

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 5

REKURSIF



Oleh: Dimas Fanny Hebrasianto Permadi

NAMA: Dimas Ramadhani

NIM: 103112400065

KELAS: IF-12-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

1. Pengertian Rekursif

Rekursif adalah teknik dalam pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan masalah. Jadi, daripada menggunakan perulangan seperti for atau while, kita bisa menggunakan rekursif untuk melakukan tugas yang berulang.

2. Komponen Rekursif

- a. Base Case (Kondisi Berhenti)
 - Ini adalah syarat di mana fungsi rekursif akan berhenti memanggil dirinya sendiri. Tanpa base case, program akan berjalan terus-menerus atau tak berujung
 - Contoh: Jika kita ingin mencetak angka 1 sampai 5 base case-nya adalah saat angka mencapai 5, fungsi berhenti.
- b. Recursive Case (Proses Pemanggilan Ulang)
 - Bagian ini adalah instruksi di mana fungsi memanggil dirinya sendiri.
 - Contoh: Jika angka belum mencapai 5, panggil fungsi lagi dengan angka berikutnya.

3. Proses Forward dan Backward pada Rekursif

Proses rekursif memiliki dua arah:

1. Forward (Maju): Fungsi terus memanggil dirinya sendiri sampai memenuhi base case.
2. Backward (Mundur): Setelah base case tercapai, fungsi kembali menyelesaikan proses dari pemanggilan terakhir hingga pertama.

Contoh alur pemanggilan fungsi:

Jika kita memanggil cetak(1) sampai cetak(5), maka prosesnya seperti:

1. Cetak(1) → cetak(2) → cetak(3) → cetak(4) → cetak(5)
(Forward)
2. Cetak(5) selesai, lalu kembali ke cetak(4), cetak(3), cetak(2), cetak(1) (Backward)

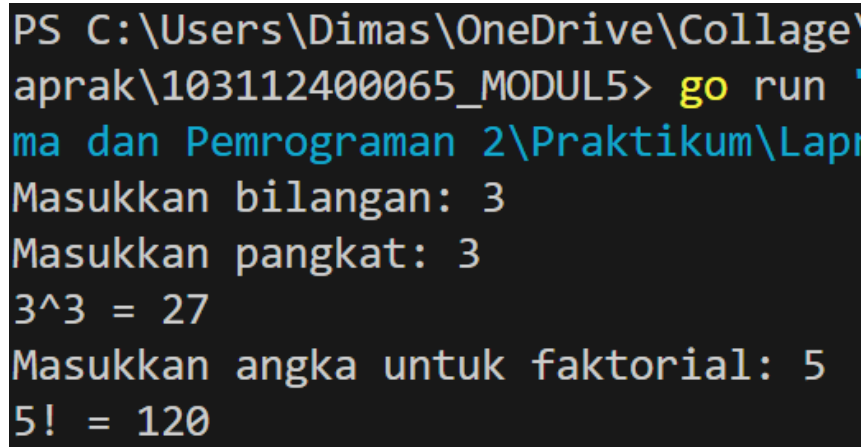
II. GUIDED

1. Nomor 1

- **Source Code:**

```
package main
import "fmt"
// Fungsi iteratif untuk menghitung pangkat (base^exp)
func pangkatIteratif(base, exp int) int {
    hasil := 1
    for i := 0; i < exp; i++ {
        hasil *= base
    }
    return hasil
}
// Fungsi iteratif untuk menghitung faktorial (n!)
func faktorialIteratif(n int) int {
    hasil := 1
    for i := 2; i <= n; i++ {
        hasil *= i
    }
    return hasil
}
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, pangkatIteratif(base, exp))
    // Input Faktorial
    fmt.Print("Masukkan angka untuk faktorial: ")
    fmt.Scanln(&n)
    fmt.Printf("%d! = %d\n", n, faktorialIteratif(n))
}
```

- **Screenshot Hasil Program:**



```
PS C:\Users\Dimas\OneDrive\Collage\
aparak\103112400065_MODUL5> go run
ma dan Pemrograman 2\Praktikum\Lap
Masukkan bilangan: 3
Masukkan pangkat: 3
3^3 = 27
Masukkan angka untuk faktorial: 5
5! = 120
```

- **Penjelasan:**

Program ini bertujuan untuk menghitung bilangan berpangkat dan bilangan faktorial variabel inputan kita yaitu base, exp dan n tipe int. Yang di mana variabel base sebagai angka yang akan di pangkatkan, lalu exp adalah pangkat dari base dan n adalah bilangan yang akan di faktorialkan.

