

**‘ LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 6

STRUCT DAN ARRAY



Oleh:

SAVILA NUR FADILLA

103112400031

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

6.1 Tipe Bentuk

Tipe bentuk memungkinkan pemrograman mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipe bentuk dapat dibedakan jadi dua jenis, yaitu Alias dan Struct.

1) Alias (Type)

Bahasa pemrograman pada umumnya mengizinkan pemrograman untuk mengubah nama suatu tipe data dengan nama baru yang lebih ringkas dan familiar. Sebagai contoh "integer" dapat dirubah dengan nama alias "bilangan ". Caranya dengan menggunakan kata kunci "type".

```
package main
```

```
type <nama alias> <tipe data>
```

```
func main(){  
...  
}
```

2) Struct atau Record

Stucture memungkinkan pemrograman mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari stucture tersebut.

```
package main
```

```
type <nama struct> struct {  
    <field 1> <tipe data>  
    <field 2> <tipe data>  
    <field 3> <tipe data>  
}
```

6.2 Array

Array adalah kumpulan data bertipe sama, yang disimpan dalam sebuah variabel. Array memiliki kapasitas yang nilainya ditentukan saat pembuatan, menjadikan elemen/data yang disimpan di array tersebut jumlahnya tidak boleh melebihi yang sudah dialokasikan. Default nilai tiap elemen array pada awalnya tergantung dari tipe datanya. Jika int maka tiap element zero value-nya adalah 0, jika bool maka false, dan seterusnya. Setiap elemen array

memiliki indeks berupa angka yang merepresentasikan posisi urutan elemen tersebut. Indeks array dimulai dari 0.

6.3 Slice

Slice adalah *reference* elemen array. Slice bisa dibuat, atau bisa juga dihasilkan dari manipulasi sebuah array ataupun slice lainnya. Karena slice merupakan data *reference*, menjadikan perubahan data di tiap elemen slice akan berdampak pada slice lain yang memiliki alamat memori yang sama.

6.4 Map

Map adalah tipe data asosiatif yang ada di Go berbentuk *key-value pair*. Data/value yang disimpan di map selalu disertai dengan key. Key harus unik, karena digunakan sebagai penanda untuk pengaksesan value yang disimpan di map. Map mirip seperti slice, hanya saja penanda yang digunakan untuk pengaksesan bukanlah index numerik, melainkan bisa dalam tipe data apapun sesuai dengan yang diinginkan.

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 :

```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL6> go run "c:\103112400031_MODUL6\103112400031_guided1.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 23-04-2025 19:44
=====
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 bertujuan untuk mensimulasikan pembuatan struk belanja sederhana menggunakan bahasa Go. data barang (nama, harga, jumlah) langsung dimasukkan dalam kode program (bukan input dari pengguna, tapi program ini dirancang agar bisa menerima input dari pengguna. Dalam program ada dua struct utama, yaitu struct untuk barang dalam struk belanja dan struct untuk struct belanja. Data barang dimasukkan ke dalam program dan program akan menghitung total belanja. Output akan menampilkan struk 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 :

```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL6> go run "c:\103112400031_MODUL6\103112400031_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

```


Penjelasan : Program ini bertujuan untuk menampilkan, menghitung, dan menganalisis data nilai mahasiswa menggunakan array satu dimensi dan dua dimensi dalam bahasa Go. Program ini tidak meminta input dari pengguna karena dalam program sudah tertera nilai nilainya. Program akan menghitung rata rata nilai dan menentukan nilai tertinggi dan terendah. Output berupa daftar nilai mahasiswa, rata rata nilai, nilai tertinggi, nilai terendah, dan ada juga contoh array dua dimensinya (nilai ujian mahasiswa untuk dua mata kuliah dalam bentuk array dua dimensi).

III. UNGUIDED

1.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    pusat Titik
    r    int
}

// fungsi untuk menghitung jarak antara dua titik
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))
}

// fungsi untuk mengecek apakah titik berada di dalam lingkaran
func didalam(c Lingkaran, p Titik) bool {
    return jarak(c.pusat, p) <= float64(c.r)
}

// fungsi utama
func main() {
    var cx1, cy1, r1 int
    var cx2, cy2, r2 int
    var x, y int

    fmt.Scan(&cx1, &cy1, &r1)
    fmt.Scan(&cx2, &cy2, &r2)
    fmt.Scan(&x, &y)

    ling1 := Lingkaran{Titik{cx1, cy1}, r1}
```

```

ling2 := Lingkaran{Titik{cx2, cy2}, r2}
titik := Titik{x, y}

diLingkaran1 := didalam(ling1, titik)
diLingkaran2 := didalam(ling2, titik)

if diLingkaran1 && diLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if diLingkaran1 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if diLingkaran2 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

Output :

