

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL V

REKURSIF



Oleh :

NAMA : Felix Pedrosa Valentino

NIM : 103112400056

KELAS : IF – 12 – 01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Fungsi rekursif adalah sebuah fungsi yang memanggil atau mengeksekusi dirinya sendiri. Fungsi ini dapat dianggap sebagai salah satu cara untuk melakukan perulangan. Dalam penulisan kode aplikasi, adakalanya menggunakan fungsi rekursif menjadi lebih mudah dan efisien. Salah satu contoh sederhana penerapan fungsi rekursif adalah dalam perhitungan faktorial.

Saat membuat fungsi rekursif, penting untuk memastikan bahwa fungsi tersebut memiliki kondisi untuk berhenti. Jika tidak, hal ini dapat menyebabkan error stack overflow, yakni situasi di mana fungsi terus-menerus memanggil dirinya sendiri tanpa henti. Oleh karena itu, biasanya fungsi rekursif tidak langsung memanggil dirinya sendiri, melainkan tergantung pada kondisi tertentu yang telah ditentukan.

Berikut adalah beberapa catatan penting mengenai teknik rekursif :

- Teknik rekursif merupakan alternatif untuk menggantikan struktur kontrol perulangan dengan memanfaatkan subprogram, baik itu fungsi maupun prosedur.
- Untuk menghentikan proses rekursif, digunakan percabangan (if-then).
- Base-case adalah kondisi yang menandai berakhirnya proses rekursif. Mengetahui base-case merupakan hal terpenting yang harus diperhatikan saat membuat program rekursif, karena tidak mungkin membangun program rekursif tanpa pemahaman tentang base-case terlebih dahulu.
- Recursive-case adalah kondisi di mana subprogram memanggil dirinya sendiri. Recursive-case adalah kebalikan atau komplemen dari base-case.
- Setiap algoritma rekursif selalu memiliki padanan dalam bentuk algoritma iteratif.

Komponen Rekursif

Algoritma rekursif terdiri dari dua komponen utama yang sangat penting:

1. Base-case (Basis)

Bagian yang berfungsi untuk menghentikan proses rekursif. Base-case adalah elemen krusial dalam suatu algoritma rekursif, karena tanpa adanya basis yang jelas, algoritma tersebut dapat berlanjut tanpa henti.

2. Recursive-case

Bagian yang melakukan pemanggilan subprogram. Di sinilah proses rekursif berlangsung, di mana fungsi memanggil dirinya sendiri dengan argumen yang berbeda.

II. GUIDED

1. Guided 1

Source Code :

```
// Felix Pedrosa V

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))
}

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\alpro2_week4_modul5\103112400056_Guided1\103112400056_Guided1.go"
Masukkan bilangan: 5
Masukkan pangkat: 5
5^5 = 3125
Masukkan angka untuk faktorial: 5
5! = 120

```

Penjelasan Program :

Program di atas merupakan program yang menggunakan bahasa Go dan bertujuan untuk menghitung pangkat dan faktorial dari bilangan yang dimasukkan oleh pengguna. Terdapat dua fungsi utama dalam program ini yaitu pangkatIteratif dan faktorialIteratif. Fungsi pangkatIteratif menerima dua parameter, yakni bilangan dasar (base) dan eksponen (exp), dan menghitung hasil pangkat dengan mengalikan bilangan dasar sebanyak eksponen kali dalam sebuah loop. Sementara itu, fungsi faktorialIteratif berfungsi untuk menghitung faktorial dari bilangan bulat positif (n) dengan cara mengalikan semua bilangan bulat dari 1 hingga n.

Di dalam fungsi main, program akan meminta pengguna untuk memasukkan bilangan dan pangkat yang diinginkan, lalu menampilkan hasil perhitungan pangkat melalui fungsi pangkatIteratif. Selanjutnya, program meminta pengguna untuk memasukkan angka yang ingin dihitung faktorialnya dan menampilkan hasilnya menggunakan fungsi faktorialIteratif.

2. Guided 2

Source Code :

```

// Felix Pedrosa V

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))
}

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week4_modul5\103112400056_Guided2\103112400056_Guided2.go"
Masukkan bilangan: 5
Masukkan pangkat: 5
5^5 = 3125
Masukkan angka untuk faktorial: 5
5! = 120

```

Penjelasan Program :

Program di atas merupakan program yang menggunakan bahasa Go dan bertujuan untuk menghitung pangkat dan faktorial dari bilangan yang dimasukkan oleh pengguna dengan memanfaatkan pendekatan rekursif. Terdapat dua fungsi utama yang berperan dalam program ini yaitu pangkatRekursif dan faktorialRekursif.

