

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 6
“STRUCK & ARRAY”



DISUSUN OLEH:
Muhammad Shabrian Fadly
103112400087
S1 IF-12-01

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

I. DASAR TEORI

STRUCT & ARRAY

Struct adalah kumpulan definisi variabel (atau property) dan atau fungsi (atau method), yang dibungkus sebagai tipe data baru dengan nama tertentu. Property dalam struct, tipe datanya bisa bervariasi. Mirip seperti map, hanya saja key-nya sudah didefinisikan di awal, dan tipe data tiap itemnya bisa berbeda. Dari sebuah struct, kita bisa buat variabel baru, yang memiliki atribut sesuai skema struct tersebut.

Dengan memanfaatkan struct, penyimpanan data yang sifatnya kolektif menjadi lebih mudah, lebih rapi, dan mudah untuk dikelola.

Sementara array digunakan untuk menyimpan beberapa nilai dengan tipe data yang sama ke dalam satu variabel, struct digunakan untuk menyimpan beberapa nilai dengan tipe data berbeda ke dalam satu variabel.

II. GUIDED

Source Code Guided 1:

```
// 103112400087_muhammad shabrian

package main

import (
    "fmt"
    "time"
)

type Item struct {
    Name    string
    Price   float64
    Quantity int
}

type Receipt struct {
    StoreInfo string
    Date       time.Time
    Items      []Item
    TotalAmount float64
}

func (r *Receipt) CalculateTotal() {
    var total float64

    for _, item := range r.Items {
        total += item.Price * float64(item.Quantity)
    }

    r.TotalAmount = total
}
```

```

}

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 :

```

PS D:\Coding manja> go run "d:\Coding manja\semester2\modul6\tempCodeRunnerFile.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 19:27
=====
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!

```

Deskripsi Program:

Program ini menggunakan struct untuk membuat sistem nota belanja. Struct Item menyimpan data nama barang, harga, dan jumlah. Struct Receipt menyimpan informasi toko, waktu pembelian, daftar item (dalam bentuk slice dari Item), dan total belanja. Program menghitung total belanja dengan menjumlahkan hasil kali harga dan jumlah untuk setiap item, lalu mencetak nota lengkap dengan format yang rapi. Ini menunjukkan penerapan struct bersarang dan penggunaan method pada struct di Go.

Source Code Guided 2:

```
// 103112400087_muhammad shabrian

package main

import (
    "fmt"
)

func main() {
    nilaiMahasiswa := [5]int{85, 90, 78, 88, 95}

    fmt.Println("Data Nilai Mahasiswa:")

    fmt.Println("=====")

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

    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)

    fmt.Println("\nContoh Array 2 Dimensi:")
    fmt.Println("=====")
    nilaiUjian := [3][2]int{
        {80, 85},
        {90, 75},
        {70, 95},
    }

    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:

```
PS D:\Coding manja> go run "d:\Coding manja\semester2\modul6\guided2.go"
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
```

Deskripsi Program:

Program ini menggunakan array untuk menyimpan nilai mahasiswa. Setelah menampilkan semua nilai, program menghitung rata-rata, nilai tertinggi, dan nilai terendah. Program juga menampilkan array dua dimensi yang berisi nilai ujian dua mata pelajaran (Matematika dan Bahasa) untuk beberapa mahasiswa.

