

**ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6**

**ARRAY DAN STRUCK**



Oleh:

MUHAMMAD GAMEL AL GHIFARI

103112400028

12IF-01

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## **I. DASAR TEORI**

Pemrograman dengan bahasa Go (Golang) menekankan prinsip kesederhanaan, efisiensi, dan keunggulan dalam menangani konkruensi. Bahasa ini memiliki fitur-fitur utama seperti penentuan tipe data saat kompilasi, pengelolaan memori otomatis melalui *garbage collection*, serta kemampuan untuk mengembalikan lebih dari satu nilai dari suatu fungsi. Dalam praktiknya, Go memanfaatkan konsep dasar pemrograman seperti perulangan, rekursi, percabangan, serta struktur data seperti array dan slice. Selain itu, validasi input merupakan aspek penting guna mencegah terjadinya kesalahan dalam program.

## II. UNGUIDED

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:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\103112400028_MODUL6\tempCodeRunnerFile.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 23:16
=====
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:

1. Struct Item: Merepresentasikan barang dengan atribut nama, harga, dan jumlah.
2. Struct Receipt: Merepresentasikan struk belanja dengan atribut informasi toko, tanggal, daftar barang, dan total belanja.
3. Method CalculateTotal: Menghitung total harga semua barang dalam struk.
4. Method PrintReceipt: Mencetak struk belanja dengan format rapi, termasuk daftar barang, total belanja, dan ucapan terima kasih.
5. Fungsi [main](#): Membuat objek Receipt, menghitung total belanja, dan mencetak struk.

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:\LAPRAK\> go run "d:\LAPRAK\103112400028_MODUL6\tempCodeRunnerFile.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 ini mencetak data nilai mahasiswa, statistik sederhana (rata-rata, nilai tertinggi, dan terendah), serta nilai ujian dalam format yang rapi.

### III. GUIDED

1.

```
// MUAHMMAD GAMEL AL GHIFARI
// 103112400028

package main

import (
    "fmt"
    "math"
)

func lingkaran(x1, y1, x2, y2 int) float64 {
    return math.Sqrt(math.Pow(float64(x2-x1), 2) + math.Pow(float64(y2-y1),
2))
}

func main() {
    var x1, y1, r1 int
    fmt.Scan(&x1, &y1, &r1)

    var x2, y2, r2 int
    fmt.Scan(&x2, &y2, &r2)

    var xt, yt int
    fmt.Scan(&xt, &yt)

    jarakKeLingkaran1 := lingkaran(x1, y1, xt, yt)
    jarakKeLingkaran2 := lingkaran(x2, y2, xt, yt)
```

```
if jarakKeLingkaran1 <= float64(r1) && jarakKeLingkaran2 <= float64(r2) {  
    fmt.Println("Titik di dalam lingkaran 1 dan 2")  
} else if jarakKeLingkaran1 <= float64(r1) {  
    fmt.Println("Titik di dalam lingkaran 1")  
} else if jarakKeLingkaran2 <= float64(r2) {  
    fmt.Println("Titik di dalam lingkaran 2")  
} else {  
    fmt.Println("Titik di luar lingkaran 1 dan 2")  
}  
}
```

OUTPUT:

```
PS D:\LAPRAK\> go run "d:\LAPRAK\103112400028_MODULE6\guided1.go"  
1 1 3  
4 5 6  
1 2  
Titik di dalam lingkaran 1 dan 2
```

DESKRIPSI:

Program ini menentukan posisi sebuah titik terhadap dua lingkaran berdasarkan jarak titik ke pusat lingkaran dan jari-jari lingkaran..

2.

```
package main
```

```

import (
    "fmt"
    "math"
)

func main() {
    var n int

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

    arr := make([]int, n)

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

    for {
        fmt.Println("\n=====")
    )

        fmt.Println("  SELAMAT DATANG DI MENU ARRAY  ")
        fmt.Println("=====")
        fmt.Println("1. TAMPILKAN SEMUA ARRAY")
    }
}