Program ini menggunakan prosedur dengan nama pangkatIteratif bertipe int, dengan parameter base, exp int. Di dalam prosedur akan mendeklarasikan variabel hasil sama dengan 1. Lalu ada perulangan dengan iterasi 0 dan batasan kurang dari variabel exp. Di dalam Perulangan akan mengalikan dan pemberian nilai pada variabel hasil dengan variabel base. Setelahnya akan mengembalikan variabel hasil.

Program ini menggunakan prosedur dengan nama faktorialIteratif bertipe int, dengan parameter n int. Di dalam prosedur akan mendeklarasikan variabel hasil sama dengan 1. Lalu ada perulangan dengan iterasi 2 dan batasan kurang dari variabel n. Di dalam Perulangan akan mengalikan dan pemberian nilai pada variabel hasil dengan variabel i. Setelahnya akan mengembalikan variabel hasil.

Masuk ke program utama, dengan mendeklarasikan base, exp dan n tipe int. Program mencetak kalimat lalu membaca inputan diulang sebanyak 3 kali sesuai pada SOURCE CODE. Lalu program akan mencetak dan menjalankan prosedur pangkatIteratif(base, exp). Setelahnya program akan mencetak kalimat dan membaca inputan,

disini inputan untuk faktorial. Lalu program akan mencetak dan menjalankan prosedur faktorialIteratif(n).

2. Nomor 2

- Source Code:

```
package main
import "fmt"
// Fungsi rekursif untuk menghitung pangkat (base^exp)
func pangkatRekursif(base, exp int) int {
    if exp == 0 {
        return 1
    }
    return base * pangkatRekursif(base, exp-1)
}
// Fungsi rekursif untuk menghitung faktorial (n!)
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))
}
```

- **Screenshot Hasil Program:**

```
PS C:\Users\Dimas\OneDrive\Collage
dan Pemrograman 2\Praktikum\Lapra
> go run "c:\Users\Dimas\OneDrive\
goritma dan Pemrograman 2\Praktiku
_MODUL5\103112400065_Guided2.go"
Masukkan bilangan: 3
Masukkan pangkat: 3
3^3 = 27
Masukkan angka untuk faktorial: 5
5! = 120
```

- **Penjelasan:**

Program ini bertujuan untuk menghitung bilangan berpangkat dan bilangan faktorial variabel inputan kita yaitu base, exp dan n tipe int. Yang di mana variabel base sebagai angka yang akan di pangkatkan, lalu exp adalah pangkat dari base dan n adalah bilangan yang akan di faktorialkan.

Program ini menggunakan prosedur dengan nama pangkatRekursif bertipe int, dengan parameter base, exp int. Di dalam prosedur akan melakukan percabangan dengan kondisi $exp = 0$, return 1. Di luar percabangan akan mengembalikan $base \times pangkatRekursif(base, exp - 1)$.

Program ini menggunakan prosedur dengan nama faktoriaRekursif bertipe int, dengan parameter n int. Di dalam prosedur akan melewati percabangan dengan kondisi $n = 0$ atau $n = 1$, maka akan mengembalikan nilai 1. Di luar percabangan akan mengembalikan $n \times faktorialRekursif(n - 1)$.

Masuk ke program utama, dengan mendeklarasikan base, exp dan n tipe int. Program mencetak kalimat lalu membaca inputan diulang sebanyak 3 kali sesuai pada SOURCE CODE. Lalu program akan mencetak dan menjalankan prosedur pangkatRekursif(base, exp). Setelahnya program akan mencetak kalimat dan membaca inputan,

disini inputan untuk faktorial. Lalu program akan mencetak dan menjalankan prosedur faktorial(n).

