

Nama : Sheila Stephanie Anindya

NIM : 103112400086

1. Source Code:

```
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")
    }
}
```

Output:

```
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code> go run "c:\Sheila Steph
Contoh if-else:
Nilai B
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>
```

1. Source Code:

```
package main

import "fmt"

func main() {

    fmt.Println("\nContoh for sebagai while:")
    counter := 1
    for counter <= 5 {
        fmt.Println("Iterasi ke-", counter)
        counter++
    }
}
```

Output:

```
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code> go run "c:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code\main.go"

Contoh for sebagai while:
Iterasi ke- 1
Iterasi ke- 2
Iterasi ke- 3
Iterasi ke- 4
Iterasi ke- 5
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>
```

1. Source Code:

```
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(buah, item) // (6) Lengkapi bagian ini untuk mencetak "Buah pada
index X adalah Y"
    }
}

```

Output:

```

PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code> go run "c:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code\main.go"

Contoh for dengan range:
[Apel Mangga Jeruk Pisang] Apel
[Apel Mangga Jeruk Pisang] Mangga
[Apel Mangga Jeruk Pisang] Jeruk
[Apel Mangga Jeruk Pisang] Pisang
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>

```

1. Source Code:

```

package main

import "fmt"

func main() {

    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:

```

PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code> go run "c:\Sheila Stephanie
Contoh switch-case:
Hari kerja
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>

```

2. Source Code:

```

package main

import (
    "fmt"
)

// Fungsi dengan parameter dan return value
func hitungLuasLingkaran(jariJari float64) float64 {
    return (phi * (r * r))
}

```

```

}

// Fungsi dengan multiple return values
func minMax(angka []int) (int, int) {
    if len(angka) == 0 {
        _____ // (2) Pastikan return value yang benar jika array
        kosong
    }

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

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

        } else {
        }

        if nilai > max {
            _____ // (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 {
        _____ // (5) Lengkapi agar min selalu mendapat nilai
        terkecil
    }
    if nilai > max {
        _____ // (6) Lengkapi agar max selalu mendapat
        nilai terbesar
    }
    total += nilai
}

avg = total / _____ // (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 {
        _____ // (8) Lengkapi proses penjumlahan
    }
    return total
}

func main() {
    // Contoh penggunaan fungsi dengan return value
    radius := 7.0
    luas := _____ // (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 := _____ // (10) Panggil fungsi
minMax dengan parameter yang benar
fmt.Printf("Nilai minimum: %d, Nilai maksimum: %d\n", minimal, maksimal)
}

```

Output:

```

PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code> go run "c:\Sheila Stephanie Anindya\SMT 2\A
# command-line-arguments
latihan 5\tempCodeRunnerFile.go:9:10: undefined: phi
latihan 5\tempCodeRunnerFile.go:9:17: undefined: r
latihan 5\tempCodeRunnerFile.go:15:3: undefined: _____
latihan 5\tempCodeRunnerFile.go:27:4: undefined: _____
latihan 5\tempCodeRunnerFile.go:46:4: undefined: _____
latihan 5\tempCodeRunnerFile.go:49:4: undefined: _____
latihan 5\tempCodeRunnerFile.go:54:16: undefined: _____
latihan 5\tempCodeRunnerFile.go:61:9: declared and not used: nilai
latihan 5\tempCodeRunnerFile.go:62:3: undefined: _____
latihan 5\tempCodeRunnerFile.go:70:10: undefined: _____

```

3. Source Code:

```

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("%d\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.Printf("Mata Kuliah  : %s\n", matkul)  
    }  
}
```

// Prosedur dengan slice parameter

```
func tampilkanDaftarMahasiswa(daftar []string) {  
    fmt.Println("Daftar Mahasiswa:")  
    for i, nama := range daftar {  
        fmt.Println(daftar, nama)  
    }  
}
```



```

}

func main() {
    var nilai int

    nilai := 75
    fmt.Printf("Nilai awal: %d\n", nilai)
    fmt.Printf("Ubah Nilai: %d\n", ubahNilai)
    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,
        },
    }
    fmt.Println(mhs, nilai)
}

```

Output:

```

PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code> go run "c:\Sheila Stephanie Anindya\SMT
# command-line-arguments
latihan 5\103112400086_3.go:38:14: declared and not used: nilai
latihan 5\103112400086_3.go:46:6: declared and not used: i
latihan 5\103112400086_3.go:54:8: no new variables on left side of :=

```

4. Source Code:

```
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 faktorial()
}

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

// 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 ", fibonacci()) // (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))
}

```

Output:

```

PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code> go run "c:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code\latihan 5\103112400086_4.go"
# command-line-arguments
latihan 5\103112400086_4.go:14:9: not enough arguments in call to faktorial
    have ()
    want (int)
latihan 5\103112400086_4.go:22:9: not enough arguments in call to fibonacci
    have ()

```