

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 7

MATERI



Oleh:

MUHAMMAD ZAKY MUBAROK

103112400073

KELAS

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

MODUL 7. STRUCK & ARRAY

II. GUIDED

```
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()
}

```

```

PS D:\ALPKU 2\1031124000\3_MODUL1> go run d:\ALPKU
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 23-04-2025 15:00
=====
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
=====

```

Penjelasan :

Kode di atas adalah program dalam bahasa Go untuk membuat dan mencetak struk belanja. Berikut penjelasan singkatnya:

1. **Struct Item:** Merepresentasikan barang dalam struk, dengan atribut seperti nama barang (Name), harga satuan (Price), dan jumlah barang (Quantity).
2. **Struct Receipt:** Merepresentasikan struk belanja dengan informasi toko (StoreInfo), tanggal transaksi (Date), daftar barang (Items), dan total belanja (TotalAmount).
3. **Method CalculateTotal:** Menghitung total harga belanja dengan menjumlahkan harga barang dikalikan dengan jumlahnya untuk setiap item di daftar barang.
4. **Method PrintReceipt:** Mencetak struk belanja ke layar dengan format yang rapi, termasuk nama barang, harga satuan, jumlah, subtotal, dan total keseluruhan. Juga menampilkan informasi toko dan tanggal transaksi.
5. **Fungsi main:**
 - Membuat instance Receipt dengan informasi toko, tanggal saat ini, dan daftar barang.
 - Menghitung total belanja dengan memanggil CalculateTotal.
 - Mencetak struk dengan memanggil PrintReceipt.

III. GUIDED

```
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])
}
}

```

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 :

Kode di atas adalah program dalam bahasa Go yang berfungsi untuk mengolah dan menampilkan data nilai mahasiswa. Berikut penjelasan singkatnya:

- 1. Array nilaiMahasiswa:**
Berisi data nilai dari 5 mahasiswa.
- 2. Menampilkan Nilai Per Mahasiswa:**
Program menggunakan perulangan for untuk mencetak nilai masing-masing mahasiswa.
- 3. Menghitung Rata-rata Nilai:**
Total semua nilai dijumlahkan, lalu dibagi dengan jumlah mahasiswa untuk mendapatkan rata-rata.
- 4. Mencari Nilai Tertinggi dan Terendah:**
Dengan membandingkan setiap nilai dalam array, program menemukan nilai tertinggi dan terendah.
- 5. Array 2 Dimensi nilaiUjian:**
Berisi nilai ujian mahasiswa dalam dua mata kuliah (Matematika dan Bahasa). Program menampilkan nilai masing-masing mahasiswa untuk setiap mata kuliah.

Output program ini mencakup:

- Nilai setiap mahasiswa.
- Rata-rata nilai.
- Nilai tertinggi dan terendah.
- Nilai ujian dalam format dua dimensi untuk masing-masing mata kuliah.

IV. UNGUIDED I

```
//Muhammad Zaky Mubarok
package main

import "fmt"

type Titik struct {
    x, y int
}

type Lingkaran struct {
    pusat    Titik
    jariJari int
}

func (l Lingkaran) berisi(t Titik) bool {
    dx := t.x - l.pusat.x
    dy := t.y - l.pusat.y
    jarakKuadrat := dx*dx + dy*dy
    jariJariKuadrat := l.jariJari * l.jariJari
    return jarakKuadrat <= jariJariKuadrat
}

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

    fmt.Scan(&l1.pusat.x, &l1.pusat.y, &l1.jariJari)
    fmt.Scan(&l2.pusat.x, &l2.pusat.y, &l2.jariJari)
    fmt.Scan(&t.x, &t.y)

    dil1 := l1.berisi(t)
    dil2 := l2.berisi(t)

    if dil1 && dil2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dil1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if dil2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

```

PS D:\ALPRO 2\103112400073_MODUL7
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\ALPRO 2\103112400073_MODUL7
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS D:\ALPRO 2\103112400073_MODUL7
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS D:\ALPRO 2\103112400073_MODUL7
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
PS D:\ALPRO 2\103112400073_MODUL7
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2

```

Penjelasan :

Kode di atas adalah program dalam bahasa Go yang menggunakan struktur data (struct) untuk memeriksa apakah suatu titik berada di dalam satu atau dua lingkaran. Berikut penjelasan singkatnya:

1. **Struct Titik:**

Merepresentasikan koordinat suatu titik dengan atribut x dan y.

2. **Struct Lingkaran:**

Merepresentasikan lingkaran dengan atribut pusat (bertipe Titik) dan jariJari (bertipe int).

3. **Method berisi:**

Fungsi dalam struct Lingkaran untuk memeriksa apakah suatu Titik berada di dalam lingkaran. Penghitungan dilakukan dengan membandingkan jarak kuadrat titik ke pusat lingkaran dengan kuadrat jari-jari lingkaran.

4. **Fungsi main:**

- Membaca input untuk dua lingkaran (l1 dan l2) dan satu titik (t).
- Memeriksa apakah titik tersebut berada di dalam satu atau kedua lingkaran dengan memanggil method berisi.
- Menampilkan hasilnya, apakah titik berada di dalam lingkaran 1, lingkaran 2, keduanya, atau tidak ada lingkaran.

V. UNGUIDED II

