

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6
STRUCK & ARRAY**



Oleh:

Muhammad Faris Rachmadi

103112400079

IF12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

6.1 Tipe Bentuk

1) Alias (Type)

Alias dalam bahasa pemrograman merujuk pada pemberian nama baru untuk sebuah tipe data yang sudah ada. Hal ini memungkinkan pemrogram untuk membuat nama yang lebih mudah dipahami atau lebih ringkas sesuai kebutuhan aplikasi yang sedang dikembangkan. Dalam hal ini, tipe data yang digunakan tetap sama, tetapi dengan nama yang lebih sesuai. Sebagai contoh, dalam bahasa pemrograman seperti Go, tipe data integer dapat diberikan alias seperti `bilangan` menggunakan kata kunci `type`. Hal ini memudahkan pemrograman, terutama dalam konteks pengembangan aplikasi yang memerlukan tipe data yang lebih bermakna.

2) Struct atau Record

Struct, atau yang sering disebut sebagai record dalam beberapa bahasa pemrograman, adalah tipe data yang memungkinkan pengelompokan beberapa data yang memiliki keterkaitan tertentu menjadi satu kesatuan. Setiap data yang disimpan dalam struct disebut sebagai field. Masing-masing field ini dapat memiliki tipe data yang berbeda, dan dengan menggunakan struktur ini, kita dapat mendeskripsikan objek yang lebih kompleks.

Dalam bahasa pemrograman seperti Go, struktur memungkinkan pembuatan tipe data baru yang terdiri dari beberapa elemen. Elemen-elemen tersebut dapat berupa berbagai tipe data, yang memudahkan representasi data yang lebih kompleks.

Keunikan dari struct adalah bahwa kesamaan tipe antara dua variabel yang berjenis struct tidak hanya dilihat dari nama variabelnya, tetapi lebih kepada urutan dan jenis field yang ada di dalamnya. Dua variabel dengan nama dan tipe field yang sama, serta urutan yang sama, dianggap memiliki tipe yang sama.

3) Array

Array adalah tipe data yang digunakan untuk menyimpan sekumpulan nilai yang bertipe sama dalam satu wadah atau variabel. Setiap elemen dalam array dapat diakses menggunakan indeks yang dimulai dari angka 0. Dalam bahasa pemrograman Go, array didefinisikan dengan menentukan ukuran array dan tipe data elemen yang disimpan.

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()
}

```

Output:

```

PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 6> go run "c:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 26-04-2025 13:23
=====
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 Go ini mencetak struk belanja dengan menggunakan dua struct: Item untuk data barang (nama, harga, jumlah) dan Receipt untuk data struk (informasi toko, tanggal, daftar barang, total belanja). Method CalculateTotal() menghitung total harga, sementara PrintReceipt() mencetak struk dengan format yang rapi. Fungsi main() menyusun struk belanja dengan beberapa item dan mencetaknya setelah perhitungan total dilakukan.

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
        }
    }

    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:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 6> go run "c:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 6\main.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 Go ini mengelola data nilai mahasiswa menggunakan array, menghitung rata-rata nilai, serta mencari nilai tertinggi dan terendah.

Program juga menampilkan contoh array 2 dimensi untuk menyimpan dan menampilkan nilai ujian mahasiswa dalam dua mata kuliah (Matematika dan Bahasa). Setiap langkah dihitung dan ditampilkan dengan format yang jelas.

III. UNGUIDED

Unguided 1

Code:

```
package main

// Muhammad Faris Rachmadi
// 103112400079

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    center 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.center, p) <= float64(c.radius)
}

func main() {
    var c1, c2 Lingkaran
    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r):")
    fmt.Scan(&c1.center.x, &c1.center.y, &c1.radius)

    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r):")
    fmt.Scan(&c2.center.x, &c2.center.y, &c2.radius)

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

    diDalam1 := didalam(c1, p)
    diDalam2 := didalam(c2, p)
```

```

switch {
case diDalam1 && diDalam2:
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
case diDalam1:
    fmt.Println("Titik di dalam lingkaran 1")
case diDalam2:
    fmt.Println("Titik di dalam lingkaran 2")
default:
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

Output:

```

PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 6> go run "c:\Users\Faris\
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

```

Deskripsi:

Program Go ini menentukan apakah sebuah titik berada di dalam satu atau dua lingkaran. Program menggunakan dua struct: Titik untuk menyimpan koordinat titik, dan Lingkaran untuk menyimpan pusat dan radius lingkaran. Fungsi jarak menghitung jarak antara dua titik, sementara fungsi didalam memeriksa apakah titik berada dalam lingkaran berdasarkan jarak ke pusat lingkaran. Program meminta input koordinat untuk dua lingkaran dan satu titik, kemudian menampilkan hasil apakah titik berada di dalam kedua lingkaran, salah satu lingkaran, atau di luar keduanya.

Unguided 2

Code:

```
package main

// Muhammad Faris Rachmadi
// 103112400079

import (
    "fmt"
    "math"
)

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

    arr := make([]int, N)

    for i := 0; i < N; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i+1)
        fmt.Scan(&arr[i])
    }

    fmt.Println("\nIsi 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()

    var sum int
    for i := 0; i < N; i++ {
        sum += arr[i]
    }
    rataRata := float64(sum) / float64(N)
    fmt.Printf("Rata-rata: %.2f\n", rataRata)
```

