

# LATIHAN SOAL

NOMOR 1

Package main

```
import (  
    "fmt"  
)
```

```
func main() {  
    // 1. Struktur kondisional if-else  
    nilai := 85  
    fmt.Println("Contoh if-else:")  
    if nilai >= 90 {  
        fmt.Println("NILAI A")  
    } else if nilai >= 80 {  
        fmt.Println("Nilai B")  
    } else if nilai >= 70 {  
        fmt.Println("NILAI C")  
    } else if nilai >= 60 {  
        fmt.Println("Nilai D")  
    } else {  
        fmt.Println("NILAI E")  
    }  
}
```

package main

```
import "fmt"
```

```
func main() {
```

```

    fmt.Println("\nContoh for sebagai while:")

    counter := 1

    for counter <= 5 {
        fmt.Println(counter) // (4) Lengkapi bagian ini untuk mencetak "Iterasi ke-X"
        counter++
    }

}

package main

import "fmt"

func main() {
    fmt.Println("\nContoh for dengan range:")

    buah := []string{"Apel", "Mangga", "Jeruk", "Pisang"}

    for _, item := range buah { // (5) Lengkapi bagian ini agar mencetak indeks dan nama buah
        fmt.Println(len(item)) // (6) Lengkapi bagian ini untuk mencetak "Buah pada index X adalah Y"
    }
}

```

```

package main

```

```

import "fmt"

```

```

func main() {
    fmt.Println("\nContoh switch-case:")

    hari := "Senin"

    switch hari {
    case "Senin":
        fmt.Println("HARI KERJA") // (7) Lengkapi bagian ini untuk mencetak "Hari kerja"
    case "Selasa":

```

```
        fmt.Println("Hari kerja")
    case "Rabu":
        fmt.Println("hari kerja") // (8) Lengkapi bagian ini agar hari kerja lengkap
    case "Kamis":
        fmt.Println("Hari kerja")
    case "Jumat":
        fmt.Println("HARI KERJA") // (9) Lengkapi bagian ini untuk mencetak "Hari kerja"
    case "Sabtu", "Minggu":
        fmt.Println("hari libur") // (10) Lengkapi bagian ini untuk mencetak "Hari libur"
    default:
        fmt.Println("Hari tidak valid")
    }
}
```

OUTPUT:

```

PS D:\> go run d:\1031124000042_LATSOL\1a.go
Contoh if-else:
Nilai B
PS D:\> go run d:\1031124000042_LATSOL\1b.go

Contoh for sebagai while:
1
2
3
4
5
PS D:\> go run d:\1031124000042_LATSOL\1c.go

Contoh for dengan range:
4
6
5
6
PS D:\> go run d:\1031124000042_LATSOL\1d.go

Contoh switch-case:
HARI KERJA
PS D:\> █

```

NOMOR 2

```
package main
```

```
import (
```

```
    "fmt"
```

```
    "math"
```

```
)
```

```
// Fungsi dengan parameter dan return value
```

```
func hitungLuasLingkaran(jariJari float64) float64 {
```

```
    return math.Pi * jariJari * jariJari // (1) Lengkapi rumus luas lingkaran
```

```
}
```

// Fungsi dengan multiple return values

```
func minMax(angka []int) (int, int) {  
    if len(angka) == 0 {  
        return 0, 0 // (2) Pastikan return value yang benar jika array kosong  
    }  
  
    min := angka[0]  
    max := angka[0]  
  
    for _, nilai := range angka {  
        if nilai < min {  
            min = nilai // (3) Lengkapi agar min selalu mendapat nilai terkecil  
        }  
        if nilai > max {  
            max = nilai // (4) Lengkapi agar max selalu mendapat nilai terbesar  
        }  
    }  
  
    return min, max  
}
```

// Fungsi dengan named return values

```
func hitungStatistik(angka []float64) (min, max, avg float64) {  
    if len(angka) == 0 {  
        return 0, 0, 0  
    }  
  
    min = angka[0]  
    max = angka[0]  
    var total float64 = 0
```

```

for _, nilai := range angka {
    if nilai < min {
        min = nilai // (5) Lengkapi agar min selalu mendapat nilai terkecil
    }
    if nilai > max {
        max = nilai // (6) Lengkapi agar max selalu mendapat nilai terbesar
    }
    total += nilai
}

avg = total / float64(len(angka)) // (7) Lengkapi perhitungan rata-rata
return          // implisit return untuk named return values
}

// Fungsi dengan variadic parameter
func jumlahkan(angka ...int) int {
    total := 0
    for _, nilai := range angka {
        total += nilai // (8) Lengkapi proses penjumlahan
    }
    return total
}

func main() {
    // Contoh penggunaan fungsi dengan return value
    radius := 7.0
    luas := hitungLuasLingkaran(radius) // (9) Panggil fungsi hitungLuasLingkaran dengan
parameter yang benar
    fmt.Printf("Luas lingkaran dengan jari-jari %.1f adalah %.2f\n", radius, luas)
}

```

```
// Contoh penggunaan fungsi dengan multiple return values

data := []int{23, 45, 12, 67, 34, 8}

minimal, maksimal := minMax(data) // (10) Panggil fungsi minMax dengan parameter yang benar

fmt.Printf("Nilai minimum: %d, Nilai maksimum: %d\n", minimal, maksimal)

}
```

OUTPUT:

```
PS D:\> go run d:\1031124000042_LATSOL\2.go
Luas lingkaran dengan jari-jari 7.0 adalah 153.94
Nilai minimum: 8, Nilai maksimum: 67
PS D:\> 
```