```
//Muhammad Zaky Mubarok
package main

import "fmt"

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

    data := make([]int, jumlahElemen)
    for i := 0; i < jumlahElemen; i++ {
        fmt.Printf("Data ke-%d: ", i)
        fmt.Scan(&data[i])
    }

    fmt.Println("a. Seluruh isi array:", data)

    fmt.Print("b. Elemen dengan indeks ganjil: ")
    for i := 1; i < len(data); i += 2 {
        fmt.Print(data[i], " ")
    }
    fmt.Println()

    fmt.Print("c. Elemen dengan indeks genap: ")
    for i := 0; i < len(data); i += 2 {
        fmt.Print(data[i], " ")
    }
    fmt.Println()

    var kelipatan int
    fmt.Print("Masukkan nilai x (kelipatan indeks): ")
    fmt.Scan(&kelipatan)
    fmt.Printf("d. Elemen dengan indeks kelipatan %d: ", kelipatan)
    for i := 0; i < len(data); i++ {
        if i%kelipatan == 0 {
            fmt.Print(data[i], " ")
        }
    }
}
```

```

    }
    fmt.Println()

    var indeksHapus int
    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&indeksHapus)
    if indeksHapus >= 0 && indeksHapus < len(data) {
        data = append(data[:indeksHapus], data[indeksHapus+1:]...)
    }
    fmt.Println("e. Array setelah penghapusan:", data)

    var total int
    for _, nilai := range data {
        total += nilai
    }
    rataRata := float64(total) / float64(len(data))
    fmt.Printf("f. Rata-rata: %.2f\n", rataRata)

    var jumlahSelisihKuadrat float64
    for _, nilai := range data {
        selisih := float64(nilai) - rataRata
        jumlahSelisihKuadrat += selisih * selisih
    }
    fmt.Printf("g. Simpangan baku kuadrat: %.2f\n",
        jumlahSelisihKuadrat/float64(len(data)))

    var angkaCari, frekuensi int
    fmt.Print("Masukkan angka yang ingin dihitung frekuensinya: ")
    fmt.Scan(&angkaCari)
    for _, nilai := range data {
        if nilai == angkaCari {
            frekuensi++
        }
    }
    fmt.Printf("h. Frekuensi kemunculan angka %d: %d kali\n",
        angkaCari, frekuensi)
}

```

```

PS D:\ALPRO 2\103112400073_MODUL7> go run "d:\ALPRO 2\103112400073_MODUL7\main.go"
Masukkan jumlah elemen array: 5
Data ke-0: 1
Data ke-1: 2
Data ke-2: 3
Data ke-3: 4
Data ke-4: 1
a. Seluruh isi array: [1 2 3 4 1]
b. Elemen dengan indeks ganjil: 2 4
c. Elemen dengan indeks genap: 1 3 1
Masukkan nilai x (kelipatan indeks): 2
d. Elemen dengan indeks kelipatan 2: 1 3 1
Masukkan indeks yang ingin dihapus: 2
e. Array setelah penghapusan: [1 2 4 1]
f. Rata-rata: 2.00
g. Simpangan baku kuadrat: 1.50
Masukkan angka yang ingin dihitung frekuensinya: 1
h. Frekuensi kemunculan angka 1: 2 kali

```

Penjelasan :

Kode di atas adalah program dalam bahasa Go untuk mengolah data array dengan beberapa fungsi analisis. Berikut penjelasan singkatnya:

1. **Input Elemen Array:**
Program meminta pengguna memasukkan jumlah elemen array dan data untuk setiap elemen.
2. **a. Menampilkan seluruh isi array:**
Cetak semua elemen array yang dimasukkan oleh pengguna.
3. **b. Menampilkan elemen dengan indeks ganjil:**
Cetak elemen dengan indeks ganjil menggunakan perulangan.
4. **c. Menampilkan elemen dengan indeks genap:**
Cetak elemen dengan indeks genap menggunakan perulangan.
5. **d. Menampilkan elemen dengan indeks kelipatan x:**
Pengguna memasukkan nilai x , lalu elemen dengan indeks kelipatan x dicetak.
6. **e. Menghapus elemen pada indeks tertentu:**
Pengguna memilih indeks untuk dihapus, dan program memodifikasi array sesuai pilihan tersebut.
7. **f. Menghitung rata-rata nilai:**
Hitung rata-rata nilai dari elemen dalam array.

8. **g. Menghitung simpangan baku kuadrat (tanpa akar kuadrat):**
Program menghitung simpangan baku kuadrat dengan formula deviasi kuadrat tanpa fungsi akar.
9. **h. Menghitung frekuensi kemunculan angka tertentu:**
Pengguna memilih angka, dan program menghitung berapa kali angka tersebut muncul dalam array.

VI. UNGUIDED III