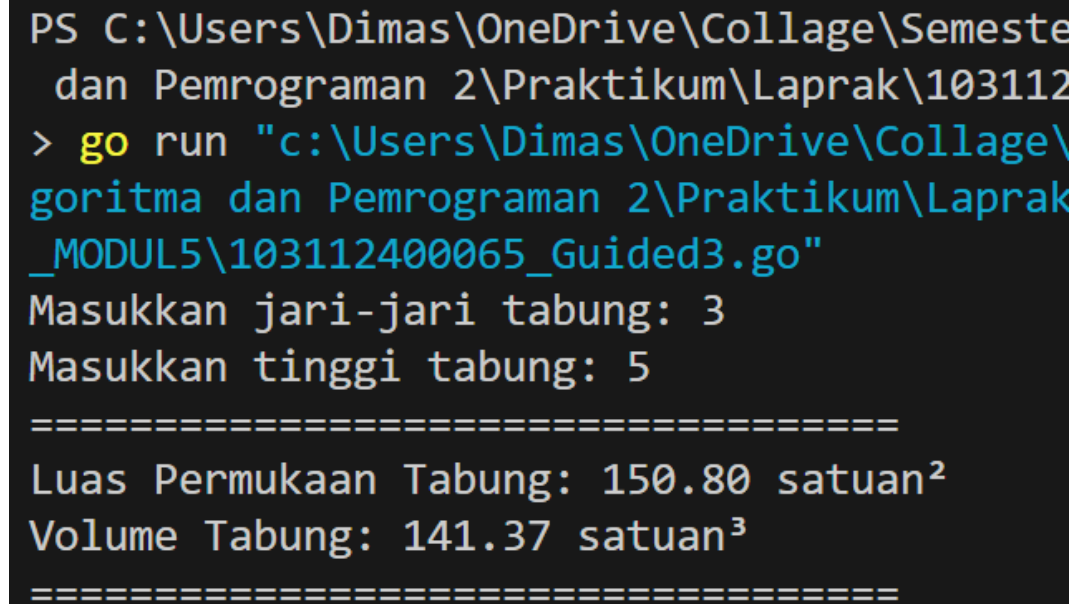
3. Nomor 3

- Source Code:

```
package main
import (
    "fmt"
    "math"
)
// Fungsi untuk menghitung luas permukaan tabung
func luasPermukaanTabung(r, t float64) float64 {
    return 2 * math.Pi * r * (r + t)
}
// Fungsi untuk menghitung volume tabung
func volumeTabung(r, t float64) float64 {
    return math.Pi * math.Pow(r, 2) * t
}
func main() {
    var r, t float64
    // Input jari-jari dan tinggi tabung dengan validasi
    fmt.Print("Masukkan jari-jari tabung: ")
    _, errR := fmt.Scan(&r)
    fmt.Print("Masukkan tinggi tabung: ")
    _, errT := fmt.Scan(&t)
    // Memeriksa apakah input valid
    if errR != nil || errT != nil {
        fmt.Println("Input tidak valid! Harap masukkan angka yang benar.")
        return
    }
    // Memeriksa apakah jari-jari dan tinggi bernilai positif
    if r <= 0 || t <= 0 {
        fmt.Println("Jari-jari dan tinggi tabung harus lebih dari nol.")
        return
    }
    // Menghitung luas permukaan dan volume
    luas := luasPermukaanTabung(r, t)
    volume := volumeTabung(r, t)
    // Menampilkan hasil
    fmt.Println("=====")
    fmt.Printf("Luas Permukaan Tabung: %.2f satuan²\n", luas)
```

```
fmt.Printf("Volume Tabung: %.2f satuan³\n", volume)
fmt.Println("=====")
}
```

- **Screenshoot Hasil Program:**



```
PS C:\Users\Dimas\OneDrive\Collage\Semester 5\Algoritma dan Pemrograman 2\Praktikum\Labrak\103112400065_Guided3.go
> go run "c:\Users\Dimas\OneDrive\Collage\Semester 5\Algoritma dan Pemrograman 2\Praktikum\Labrak\103112400065_Guided3.go"
Masukkan jari-jari tabung: 3
Masukkan tinggi tabung: 5
=====
Luas Permukaan Tabung: 150.80 satuan²
Volume Tabung: 141.37 satuan³
=====
```

- **Penjelasan:**

Program ini bertujuan untuk menghitung luas permukaan tabung dan volume tabung lalu variabel inputan seperti r dan t tipe float64. Bilangan untuk menggambarkan jari-jari akan disimpan pada variabel r sedangkan untuk bilangan tinggi tabung akan disimpan pada variabel t.