Fungsi pangkatRekursif bertugas menghitung pangkat dari suatu bilangan dasar (base) dengan eksponen (exp), dengan cara memanggil dirinya sendiri berulang kali hingga eksponen mencapai nol, yang pada titik ini hasilnya menjadi 1. Sebaliknya, fungsi faktorialRekursif digunakan untuk menghitung faktorial dari bilangan bulat positif (n) dengan cara yang serupa, di mana ia memanggil dirinya sendiri hingga mencapai nilai 0 atau 1 yang keduanya juga menghasilkan 1.

Dalam fungsi main, program meminta pengguna untuk memasukkan bilangan dan pangkat yang diinginkan. Setelah itu, hasil perhitungan pangkat akan ditampilkan menggunakan fungsi pangkatRekursif. Selanjutnya, pengguna akan diminta untuk memasukkan angka yang akan dihitung faktorialnya, dan hasilnya akan ditampilkan menggunakan fungsi faktorialRekursif.

III. UNGUIDED

1. UnGuided 1

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"

// Fungsi rekursif untuk menghitung suku ke-n dari deret Fibonacci
func fibonacci(n int) int {
    if n == 0 {
        return 0
    } else if n == 1 {
        return 1
    } else {
        return fibonacci(n-1) + fibonacci(n-2)
    }
}

func main() {
    // Mencetak deret Fibonacci hingga suku ke-10
    for i := 0; i <= 10; i++ {
        fmt.Printf("Fibonacci(%d) = %d\n", i, fibonacci(i))
    }
}
```

Output :

```
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week4_modul5\103112400056_Unguided1\103112400056_Unguided1.go"
Fibonacci(0) = 0
Fibonacci(1) = 1
Fibonacci(2) = 1
Fibonacci(3) = 2
Fibonacci(4) = 3
Fibonacci(5) = 5
Fibonacci(6) = 8
Fibonacci(7) = 13
Fibonacci(8) = 21
Fibonacci(9) = 34
Fibonacci(10) = 55
```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa pemrograman Go dan berfungsi untuk menghitung nilai dari suku-suku deret Fibonacci dengan pendekatan rekursif. Deret Fibonacci dimulai dengan angka 0 dan 1, di mana setiap suku berikutnya diperoleh dengan menjumlahkan dua suku

sebelumnya. Di dalam program ini, terdapat sebuah fungsi bernama `fibonacci` yang menerima parameter integer n dan mengembalikan nilai sesuai dengan rumus dasar deret Fibonacci. Jika nilai n adalah 0, fungsi akan mengembalikan 0; jika n adalah 1, maka hasilnya adalah 1. Untuk nilai n yang lebih besar, fungsi ini akan memanggil dirinya sendiri dua kali, dengan parameter $n-1$ dan $n-2$, kemudian menjumlahkan hasilnya. Di dalam fungsi utama `main`, program akan mencetak deret Fibonacci mulai dari suku ke-0 hingga suku ke-10 dengan format yang jelas, sehingga pengguna dapat dengan mudah melihat nilai setiap suku dalam deret tersebut. Hasil akhirnya adalah deret yang secara jelas mencerminkan karakteristik dari deret Fibonacci.

2. UnGuided 2

Source Code :

```
// Felix Pedrosa V

package main

import "fmt"

// Fungsi rekursif untuk menampilkan bintang
func printStars( $n$  int,  $current$  int) {
    if  $current > n$  {
        return
    }
    // Mencetak bintang sesuai dengan angka current
    for  $i := 0$ ;  $i < current$ ;  $i++$  {
        fmt.Print("*")
    }
    fmt.Println() // Pindah ke baris berikutnya
    printStars( $n$ ,  $current+1$ ) // Panggil fungsi rekursif untuk current yang
    berikutnya
}

func main() {
    var  $N$  int
    fmt.Scan(& $N$ ) // Membaca input dari user

    printStars( $N$ , 1) // Memanggil fungsi untuk mulai dari 1
}
```

```
}
```

Output :

```
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week4_modul5\10311240056_Unguided2\10311240056_Unguided2.go"
5
*
**
***
****
*****
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week4_modul5\10311240056_Unguided2\10311240056_Unguided2.go"
1
*
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week4_modul5\10311240056_Unguided2\10311240056_Unguided2.go"
3
*
**
***
```

Penjelasan Program :

Program diatas ditulis dalam bahasa pemrograman Go dan berfungsi untuk menampilkan pola bintang berbentuk segitiga di layar. Proses dimulai dengan mendefinisikan fungsi rekursif bernama printStars, yang memiliki dua parameter yaitu n, yang menunjukkan jumlah total baris bintang yang ingin dicetak, dan current, yang menunjukkan jumlah bintang yang akan dicetak pada baris saat ini.

Fungsi ini terlebih dahulu memeriksa apakah nilai current sudah melebihi n ; jika iya, maka fungsi akan berhenti. Sebaliknya, jika current masih dalam batas yang ditentukan, program akan mencetak bintang sebanyak nilai current menggunakan loop for , kemudian berlanjut ke baris baru. Selanjutnya, fungsi printStars akan dipanggil kembali dengan current yang ditingkatkan satu, sehingga baris berikutnya akan menampilkan jumlah bintang yang lebih banyak.

Di dalam fungsi main, program meminta pengguna untuk memasukkan nilai integer N , yang kemudian digunakan untuk memulai proses pencetakan bintang dari baris pertama. Sebagai contoh, jika pengguna memasukkan angka 5, program akan mencetak 5 baris bintang, dengan baris pertama berisi 1 bintang, baris kedua 2 bintang, dan seterusnya hingga baris kelima yang berisi 5 bintang.

3. UnGuided 3

Source Code :