```

var squaredDiffSum float64
for i := 0; i < N; i++ {
    squaredDiffSum += math.Pow(float64(arr[i])-rataRata, 2)
}
simpanganBakuTanpaAkar := squaredDiffSum / float64(N)
fmt.Printf("Simpangan baku (tanpa akar): %.2f\n",
simpanganBakuTanpaAkar)

var angka int
fmt.Print("Masukkan angka untuk dihitung frekuensinya: ")
fmt.Scan(&angka)
frekuensi := 0
for i := 0; i < N; i++ {
    if arr[i] == angka {
        frekuensi++
    }
}
fmt.Printf("Frekuensi angka %d: %d\n", angka, frekuensi)
}

```

Output:

```

PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 6> go run "c:\Users\Faris\Documents
Masukkan jumlah elemen: 5
Masukkan elemen ke-1: 1
Masukkan elemen ke-2: 2
Masukkan elemen ke-3: 3
Masukkan elemen ke-4: 4
Masukkan elemen ke-5: 5

Isi array: [1 2 3 4 5]
Indeks ganjil: 2 4
Indeks genap: 1 3 5
Rata-rata: 3.00
Simpangan baku (tanpa akar): 2.00
Masukkan angka untuk dihitung frekuensinya: 4
Frekuensi angka 4: 1

```

Deskripsi:

Program Go ini melakukan beberapa operasi statistik dasar pada sebuah array yang diisi oleh pengguna. Pertama, program meminta input jumlah elemen dan elemen-elemen array. Kemudian, program menampilkan elemen dengan indeks ganjil dan genap, menghitung rata-rata nilai dalam array, serta menghitung simpangan baku tanpa akar. Selain itu, program juga meminta pengguna untuk memasukkan angka tertentu dan menghitung frekuensinya dalam array. Semua hasil tersebut ditampilkan dalam format yang mudah dibaca oleh pengguna.

Unguided 3

Code:

```
package main

// Muhammad Faris Rachmadi
// 103112400079

import (
    "fmt"
)

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

    fmt.Print("Masukkan nama klub A: ")
    fmt.Scan(&klubA)
    fmt.Print("Masukkan nama klub B: ")
    fmt.Scan(&klubB)

    fmt.Println("\nPertandingan dimulai!")

    pertandingan = 1
    for {
        fmt.Printf("Pertandingan %d: ", pertandingan)
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            fmt.Println("Pertandingan selesai")
            break
        }

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

    fmt.Println("\nHasil pertandingan:")
}
```

```
for i := 0; i < len(hasil); i++ {  
    fmt.Printf("Hasil %d: %s\n", i+1, hasil[i])  
}  
}
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 6> go run "c:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 6\Unguided\Unguided_  
Masukkan nama klub A: MU  
Masukkan nama klub B: Inter  
  
Pertandingan dimulai!  
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  
Pertandingan selesai  
  
Hasil pertandingan:  
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 Go ini mensimulasikan serangkaian pertandingan antara dua klub sepak bola. Pengguna diminta untuk memasukkan nama klub A dan klub B, kemudian mencatat skor setiap pertandingan hingga pertandingan kesembilan atau hingga ada skor negatif yang menunjukkan pertandingan selesai. Program menyimpan hasil setiap pertandingan, apakah kemenangan klub A, klub B, atau seri (draw). Setelah semua pertandingan selesai, program menampilkan hasil pertandingan untuk setiap ronde yang telah berlangsung.

Unguided 4

Code:

```
package main

// Muhammad Faris Rachmadi
// 103112400079
import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var input string
    fmt.Print("Masukkan teks: ")
    fmt.Scan(&input)
    *n = len(input)

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

func cetakArray(t tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Print(string(t[i]))
    }
    fmt.Println()
}

func balikkanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        t[i], t[n-1-i] = t[n-1-i], t[i]
    }
}

func palindrome(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 m int

    isiArray(&tab, &m)

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

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

    if palindrome(tab, m) {
        fmt.Println("Palindrome : true")
    } else {
        fmt.Println("Palindrome : false")
    }
}

```

Output:

```

PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 6> go run "c:\Users\Faris\Document
Masukkan teks: SENANG
Teks : SENANG
Reverse teks : GNANES
Palindrome : false
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 6> go run "c:\Users\Faris\Document
Masukkan teks: KATAK
Teks : KATAK
Reverse teks : KATAK
Palindrome : true

```

Deskripsi:

Program ini mencetak deret angka berdasarkan aturan berikut: jika angka genap, bagi dengan 2; jika angka ganjil, kalikan dengan 3 dan tambahkan 1. Proses ini terus berlanjut hingga angka mencapai 1. Program meminta input berupa angka dan memeriksa apakah angka tersebut berada dalam rentang 1 hingga 999,999. Jika valid, program akan mencetak deret angka sesuai dengan aturan tersebut. Jika input tidak valid, program akan menampilkan pesan kesalahan.

IV. KESIMPULAN

Praktikum ini mengajarkan penerapan konsep dasar *struct* dan *array* dalam bahasa pemrograman Go. *Struct* digunakan untuk mengelompokkan data yang memiliki hubungan, seperti pada pembuatan struk belanja dan pengecekan posisi titik dalam lingkaran, sementara *array* digunakan untuk menyimpan dan memproses sekumpulan data bertipe sama, seperti nilai mahasiswa dan hasil pertandingan. Dengan memanfaatkan kedua struktur data ini, praktikum ini menunjukkan bagaimana mengelola data kompleks, menghitung statistik, dan menyelesaikan masalah praktis dengan cara yang lebih terstruktur dan efisien.

V. REFRENSI

MODUL 7 PRAKTIKUM ALGORITMA PEMROGRAMAN

2 –STRUCK DAN ARRAY