

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 5

MATERI



Oleh:

MUHAMMAD ZAKY MUBAROK

103112400073

KELAS

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

MODUL 5. REKURSIF

II. GUIDED RB

```
package main

import "fmt"

func pangkatIteratif(base, exp int) int {
    hasil := 1

    for i := 0; i < exp; i++ {
        hasil *= base
    }

    return hasil
}

func faktorialIteratif(n int) int {
    hasil := 1
    for i := 2; i <= n; i++ {
        hasil *= i
    }
    return hasil
}

func main() {
    var base, exp, n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&base)
    fmt.Print("Masukkan pangkat: ")
    fmt.Scanln(&exp)

    fmt.Printf("%d^%d = %d\n", base, exp, pangkatIteratif(base,
exp))
    fmt.Print("Masukkan angka untuk faktorial: ")
    fmt.Scanln(&n)
    fmt.Printf("%d! = %d\n", n, faktorialIteratif(n))
}
```

```
PS D:\ALPRO 2\103112400073_MODUL5> go r
Masukkan bilangan: 10
Masukkan pangkat: 2
10^2 = 100
Masukkan angka untuk faktorial: 5
5! = 120
PS D:\ALPRO 2\103112400073_MODUL5> █
```

Penjelasan :

Program ini berfungsi untuk menghitung **pangkat** dan **faktorial** sebuah bilangan menggunakan metode iteratif. Berikut penjelasannya:

1. **Fungsi** pangkatIteratif(base, exp):

- Menghitung pangkat dengan mengalikan base sebanyak exp kali dalam sebuah loop.

2. **Fungsi** faktorialIteratif(n):

- Menghitung faktorial dengan mengalikan semua bilangan dari 1 hingga n secara berulang dalam sebuah loop.

3. **Fungsi** main:

- Meminta pengguna memasukkan angka base dan exp untuk pangkat, lalu menghitung dan mencetak hasilnya.
- Meminta angka lain untuk faktorial, lalu menghitung dan mencetak hasilnya.

Contoh Output: Jika input adalah base = 2, exp = 3, maka output pangkat adalah $2^3 = 8$. Jika input untuk faktorial adalah n = 5, maka outputnya $5! = 120$

III. GUIDED REKURSIF

```
package main

import "fmt"

func pangkatRekursif(base, exp int) int {
    if exp == 0 {
        return 1
    }
    return base * pangkatRekursif(base, exp-1)
}

func faktorialRekursif(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    return n * faktorialRekursif(n-1)
}

func main() {
    var base, exp, n int
    // Input pangkat
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&base)
    fmt.Print("Masukkan pangkat: ")
    fmt.Scanln(&exp)

    fmt.Printf("%d^%d = %d\n", base, exp, pangkatRekursif(base, exp))

    // Input faktorial
    fmt.Print("Masukkan angka untuk faktorial: ")
    fmt.Scanln(&n)

    fmt.Printf("%d! = %d\n", n, faktorialRekursif(n))
}
```

PS D:\ALPRO 2\103112400073_MODUL5> g

Masukkan bilangan: 10

Masukkan pangkat: 2

10^2 = 100

Masukkan angka untuk faktorial: 5

5! = 120

Penjelasan :

Program ini menghitung pangkat dan faktorial menggunakan metode rekursif. Berikut ringkasan singkatnya:

1. Fungsi pangkatRekursif(base, exp):
 - Menghitung pangkat dengan memanggil dirinya sendiri (rekursif) hingga nilai eksponen (exp) mencapai 0, menghasilkan hasil akhir.
2. Fungsi faktorialRekursif(n):
 - Menghitung faktorial dengan memanggil dirinya sendiri untuk mengalikan angka dari n hingga 1.
3. Fungsi main():
 - Meminta input pengguna untuk bilangan (base) dan pangkat (exp), lalu mencetak hasil pangkat.
 - Meminta input bilangan untuk faktorial (n), lalu mencetak hasil faktorial.

Contoh Output:

- Jika input base = 2, exp = 3, hasilnya: $2^3 = 8$.
- Jika input n = 5, hasilnya: $5! = 120$

IV. UNGUIDED I

```
//Muhammad Zaky Mubarak
package main

import "fmt"

func fibonacci(n int) int {
    if n == 0 {
        return 0
    } else if n == 1 {
        return 1
    }

    return fibonacci(n-1) + fibonacci(n-2)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n (hingga suku ke-n): ")
    fmt.Scanln(&n)

    fmt.Println("Deret Fibonacci:")
    for i := 0; i <= n; i++ {
        fmt.Print(fibonacci(i), " ")
    }
}
```

```
Masukkan nilai n (hingga suku ke-n): 10
Deret Fibonacci:
0 1 1 2 3 5 8 13 21 34 55
```

Penjelasan :

Program ini menghasilkan deret Fibonacci hingga suku ke-n menggunakan metode **rekursif**. Berikut penjelasannya:

1. **Fungsi** fibonacci(n):

- Mengembalikan nilai Fibonacci untuk suku ke-n.
- Jika $n = 0$, hasilnya 0. Jika $n = 1$, hasilnya 1.
- Untuk $n > 1$, nilai dihitung dengan menambahkan hasil rekursif dari dua suku sebelumnya ($n-1$ dan $n-2$).