```
//Muhammad Zaky Mubarak
package main

import "fmt"

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

    fmt.Print("Klub A : ")
    fmt.Scanln(&klubA)

    fmt.Print("Klub B : ")
    fmt.Scanln(&klubB)

    pertandingan := 1
    for {
        fmt.Printf("Pertandingan %d : ", 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++
    }

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

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



```
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 :

Kode ini adalah program untuk mencatat hasil pertandingan antara dua klub sepak bola dalam bahasa Go. Berikut penjelasan singkatnya:

1. **Input Nama Klub:**
Program meminta pengguna memasukkan nama dua klub: `klubA` dan `klubB`.
2. **Input Skor Pertandingan:**
Dalam setiap iterasi, pengguna memasukkan skor masing-masing klub (`skorA` dan `skorB`) untuk pertandingan tertentu.

3. Penentuan Pemenang:

- Jika `skorA > skorB`, pemenang pertandingan adalah `klubA`.
- Jika `skorB > skorA`, pemenang adalah `klubB`.
- Jika skor sama, hasilnya adalah "Draw" (seri).

4. Menghentikan Input:

Program berhenti meminta skor jika ditemukan input skor negatif (`skorA < 0` atau `skorB < 0`).

5. Output Hasil Pertandingan:

Setelah semua input selesai, program mencetak hasil setiap pertandingan, termasuk pemenangnya.

6. Akhir Program:

Program menampilkan pesan "Pertandingan selesai" setelah semua hasil diproses.

VII. UNGUIDED IV

```
//Muhammad Zaky Mubarak
package main

import (
    "fmt"
)

const NMAX int = 127
type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var ch rune
    *n = 0
    fmt.Println("Masukkan karakter (akhiri dengan titik):")
    for {
        fmt.Scanf("%c", &ch)
        if ch == '.' || *n >= NMAX {
            break
        }
        (*t)[*n] = ch
        *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 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)

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

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

```

Masukkan karakter (akhiri dengan titik):
KATAK.

Teks: KATAK

Palindrome? true

Reverse teks: KATAK

PS D:\ALPRO 2\103112400073_MODUL7> go run

Masukkan karakter (akhiri dengan titik):
SENANG.

Teks: SENANG

Palindrome? false

Reverse teks: GNANES

Penjelasan :

Kode ini adalah program dalam bahasa Go yang berfungsi untuk memproses dan memanipulasi array karakter (tabel karakter). Berikut penjelasan singkatnya:

1. **Const dan Struct:**
 - `NMAX` mendefinisikan jumlah maksimum karakter dalam array (127).
 - `tabel` adalah array dengan tipe `rune` untuk menyimpan karakter.
2. **Fungsi `isiArray`:**
 - Mengisi array karakter (`tabel`) berdasarkan input pengguna hingga karakter titik (.) atau mencapai batas maksimum (`NMAX`).
3. **Fungsi `cetakArray`:**
 - Menampilkan isi array karakter ke layar.
4. **Fungsi `balikanArray`:**
 - Membalikkan isi array, mengatur urutan karakter dari awal ke akhir.
5. **Fungsi `palindrome`:**
 - Mengecek apakah array adalah palindrome, yaitu apakah karakter array sama jika dibaca dari depan ke belakang.
6. **Fungsi `main`:**
 - Memanggil fungsi `isiArray` untuk menerima input karakter.
 - Menampilkan teks awal menggunakan `cetakArray`.
 - Memeriksa apakah teks adalah palindrome menggunakan fungsi `palindrome`.
 - Membalikkan teks dan menampilkan hasilnya menggunakan fungsi `balikanArray`.

VIII. KESIMPULAN

Kesimpulan Modul 7: STRUCT & ARRAY

memperkenalkan dan memperdalam pemahaman mengenai penggunaan array, struct, type alias, serta konsep manipulasi data dalam konteks algoritma dan pemrograman menggunakan bahasa Go (Golang). Hal-hal penting yang dipelajari dalam modul ini meliputi:

1. Tipe Bentuk

- **Alias (type):** Digunakan untuk memberi nama baru pada tipe data yang sudah ada agar lebih mudah dibaca dan digunakan.
- **Struct (structure):** Mengelompokkan beberapa nilai/data yang saling berkaitan dalam satu entitas. Berguna untuk merepresentasikan objek dengan beberapa atribut (field).

2. Array

- **Array** adalah struktur data yang menyimpan sekumpulan nilai dengan tipe yang sama dan ukuran tetap.
- **Operasi dasar pada array:** pengisian data, akses indeks, pengubahan elemen, iterasi.
- Terdapat array statis (dengan ukuran tetap) dan slice (dinamis, meskipun tidak dibahas dalam praktikum ini).

REFERENSI

MODUL 7. STRUCT & ARRAY