

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 5
REKURSIF**



Oleh:

Muhamad Faza Fahri Aziz

103112400072

12 IF 01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. DASAR TEORI

Rekursi dalam Pemrograman

Rekursi adalah teknik dalam pemrograman di mana suatu fungsi memanggil dirinya sendiri untuk menyelesaikan suatu masalah dengan membaginya menjadi sub-masalah yang lebih kecil hingga mencapai kondisi dasar (base-case).

Komponen Rekursi

1. **Base-case** → Kondisi yang menghentikan rekursi agar tidak berjalan tanpa batas.
2. **Recursive-case** → Bagian di mana fungsi memanggil dirinya sendiri dengan parameter yang semakin mendekati base-case.

Cara Kerja Rekursi

- **Forward** → Fungsi terus memanggil dirinya sendiri hingga mencapai base-case.
- **Backward** → Setelah mencapai base-case, fungsi mulai kembali ke pemanggil sebelumnya.

Keunggulan Rekursi

- Memudahkan penyelesaian masalah yang kompleks.
- Kode lebih sederhana dan mudah dibaca untuk permasalahan yang bersifat rekursif.

Kelemahan Rekursi

- Menggunakan lebih banyak memori dibandingkan iterasi.
- Jika tidak memiliki base-case yang tepat, dapat menyebabkan infinite loop atau stack overflow.

Contoh Penggunaan Rekursi

- Menemukan faktor suatu bilangan.
- Menghitung faktorial.
- Menghitung perpangkatan.
- Algoritma pencarian dan sorting seperti QuickSort dan MergeSort.

II. GUIDED

Contoh 1

```

package main
import "fmt"

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

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

    return hasil
}

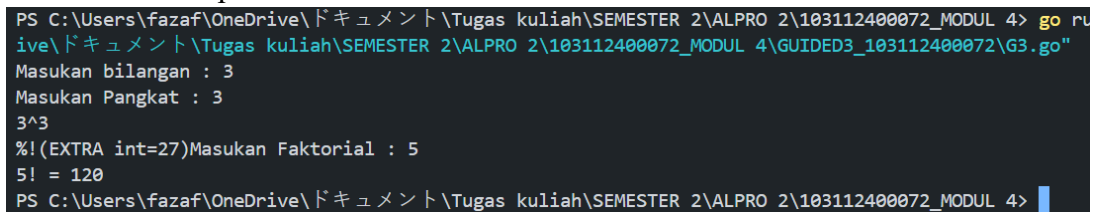
func faktorialIteratif(n int) int
{ hasil := 1 for i := 2; i <= n;
  i++ { hasil *= i
    }
  return hasil
}

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))
}

```

Screenshots Output



```

PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4> go run
ive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4\GUIDED3_103112400072\G3.go"
Masukan bilangan : 3
Masukan Pangkat : 3
3^3
%(EXTRA int=27)Masukan Faktorial : 5
5! = 120
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4>

```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menggunakan rekursi untuk menghitung perpangkatan dan faktorial. Fungsi pangkatRekursif menghitung hasil perpangkatan dengan mengalikan basis berulang kali hingga eksponennya mencapai nol. Sementara itu, fungsi faktorialRekursif menentukan nilai faktorial dengan mengalikan angka secara berurutan hingga mencapai satu. Di dalam fungsi main, pengguna diminta memasukkan bilangan dan pangkatnya, serta angka untuk faktorial, kemudian hasil perhitungan ditampilkan.

Contoh 2

```
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

    fmt.Print("masukkan bilangan : ")
    fmt.Scanln(&base)
    fmt.Print("masukkan pangkat : ")
    fmt.Scanln(&exp)
```

```

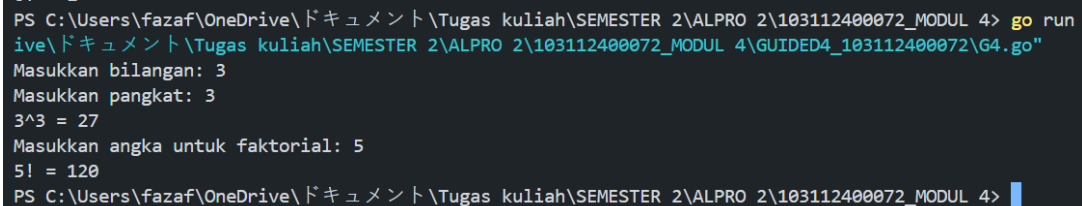
    fmt.Println("%d pangkat %d = %d\n", base, exp, pangkatRekursif(base, exp))

    fmt.Println("masukkan angka untuk faktorial : ")
    fmt.Scanln(&n)

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

```

Screenshots Output



```

PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4> go run
ive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4\GUIDED4_103112400072\G4.go
Masukkan bilangan: 3
Masukkan pangkat: 3
3^3 = 27
Masukkan angka untuk faktorial: 5
5! = 120
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4>

```

// Foto hasil dari menjalankan code

Program ini menggunakan metode iteratif untuk menghitung perpangkatan dan faktorial. Fungsi pangkatIteratif menghitung perpangkatan dengan mengalikan basis sebanyak exp kali dalam perulangan for, sedangkan fungsi faktorialIteratif menghitung faktorial dengan mengalikan angka secara berurutan dari 2 hingga n. Dalam fungsi main, pengguna diminta memasukkan bilangan dan pangkatnya, serta angka untuk faktorial, lalu hasil perhitungan ditampilkan.