```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL6> go run "c:\103112400031_MODUL6\10311
2400031_unguided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\103112400031_MODUL6> go run "c:\103112400031_MODUL6\103112400031_unguided1.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\103112400031_MODUL6> go run "c:\103112400031_MODUL6\103112400031_unguided1.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS C:\103112400031_MODUL6> go run "c:\103112400031_MODUL6\103112400031_unguided1.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2

```

Penjelasan : Program ini bertujuan untuk menentukan posisi titik. Program meminta kita menginput inputan yang terdiri dari 3 baris, baris pertama koordinat pusat dan jari-jari lingkaran pertama (cx1, cy1, r1), baris kedua koordinat pusat dan jari-jari lingkaran kedua (cx2, cy2, r2), baris ketiga koordinat titik uji (x, y). Ada dua struct (Titik dan Lingkaran). Kemudian ada fungsi jarak untuk menghitung jarak antara dua titik dan fungsi diDalam untuk mengecek apakah titik berada di dalam lingkaran. Output berupa keterangan posisi titik ada di dalam lingkaran 1, lingkaran 2, keduanya, atau berada di luar lingkaran.

2.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import (
    "fmt"
    "math"
)

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

    array := make([]int, n)
    for i := range array {
        fmt.Printf("Elemen [%v]: ", i)
        fmt.Scan(&array[i])
    }

    fmt.Println("\nSemua elemen:", array)

    fmt.Print("Indeks ganjil: ")
    for i := 1; i < n; i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()

    fmt.Print("Indeks genap: ")
    for i := 0; i < n; i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()

    var x int
    fmt.Print("\nKelipatan indeks x: ")
    fmt.Scan(&x)
    for i := 0; i < n; i++ {
        if i%x == 0 {
            fmt.Print(array[i], " ")
        }
    }
}
```

```

fmt.Println()

var hapus int
fmt.Print("\nIndeks yang dihapus: ")
fmt.Scan(&hapus)
if hapus >= 0 && hapus < len(array) {
    array = append(array[:hapus], array[hapus+1:]...)
}
fmt.Println("array setelah hapus:", array)

total := 0
for _, v := range array {
    total += v
}
rata := float64(total) / float64(len(array))
fmt.Printf("\nRata-rata: %.2f\n", rata)

var deviasi float64
for _, v := range array {
    deviasi += math.Pow(float64(v)-rata, 2)
}
fmt.Printf("Standar deviasi: %.2f\n",
math.Sqrt(deviasi/float64(len(array))))

var cari, frek int
fmt.Print("\nBilangan yang dicari frekuensinya: ")
fmt.Scan(&cari)
for _, v := range array {
    if v == cari {
        frek++
    }
}
fmt.Printf("Frekuensi %v: %v kali\n", cari, frek)
}

```

Output :

```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL6> go run "c:\103112400031_MODUL6\103112400031_unguided2.go"
Jumlah elemen: 5
Elemen [0]: 20
Elemen [1]: 30
Elemen [2]: 10
Elemen [3]: 40
Elemen [4]: 50

Semua elemen: [20 30 10 40 50]
Indeks ganjil: 30 40
Indeks genap: 20 10 50

Kelipatan indeks x: 2
20 10 50

Indeks yang dihapus: 1
array setelah hapus: [20 10 40 50]

Rata-rata: 30.00
Standar deviasi: 15.81

Bilangan yang dicari frekuensinya: 50
Frekuensi 50: 1 kali
```

Penjelasan : Program ini bertujuan untuk mengisi array sebanyak N elemen nilai. Asumsikan array memiliki kapasitas penyimpanan data sejumlah elemen tertentu. Program dapat menampilkan beberapa informasi berikut:

- a. Menampilkan keseluruhan isi dari array.
- b. Menampilkan elemen-elemen array dengan indeks ganjil saja.
- c. Menampilkan elemen-elemen array dengan indeks genap saja (asumsi indek ke-0 adalah genap).
- d. Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x. x bisa diperoleh ddari masukan pengguna.
- e. Menghapus elemen array pada indeks tertentu, asumsi indeks yang hapus selalu valid. Tampilkan keseluruhan isi dari arraynya, pastikan data yang dihapus tidak tampil.
- f. Menampilkan rata-rata dari bilangan yang ada di dalam array.
- g. Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut.
- h. Menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi.

