

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6
STRUCT ARRAY**



Oleh:

Achmad Zulvan Nur Hakim

103112400070

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

1. Struct

Struct adalah tipe bentukan (composite type) dalam bahasa Go yang digunakan untuk mengelompokkan beberapa data yang saling berkaitan dalam satu kesatuan. Struct sangat berguna untuk merepresentasikan objek nyata yang memiliki beberapa atribut.

Kelebihan Struct

- Memudahkan pengelompokan data yang memiliki relasi.
- Meningkatkan keterbacaan dan organisasi kode.
- Bisa dipakai sebagai parameter maupun return value dalam fungsi.

2. Array

Array adalah struktur data yang menyimpan sejumlah elemen dengan tipe data yang sama dan memiliki ukuran tetap. Dalam Go, ukuran array harus didefinisikan saat deklarasi, dan tidak dapat diubah setelahnya.

II. GUIDED

Code 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:\103112400070> go run "d:\103112400070\103112400070_Modul6\Guided\1\1.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 21:48
=====
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 ini adalah aplikasi untuk mencetak struk belanja yang mencakup informasi tentang toko, tanggal transaksi, daftar barang yang dibeli, harga, jumlah, dan total belanja.

Code 2:

```
package main

import (
    "fmt"
)

func main() {
    // Deklarasi dan inisialisasi array nilai mahasiswa
    nilaiMahasiswa := []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:\103112400070> go run "d:\103112400070\103112400070_Modul6\Guided\2\2.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

```

Penjelasan:

Program ini menghitung rata-rata, nilai tertinggi, dan terendah dari data nilai mahasiswa, serta menampilkan hasilnya. Selain itu, program juga menggunakan array dua dimensi untuk menyimpan dan menampilkan nilai ujian di dua mata kuliah untuk setiap mahasiswa.

III. UNGUIDED

Code 1:

```
//Achmad Zulvan Nur Hakim 103112400070
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type L struct {
    cen Titik
    r int
}

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

func didalam(l L, p Titik) bool {
    return jarak(p, l.cen) <= float64(l.r)
}

func main() {
    var L1, L2 L
    var titik Titik

    fmt.Scan(&L1.cen.x, &L1.cen.y, &L1.r)
    fmt.Scan(&L2.cen.x, &L2.cen.y, &L2.r)
    fmt.Scan(&titik.x, &titik.y)

    dlmL1 := didalam(L1, titik)
    dlmL2 := didalam(L2, titik)

    if dlmL1 && dlmL2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dlmL1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if dlmL2 {
        fmt.Println("Titik di dalam lingkaran 2")
    }
}
```

```

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

```

Output:

```

PS D:\103112400070> go run "d:\103112400070\103112400070_Modul6\Unguided\1\1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\103112400070> go run "d:\103112400070\103112400070_Modul6\Unguided\1\1.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS D:\103112400070> go run "d:\103112400070\103112400070_Modul6\Unguided\1\1.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS D:\103112400070> go run "d:\103112400070\103112400070_Modul6\Unguided\1\1.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2

```

Penjelasan:

Program ini menentukan apakah sebuah titik berada di dalam satu atau lebih lingkaran berdasarkan koordinat pusat dan jari-jari lingkaran. Program ini menggunakan tipe data struct untuk merepresentasikan titik dan lingkaran, serta fungsi untuk menghitung jarak dan mengecek apakah titik berada dalam lingkaran.

Code 2:

```
//Achmad Zulnan Nur Hakim 103112400070
package main

import (
    "fmt"
    "math"
)

func main() {
    numb := []int{1, 2, 3, 4, 5, 6, 7, 8, 9, 0}

    tampilkanArray := func(label string, arr []int) {
        fmt.Printf("%s: ", label)
        for i, n := range arr {
            if i > 0 {
                fmt.Print(" ")
            }
            fmt.Print(n)
        }
        fmt.Println()
    }

    tampilkanArray("Array awal", numb)

    fmt.Print("Indeks ganjil: ")
    for i := 1; i < len(numb); i += 2 {
        if i > 1 {
            fmt.Print(" ")
        }
        fmt.Print(numb[i])
    }
    fmt.Println()

    fmt.Print("Indeks genap: ")
    for i := 0; i < len(numb); i += 2 {
        if i > 0 {
            fmt.Print(" ")
        }
        fmt.Print(numb[i])
    }
    fmt.Println()

    x := 3
    fmt.Printf("Indeks kelipatan %d: ", x)
```

```

    pertama := true
    for i := 0; i < len(numb); i++ {
        if i%x == 0 {
            if !pertama {
                fmt.Print(" ")
            }
            fmt.Print(numb[i])
            pertama = false
        }
    }
    fmt.Println()

    del := 4
    if del >= 0 && del < len(numb) {
        numb = append(numb[:del], numb[del+1:]...)
        tampilkanArray(fmt.Sprintf("Setelah hapus indeks %d", del),
numb)
    } else {
        fmt.Println("Indeks tidak nvalid - tidak ada yang dihapus")
    }

    if len(numb) > 0 {
        total := 0
        for _, n := range numb {
            total += n
        }
        rata := float64(total) / float64(len(numb))
        fmt.Printf("Rata-rata: %.2f\n", rata)

        var sum float64
        for _, n := range numb {
            selisih := float64(n) - rata
            sum += selisih * selisih
        }
        deniasi := math.Sqrt(sum / float64(len(numb)))
        fmt.Printf("Standar deviasi: %.2f\n", deniasi)

        min, max := numb[0], numb[0]
        for _, n := range numb {
            if n < min {
                min = n
            }
            if n > max {
                max = n
            }
        }
    }

