

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 7
STRUCT DAN ARRAY



Oleh:

Pratama Bintang Daniswara

103112400051

12 IF 01

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

I. Dasar Teori

- **Array**

Array adalah struktur data yang menyimpan sekumpulan elemen dengan tipe data yang sama dan ukuran tetap. Artinya, jumlah elemen harus ditentukan di awal dan tidak bisa diubah selama program berjalan. Setiap elemen dalam array memiliki indeks yang dimulai dari nol, sehingga kita bisa mengakses atau mengubah nilai elemen tertentu berdasarkan posisinya. Dalam praktiknya, array cocok digunakan ketika jumlah data sudah pasti dan tidak berubah-ubah.

- **Struct**

Struct (structure) adalah tipe data bentukan yang digunakan untuk mengelompokkan beberapa data dengan tipe yang berbeda menjadi satu kesatuan. Struct berguna untuk merepresentasikan objek atau entitas nyata yang memiliki beberapa atribut, seperti data mahasiswa yang terdiri dari nama, umur, dan jurusan. Dengan struct, kita dapat menyimpan dan mengorganisasi data secara lebih terstruktur dan mudah diakses berdasarkan nama field-nya.

- **Slice**

Slice adalah versi fleksibel dari array yang ukurannya dapat berubah selama program berjalan. Berbeda dengan array, slice tidak perlu dideklarasikan dengan ukuran tetap dan dapat bertambah atau berkurang sesuai kebutuhan. Slice sebenarnya adalah referensi ke bagian dari array di belakang layar, sehingga lebih hemat memori dan efisien untuk manipulasi data. Slice sangat umum digunakan dalam bahasa Go karena lebih praktis dibanding array biasa.

- **Map**

Map adalah struktur data yang menyimpan pasangan key-value, di mana setiap nilai (value) dipetakan dengan sebuah kunci unik (key). Map digunakan ketika kita ingin mengakses data berdasarkan nama atau identitas tertentu, bukan berdasarkan urutan atau indeks. Misalnya, kita bisa menyimpan nilai siswa berdasarkan nama mereka sebagai key. Map sangat efisien untuk pencarian data karena menggunakan hashing di belakang layar.

II. Guided

Contoh 1

```
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 D:\Coding\ALPRO\103112400051_MODUL6> go run "
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 17:06
=====
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 D:\Coding\ALPRO\103112400051_MODUL6>

```

Deskripsi : Program ini ditulis dengan bahasa Go untuk mencetak struk belanja. Ada dua struktur utama: Item untuk menyimpan data barang, dan Receipt untuk menyimpan info toko, tanggal, daftar belanja, serta total harganya. Fungsi CalculateTotal() dipakai untuk menghitung total belanja, lalu PrintReceipt() mencetak struk ke layar dengan format yang rapi. Di bagian main(), data dimasukkan lalu diproses dan ditampilkan.

Contoh 2

```
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 D:\Coding\ALPRO\103112400051_MODUL6> go r
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 D:\Coding\ALPRO\103112400051_MODUL6> 

```

Deskripsi : Program ini menampilkan data nilai lima mahasiswa, menghitung rata-rata nilai, serta mencari nilai tertinggi dan terendah menggunakan array satu dimensi. Selanjutnya, program juga memberikan contoh penggunaan array dua dimensi untuk menyimpan dan menampilkan nilai ujian dua mata kuliah (Matematika dan Bahasa) dari tiga mahasiswa secara terstruktur.

III. Unguided

Soal 1

```
package main

import "fmt"

type Titik struct {
    x,y int
}

type Lingkaran struct {
    pusat Titik
    r int
}

func jarakKuadrat(a, b Titik) int {
    dx := a.x - b.x
    dy := a.y - b.y
    return dx*dx + dy*dy
}

func dalamLingkaran(l Lingkaran, t Titik) bool {
    return jarakKuadrat(l.pusat, t) <= l.r*l.r
}

func main() {
    var l1, l2 Lingkaran
    var titik Titik

    fmt.Scan(&l1.pusat.x, &l1.pusat.y, &l1.r)
    fmt.Scan(&l2.pusat.x, &l2.pusat.y, &l2.r)
    fmt.Scan(&titik.x, &titik.y)

    inL1 := dalamLingkaran(l1, titik)
    inL2 := dalamLingkaran(l2, titik)

    switch {
    case inL1 && inL2:
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    case inL1:
        fmt.Println("Titik di dalam lingkaran 1")
    case inL2:
        fmt.Println("Titik di dalam lingkaran 2")
    default:
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

Output :