III. UNGUIDED

Source Code Unguided 1:

```
// 103112400087_muhammad shabrian

package main

import (
    "fmt"
    "math"
)

type titik struct {
    x, y int
}

type lingkaran struct {
    pusat titik
    radius int
}

func jarak(p, q titik) float64 {
    return math.Sqrt(float64((p.x-q.x)*(p.x-q.x) + (p.y-q.y)*(p.y-q.y)))
}

func didalam(c lingkaran, p titik) bool {
    return jarak(c.pusat, p) <= float64(c.radius)
}

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

    fmt.Println("Masukkan pusat dan radius lingkaran 1 (x y r): ")
    fmt.Scan(&l1.pusat.x, &l1.pusat.y, &l1.radius)

    fmt.Println("Masukkan pusat dan radius lingkaran 2 (x y r): ")
}
```

```

fmt.Scan(&l2.pusat.x, &l2.pusat.y, &l2.radius)

fmt.Print("Masukkan titik sembarang (x y): ")

fmt.Scan(&p.x, &p.y)

inL1 := didalam(l1, p)

inL2 := didalam(l2, p)

if inL1 && inL2 {

    fmt.Println("Titik di dalam lingkaran 1 dan 2")

} else if inL1 {

    fmt.Println("Titik di dalam lingkaran 1")

} else if inL2 {

    fmt.Println("Titik di dalam lingkaran 2")

} else {

    fmt.Println("Titik di luar lingkaran 1 dan 2")

}

}

```

Output:

```

PS D:\Coding manja> go run "d:\Coding manja\semester2\modul6\unguided1.go"
Masukkan pusat dan radius lingkaran 1 (x y r): 1 2 3
Masukkan pusat dan radius lingkaran 2 (x y r): 4 5 6
Masukkan titik sembarang (x y): 7 8
Titik di dalam lingkaran 2
PS D:\Coding manja> go run "d:\Coding manja\semester2\modul6\unguided1.go"
Masukkan pusat dan radius lingkaran 1 (x y r): 5 10 15
Masukkan pusat dan radius lingkaran 2 (x y r): -15 4 20
Masukkan titik sembarang (x y): 0 0
Titik di dalam lingkaran 1 dan 2

```

Deskripsi Program: Program ini digunakan untuk menentukan posisi sebuah titik terhadap dua buah lingkaran. Input berupa koordinat pusat dan jari-jari dari dua lingkaran serta koordinat titik sembarang. Program menghitung jarak antara titik dan pusat lingkaran menggunakan rumus Euclidean dan menentukan apakah titik tersebut berada di dalam satu atau kedua lingkaran.

Source Code Unguided 2 :

```
// 103112400087_muhammad shabrian

package main

import "fmt"

func main() {

    var n int

    fmt.Print("Masukkan jumlah elemen: ")

    fmt.Scan(&n)

    arr := make([]int, n)

    fmt.Println("Masukkan elemen:")

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

        fmt.Scan(&arr[i])

    }

    fmt.Println("Isi array:", arr)

    fmt.Print("Indeks ganjil: ")

    for i := 1; i < n; i += 2 {

        fmt.Print(arr[i], " ")

    }

    fmt.Println()

    fmt.Print("Indeks genap: ")

    for i := 0; i < n; i += 2 {

        fmt.Print(arr[i], " ")

    }

    fmt.Println()

}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\semester2\modul6\unguided2.go"
Masukkan jumlah elemen: 4
Masukkan elemen:
1
2
3
4
Isi array: [1 2 3 4]
Indeks ganjil: 2 4
Indeks genap: 1 3
```

Deskripsi Program:

Program ini meminta input sejumlah bilangan bulat ke dalam array, kemudian menampilkan isi array secara lengkap, menampilkan elemen dengan indeks ganjil dan indeks genap secara terpisah.