III. UNGUIDED

Soal 1

```
// Muhamad faza fahri aziz || 103112400072

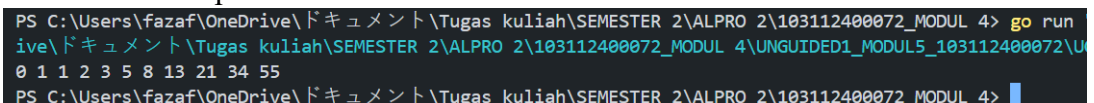
package main

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() {
    n := 10
    for i := 0; i <= n; i++ {
        fmt.Print(fibonacci(i), " ")
    }
    fmt.Println()
}
```

Screenshots Output



```
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4> go run
ive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4\UNGUIDED1_MODULE5_103112400072\U
0 1 1 2 3 5 8 13 21 34 55
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4>
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menampilkan deret Fibonacci dengan metode rekursif. Fungsi fibonacci(ok) menghitung nilai Fibonacci ke-ok dengan menjumlahkan dua nilai sebelumnya hingga mencapai kasus dasar, yaitu 0 atau 1. Dalam fungsi main(), pengguna memasukkan sebuah angka q, kemudian program mencetak deret Fibonacci sebanyak q angka dengan memanggil fibonacci(i) dalam perulangan.

Soal 2

```
//103112400072_Muhamad faza Fahri aziz
```

```
package main
```

```
import (  
    "fmt"  
)
```

```
func cetakBintang(n int, i int) {  
    if i > n {  
        return  
    }  
    fmt.Println(repeat("*", i))  
    cetakBintang(n, i+1)  
}
```

```
func repeat(char string, count int) string {  
    if count == 0 {  
        return ""  
    }  
    return char + repeat(char, count-1)  
}
```

```
func main() {  
    var n int  
    fmt.Print("Masukkan jumlah baris: ")  
    fmt.Scan(&n)  
    cetakBintang(n, 1)  
}
```

Screenshots Output

```
Masukkan jumlah baris: 5  
*  
**  
***  
****  
*****  
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\AL  
ive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODU  
Masukkan jumlah baris: 1  
*
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menampilkan pola segitiga bintang menggunakan perulangan bersarang. Fungsi cetakBintang(n) terdiri dari dua perulangan, di mana perulangan luar menentukan jumlah baris, sementara perulangan dalam mencetak bintang sesuai dengan jumlah baris tersebut. Dalam fungsi main(), pengguna memasukkan nilai n, lalu program memanggil cetakBintang(n) untuk mencetak pola bintang sebanyak n baris.

Soal 3

```
//103112400072 || Muhamad faza Fahri aziz
package main

import (
    "fmt"
)

func cetakFaktor(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Printf("%d ", i)
    }
    cetakFaktor(n, i+1)
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)
    fmt.Printf("Faktor dari %d: ", n)
    cetakFaktor(n, 1)
    fmt.Println()
}
```

Screenshots Output


```
Masukkan bilangan: 5
Faktor dari 5: 1 5
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4>
ive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4\UG5_M5.go"
Masukkan bilangan: 12
Faktor dari 12: 1 2 3 4 6 12
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4>
```

// Foto hasil dari menjalankan code

Deskripsi : Program ini menentukan dan menampilkan faktor dari suatu bilangan dengan metode rekursif. Fungsi cariFaktorial(angka, pembagi) mengecek apakah pembagi dapat membagi angka tanpa sisa. Jika iya, nilai tersebut dicetak, kemudian fungsi dipanggil kembali dengan pembagi + 1 hingga melebihi angka. Dalam fungsi main(), pengguna memasukkan bilangan positif n, lalu program menjalankan cariFaktorial(n, 1) untuk mencetak semua faktor dari n.

IV. KESIMPULAN

Materi ini membahas konsep rekursi dan iterasi dalam pemrograman, khususnya dalam bahasa Go. Rekursi adalah teknik di mana sebuah fungsi memanggil dirinya sendiri hingga mencapai kondisi dasar (base-case), sedangkan iterasi menggunakan perulangan untuk menyelesaikan suatu permasalahan. Beberapa penerapan dari kedua metode ini yang dibahas meliputi perhitungan faktorial, perpangkatan, deret Fibonacci, pencarian faktor bilangan, serta pencetakan pola bintang. Rekursi sering memberikan solusi yang lebih sederhana dan elegan untuk masalah yang bersifat berulang, namun dapat mengonsumsi lebih banyak memori dibandingkan dengan iterasi. Oleh karena itu, pemilihan antara rekursi dan iterasi harus mempertimbangkan efisiensi serta kebutuhan program.

V. REFERENSI

MODUL 5 REKURSIF ALGORITMA PEMROGRAMAN