Program ini menggunakan prosedur dengan nama luasPermukaanTabung bertipe float64, dengan parameter r dan t tipe float64. Prosedur ini menghitung luas permukaan tabung menggunakan rumus $Luas\ Permukaan = 2 \times \pi \times r \times (r + t)$. Lalu mengembalikan variabel hasil perhitungan luas permukaan tabung bertipe float64.

Program ini menggunakan prosedur dengan nama volumeTabung bertipe float64, dengan parameter r dan t tipe float64. Prosedur ini menghitung volume tabung menggunakan rumus $Volume = \pi \times r^2 \times t$. Lalu akan mengembalikan nilai hasil perhitungan volume tabung bertipe float64.

Masuk ke program utama, mendeklarasikan variabel r dan t tipe float64. Inputan data, program meminta pengguna memasukkan jari-jari dan tinggi tabung. Lalu akan melalui percabangan jika input tidak valid program akan mencetak "Input tidak valid! Harap masukkan angka yang benar." Namun jika r atau t bernilai kurang dari atau sama dengan 0, program mencetak kalimat "Jari-jari dan tinggi tabung harus lebih dari nol." Jika input valid, perhitungan akan dilanjutkan. Proses perhitungan pertama menghitung luas permukaan menggunakan `luasPermukaanTabung(r, t)` lalu menghitung volume menggunakan `volumeTabung(r, t)`. Output hasilnya akan menampilkan hasil perhitungan luas permukaan dan volume sesuai format pada SOURCE CODE.

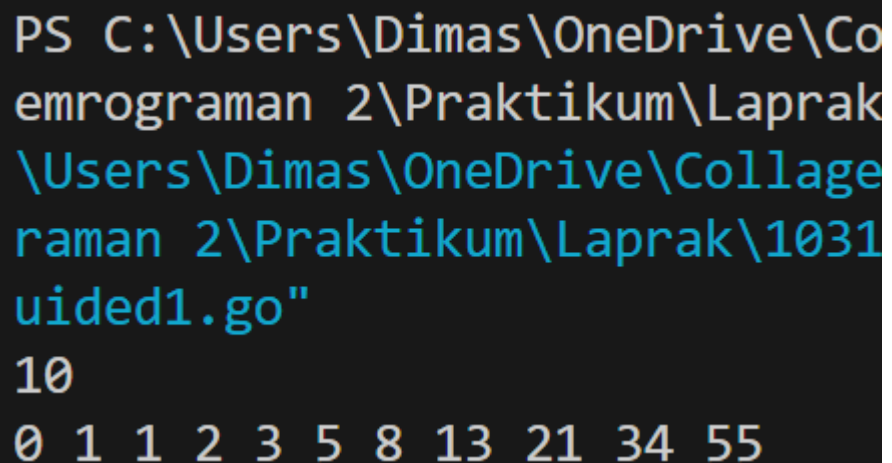
III. UNGUIDED

1. Nomor 1

- **Source Code:**

```
// Dimas Ramadhani
// 103112400065
package main
import "fmt"
func deretFibonacci(x int) int {
    if x == 0 {
        return 0
    } else if x == 1 {
        return 1
    } else {
        return deretFibonacci(x-1)+deretFibonacci(x-2)
    }
}
func main() {
    var a int
    fmt.Scan(&a)
    for i := 0; i <= a; i++ {
        fmt.Print(deretFibonacci(i), " ")
    }
}
```

- **Screenshot Hasil Pembahasan:**



```
PS C:\Users\Dimas\OneDrive\College\raman 2\Praktikum\Laprak\103112400065> go run unguided1.go
10
0 1 1 2 3 5 8 13 21 34 55
```

- **Penjelasan:**

Program ini bertujuan untuk menampilkan deret Fibonacci hingga angka ke- a yang dimasukkan oleh pengguna. Program menggunakan rekursi untuk menghitung nilai pada setiap posisi deret Fibonacci.

