

ALGORITMA DAN PEMROGRAMAN 2

LATIHAN SOAL 2



Oleh:

M.HANIF AL FAIZ

103112400042

12IF-01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I. DASAR TEORI

Pemrograman menggunakan bahasa Go (Golang) menekankan pada kesederhanaan, efisiensi, dan kemampuan dalam menangani konkurensi. Bahasa ini memiliki fitur-fitur utama seperti penentuan tipe data pada kompilasi, pengelolaan memori otomatis (garbage collection), serta kemampuan mengembalikan lebih dari satu nilai dari fungsi. Dalam penerapannya, Go memanfaatkan konsep-konsep dasar dalam pemrograman seperti looping, rekursi, kondisional, dan struktur data seperti array dan slice. Selain itu, validasi input menjadi bagian penting untuk menghindari kesalahan dalam program.

II. UNGUIDED

TYPE A:

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:\LAPRAKKKK> go run "d:\LAPRAKKKK\103112400042_MHANIFALFAIZ\tempCodeRunnerFile.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 24-04-2025 22:39
=====
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 [Name](#), [Price](#), dan [Quantity](#).
2. **Struct Receipt:** Merepresentasikan struk belanja dengan atribut [StoreInfo](#), [Date](#), [Items](#) (daftar barang), dan TotalAmount.
3. **Method CalculateTotal:** Menghitung total belanja dengan menjumlahkan harga setiap barang dikalikan jumlahnya.
4. **Method PrintReceipt:** Mencetak struk belanja ke konsol dengan format rapi, termasuk informasi toko, tanggal, daftar barang, total belanja, dan ucapan terima kasih.
5. **Fungsi main:** Membuat objek Receipt dengan data toko, tanggal, dan daftar barang, lalu memanggil CalculateTotal untuk menghitung total dan PrintReceipt untuk 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:\LAPRAKKKK> go run "d:\LAPRAKKKK\103112400042_MHANIFALFAIZ\unguided2.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:

1. Array Satu Dimensi:

- Menyimpan nilai 5 mahasiswa.

- Menghitung rata-rata, nilai tertinggi, dan nilai terendah.

2. Array Dua Dimensi:

- Menyimpan nilai 3 mahasiswa dalam 2 mata kuliah (Matematika dan Bahasa).
- Menampilkan nilai setiap mahasiswa untuk kedua mata kuliah.

III. GUIDED

1.

```
// M.HANIF AL FAIZ
// 103112400042

package main

import (
    "fmt"
    "math"
)

func jarak(x1, y1, x2, y2 int) float64 {
    return math.Sqrt(float64((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1)))
}

func main() {
    var x1, y1, r1 int
    var x2, y2, r2 int
    var xt, yt int
```

```

    fmt.Scan(&x1, &y1, &r1)

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

    fmt.Scan(&xt, &yt)

    jarakKeLingkaran1 := jarak(x1, y1, xt, yt)
    jarakKeLingkaran2 := jarak(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:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\103112400042_MHANIFALFAIZ\guided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1

```

DESKRIPSI:

1. **Fungsi [jarak](#)**: Menghitung jarak antara dua titik menggunakan rumus Euclidean.
2. **Input**: Koordinat pusat dan jari-jari dua lingkaran ([x1, y1, r1](#) dan [x2, y2, r2](#)) serta koordinat titik ([xt, yt](#)).
3. **Logika**:
 - Menghitung jarak titik ke pusat masing-masing lingkaran.
 - Mengevaluasi apakah titik berada di dalam salah satu atau kedua lingkaran, atau di luar keduanya.
4. **Output**: Pesan posisi titik relatif terhadap lingkaran.