```

```

        fmt.Printf("Nilai minimum: %d\n", min)
        fmt.Printf("Nilai maksimum: %d\n", max)
    }

    find := 0
    freq := 0
    for _, n := range numb {
        if n == find {
            freq++
        }
    }
    fmt.Printf("Frekuensi angka %d: %d kali\n", find, freq)
}

```

Output:

```

PS D:\103112400070> go run "d:\103112400070\103112400070_Modul6\Unguided\2\2.go"
Array awal: 1 2 3 4 5 6 7 8 9 0
Indeks ganjil: 2 4 6 8 0
Indeks genap: 1 3 5 7 9
Indeks kelipatan 3: 1 4 7 0
Setelah hapus indeks 4: 1 2 3 4 6 7 8 9 0
Rata-rata: 4.44
Standar deviasi: 3.02
Nilai minimum: 0
Nilai maksimum: 9
Frekuensi angka 0: 1 kali

```

Penjelasan:

Program ini bekerja dengan sebuah array integer dan melakukan beberapa operasi untuk menganalisis dan memodifikasi data dalam array tersebut. Program ini mencetak nilai array, memisahkan angka berdasarkan indeks ganjil dan genap, mencari angka pada indeks kelipatan tertentu, menghapus elemen di indeks tertentu, serta menghitung rata-rata, standar deviasi, nilai minimum, maksimum, dan frekuensi kemunculan angka dalam array.

Code 3:

```
//Achmad Zulvan Nur Hakim 103112400070
package main

import "fmt"

func main() {
    var klubA, klubB string
    fmt.Print("Klub A : ")
    fmt.Scanln(&klubA)
    fmt.Print("Klub B : ")
    fmt.Scanln(&klubB)
    fmt.Println()
    var skorA, skorB []int
    pertandinganKe := 1

    for {
        var a, b int
        fmt.Printf("Pertandingan %d : ", pertandinganKe)
        _, err := fmt.Scanln(&a, &b)

        if err != nil || a < 0 || b < 0 {
            break
        }

        skorA = append(skorA, a)
        skorB = append(skorB, b)
        pertandinganKe++
    }

    for i := 0; i < len(skorA); i++ {
        if skorA[i] > skorB[i] {
            fmt.Printf("Hasil %d : %s\n", i+1, klubA)
        } else if skorB[i] > skorA[i] {
            fmt.Printf("Hasil %d : %s\n", i+1, klubB)
        } else {
            fmt.Printf("Hasil %d : Draw\n", i+1)
        }
    }

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

Output:

```
PS D:\103112400070> go run "d:\103112400070\103112400070_Modul6\Unguided\3\3.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
```

Penjelasan:

Program ini meminta input nama dua klub sepak bola dan mencatat skor setiap pertandingan antara keduanya. Setelah mencatat semua hasil pertandingan yang valid, program akan mencetak hasil pertandingan yang menunjukkan klub mana yang menang atau jika terjadi seri, serta memberi tahu bahwa pertandingan telah selesai.

Code 4:

```
//Achmad Zulvan Nur Hakim 103112400070
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

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

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

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

func isPalindrom(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

    fmt.Print("Teks : ")
    isiArray(&tab, &m)

    copyTab := tab

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

    fmt.Printf("Palindrom ? %t\n", isPalindrom(copyTab, m))
}

```

Output:

```

PS D:\103112400070> go run "d:\103112400070\103112400070_Modul6\Unguided\4\4.go"
Teks : K A T A K.
Reverse : K A T A K
Palindrom ? true
PS D:\103112400070> go run "d:\103112400070\103112400070_Modul6\Unguided\4\4.go"
Teks : S E N A N G.
Reverse : G N A N E S
Palindrom ? false

```

Penjelasan:

Program ini membaca karakter satu per satu dari input hingga menemukan tanda titik (.), lalu menyimpan karakter tersebut ke dalam array tanpa menyertakan spasi atau baris baru. Setelah itu, program membalik urutan karakter dan mencetaknya, serta memeriksa apakah karakter yang dimasukkan membentuk sebuah palindrom (teks yang dibaca dari depan dan belakang tetap sama).

IV. KESIMPULAN

V. REFERENSI

MODUL 6. Struct dan Array Algoritma dan Pemrograman 2