```
PS D:\Coding\ALPRO\103112400051_MODUL6> go
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\Coding\ALPRO\103112400051_MODUL6> go
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS D:\Coding\ALPRO\103112400051_MODUL6> go
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS D:\Coding\ALPRO\103112400051_MODUL6> go
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
PS D:\Coding\ALPRO\103112400051_MODUL6> 
```

Deskripsi : Program ini dibuat untuk mengecek apakah sebuah titik berada di dalam, di luar, atau tepat di salah satu dari dua lingkaran. Pertama, pengguna diminta memasukkan data untuk dua lingkaran (pusat dan jari-jari) serta koordinat sebuah titik. Program lalu menghitung jarak kuadrat antara titik dan pusat masing-masing lingkaran, dan membandingkannya dengan kuadrat jari-jari lingkaran tersebut. Berdasarkan hasil perbandingan itu, program akan mencetak apakah titik tersebut berada di dalam lingkaran pertama, kedua, keduanya, atau tidak berada di dalam lingkaran manapun.

Soal 2

```
package main

import "fmt"

func main() {
    const maxSize = 100
    var arr [maxSize]int
    var n, x, index, num int

    fmt.Print("Masukkan jumlah elemen : ")
    fmt.Scan(&n)

    fmt.Println("Masukkan", n, "elemen array:")
    for i := 0; i < n; i++ {
        fmt.Scan(&arr[i])
    }

    // a. Tampilkan seluruh array
    fmt.Println("\na. Isi array lengkap:")
    for i := 0; i < n; i++ {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    // b. Indeks ganjil
    fmt.Println("\nb. Elemen indeks ganjil:")
    for i := 1; i < n; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    // c. Indeks genap
    fmt.Println("\nc. Elemen indeks genap:")
    for i := 0; i < n; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    // d. Kelipatan x
    fmt.Print("\nd. Masukkan nilai x untuk indeks kelipatan: ")
    fmt.Scan(&x)
    fmt.Println("Elemen dengan indeks kelipatan", x, ":")
    for i := 0; i < n; i++ {
        if i%x == 0 {
            fmt.Print(arr[i], " ")
        }
    }
    fmt.Println()
}
```

```

// e. Hapus elemen
fmt.Print("\ne. Masukkan indeks yang akan dihapus (0-", n-1, "): ")
fmt.Scan(&index)
for i := index; i < n-1; i++ {
    arr[i] = arr[i+1]
}
n--
fmt.Println("Array setelah penghapusan:")
for i := 0; i < n; i++ {
    fmt.Print(arr[i], " ")
}
fmt.Println()

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

// g. Standar deviasi
var variance float64
avg := float64(sum) / float64(n)
for i := 0; i < n; i++ {
    diff := float64(arr[i]) - avg
    variance += diff * diff
}
fmt.Printf("g. Standar deviasi : %.2f\n", sqrtApprox(variance/float64(n)))

// h. Frekuensi bilangan
fmt.Print("\nh. Masukkan bilangan yang dicari: ")
fmt.Scan(&num)
count := 0
for i := 0; i < n; i++ {
    if arr[i] == num {
        count++
    }
}
fmt.Println("Frekuensi", num, ":", count)
}

func sqrtApprox(x float64) float64 {
    z := 1.0
    for i := 0; i < 10; i++ {
        z -= (z*z - x) / (2 * z)
    }
    return z
}

```

Output :