Program ini menggunakan prosedur dengan nama `deretFibonacci` bertipe `int`, dengan parameter x `int`, dengan x menunjukkan posisi angka terakhir. Prosedur ini berfungsi menghitung nilai Fibonacci menggunakan rekursif, jika $x = 0$ maka fungsi mengembalikan 0 (nilai pertama dalam deret). Jika $x = 1$, maka fungsi mengembalikan 1 (nilai kedua dalam deret). Jika $x > 1$, maka fungsi memanggil dirinya sendiri untuk menjumlahkan dua nilai sebelumnya $F(x) = F(x - 1) + F(x - 2)$ lalu mengembalikan nilai Fibonacci pada posisi ke- x bertipe `int`.

Masuk ke program utama, dengan mendeklarasikan variabel a tipe `int` berguna untuk menyimpan batas jumlah deret Fibonacci yang diinput oleh pengguna. Lalu Program akan melakukan perulangan dari $i = 0$ hingga $i \leq a$. Setiap iterasi memanggil fungsi `deretFibonacci(i)` untuk menghitung nilai Fibonacci di posisi tersebut. Hasilnya dicetak di baris yang sama dengan spasi di antara setiap angka.

2. Nomor 2

- **Source Code:**

```
// Dimas Ramadhani
// 103112400065
package main
import "fmt"
func buatSegitiga(x int) {
    if x == 0 {
        return
    }
    buatSegitiga(x - 1)
    for i := 1; i <= x; i++ {
        fmt.Print("*")
    }
    fmt.Println()
}
func main() {
    var a int
    fmt.Scan(&a)
    buatSegitiga(a)
}
```

- **Screenshot Hasil Pembahasan:**

```
PS C:\Users\Dimas\emrograman 2\Users\Dimas\emrograman 2\Praktikum\src> go run provided2.go
5
*
**
***
****
*****

PS C:\Users\Dimas\emrograman 2\Users\Dimas\emrograman 2\Praktikum\src> go run provided2.go
1
*

PS C:\Users\Dimas\emrograman 2\Users\Dimas\emrograman 2\Praktikum\src> go run provided2.go
3
*
**
***
```

- **Penjelasan:**

Program ini bertujuan untuk mencetak segitiga bintang dengan tinggi sesuai angka yang diinput oleh pengguna. Program menggunakan rekursi dalam Prosedur `buatSegitiga` untuk mencetak baris bintang secara bertahap dari atas ke bawah.

Program ini menggunakan prosedur dengan nama `buatSegitiga`, dengan parameter `x int`, berfungsi untuk menentukan tinggi segitiga yang akan dicetak. Prosedur ini menggunakan pendekatan rekursif untuk mencetak bintang. Baris rekursi, jika $x = 0$ maka prosedur berhenti (tidak mencetak apa pun). Panggilan rekursif, memanggil prosedur `buatSegitiga(x-1)` untuk mencetak baris sebelumnya. Cetak baris, setelah pemanggilan rekursif selesai, mencetak `x` bintang dalam satu baris. Rekursi berjalan dari `x` ke 1, memastikan baris pertama yang dicetak memiliki 1 bintang, diikuti 2, dan seterusnya sampai `x`.

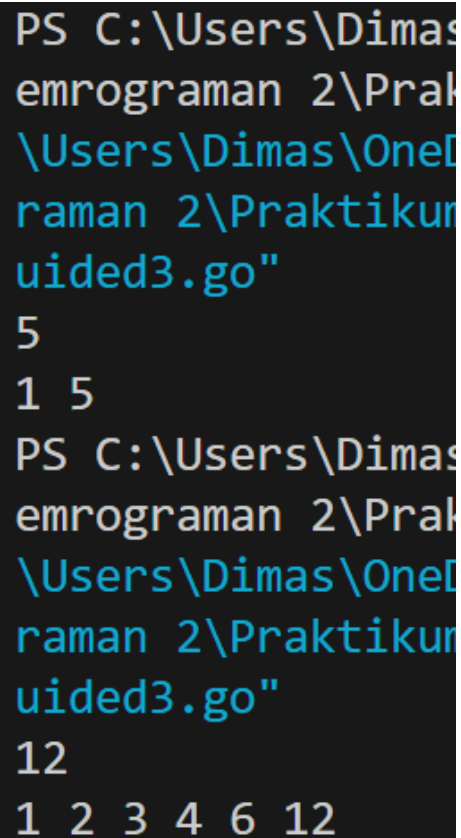
Masuk ke program utama, dengan mendeklarasikan variabel `a` tipe `int`, berfungsi untuk menyimpan tinggi segitiga yang diinput oleh pengguna. Program akan meminta input dari pengguna. Setelah itu program memanggil prosedur `buatSegitiga(a)` untuk mencetak segitiga bintang.