2.

```
// M.HANIF AL FAIZ
// 1031124000042

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("\nMenu Operasi Array:")
```

```
    fmt.Println("a. Tampilkan seluruh isi array")
```

```
    fmt.Println("b. Tampilkan elemen dengan indeks ganjil")
```

```
    fmt.Println("c. Tampilkan elemen dengan indeks genap")
```

```
    fmt.Println("d. Tampilkan elemen dengan indeks kelipatan x")
```

```
    fmt.Println("e. Hapus elemen pada indeks tertentu")
```

```
    fmt.Println("f. Hitung rata-rata array")
```

```
    fmt.Println("g. Hitung standar deviasi array")
```

```
    fmt.Println("h. Hitung frekuensi bilangan tertentu")
```

```
    fmt.Println("x. Keluar")
```

```
var choice string
```

```
fmt.Print("Pilihan Anda: ")
```

```
fmt.Scan(&choice)
```

```
switch choice {  
    case "a":  
        displayAll(arr)  
    case "b":  
        displayOddIndex(arr)  
    case "c":  
        displayEvenIndex(arr)  
    case "d":  
        displayMultiples(arr)  
    case "e":  
        arr = deleteElement(arr)  
    case "f":  
        calculateAverage(arr)  
    case "g":  
        calculateStdDev(arr)  
    case "h":  
        calculateFrequency(arr)  
    case "x":  
        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 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 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 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)

    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:

```
PS D:\LAPRAK\> go run "d:\LAPRAK\103112400042_MHANIFALFAIZ\guided2.go"
Masukkan jumlah elemen array: 4
Masukkan elemen array:
Elemen ke-1: 2
Elemen ke-2: 2
Elemen ke-3: 2
Elemen ke-4: 2

Menu Operasi Array:
a. Tampilkan seluruh isi array
b. Tampilkan elemen dengan indeks ganjil
c. Tampilkan elemen dengan indeks genap
d. Tampilkan elemen dengan indeks kelipatan x
e. Hapus elemen pada indeks tertentu
f. Hitung rata-rata array
g. Hitung standar deviasi array
h. Hitung frekuensi bilangan tertentu
x. Keluar
Pilihan Anda: c

Elemen dengan indeks genap:
Indeks 0: 2
Indeks 2: 2

Menu Operasi Array:
a. Tampilkan seluruh isi array
b. Tampilkan elemen dengan indeks ganjil
c. Tampilkan elemen dengan indeks genap
d. Tampilkan elemen dengan indeks kelipatan x
e. Hapus elemen pada indeks tertentu
f. Hitung rata-rata array
g. Hitung standar deviasi array
h. Hitung frekuensi bilangan tertentu
x. Keluar
Pilihan Anda: █
```

DESKRIPSI:

1. Input Data:

- Pengguna memasukkan jumlah elemen array dan nilai-nilai elemen array.

2. Menu Operasi:

- Program menyediakan berbagai operasi pada array:
 - **a:** Menampilkan seluruh isi array.
 - **b:** Menampilkan elemen dengan indeks ganjil.
 - **c:** Menampilkan elemen dengan indeks genap.
 - **d:** Menampilkan elemen dengan indeks kelipatan tertentu.
 - **e:** Menghapus elemen pada indeks tertentu.
 - **f:** Menghitung rata-rata elemen array.
 - **g:** Menghitung standar deviasi elemen array.
 - **h:** Menghitung frekuensi kemunculan bilangan tertentu.
 - **x:** Keluar dari program.

3. Fungsi-Fungsi:

- [displayAll](#): Menampilkan semua elemen array.
- [displayOddIndex](#): Menampilkan elemen pada indeks ganjil.
- [displayEvenIndex](#): Menampilkan elemen pada indeks genap.
- [displayMultiples](#): Menampilkan elemen pada indeks kelipatan tertentu.
- [deleteElement](#): Menghapus elemen pada indeks tertentu dan mengembalikan array baru.
- [calculateAverage](#): Menghitung rata-rata elemen array.
- [calculateStdDev](#): Menghitung standar deviasi elemen array.
- [calculateFrequency](#): Menghitung frekuensi kemunculan bilangan tertentu dalam array.

4. Logika Program:

- Program berjalan dalam loop hingga pengguna memilih opsi keluar ([x](#)).
- Setiap pilihan menu memanggil fungsi yang sesuai untuk memproses array.

3.