NOMOR 3

```
package main
```

```
import (
    "fmt"
)
```

```
func tampilkanHeader() {
    fmt.Print("=====") // (1) Lengkapi untuk mencetak garis atas

    fmt.Println("    PROGRAM MAHASISWA    ")

    fmt.Print("=====") // (2) Lengkapi untuk mencetak garis bawah
}
```

```
// Prosedur dengan parameter value

func tampilkanInfo(nama string, nim string, jurusan string) {
    fmt.Println("Informasi Mahasiswa:")

    fmt.Printf("Nama   : %s\n", nama)

    fmt.Printf("Nim    : %s\n", nim) // dengan format yang benar
}
```

```
    fmt.Printf("Jurusan : %s\n", jurusan)
}
```

```
// Prosedur dengan parameter pointer
```

```
func ubahNilai(nilai *int) {
    *nilai += 10
    return nilaingkapi agar mencetak nilai setelah diubah
}
```

```
// Prosedur dengan struct parameter
```

```
type Mahasiswa struct {
    Nama   string
    NIM    string
    Jurusan string
    Nilai  map[string]int
}
```

```
func tampilkanNilai(mhs Mahasiswa) {
    fmt.Printf("Nilai mahasiswa %s:\n", mhs.Nama)
    for matkul, nilai := range mhs.Nilai {

    }
}
```

```
// Prosedur dengan slice parameter
```

```
func tampilkanDaftarMahasiswa(daftar []string) {
    fmt.Println("Daftar Mahasiswa:")
    for i, nama := range daftar {
        fmt.Print(nama,nim)// (6) Lengkapi agar mencetak nomor dan nama mahasiswa
    }
}
```



```

func main() {
    // Memanggil prosedur tanpa parameter
    _____ // (7) Lengkapi agar memanggil prosedur tampilkanHeader

    // Memanggil prosedur dengan parameter value
    _____ // (8) Lengkapi agar memanggil prosedur tampilkanInfo dengan
data yang sesuai

    // Memanggil prosedur dengan parameter pointer
    nilai := 75
    fmt.Printf("Nilai awal: %d\n", nilai)
    _____ // (9) Lengkapi agar memanggil prosedur ubahNilai dengan
parameter yang benar
    fmt.Printf("Nilai akhir: %d\n", nilai)

    // Memanggil prosedur dengan struct parameter
    mhs := Mahasiswa{
        Nama: "Ani Wijaya",
        NIM: "87654321",
        Jurusan: "Sistem Informasi",
        Nilai: map[string]int{
            "Algoritma": 85,
            "Basis Data": 90,
            "Pemrograman Web": 78,
            "Struktur Data": 82,
        },
    }

    _____ // (10) Lengkapi agar memanggil prosedur tampilkanNilai dengan
parameter yang sesuai
}

```

OUTPUT:

Nomor 4

```
package main
```

```
import (
```

```
    "fmt"
```

```
)
```

```
// Rekursif untuk menghitung faktorial
```

```
func faktorial(n int) int {
```

```
    // Basis/kondisi penghentian rekursi
```

```
    if n == 0 || n == 1 {
```

```
        return 1
```

```
    }
```

```
    // Langkah rekursif
```

```
    return _____ // (1) Lengkapi bagian ini
```

```
}
```

```
// Rekursif untuk menghitung bilangan Fibonacci
```

```
func fibonacci(n int) int {
```

```
    if n <= 1 {
```

```
        return n
```

```
    }
```

```
    return _____ // (2) Lengkapi bagian ini
```

```
}
```

```
// Rekursif untuk menghitung pangkat
```

```
func pangkat(base int, eksponen int) int {
```

```
    if eksponen == 0 {
```

```
        return 1
```

```
    }
```

```
        return _____ // (3) Lengkapi bagian ini
    }
```

// Rekursif untuk mengecek palindrome

```
func isPalindrome(s string) bool {
    if len(s) <= 1 {
        return true
    }
    if s[0] != s[len(s)-1] {
        return false
    }
    return _____ // (4) Lengkapi bagian ini
}
```

// Rekursif dengan helper function (untuk menghitung jumlah elemen array)

```
func sum(arr []int) int {
    return _____ // (5) Lengkapi bagian ini
}
```

```
func sumHelper(arr []int, index int) int {
    if index >= len(arr) {
        return 0
    }
    return _____ // (6) Lengkapi bagian ini
}
```

```
func main() {
    // Contoh penggunaan rekursif faktorial
    fmt.Printf("Faktorial 5 = %d\n", faktorial(5))

    // Contoh penggunaan rekursif fibonacci
}
```

```
fmt.Println("Deret Fibonacci:")
```

```
for i := 0; i < 10; i++ {
```

```
    fmt.Printf("%d ", _____) // (7) Lengkapi bagian ini
```

```
}
```

```
fmt.Println()
```

```
// Contoh penggunaan rekursif pangkat
```

```
fmt.Printf("2 pangkat 8 = %d\n", _____) // (8) Lengkapi bagian ini
```

```
// Contoh penggunaan rekursif palindrome
```

```
kata1 := "katak"
```

```
kata2 := "mobil"
```

```
fmt.Printf("Apakah '%s' palindrome? %t\n", kata1, _____) // (9)
```

Lengkapi bagian ini

```
fmt.Printf("Apakah '%s' palindrome? %t\n", kata2, _____) // (10)
```

Lengkapi bagian ini

```
// Contoh penggunaan rekursif dengan helper function
```

```
angka := []int{1, 2, 3, 4, 5}
```

```
fmt.Printf("Jumlah elemen array = %d\n", sum(angka))
```

```
}
```