

Soal

1. Struktur Kontrol

```
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") // (1) Lengkapi bagian ini untuk mencetak
        "Nilai A"
    } else if nilai >= 80 {
        fmt.Println("Nilai B")
    } else if nilai >= 70 {
        fmt.Println("Nilai C") // (2) Lengkapi bagian ini untuk mencetak
        "Nilai C"
    } else if nilai >= 60 {
        fmt.Println("Nilai D")
    } else {
        fmt.Println("Nilai E") // (3) Lengkapi bagian ini untuk mencetak
        "Nilai E"
    }

    // 2. Struktur perulangan for (seperti while)
    fmt.Println("\nContoh for sebagai while:")
    counter := 1
    for counter <= 5 {
        fmt.Printf("Iterasi ke-%d\n", counter) // (4) Lengkapi bagian ini
        untuk mencetak "Iterasi ke-X"
        counter++
    }

    // 3. Struktur perulangan for dengan range
    fmt.Println("\nContoh for dengan range:")
    buah := []string{"Apel", "Mangga", "Jeruk", "Pisang"}
    for index, item := range buah { // (5) Lengkapi bagian ini agar
        mencetak indeks dan nama buah
        fmt.Printf("Buah pada index %d adalah %s\n", index, item) //
        (6) lengkapi bagian ini untuk emncetak "Buah pada index X adalah Y"
    }
}
```

```

// 4. Struktur switch-case
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")
}

```

```

}

```

```
PS D:\aru\perkuliahan\Semester 2\Alpro Teori & Praktek\Modul
han\Semester 2\Alpro Teori & Praktek\Modul 3\coso1\coso1.go"
Contoh if-else:
Nilai B

Contoh for sebagai while:
Iterasi ke-1
Iterasi ke-2
Iterasi ke-3
Iterasi ke-4
Iterasi ke-5

Contoh for dengan range:
Buah pada index 0 adalah Apel
Buah pada index 1 adalah Mangga
Buah pada index 2 adalah Jeruk
Buah pada index 3 adalah Pisang

Contoh switch-case:
Hari kerja
PS D:\aru\perkuliahan\Semester 2\Alpro Teori & Praktek\Modul
```

2. Fungsi

```
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)
}

```

```

PS D:\aru\perkuliahan\Semester 2\Alpro Teori & Praktek\Modul 3> go run "d:\aru\perkuliahan\Semester 2\Alpro Teori & Praktek\Modul 3\coso 2\coso2.go"
Luas lingkaran dengan jari-jari 7.0 adalah 153.94
Nilai minimum: 8, Nilai maksimum: 67
PS D:\aru\perkuliahan\Semester 2\Alpro Teori & Praktek\Modul 3>

```

3. Prosedur

```

package main

import (
    "fmt"
)

// Prosedur sederhana tanpa parameter
func tampilkanHeader() {
    fmt.Println("=====") // (1)Lengkapi untuk
    mencetak garis atas
    fmt.Println("        PROGRAM MAHASISWA        ")
    fmt.Println("=====") // (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) // (3)Lengkapi untuk mencetak NIM
    dengan format yang benar
    fmt.Printf("Jurusan  : %s\n", jurusan)
}

// Prosedur dengan parameter pointer
func ubahNilai(nilai *int) {
    *nilai += 10
    fmt.Printf("Nilai setelah diubah: %d\n", *nilai) // (4)Lengkapi 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 {
        fmt.Printf("%s: %d\n", matkul, nilai) // (5)Lengkapi agar
        mencetak nama mata kuliah dan nilai
    }
}

// Prosedur dengan slice parameter
func tampilkanDaftarMahasiswa(daftar []string) {
    fmt.Println("Daftar Mahasiswa:")
    for i, nama := range daftar {
        fmt.Printf("%d. %s\n", i+1, nama) // (6)Lengkapi agar mencetak
        nomor dan nama mahasiswa
    }
}

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