```

```
fmt.Println("2. TAMPILKAN SEMUA ARRAY GENAP")

fmt.Println("3. TAMPILKAN SEMUA ARRAY GANJIL")

fmt.Println("4. TAMPILKAN ARRAY DENGAN INDEKS KELIPATAN
X")

fmt.Println("5. HAPUS ELEMEN PADA INDEKS TERTENTU")

fmt.Println("6. HITUNG RATA-RATA ARRAY")

fmt.Println("7. HITUNG STANDAR DEVIASI ARRAY")

fmt.Println("8. HITUNG FREKUENSI BILANGAN TERTENTU")

fmt.Println("9. KELUAR")

fmt.Print("Pilihan Anda: ")

var choice int

fmt.Scan(&choice)

switch choice {

case 1:

    displayAll(arr)

case 2:

    displayEvenIndex(arr)

case 3:

    displayOddIndex(arr)

case 4:

    displayMultiples(arr)

case 5:

    arr = deleteElement(arr)
```

```
case 6:
    calculateAverage(arr)

case 7:
    calculateStdDev(arr)

case 8:
    calculateFrequency(arr)

case 9:
    fmt.Println("Keluar dari program.")
    return

default:
    fmt.Println("Pilihan tidak valid!")
}
}
}

func displayAll(arr []int) {
    fmt.Println("\nIsi array:")
    for i, val := range arr {
        fmt.Printf("Indeks %d: %d\n", i, val)
    }
}

func displayEvenIndex(arr []int) {
    fmt.Println("\nElemen dengan indeks genap:")
```

```
for i := 0; i < len(arr); i += 2 {  
    fmt.Printf("Indeks %d: %d\n", i, arr[i])  
}  
}  
  
func displayOddIndex(arr []int) {  
    fmt.Println("\nElemen dengan indeks ganjil:")  
    for i := 1; i < len(arr); i += 2 {  
        fmt.Printf("Indeks %d: %d\n", i, arr[i])  
    }  
}  
  
func displayMultiples(arr []int) {  
    var x int  
    fmt.Print("Masukkan nilai x: ")  
    fmt.Scan(&x)  
  
    fmt.Printf("\nElemen dengan indeks kelipatan %d:\n", x)  
    for i := 0; i < len(arr); i++ {  
        if i%x == 0 {  
            fmt.Printf("Indeks %d: %d\n", i, arr[i])  
        }  
    }  
}
```

```
func deleteElement(arr []int) []int {  
    var index int  
  
    fmt.Print("Masukkan indeks yang akan dihapus: ")  
    fmt.Scan(&index)  
  
    if index < 0 || index >= len(arr) {  
        fmt.Println("Indeks tidak valid!")  
        return arr  
    }  
  
    newArr := append(arr[:index], arr[index+1:]...)  
  
    fmt.Println("\nArray setelah penghapusan:")  
    for i, val := range newArr {  
        fmt.Printf("Indeks %d: %d\n", i, val)  
    }  
  
    return newArr  
}  
  
func calculateAverage(arr []int) {  
    if len(arr) == 0 {  
        fmt.Println("Array kosong")  
    }  
}
```



```
        return
    }

    sum := 0
    for _, val := range arr {
        sum += val
    }
    average := float64(sum) / float64(len(arr))
    fmt.Printf("\nRata-rata: %.2f\n", average)
}

func calculateStdDev(arr []int) {
    if len(arr) == 0 {
        fmt.Println("Array kosong")
        return
    }

    sum := 0
    for _, val := range arr {
        sum += val
    }

    mean := float64(sum) / float64(len(arr))

    variance := 0.0
```

```
    for _, val := range arr {  
        variance += math.Pow(float64(val)-mean, 2)  
    }  
    variance /= float64(len(arr))  
  
    stdDev := math.Sqrt(variance)  
    fmt.Printf("\nStandar deviasi: %.2f\n", stdDev)  
}  
  
func calculateFrequency(arr []int) {  
    var target int  
    fmt.Print("Masukkan bilangan yang akan dihitung frekuensinya: ")  
    fmt.Scan(&target)  
  
    count := 0  
    for _, val := range arr {  
        if val == target {  
            count++  
        }  
    }  
    fmt.Printf("\nFrekuensi bilangan %d: %d\n", target, count)  
}
```

## OUTPUT:

```
Elemen ke-2: 3

=====
                SELAMAT DATANG DI MENU ARRAY
=====

1. TAMPILKAN SEMUA ARRAY
2. TAMPILKAN SEMUA ARRAY GENAP
3. TAMPILKAN SEMUA ARRAY GANJIL
4. TAMPILKAN ARRAY DENGAN INDEKS KELIPATAN X
5. HAPUS ELEMEN PADA INDEKS TERTENTU
6. HITUNG RATA-RATA ARRAY
7. HITUNG STANDAR DEVIASI ARRAY
8. HITUNG FREKUENSI BILANGAN TERTENTU
9. KELUAR
Pilihan Anda: 8
Masukkan bilangan yang akan dihitung frekuensinya: 3

Frekuensi bilangan 3: 1

=====
                SELAMAT DATANG DI MENU ARRAY
=====

1. TAMPILKAN SEMUA ARRAY
2. TAMPILKAN SEMUA ARRAY GENAP
3. TAMPILKAN SEMUA ARRAY GANJIL
4. TAMPILKAN ARRAY DENGAN INDEKS KELIPATAN X
5. HAPUS ELEMEN PADA INDEKS TERTENTU
6. HITUNG RATA-RATA ARRAY
7. HITUNG STANDAR DEVIASI ARRAY
8. HITUNG FREKUENSI BILANGAN TERTENTU
9. KELUAR
Pilihan Anda: █
```

## DESKRIPSI:

Program ini adalah aplikasi interaktif untuk melakukan berbagai operasi pada array bilangan bulat.