3. Nomor 3

- **Source Code:**

```
// Dimas Ramadhani
// 103112400065
package main
import "fmt"
func faktorBilangan(x, i int) {
    if i > x {
        return
    }
    if x%i == 0 {
        fmt.Print(i, " ")
    }
    faktorBilangan(x, i+1)
}
func main() {
    var a int
    fmt.Scan(&a)
    faktorBilangan(a, 1)
}
```

- **Screenshoot Hasil Pembahasan:**



```
PS C:\Users\Dimas\OneDrive\Documents\emrograman 2\Praktikum\Users\Dimas\OneDrive\Documents\emrograman 2\Praktikum\Guided3.go"
5
1 5
PS C:\Users\Dimas\OneDrive\Documents\emrograman 2\Praktikum\Users\Dimas\OneDrive\Documents\emrograman 2\Praktikum\Guided3.go"
12
1 2 3 4 6 12
```


- **Penjelasan:**

Program ini bertujuan untuk menampilkan faktor-faktor dari suatu bilangan yang diinput oleh pengguna. Faktor bilangan adalah angka-angka yang habis membagi bilangan tersebut.

Program ini menggunakan prosedur dengan nama faktorBilangan, dengan parameter variabel x dan i tipe int, untuk variabel x berfungsi sebagai tempat bilangan yang akan dicari faktornya sedangkan variabel i berguna untuk Nilai iterator (dimulai dari 1). Fungsi prosedur ini menggunakan metode rekursi untuk mencetak semua faktori dari bilangan x. Baris rekursi, jika i lebih besar dari x maka fungsi berhenti. Proses Rekursif, Jika x habis dibagi i maka i dicetak sebagai faktor. Langkah Rekursi, memanggil dirinya sendiri dengan i+1 hingga $i > x$.

Masuk ke program utama, dengan mendeklarasikan variabel a tipe int, untuk menyimpan bilangan yang akan dicari faktornya. Lalu program akan meminta input dari pengguna. Lalu program akan memanggil prosedur faktorBilangan(a, 1) yang di mana memulai pencarian faktor dari angka 1 hingga a

IV. KESIMPULAN

Rekursif adalah cara dalam pemrograman di mana sebuah fungsi bisa memanggil dirinya sendiri. Teknik ini berguna untuk menyelesaikan masalah yang bisa dipecah menjadi bagian-bagian yang lebih kecil dan berulang.

Agar rekursif berjalan dengan benar, ada dua hal penting yang harus diperhatikan:

1. Base Case (Kondisi Berhenti): Ini adalah syarat agar proses berhenti dan tidak berjalan terus-menerus.
2. Recursive Case (Pemanggilan Ulang): Bagian ini adalah tempat di mana fungsi memanggil dirinya sendiri untuk melanjutkan proses.

Rekursif sering digunakan dalam situasi yang membutuhkan langkah-langkah berulang, seperti menghitung faktorial, menghasilkan deret Fibonacci, atau mencari faktor bilangan. Walaupun rekursif bisa menggantikan perulangan biasa (seperti for atau while), dalam beberapa kasus, perulangan lebih efisien dan cepat.

Intinya, rekursif adalah alat yang kuat jika digunakan dengan benar, terutama saat menangani masalah yang memiliki pola berulang atau bertingkat.

V. REFERENSI

Prayogo, N. A. (2024). *Dasar Pemrograman Golang* (Versi 4.0.20240830). Retrieved from <https://github.com/novalagung/dasarpemrogramangolang>