-3: 4

Data ke-4: 5

===== HASIL SEMUA OPERASI =====

a. Semua elemen array:

Isi array: [1 2 3 4 5]

b. Elemen dengan indeks ganjil:

Index 1: 2

Index 3: 4

c. Elemen dengan indeks genap:

Index 0: 1

Index 2: 3

Index 4: 5

d. Elemen dengan indeks kelipatan 3:

Index 0: 1

Index 3: 4

e. Hapus elemen pada indeks 1:

Array sebelum dihapus: [1 2 3 4 5]

Array setelah dihapus: [1 3 4 5]

f. Rata-rata elemen: 3.00

g. Standar deviasi elemen: 1.41

h. Frekuensi setiap bilangan dalam array:

Bilangan 2 muncul 1 kali

Bilangan 3 muncul 1 kali

Bilangan 4 muncul 1 kali

Bilangan 5 muncul 1 kali

Bilangan 1 muncul 1 kali

Program selesai.



### **Penjelasan:**

Program ini digunakan untuk memproses dan menganalisis data dalam sebuah array. Fungsinya meliputi: menampilkan seluruh isi array yang telah diinput, menampilkan elemen-elemen dengan indeks ganjil, indeks genap, serta indeks kelipatan bilangan tertentu (berdasarkan input pengguna). Selain itu, program ini juga dapat menghapus elemen pada indeks tertentu dan memastikan elemen tersebut tidak lagi tampil, menghitung rata-rata nilai dalam array, menghitung standar deviasi (simpangan baku), serta menampilkan frekuensi kemunculan suatu bilangan tertentu dalam array.

### **• UNGUIDED 3**

#### **Code:**

```
package main

import "fmt"

func main() {
    var jumPer int
    fmt.Print("Masukkan jumlah pertandingan: ")
    fmt.Scan(&jumPer)

    var hasil string

    for i := 1; i <= jumPer; i++ {
        var mu, inter int
        fmt.Scan(&mu, &inter)

        fmt.Printf("Pertandingan %d : %d %d\n", i, mu, inter)

        if mu > inter {
            hasil += fmt.Sprintf("Hasil %d : MU\n", i)
        } else if mu < inter {
            hasil += fmt.Sprintf("Hasil %d : Inter\n", i)
        } else {
            hasil += fmt.Sprintf("Hasil %d : Draw\n", i)
        }
    }
```

```

    }

    fmt.Print(hasil)
    fmt.Println("Pertandingan selesai")
}

```

## Output:

```

Masukkan jumlah pertandingan: 9
2 0
Pertandingan 1 : 2 0
1 2
Pertandingan 2 : 1 2
2 2
Pertandingan 3 : 2 2
0 1
Pertandingan 4 : 0 1
3 2
Pertandingan 5 : 3 2
1 0
Pertandingan 6 : 1 0
5 2
Pertandingan 7 : 5 2
2 3
Pertandingan 8 : 2 3
-1 2
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
Hasil 9 : Inter
Pertandingan selesai
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\MODUL6_103112400072>

```

## Penjelasan:

Program ini digunakan untuk menampilkan klub yang menjadi pemenang pada setiap pertandingan yang telah berlangsung.

### • UNGUIDED 4

#### Code:

```
package main
```

```

import (
    "bufio"
    "fmt"
    "os"

```

```

    "strings"
)

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    reader := bufio.NewReader(os.Stdin)
    fmt.Print("Teks: ")
    input, _ := reader.ReadString('\n')
    input = strings.TrimSpace(input)

    *n = len(input)
    for i := 0; i < *n && i < NMAX; i++ {
        (*t)[i] = rune(input[i])
    }
}

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

func palindrom(t tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t[i] != t[n-1-i] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel

```

```

var n int

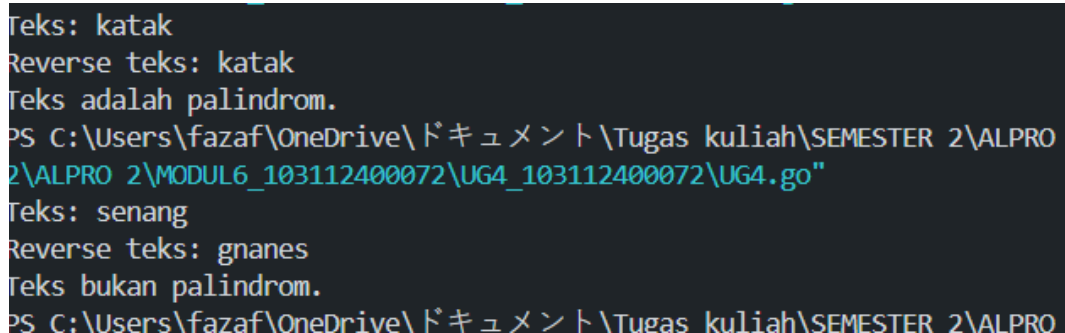
isiArray(&tab, &n)

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

if palindrom(tab, n) {
    fmt.Println("Teks adalah palindrom.")
} else {
    fmt.Println("Teks bukan palindrom.")
}
}

```

### Output:



```

Teks: katak
Reverse teks: katak
Teks adalah palindrom.
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO
2\ALPRO 2\MODUL6_103112400072\UG4_103112400072\UG4.go"
Teks: senang
Reverse teks: gnanes
Teks bukan palindrom.
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO

```

### Penjelasan:

Program di atas digunakan untuk memeriksa apakah teks yang diinputkan merupakan palindrom, yaitu teks yang dibaca sama baik dari depan maupun dari belakang.

#### **IV. KESIMPULAN**

Array memiliki berbagai manfaat, antara lain mempermudah penyusunan data, membuat data lebih terstruktur dan mudah dibaca, memudahkan proses akses terhadap elemen data, menyederhanakan penulisan program tanpa mengurangi estetika, serta meningkatkan efisiensi dalam pengolahan dan penggunaan data.

## **REFERENSI**

MODUL 6 STRUCT & ARRAY