```

```
// Memanggil prosedur dengan parameter value
tampilkanInfo("Budi Santoso", "12345678", "Teknik Informatika") //
(8)Lengkapi agar memanggil prosedur tampilkanInfo
```

```
// Memanggil prosedur dengan parameter pointer
nilai := 75
fmt.Printf("Nilai awal: %d\n", nilai)
ubahNilai(&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,
    },
}
```

```
tampilkanNilai(mhs) // (10)Lengkapi agar memanggil prosedur
tampilkanNilai dengan parameter yang sesuai
```

```
}
```

```
PS D:\aru\perkuliahan\Semester 2\Alpro Teori & Praktek\Modul 3> go
han\Semester 2\Alpro Teori & Praktek\Modul 3\coso 3\coso3.go"
```

```
=====
PROGRAM MAHASISWA
=====
```

```
Informasi Mahasiswa:
Nama      : Budi Santoso
NIM       : 12345678
Jurusan   : Teknik Informatika
Nilai awal: 75
Nilai setelah diubah: 85
Nilai akhir: 85
Nilai mahasiswa Ani Wijaya:
Pemrograman Web: 78
Struktur Data: 82
Algoritma: 85
Basis Data: 90
PS D:\aru\perkuliahan\Semester 2\Alpro Teori & Praktek\Modul 3>
```


4. Rekursif

```
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 n * faktorial(n-1) // (1) Lengkapi bagian ini  
}
```

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

```
func fibonacci(n int) int {  
    if n <= 1 {  
        return n  
    }  
    return fibonacci(n-1) + fibonacci(n-2) // (2) Lengkapi bagian ini  
}
```

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

```
func pangkat(base int, eksponen int) int {  
    if eksponen == 0 {  
        return 1  
    }  
    return base * pangkat(base, eksponen) // (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 isPalindrome(s[1 : len(s)-1]) // (4) Lengkapi bagian ini
}

// Rekursif dengan helper function (untuk menghitung jumlah elemen array)
func sum(arr []int) int {
    return sumHelper(arr, 0) // (5) Lengkapi bagian ini
}

func sumHelper(arr []int, index int) int {
    if index >= len(arr) {
        return 0
    }
    return arr[index] + sumHelper(arr, index+1) // (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 ", fibonacci(i)) // (7) Lengkapi bagian ini
    }
    fmt.Println()

    // Contoh penggunaan rekursif pangkat
    fmt.Printf("2 pangkat 8 = %d\n", pangkat(2, 8)) // (8) Lengkapi bagian ini
}

```

```
// Contoh penggunaan rekursif palindrome
kata1 := "katak"
kata2 := "mobil"
fmt.Printf("Apakah '%s' palindrome? %t\n", kata1, isPalindrome(kata1)) //
(9) Lengkapi bagian ini
    fmt.Printf("Apakah '%s' palindrome? %t\n", kata2, isPalindrome(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))
}
```

```
PS D:\aru\perkuliahan\Semester 2\Alpro Teori & Praktek\Modul 3> go run "d:\aru\perkuliahan\Semester 2\Alpro Teori & Praktek\Modul 3\coso 4\coso4.go"
Faktorial 5 = 120
Deret Fibonacci:
0 1 1 2 3 5 8 13 21 34
runtime: goroutine stack exceeds 1000000000-byte limit
runtime: sp=0xc0200e1390 stack=[0xc0200e0000, 0xc0400e0000]
fatal error: stack overflow

runtime stack:
runtime.throw({0xfbd88c?, 0x124?})
        C:/Program Files/Go/src/runtime/panic.go:1067 +0x4d fp=0xed3dbffc90 sp=0xed3dbffc60 pc=0xf64d6d
runtime.newstack()
        C:/Program Files/Go/src/runtime/stack.go:1117 +0x5cc fp=0xed3dbffdd0 sp=0xed3dbffc90 pc=0xf4b50c
runtime.morestack()
        C:/Program Files/Go/src/runtime/asm_amd64.s:621 +0x79 fp=0xed3dbffdd8 sp=0xed3dbffdd0 pc=0xf6a059

goroutine 1 gp=0xc000058000 m=0 mp=0x1069600 [running]:
main.pangkat(0x2, 0x8?)
        d:/aru/perkuliahan/Semester 2/Alpro Teori & Praktek/Modul 3/coso 4/coso4.go:26 +0x46 fp=0xc0200e13a0 sp=0xc0200e1390 pc=0xf99c86
main.pangkat(0x2, 0x0?)
        d:/aru/perkuliahan/Semester 2/Alpro Teori & Praktek/Modul 3/coso 4/coso4.go:30
```