3.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import "fmt"

func main() {
    var klubA, klubB string
    fmt.Print("Klub A : ")
    fmt.Scan(&klubA)
    fmt.Print("Klub B : ")
    fmt.Scan(&klubB)

    var pemenang []string
    pertandingan := 1

    for {
        var skorA, skorB int
        fmt.Printf("Pertandingan %v : ", pertandingan)
        fmt.Scan(&skorA, &skorB)

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

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

    fmt.Println()
    for i, hasil := range pemenang {
        fmt.Printf("Hasil %-2d : %s\n", i+1, hasil)
    }

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

Output :

```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL6> go run "c:\103112400031_MODUL6\103112400031_unguided3.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 bertujuan untuk menyimpan dan menampilkan nama nama klub yang memenangkan pertandingan bola pada suatu grup pertandingan. Program meminta kita menginput nama nama klub yang bertanding, kemudian meminta menginput skor hasil pertandingan kedua klub tersebut. Yang disimpan dalam array hanya nama nama klub yang menang saja. Program akan berhenti meminta input skor jika kita memasukkan angka negative, atau yang kurang dari nol. Ouput berupa daftar klub yang memenangkan pertandingan.

4.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

type Tabel struct {
    tab tabel
    m int
}

// fungsi untuk mengisi array karakter
func isiArray(t *Tabel, n *int) {
    var c rune
    *n = 0
    fmt.Print("Masukkan karakter (akhiri dengan tanda TITIK) : ")
    for {
        fmt.Scanf("%c", &c)
        if c == '.' || *n >= NMAX {
            break
        }
        t.tab[*n] = c
        *n = *n + 1
    }
}

// fungsi untuk mencetak isi array
func cetakArray(t Tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Printf("%c", t.tab[i])
    }
    fmt.Println()
}

// fungsi untuk mereverse isi array
func balikanArray(t *Tabel, n int) {
    for i := 0; i < n/2; i++ {
        t.tab[i], t.tab[n-1-i] = t.tab[n-1-i], t.tab[i]
    }
}
```

```

    }
}

// fungsi untuk mengecek apakah isi array adalah palindrome
func palindrome(t Tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t.tab[i] != t.tab[n-1-i] {
            return false
        }
    }
    return true
}

func main() {
    var tab Tabel
    var m int

    isiArray(&tab, &m)

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

    var reversed Tabel = tab
    balikanArray(&reversed, m)

    fmt.Print("Reverse teks  : ")
    cetakArray(reversed, m)

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

```

Output :

```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL6> go run "c:\103112400031_MODUL6\103112400031_unguided4.go"
Masukkan karakter (akhiri dengan tanda TITIK) : S E N A N G.
Teks      : S E N A N G
Reverse teks : G N A N E S
Palindrome ? false
PS C:\103112400031_MODUL6> go run "c:\103112400031_MODUL6\103112400031_unguided4.go"
Masukkan karakter (akhiri dengan tanda TITIK) : K A T A K.
Teks      : K A T A K
Reverse teks : K A T A K
Palindrome ? true
```

Penjelasan : Program ini bertujuan untuk membalikkan urutan isi array dan memeriksa apakah membentuk palindrom. Program meminta kita menginput sebuah kata yang diakhiri dengan titik. Ada fungsi isiArray untuk mengisi array karakter, yang sebelumnya dijelaskan (akhiri dengan titik). Lalu ada fungsi cetakArray untuk mencetak array, ada fungsi balikanArray untuk mereverse isi array, dan fungsi palindrome untuk mengecek apakah palindrome atau bukan. Output berupa teks, reverse teks, dan keterangan true jika palindrome dan false jika bukan palindrome.

IV. KESIMPULAN

Berdasarkan laporan ini, dapat disimpulkan jika soal soal tersebut menggunakan struct dan array. Ada yang menggunakan struktur prosedur dan fungsi juga agar lebih terstruktur dan mudah dipahami. Struct dalam pemrograman Bahasa go digunakan untuk membuat tipe data yang lebih kompleks dengan mengelompokkan data serta menggabungkan berbagai tipe data yang berbeda. Array digunakan untuk menyimpan beberapa nilai dengan tipe data yang sama dalam satu variable.

REFERENSI

Telkom University. (2025). *Modul Praktikum Algoritma dan Pemrograman 2*.

<https://dasarpemrogramangolang.novalagung.com/>