2. **Fungsi** main():

- Meminta input pengguna untuk menentukan nilai n (suku ke berapa deret Fibonacci akan dihitung).
- Menggunakan perulangan untuk mencetak deret Fibonacci dari suku ke-0 hingga suku ke-n.

Contoh Output: Jika input $n = 5$, maka output deret adalah: 0 1 1 2 3 5

V. UNGUIDED II

```
//Muhammad Zaky Mubarak
package main

import "fmt"

func cetakBintang(jumlah int, maksimum int) {

    if jumlah > maksimum {
        return
    }

    for i := 0; i < jumlah; i++ {
        fmt.Print("*")
    }

    fmt.Println()

    cetakBintang(jumlah+1, maksimum)
}

func main() {

    var jumlahBintang int
    fmt.Print("Masukkan jumlah bintang: ")
    fmt.Scanln(&jumlahBintang)

    cetakBintang(1, jumlahBintang)
}
```

```
Masukkan jumlah bintang: 5
*
**
***
****
*****
```

Penjelasan :

Program ini mencetak pola segitiga bintang secara rekursif. Berikut penjelasan singkatnya:

1. Fungsi cetakBintang(jumlah, maksimum):
 - Mencetak sejumlah bintang sesuai nilai jumlah.
 - Setiap pemanggilan fungsi mencetak satu baris bintang, lalu memanggil dirinya sendiri dengan jumlah+1 hingga mencapai nilai maksimum.
2. Fungsi main():
 - Meminta input pengguna untuk menentukan jumlah maksimum bintang pada baris terakhir.
 - Memanggil fungsi cetakBintang dengan nilai awal 1 untuk mulai mencetak pola.

VI. UNGUIDED III

```
//Muhammad Zaky Mubarak
package main

import "fmt"

func faktor(n int, i int) {

    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    faktor(n, i+1)
}

func main() {
    var N int
    fmt.Scanln(&N)

    faktor(N, 1)
    fmt.Println()
}
```

```
PS D:\ALPRO 2\103112400073_MODUL5>
5
1 5
PS D:\ALPRO 2\103112400073_MODUL5>
12
1 2 3 4 6 12
```

Penjelasan :

1. Fungsi faktor(n, i):

- Mengecek apakah i adalah faktor dari n. Jika ya, angka i dicetak.
- Fungsi ini dipanggil secara rekursif dengan i+1 hingga i melebihi n.

2. Fungsi main():

- Meminta input bilangan N dari pengguna.
- Memanggil fungsi faktor untuk mencari dan mencetak semua faktor N, mulai dari 1 hingga N.

Contoh Output: Jika input N = 12, maka outputnya: 1 2 3 4 6 12

VII. KESIMPULAN

Modul ini membahas fungsi (function) dalam pemrograman, khususnya dalam bahasa Go dan pseudocode. Berikut adalah beberapa poin utama dari modul ini:

1. Definisi Fungsi
 - Fungsi adalah subprogram yang mengembalikan nilai setelah diproses.
 - Digunakan untuk memecah program menjadi bagian yang lebih kecil, modular, dan mudah dipahami.
 - Berbeda dengan prosedur, fungsi harus memiliki nilai kembalian (return).
2. Deklarasi dan Implementasi Fungsi
 - Ditulis di luar func main().
 - Menggunakan parameter formal sebagai input.
 - Mengembalikan nilai dengan menggunakan return.
 - Contoh implementasi diberikan dalam pseudocode dan bahasa Go.
3. Pemanggilan Fungsi
 - Dilakukan dengan menyebutkan nama fungsi dan memberikan argumen yang sesuai.
 - Nilai kembalian bisa disimpan dalam variabel atau langsung digunakan dalam operasi lain.
4. Parameter dalam Fungsi
 - Parameter Formal: Variabel yang digunakan dalam deklarasi fungsi.
 - Parameter Aktual: Nilai yang dikirimkan saat fungsi dipanggil.
 - Mendukung Pass by Value (nilai dikopi) dan Pass by Reference (menggunakan pointer untuk referensi langsung ke memori).
5. Contoh Implementasi dalam Program
 - Menggunakan fungsi untuk menghitung luas lingkaran.
 - Menggunakan fungsi untuk menghitung faktorial.
 - Implementasi fungsi untuk menghitung volume tabung dan pangkat bilangan.

6. Latihan Pemrograman

- Implementasi fungsi untuk menghitung deret Fibonacci.
- Membuat program yang mencetak pola bintang menggunakan rekursi.
- Menampilkan faktor-faktor dari suatu bilangan menggunakan fungsi rekursif.
- Menghitung hasil pangkat dari dua bilangan tanpa menggunakan fungsi math bawaan.

REFERENSI

MODUL 5. REKURSIF