```
// M.HANIF AL FAIZ
// 103112400042

package main

import "fmt"

func main() {

    var clubA, clubB string

    fmt.Print("Klub A : ")
    fmt.Scan(&clubA)
    fmt.Print("Klub B : ")
    fmt.Scan(&clubB)

    var winners []string

    for i := 1; ; i++ {

        var scoreA, scoreB int

        fmt.Printf("Pertandingan %d : ", i)

        _, err := fmt.Scan(&scoreA, &scoreB)

        if err != nil || scoreA < 0 || scoreB < 0 {

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

```

        break
    }

    var result string

    switch {
    case scoreA > scoreB:
        result = clubA
    case scoreB > scoreA:
        result = clubB
    default:
        result = "Draw"
    }

    winners = append(winners, result)

    fmt.Printf("Hasil %d : %s\n", i, result)
}

fmt.Println("\nDaftar klub yang memenangkan pertandingan:")
for i, winner := range winners {
    if winner != "Draw" {
        fmt.Printf("Pertandingan %d: %s\n", i+1, winner)
    }
}
}

```

OUTPUT:

```

Klub A : MU
Klub B : INTER
Pertandingan 1 : 2 0
Hasil 1 : MU
Pertandingan 2 : 1 2
Hasil 2 : INTER
Pertandingan 3 : 2 2
Hasil 3 : Draw
Pertandingan 4 : 0 1
Hasil 4 : INTER
Pertandingan 5 : 3 2
Hasil 5 : MU
Pertandingan 6 : 1 0
Hasil 6 : MU
Pertandingan 7 : 5 2
Hasil 7 : MU
Pertandingan 8 : 2 3
Hasil 8 : INTER
Pertandingan 9 : -1 2
Pertandingan selesai

Daftar klub yang memenangkan pertandingan:
Pertandingan 1: MU
Pertandingan 2: INTER
Pertandingan 4: INTER
Pertandingan 5: MU
Pertandingan 6: MU
Pertandingan 7: MU
Pertandingan 8: INTER

```

DESKRIPSI:

1. Input:

- Nama klub A dan klub B.
- Skor pertandingan secara berulang hingga skor negatif dimasukkan (mengakhiri input).

2. Logika:

- Membandingkan skor untuk menentukan pemenang (klub A, klub B, atau "Draw").
- Menyimpan hasil pemenang ke dalam slice winners.

3. Output:

- Menampilkan daftar klub yang memenangkan setiap pertandingan (mengabaikan hasil "Draw").

4.

```
// M.HANIF AL FAIZ
// 10311240002

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("Isi array: ")

    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

    isiArray(&tab, &m)

    fmt.Println("\nArray sebelum dibalik:")
    cetakArray(tab, m)

    if isPalindrom(tab, m) {
        fmt.Println("Array adalah palindrom")
    } else {
        fmt.Println("Array bukan palindrom")
    }
}
```

```

    }

    balikArray(&tab, m)

    fmt.Println("\nArray setelah dibalik:")
    cetakArray(tab, m)

    if isPalindrom(tab, m) {
        fmt.Println("Array adalah palindrom")
    } else {
        fmt.Println("Array bukan palindrom")
    }
}

```

OUTPUT:

```

PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\103112400042_MHANIFALFAIZ\guided4.go"
Masukkan karakter (akhiri dengan titik '.'):
KATAK.

Array sebelum dibalik:
Isi array: K A T A K
Array adalah palindrom

Array setelah dibalik:
Isi array: K A T A K
Array adalah palindrom

```

DESKRIPSI:

1. **Fungsi [isiArray](#)**: Mengisi array karakter hingga titik ('.') atau batas maksimum.
2. **Fungsi [cetakArray](#)**: Menampilkan isi array.
3. **Fungsi [balikanArray](#)**: Membalik urutan elemen array.
4. **Fungsi [isPalindrom](#)**: Mengecek apakah array adalah palindrom (simetris).

IV. KESIMPULAN

Dokumen ini menampilkan berbagai contoh penggunaan bahasa Go dalam pemrograman. Penggunaan algoritma yang efisien, seperti rekursi dalam perkalian dan pencarian bilangan ganjil, serta optimasi perhitungan, memperlihatkan bagaimana kompleksitas dapat diminimalkan. Berbagai program yang dibuat menunjukkan bagaimana Go mampu menangani berbagai masalah, dari perhitungan aritmatika dasar hingga sistem berbasis pola seperti hadiah. Validasi input dan penanganan kasus tepi meningkatkan ketangguhan program. Secara keseluruhan, Go memberikan solusi pemrograman yang efisien dan cocok untuk mahasiswa yang ingin mempelajari konsep-konsep dasar hingga menengah dalam pemrograman.

V. REFERENSI

Programmer Zaman Now. "ALGORITMA & STRUKTUR DATA - SEMESTER 2." YouTube, 2023, https://youtu.be/IO_vkyJnMas?si=4jSJTZ5Zd2Fi5Vd2.