

1.)

```
// Savila Nur Fadilla
// 10311240031

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")
    }

    // 2. Struktur perulangan for (seperti while)
    fmt.Println("\nContoh for sebagai while:")
    counter := 1
    for counter <= 5 {
        fmt.Printf("Iterasi ke-%v\n", counter)
        counter++
    }

    // 3. Struktur perulangan for dengan range
    fmt.Println("\nContoh for dengan range:")
    buah := []string{"Apel", "Mangga", "Jeruk", "Pisang"}
    for indeks, item := range buah {
        fmt.Printf("Buah pada indeks %v adalah %v\n", indeks, item)
    }

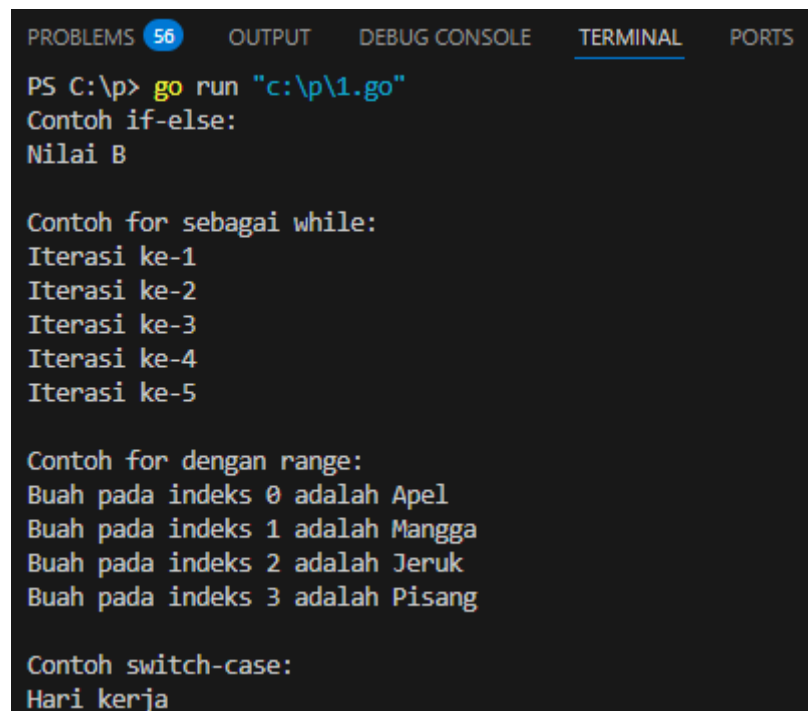
    // 4. Struktur switch-case
    fmt.Println("\nContoh switch-case:")
    hari := "Senin"
    switch hari {
    case "Senin":
        fmt.Println("Hari kerja")
    case "Selasa":
        // ...
    }
}
```

```

        fmt.Println("Hari kerja")
    case "Rabu":
        fmt.Println("Hari kerja")
    case "Kamis":
        fmt.Println("Hari kerja")
    case "Jumat":
        fmt.Println("Hari kerja")
    case "Sabtu", "Minggu":
        fmt.Println("Hari libur")
    default:
        fmt.Println("Hari tidak valid")
}
}

```

Output:



```

PROBLEMS 56 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\p> go run "c:\p\1.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 indeks 0 adalah Apel
Buah pada indeks 1 adalah Mangga
Buah pada indeks 2 adalah Jeruk
Buah pada indeks 3 adalah Pisang

Contoh switch-case:
Hari kerja

```

2.)

```
2.) package main

import (
    "fmt"
    "math"
)

// Fungsi dengan parameter dan return value
func hitungLuasLingkaran(jariJari float64) float64 {
    return math.Pi * (math.Pow(jariJari, 2))
}

// Fungsi dengan multiple return values
func minMax(angka []int) (int, int) {
    if len(angka) == 0 {
        return 0, 0
    }

    min := angka[0]
    max := angka[0]

    for _, nilai := range angka {
        if nilai < min {
            min = nilai
        }
        if nilai > max {
            max = nilai
        }
    }

    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
        }
    }
}
```

```

    }
    if nilai > max {
        max = nilai
    }
    total += nilai
}

avg = total / float64(len(angka)) //
return // implisit return untuk named return values
}

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

func main() {
    // Contoh penggunaan fungsi dengan return value
    radius := 7.0
    luas := hitungLuasLingkaran(radius)
    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)
    fmt.Printf("Nilai minimum: %d, Nilai maksimum: %d\n", minimal, maksimal)
}

```

Output:

```

PROBLEMS 56 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\p> go run "c:\p\2.go"
Luas lingkaran dengan jari-jari 7.0 adalah 153.94
Nilai minimum: 8, Nilai maksimum: 67

```

3.)

```
package main

import (
    "fmt"
)

// Prosedur sederhana tanpa parameter
func tampilkanHeader() {
    fmt.Println("=====")
    fmt.Println("          PROGRAM MAHASISWA          ")
    fmt.Println("=====")
}

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

// Prosedur dengan parameter pointer
func ubahNilai(nilai *int) {
    *nilai += 10
    fmt.Printf("Nilai setelah diubah: %v\n", *nilai)
}

// 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.Println(matkul, nilai)
    }
}

// Prosedur dengan slice parameter
func tampilkanDaftarMahasiswa(daftar []string) {
    fmt.Println("Daftar Mahasiswa:")
    for i, nama := range daftar {
```

```

        fmt.Println(i, nama)
    }
}

func main() {
    // Memanggil prosedur tanpa parameter
    tampilkanHeader()

    // Memanggil prosedur dengan parameter value
    tampilkanInfo("Ani", "87654321", "Informatika")

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

```

Output:

PROBLEMS 55 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\p> go run "c:\p\3.go"
```

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

```
=====
```

```
Informasi Mahasiswa:
```

```
Nama      : Ani
```

```
NIM       : 87654321
```

```
Jurusan  : Informatika
```

```
Nilai awal: 75
```

```
Nilai setelah diubah: 85
```

```
Nilai akhir: 85
```

```
Nilai mahasiswa Ani Wijaya:
```

```
Algoritma 85
```

```
Basis Data 90
```

```
Pemrograman Web 78
```

```
Struktur Data 82
```

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 n * faktorial(n-1)
}

// Rekursif untuk menghitung bilangan Fibonacci
func fibonacci(n int) int {
    if n <= 1 {
        return n
    }
    return fibonacci(n-1) + fibonacci(n-2)
}

// Rekursif untuk menghitung pangkat
func pangkat(base int, eksponen int) int {
    if eksponen == 0 {
        return 1
    }
    return
}

// 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 sumHelper(arr, 0)
}

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())
    }
    fmt.Println()

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

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

    // Contoh penggunaan rekursif dengan helper function
    angka := []int{1, 2, 3, 4, 5}
    fmt.Printf("Jumlah elemen array = %d\n", sum(angka))
}

```