

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 6
“STRUCT AND ARRAY”



DISUSUN OLEH:
SHEILA STEPHANIE ANINDYA
103112400086
S1 IF-12-01
S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

I. Dasar Teori

1) Struct

Struct dalam bahasa Go merupakan fitur yang memungkinkan kita untuk mengelompokkan berbagai data yang saling berkaitan dalam satu wadah. Struct adalah tipe data buatan yang berisi sejumlah field (atau properti) dengan nama dan tipe data masing-masing. Keunikan dari struct ini adalah setiap field bisa memiliki tipe data yang berbeda-beda, sehingga cocok untuk menggambarkan objek nyata yang memiliki berbagai atribut.

Penggunaan struct sangat membantu ketika kita ingin menyimpan data yang terstruktur dan saling berhubungan. Misalnya, jika ingin menyimpan informasi tentang seorang mahasiswa, lebih efisien jika kita menggabungkan nama, umur, NIM, dan status aktif ke dalam satu struct dengan nama yang sesuai, ketimbang membuat variabel terpisah untuk masing-masing informasi. Hal ini membuat program lebih mudah dibaca dan dipelihara, karena data yang berkaitan berada dalam satu unit logis.

Struct juga dapat menyertakan struct lain di dalamnya, sehingga bisa membentuk struktur data yang lebih kompleks. Meskipun Go bukan bahasa yang murni berorientasi objek dan tidak memiliki fitur inheritance seperti pada bahasa lain, struktur data bertingkat tetap bisa dibuat melalui komposisi struct.

Kalau dibandingkan dengan map, struct memiliki sejumlah perbedaan penting. Pada map, kunci (key) dan nilainya bersifat dinamis dan biasanya memiliki tipe data yang seragam untuk seluruh nilai. Sementara itu, field pada struct ditentukan sejak awal (saat penulisan kode), dan masing-masing field bisa memiliki tipe data yang berbeda. Maka dari itu, struct lebih cocok dipakai ketika data yang akan digunakan sudah pasti bentuk dan susunannya, sedangkan map lebih cocok untuk data yang lebih fleksibel dan berubah-ubah.

Struct dalam Go juga bisa dijadikan parameter maupun nilai kembali dari sebuah fungsi. Ini membuat program menjadi lebih modular dan logis, karena kita bisa mengirim dan menerima kumpulan data sebagai satu kesatuan. Dalam berbagai proyek pengembangan, struct sering dimanfaatkan untuk menyusun model data, seperti saat membuat aplikasi web, pengelolaan database, maupun pemrosesan input dari pengguna.

2) Array

Dalam bahasa pemrograman Go, array adalah struktur data yang digunakan untuk menyimpan sekumpulan elemen dengan tipe data yang sama dalam satu variabel. Setiap elemen dalam array diakses menggunakan indeks numerik yang dimulai dari nol. Panjang array ditentukan saat deklarasi dan tidak dapat diubah setelahnya, menjadikan array sebagai struktur data dengan ukuran tetap.

Karakteristik Array dalam Go :

1. Tipe Data Homogen: Semua elemen dalam array harus memiliki tipe data yang sama. Misalnya, array bertipe `int` hanya dapat menyimpan nilai-nilai integer.
2. Ukuran Tetap: Ukuran array ditentukan saat deklarasi dan menjadi bagian dari tipe array tersebut. Oleh karena itu, array `[5]int` berbeda dengan array `[10]int`, meskipun keduanya menyimpan nilai integer.
3. Inisialisasi: Array dapat diinisialisasi saat deklarasi dengan menyebutkan nilai-nilai awalnya. Jika tidak diinisialisasi, elemen-elemen array akan memiliki nilai default sesuai tipe datanya.
4. Akses Elemen: Elemen array diakses menggunakan indeks, dengan indeks pertama adalah 0. Misalnya, `array[0]` mengakses elemen pertama.
5. Imutabilitas Ukuran: Setelah dideklarasikan, ukuran array tidak dapat diubah. Untuk kebutuhan ukuran dinamis, Go menyediakan tipe data `slice`, yang merupakan abstraksi dari array dengan ukuran yang dapat berubah.

Array cocok digunakan ketika jumlah elemen yang akan disimpan sudah diketahui dan tetap. Misalnya, menyimpan nilai-nilai sensor yang jumlahnya tetap atau data statistik dengan jumlah elemen yang pasti.