.

```
//MUHAMMAD GAMEL AL GHIFARI  
// // 103112400028  
  
package main  
  
import "fmt"  
  
func main() {  
    var clubA, clubB string  
  
    fmt.Print("Masukkan nama club A: ")  
  
    fmt.Scan(&clubA)  
  
    fmt.Print("Masukkan nama club B: ")  
  
    fmt.Scan(&clubB)  
  
    var winners []string  
  
    for i := 1; ; i++ {  
        var scoreA, scoreB int  
  
        fmt.Printf("Pertandingan %d (masukkan skor A dan B): ", i)  
  
        fmt.Scan(&scoreA, &scoreB)  
  
        if scoreA < 0 || scoreB < 0 {  
            fmt.Println("Pertandingan selesai")  
            break  
        }  
    }
```

```

        if scoreA > scoreB {
            winners = append(winners, clubA)
        } else if scoreB > scoreA {
            winners = append(winners, clubB)
        } else {
            winners = append(winners, "Draw")
        }
    }

    fmt.Println("\nHasil pertandingan:")

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

```

OUTPUT:

```

Masukkan nama club A: mu
Masukkan nama club B: mc
Pertandingan 1 (masukkan skor A dan B): 1 2
Pertandingan 2 (masukkan skor A dan B): 1 2
Pertandingan 3 (masukkan skor A dan B): 0 0
Pertandingan 4 (masukkan skor A dan B): 2 1
Pertandingan 5 (masukkan skor A dan B): -1 2
Pertandingan selesai

Hasil pertandingan:
Hasil 1: mc
Hasil 2: mc
Hasil 3: Draw
Hasil 4: mu
PS D:\LAPRAKKKK>

```

#### DESKRIPSI:

Program ini mencatat hasil pertandingan antara dua klub sepak bola berdasarkan skor yang dimasukkan pengguna.

4.

```
// MUHAMMAD GAMEL AL GHIFARI
// 103112400028

package main

import (
    "fmt"
)

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var input rune
    *n = 0
    fmt.Println("Masukkan karakter (akhiri dengan titik '.'):")

    for {
        fmt.Scanf("%c", &input)
```

```

    if input == '\n' {
        continue
    }

    if input == '.' || *n >= NMAX {
        break
    }

    t[*n] = input
    (*n)++
}

```

```

func cetakArray(t tabel, n int) {
    fmt.Print("Teks: ")
    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 n int  
  
    isiArray(&tab, &n)  
  
    fmt.Println("\nTeks asli:")  
    cetakArray(tab, n)  
  
    balikanArray(&tab, n)  
    fmt.Println("\nTeks setelah dibalik:")  
    cetakArray(tab, n)  
  
    if isPalindrom(tab, n) {
```



```
        fmt.Println("\nPalindrom? true")
    } else {
        fmt.Println("\nPalindrom? false")
    }
}
```

#### OUTPUT:

```
PS D:\LAPRAK\> go run "d:\LAPRAK\103112400028_MODUL6\guided4.go"
Masukkan karakter (akhiri dengan titik '.'):
K A T A K
.

Teks asli:
Teks: K A T A K

Teks setelah dibalik:
K A T A K

Palindrom? false
PS D:\LAPRAK\> |
```

#### DESKRIPSI:

Program ini memproses array karakter untuk memeriksa apakah array adalah palindrom dan membalik urutannya.

#### **IV. KESIMPULAN**

Dokumen ini menyajikan berbagai contoh penerapan bahasa Go dalam pemrograman. Penggunaan algoritma yang efisien—seperti rekursi untuk operasi perkalian dan pencarian bilangan ganjil—serta optimasi dalam perhitungan menunjukkan bagaimana kompleksitas dapat ditekan. Beragam program yang dikembangkan memperlihatkan kemampuan Go dalam menyelesaikan berbagai persoalan, mulai dari perhitungan aritmetika sederhana hingga sistem berbasis pola seperti pemberian hadiah. Validasi input dan penanganan kasus tepi turut meningkatkan keandalan program. Secara keseluruhan, Go menawarkan solusi pemrograman yang efisien dan sangat cocok bagi mahasiswa yang ingin memahami konsep-konsep dasar hingga menengah dalam dunia pemrograman.

## **REFERENSI**

Programmer Zaman Now. "ALGORITMA & STRUKTUR DATA - SEMESTER 2." YouTube, 2023, [https://youtu.be/IO\\_vkyJnMas?si=4jSJTZ5Zd2Fi5Vd2](https://youtu.be/IO_vkyJnMas?si=4jSJTZ5Zd2Fi5Vd2).