Source Code unguided 3:

```
// 103112400087_muhammad shabrian

package main

import "fmt"

func main() {

    var klubA, klubB string

    var skorA, skorB int

    var pemenang []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 {

            pemenang = append(pemenang, klubA)

            fmt.Println("Hasil:", klubA)
```

```
    } else if skorB > skorA {  
        pemenang = append(pemenang, klubB)  
        fmt.Println("Hasil:", klubB)  
    } else {  
        pemenang = append(pemenang, "Draw")  
        fmt.Println("Hasil: Draw")  
    }  
    i++  
}  
fmt.Println("\nDaftar klub pemenang:")  
for i, klub := range pemenang {  
    fmt.Printf("Hasil %d: %s\n", i+1, klub)  
}  
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\semester2\modul6\unguided3.go"
Klub A: MU
Klub B: INTER
Pertandingan 1: 2 0
Hasil: MU
Pertandingan 2: 1 2
Hasil: INTER
Pertandingan 3: 2 2
Hasil: Draw
Pertandingan 4: 0 1
Hasil: INTER
Pertandingan 5: 3 2
Hasil: MU
Pertandingan 6: 1 0
Hasil: MU
Pertandingan 7: 5 2
Hasil: MU
Pertandingan 8: 2 3
Hasil: INTER
Pertandingan 9: -1 2

Daftar klub pemenang:
Hasil 1: MU
Hasil 2: INTER
Hasil 3: Draw
Hasil 4: INTER
Hasil 5: MU
Hasil 6: MU
Hasil 7: MU
Hasil 8: INTER
```

Deskripsi Program:

Program ini merekam hasil pertandingan antara dua klub sepak bola. Setiap pertandingan dicatat skor kedua tim, kemudian ditentukan siapa pemenangnya atau jika hasilnya seri. Program akan berhenti menerima input ketika skor yang dimasukkan negatif. Di akhir program, ditampilkan hasil pertandingan dalam urutan yang sesuai.

Source Code unguided 4:

```
// 103112400087_muhammad shabrian

package main

import "fmt"

func isPalindrom(text []rune) bool {
    for i := 0; i < len(text)/2; i++ {
        if text[i] != text[len(text)-1-i] {
            return false
        }
    }
    return true
}

func main() {
    var input string
    fmt.Print("Masukkan teks (akhiri dengan titik): ")
    fmt.Scan(&input)
    text := []rune{}
    for _, ch := range input {
        if ch == '.' {
            break
        }
        text = append(text, ch)
    }

    fmt.Print("Teks dibalik: ")
    for i := len(text) - 1; i >= 0; i-- {
```



```
        fmt.Printf("%c ", text[i])
    }

    fmt.Println()

    fmt.Println("Palindrom?", isPalindrom(text))
}
```

Output:

```
PS D:\Coding manja> go run 'd:\Coding manja\semester2\modul6\unguided4.go'
Masukkan teks (akhiri dengan titik): KATAK
Teks dibalik: K A T A K
Palindrom? true
PS D:\Coding manja> go run 'd:\Coding manja\semester2\modul6\unguided4.go'
Masukkan teks (akhiri dengan titik): SENANG
Teks dibalik: G N A N E S
Palindrom? false
```

Deskripsi Program: Program meminta input karakter satu per satu hingga karakter titik (.) dimasukkan. Setelah itu, karakter ditampilkan dalam urutan terbalik dan dicek apakah membentuk kata palindrom (dibaca dari depan dan belakang sama).

IV. KESIMPULAN

Dalam praktikum ini, mahasiswa mempelajari dan mengimplementasikan dua konsep penting dalam pemrograman menggunakan bahasa Go, yaitu **Struct** dan **Array**. Struct digunakan untuk merepresentasikan data kompleks yang terdiri dari berbagai jenis tipe data dalam satu kesatuan, sedangkan Array digunakan untuk menyimpan data dalam jumlah banyak dengan tipe yang sama.

Melalui latihan yang diberikan, mahasiswa memahami cara menyusun program dengan pendekatan berbasis struktur data dan dapat:

- Mengelola data menggunakan struct dan array.
- Melakukan operasi-operasi dasar terhadap array seperti pencarian, penghapusan, dan perhitungan statistik.
- Menerapkan logika pemrograman dalam menyelesaikan masalah nyata seperti rekap pertandingan dan validasi palindrom.

Secara keseluruhan, praktikum ini meningkatkan kemampuan mahasiswa dalam menyusun program terstruktur dan efisien dengan memanfaatkan struct dan array sebagai dasar pengelolaan data.

V. REFERENSI

https://www.w3schools.com/go/go_struct.php

<https://dasarpemrogramangolang.novalagung.com/A-struct.html>