```
// Felix Pedrosa V
```

```

package main

import "fmt"

// Fungsi rekursif untuk mencetak faktor
func faktorRekursif(n int, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    faktorRekursif(n, i+1)
}

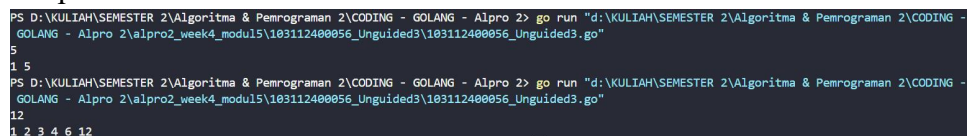
func main() {
    var N int

    // Membaca input dari pengguna
    fmt.Scan(&N)

    faktorRekursif(N, 1) // Memanggil fungsi rekursif dimulai dari 1
    fmt.Println() // Untuk newline setelah output
}

```

Output :



```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week4_modul5\103112400056_Unguided3\103112400056_Unguided3.go"
1 5
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week4_modul5\103112400056_Unguided3\103112400056_Unguided3.go"
12
1 2 3 4 6 12

```

Penjelasan Program :

Program di atas ditulis dalam bahasa pemrograman Go dan berfungsi untuk mencetak semua faktor dari sebuah bilangan bulat positif yang dimasukkan oleh pengguna. Proses dimulai dengan mendeklarasikan paket utama dan mengimpor paket `fmt` yang diperlukan untuk input dan output.

Sebuah fungsi bernama `faktorRekursif` didefinisikan untuk melakukan pencarian faktor secara rekursif. Fungsi ini menerima dua parameter: `n`, yang merupakan bilangan yang akan dicari faktornya, dan `i`, yang merupakan angka pengecekan untuk menentukan apakah `i` adalah faktor dari `n`. Dalam fungsi ini terdapat sebuah kondisi yang memeriksa apakah `i` sudah melebihi `n`, yang menandakan bahwa pencarian faktor telah selesai. Jika `i` terbukti sebagai faktor dari `n` (artinya `n` dibagi `i` memberikan sisa 0), maka nilai `i` akan dicetak. Selanjutnya, fungsi ini akan memanggil dirinya sendiri dengan parameter `i` yang ditingkatkan satu, sehingga pencarian faktor dapat dilanjutkan hingga semua angka dari 1 hingga `n` diperiksa.

Pada bagian fungsi main, program meminta pengguna untuk memasukkan sebuah bilangan bulat dan kemudian memanggil fungsi `faktorRekursif` dengan parameter `N` dan 1, sehingga pencarian faktor dimulai dari angka 1. Setelah semua faktor berhasil dicetak, program menambahkan newline untuk memperjelas output.

IV. KESIMPULAN

Kesimpulan dari modul rekursif ini menjelaskan bahwa teknik rekursif adalah metode yang sangat efektif dalam pemrograman, karena mampu menyelesaikan masalah dengan cara memecahnya menjadi sub-masalah yang lebih kecil dan serupa. Dalam pendekatan rekursif, terdapat dua komponen utama yang perlu dipahami, yaitu base-case, yang berfungsi untuk menghentikan proses rekursif, dan recursive-case, yang merupakan keadaan di mana fungsi memanggil dirinya sendiri.

Modul ini juga memberikan berbagai contoh penerapan rekursif, seperti perhitungan faktorial, deret Fibonacci, dan penjumlahan bilangan, yang memperlihatkan fleksibilitas serta kekuatan teknik ini dalam menyelesaikan beragam tantangan. Selain itu, ditekankan pentingnya memahami dan menentukan base-case sebelum mengembangkan program rekursif, karena tanpa kondisi ini, proses rekursif tidak akan pernah berhenti dan berpotensi menyebabkan kesalahan atau kegagalan pada program.

V. REFERENSI

Modul 5 - Praktikum Alpro 2

#27: Recursive Function - Belajar Golang Dari Dasar. (2022). Retrieved from <https://blog.ruangdeveloper.com/golang-recursive-function/>