```
PS D:\Coding\ALPRO\103112400051_MODUL6> go run
Masukkan jumlah elemen : 4
Masukkan 4 elemen array:
1 4 5 3

a. Isi array lengkap:
1 4 5 3

b. Elemen indeks ganjil:
4 3

c. Elemen indeks genap:
1 5

d. Masukkan nilai x untuk indeks kelipatan: 2
Elemen dengan indeks kelipatan 2 :
1 5

e. Masukkan indeks yang akan dihapus (0-3): 2
Array setelah penghapusan:
1 4 3

f. Rata-rata: 2.67
g. Standar deviasi : 1.25

h. Masukkan bilangan yang dicari: 1
Frekuensi 1 : 1
PS D:\Coding\ALPRO\103112400051_MODUL6> |
```

Deskripsi : Program ini membantu pengguna mengelola data dalam bentuk array melalui berbagai fitur interaktif. Setelah memasukkan sejumlah angka, pengguna bisa melihat isi array secara lengkap, memisahkan elemen berdasarkan posisi ganjil atau genap, dan menampilkan elemen pada indeks kelipatan tertentu. Selain itu, pengguna dapat menghapus elemen di indeks tertentu, lalu melihat hasil array yang sudah diperbarui. Program juga menghitung rata-rata dan standar deviasi dari nilai-nilai dalam array, serta memberi tahu seberapa sering sebuah angka muncul.

Soal 3

```
package main

import "fmt"

func main() {
    var klub [2]string
    var pemenang [100]string
    var skorA, skorB int
    var i int

    fmt.Print("Klub A : ")
    fmt.Scan(&klub[0])

    fmt.Print("Klub B : ")
    fmt.Scan(&klub[1])

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

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

        if skorA > skorB {
            pemenang[i] = klub[0]
        } else if skorB > skorA {
            pemenang[i] = klub[1]
        } else {
            pemenang[i] = "Draw"
        }
        i++
    }

    fmt.Println()
    for j := 0; j < i; j++ {
        fmt.Printf("Hasil %d : %s\n", j+1, pemenang[j])
    }
    fmt.Println("Pertandingan selesai")
}
```

Output :

```
PS D:\Coding\ALPRO\103112400051_MODUL6> go
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 D:\Coding\ALPRO\103112400051_MODUL6> |
```

Deskripsi : Program ini dibuat untuk mencatat hasil pertandingan antara dua klub sepak bola dengan cara yang simpel dan interaktif. Pengguna cukup memasukkan nama kedua klub, lalu memasukkan skor pertandingan satu per satu. Program otomatis menentukan siapa pemenangnya di setiap pertandingan, atau mencatatnya sebagai "Draw" kalau skornya imbang. Input berakhir ketika salah satu skor bernilai negatif, dan setelah itu, semua hasil pertandingan langsung ditampilkan sebagai rekap.

Soal 4

```
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var char rune
    *n = 0
    fmt.Print("Teks  :")
    for {
        _, err := fmt.Scanf("%c", &char)
        if err != nil || char == '!' || *n >= NMAX {
            break
        }
        if char != ' ' && char != '\n' && char != '\r' {
            t[*n] = char
            fmt.Printf(" %c", char)
            *n++
        }
    }
    fmt.Println()
}

func cetakArray(t tabel, n int, label string) {
    fmt.Print(label)
    for i := 0; i < n; i++ {
        fmt.Printf(" %c", 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 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, tabAsli tabel
    var m int

    // Isi array tab
    isiArray(&tab, &m)

    // Simpan versi asli
    copy(tabAsli[:], tab[:m])

    // Periksa palindrom
    fmt.Print("Palindrom  ? ", palindrom(tab, m), "\n\n")

    // Balikkan array dan tampilkan
    balikkanArray(&tab, m)
    cetakArray(tab, m, "Reverse teks  :")
}

```

Output :

```

PS D:\Coding\ALPRO\103112400051_MODUL6> go
Teks      :K A T A K .
  K A T A K
Palindrom  ? true

Reverse teks  : K A T A K
PS D:\Coding\ALPRO\103112400051_MODUL6> go
Teks      :S E N A N G .
  S E N A N G
Palindrom  ? false

Reverse teks  : G N A N E S
PS D:\Coding\ALPRO\103112400051_MODUL6>

```

Deskripsi : Program ini dibuat untuk membaca teks dari pengguna hingga mengetikkan tanda titik (.) sebagai penanda akhir. Teks yang dimasukkan akan disaring dari spasi dan enter, lalu dicek apakah merupakan palindrom—yaitu apakah teks tersebut terbaca sama dari depan maupun belakang. Setelah itu, program juga menampilkan versi terbalik dari teks yang sudah dimasukkan.

IV. Kesimpulan

V. Referensi

- MODUL 7. STRUCK & ARRAY