II. GUIDED

1. Source 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()
}

```

Output:

```

=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 17-04-2025 11:20
=====
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!
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>

```

Penjelasan :

Program tersebut membuat dan mencetak struk belanja dengan menggunakan struct untuk merepresentasikan item dan struk belanja, serta menyediakan metode untuk menghitung total harga dan mencetak struk secara terformat lengkap dengan informasi toko, tanggal transaksi, daftar barang, dan total belanja.

2. Source 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
    }
}

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])
}
}

```

Output:

```

Data Nilai Mahasiswa:
=====
Mahasiswa 1: 85
Mahasiswa 2: 90
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:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>

```

Penjelasan :

Program ini menghitung dan menampilkan data nilai individu, rata-rata, nilai tertinggi, dan terendah dari lima mahasiswa.

III. UNGUIDED

1. Source Code:

```
package main

import "fmt"

type titik struct {
    x, y int
}

type lingkaran struct {
    pusat titik
    jariJari int
}

func (l lingkaran) berisi(t titik) bool {
    selisihX := t.x - l.pusat.x
    selisihY := t.y - l.pusat.y
    jarakKuadrat := selisihX*selisihX + selisihY*selisihY
    jariJariKuadrat := l.jariJari * l.jariJari
    return jarakKuadrat <= jariJariKuadrat
}

func main() {
    var lingkaranMana [2]lingkaran
    var cekTitik titik

    for i := 0; i < 2; i++ {
        fmt.Printf("masukkan x, y pusat dan jari-jari lingkaran ke-%d : ", i+1)
        fmt.Scan(&lingkaranMana[i].pusat.x, &lingkaranMana[i].pusat.y,
&lingkaranMana[i].jariJari)
    }
```

```

fmt.Print("masukkan x dan y titik yang ingin dicek : ")
fmt.Scan(&cekTitik.x, &cekTitik.y)

lingkar1 := lingkaranMana[0].berisi(cekTitik)
lingkar2 := lingkaranMana[1].berisi(cekTitik)

if lingkar1 && lingkar2 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if lingkar1 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if lingkar2 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

Output:

```

masukkan x, y pusat dan jari-jari lingkaran ke-1 : 5 10 15
masukkan x, y pusat dan jari-jari lingkaran ke-2 : -15 4 20
masukkan x dan y titik yang ingin dicek : 0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code> 

```

Penjelasan :

Program ini menentukan posisi suatu titik terhadap dua buah lingkaran berdasarkan input dari pengguna. Program menggunakan struct untuk titik dan lingkaran, menghitung jarak titik ke pusat lingkaran, lalu menentukan posisinya relatif terhadap dua lingkaran yang dimasukkan.

2. Source Code:

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("masukkan jumlah elemen : ")
    fmt.Scan(&n)

    data := make([]int, n)
    var total int

    for i := 0; i < n; i++ {
        fmt.Printf("masukkan elemen ke-%d : ", i+1)
        fmt.Scan(&data[i])
        total += data[i]
    }

    rataRata := float64(total) / float64(n)
    fmt.Printf("\nRata-rata: %.2f\n", rataRata)

    var cariAngka, frekuensi int
    fmt.Print("Masukkan angka yang ingin dicari frekuensinya : ")
    fmt.Scan(&cariAngka)

    for i := 0; i < n; i++ {
        if data[i] == cariAngka {
            frekuensi++
        }
    }

    fmt.Println("Frekuensi angka", cariAngka, ":", frekuensi)
```

```
}
```

Output:

```
masukkan jumlah elemen : 5
masukkan elemen ke-1: 3
masukkan elemen ke-2: 4
masukkan elemen ke-3: 6
masukkan elemen ke-4: 2
masukkan elemen ke-5: 7

Rata-rata: 4.40
Masukkan angka yang ingin dicari frekuensinya: 4
Frekuensi angka 4 : 1
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>
```

Penjelasan :

Program ini meminta input untuk memasukkan sejumlah elemen ke dalam sebuah array, menghitung rata-rata nilai elemen, dan mencari frekuensi kemunculan angka tertentu dalam array tersebut.

3. Source Code:

```
package main

import "fmt"

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

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

    i := 1
    for {
        fmt.Printf("Pertandingan %d : ", i)
```

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

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

    if skorA > skorB {
        hasil = append(hasil, klubA)
    } else if skorB > skorA {
        hasil = append(hasil, klubB)
    } else {
        hasil = append(hasil, "Draw")
    }
    i++
}

for j, p := range hasil {
    fmt.Printf("Hasil %d : %s\n", j+1, p)
}

fmt.Println("Pertandingan selesai")
}
```

Output:

```

PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>> g
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
Pertandingan selesai
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>

```

Penjelasan :

Program ini mencatat hasil pertandingan antara dua klub sepak bola, meminta skor untuk setiap pertandingan hingga pengguna memasukkan skor negatif, yang menandakan berakhirnya input. Setelah itu, program menampilkan hasil setiap pertandingan, yaitu pemenang atau "Draw" jika skor sama, dan mengakhiri program dengan pesan "Pertandingan selesai".

4. Source Code:

```

package main

import "fmt"

const NMAX = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var Teks rune
    *n = 0
    fmt.Print("Teks    : ")
    for {
        fmt.Scanf("%c", &Teks)
    }
}

```

```

    if Teks == '.' || *n >= NMAX {
        break
    }
    if Teks != ' ' && Teks != '\n' {
        (*t)[*n] = Teks
        *n++
    }
}

func balikArray(t tabel, n int) tabel {
    var hasil tabel
    for i := 0; i < n; i++ {
        hasil[i] = t[n-1-i]
    }
    return hasil
}

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, reversed tabel
    var n int

    isiArray(&tab, &n)

    fmt.Print("Teks   : ")

```

```

    for i := 0; i < n; i++ {
        fmt.Printf("%c ", tab[i])
    }
    fmt.Println()

    reversed = balikArray(tab, n)

    fmt.Print("Reverse teks : ")
    for i := 0; i < n; i++ {
        fmt.Printf("%c ", reversed[i])
    }
    fmt.Println()

    if palindrom(tab, n) {
        fmt.Println("Palindrom ? true")
    } else {
        fmt.Println("Palindrom ? false")
    }
}

```

Output:

```

Teks      : S E N A N G .
Teks      : S E N A N G
Reverse teks : G N A N E S
Palindrom ? false
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\

```

Penjelasan :

Program ini meminta input teks dari pengguna, kemudian memeriksa apakah teks tersebut merupakan palindrom (sama dibaca dari depan dan belakang). Selain itu, program juga mengecek apakah teks tersebut palindrom dengan membandingkan karakter-karakter dari awal dan akhir. Jika semua karakter cocok, program mencetak "Palindrom ? true", sebaliknya "Palindrom ? false".

IV. RINGKASAN

Struct di Go adalah tipe data buatan yang berfungsi untuk mengelompokkan beberapa field dengan tipe data berbeda dalam satu kesatuan. Struct sangat berguna untuk merepresentasikan objek nyata yang memiliki berbagai atribut, seperti mahasiswa yang memiliki nama, NIM, umur, dan status aktif.

Dengan menggunakan struct, data yang saling berkaitan dapat disimpan bersama dalam satu entitas logis, membuat kode lebih rapi dan mudah dikelola. Go juga memungkinkan struct di dalam struct lainnya, sehingga bisa membentuk data yang kompleks. Meskipun Go tidak mendukung pewarisan seperti bahasa berorientasi objek, komposisi struct tetap bisa dimanfaatkan untuk membangun struktur data yang kuat.

Struct berbeda dari map, karena field pada struct ditentukan saat kompilasi dan tiap field bisa berbeda tipe, sedangkan map lebih fleksibel namun seragam. Struct juga bisa digunakan dalam fungsi sebagai parameter maupun return value, menjadikannya sangat berguna dalam pemodelan data di berbagai aplikasi.

Sementara Array dalam bahasa Go merupakan struktur data yang menyimpan elemen-elemen dengan tipe data yang sama dalam satu variabel. Ukuran array ditentukan saat deklarasi dan bersifat tetap, sehingga array lebih cocok digunakan untuk data yang jumlahnya sudah diketahui sebelumnya. Setiap elemen dalam array diakses menggunakan indeks yang dimulai dari nol. Karakteristik utama array antara lain: hanya dapat menyimpan data dengan tipe yang sama, ukuran tidak bisa diubah, dan nilai default akan otomatis diisi jika tidak diinisialisasi.

Meskipun array cukup efisien, penggunaannya cenderung terbatas karena sifatnya yang statis. Untuk kebutuhan yang lebih fleksibel, Go menyediakan slice—struktur data yang merupakan abstraksi dari array dan bisa berubah ukuran. Walau begitu, memahami array tetap penting sebagai dasar sebelum mempelajari slice. Array sering digunakan untuk menyimpan data tetap seperti nilai-nilai statistik, data sensor, atau daftar nilai ujian yang jumlahnya diketahui pasti.

V. DAFTAR PUSTAKA

1. Agisti, T. (2020, Januari 16). *Pengenalan Struct di Golang: Struktur Data yang Fleksibel*. Medium.
<https://medium.com/@tiar.agisti/pengenalan-struct-di-golang-struktur-data-yang-fleksibel-e6175f953b2c>
2. Novalagung. (n.d.). *A. Struct*. Dasar Pemrograman Golang.
<https://dasarpemrogramangolang.novalagung.com/A-struct.html>
3. Novalagung. (n.d.). *A. Array*. Dasar Pemrograman Golang.
<https://dasarpemrogramangolang.novalagung.com/A-array.html>