

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 5
REKURSIF**



Oleh:

SAVILA NUR FADILLA

103112400031

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

5.1 Pengantar Rekursif

Pada modul-modul sebelumnya sudah dijelaskan bahwa suatu subprogram baik fungsi atau prosedur bisa memanggil subprogram lainnya. Hal ini tidak menutup kemungkinan bahwa subprogram yang dipanggil adalah dirinya sendiri. Dalam pemrograman teknik ini dikenal dengan istilah rekursif. Rekursif secara sederhana dapat diartikan sebagai cara menyelesaikan suatu masalah dengan cara menyelesaikan sub-masalah yang identik dari masalah utama.

- Teknik rekursif merupakan salah satu alternatif untuk mengganti struktur kontrol perulangan dengan memanfaatkan subprogram (bisa fungsi ataupun prosedur).
- Untuk menghentikan proses rekursif digunakan percabangan (if-then).
- Base-case adalah kondisi proses rekursif berhenti. Base-case merupakan hal terpenting dan pertama yang harus diketahui ketika akan membuat program rekursif. Mustahil membuat program rekursif tanpa mengetahui base-case terlebih dahulu.
- Recursive-case adalah kondisi dimana proses pemanggilan dirinya sendiri dilakukan. Kondisi recursive-case adalah komplemen atau negasi dari base-case.
- Setiap algoritma rekursif selalu memiliki padanan dalam bentuk algoritma iteratif.

5.2 Komponen Rekursif

Algoritma rekursif terdiri dari dua komponen utama:

1. Base-case (Basis), yaitu bagian untuk menghentikan proses rekursif dan menjadi komponen terpenting di dalam sebuah rekursif.
2. Recursive-case, yaitu bagian pemanggilan subprogramnya.

II. GUIDED

1.) Source Code

```
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

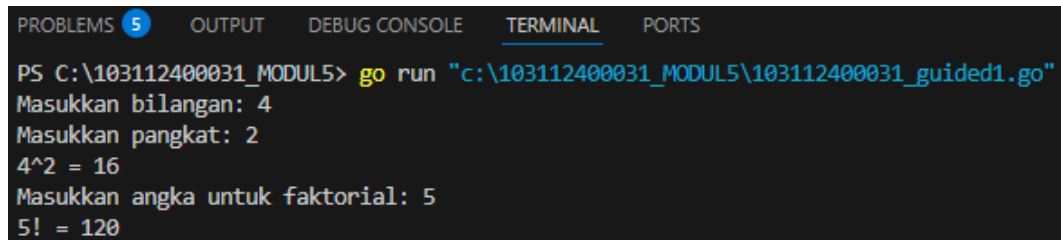
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&base)
    fmt.Print("Masukkan pangkat: ")
    fmt.Scanln(&exp)

    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))
}
```

Output :



```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL5> go run "c:\103112400031_MODUL5\103112400031_guided1.go"
Masukkan bilangan: 4
Masukkan pangkat: 2
4^2 = 16
Masukkan angka untuk faktorial: 5
5! = 120
```

Penjelasan : Program ini bertujuan untuk menghitung pangkat dan faktorial menggunakan metode iteratif. Program ini meminta input berupa bilangan dan pangkatnya, kemudian program akan menghitung hasil perhitungan pangkat tersebut menggunakan perulangan for. Selanjutnya program meminta input berupa angka untuk faktorial, lalu program akan menghitung hasil faktorial dari angka tersebut menggunakan perulangan for. Output berupa hasil perhitungan pangkat dan faktorial.

2.) Source Code

```
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

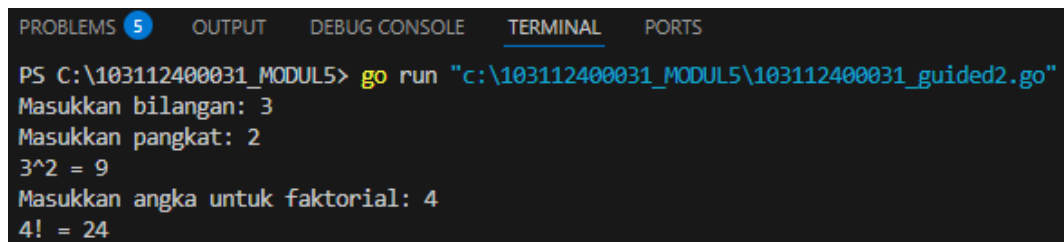
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&base)
    fmt.Print("Masukkan pangkat: ")
    fmt.Scanln(&exp)

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

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

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

Output :



```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL5> go run "c:\103112400031_MODUL5\103112400031_guided2.go"
Masukkan bilangan: 3
Masukkan pangkat: 2
3^2 = 9
Masukkan angka untuk faktorial: 4
4! = 24
```

Penjelasan : Program ini bertujuan untuk menghitung pangkat dan faktorial menggunakan metode rekursif. Program ini meminta input berupa bilangan dan pangkatnya, kemudian program akan menghitung hasil perhitungan pangkat tersebut secara rekursif (menggunakan percabangan if). Selanjutnya program meminta input berupa angka untuk faktorial, lalu program akan menghitung hasil faktorial dari angka tersebut secara rekursif (menggunakan percabangan if). Output berupa hasil perhitungan pangkat dan faktorial.

.

III. UNGUIDED

1.) Source code

```
// Savila Nur Fadilla
// 103112400031

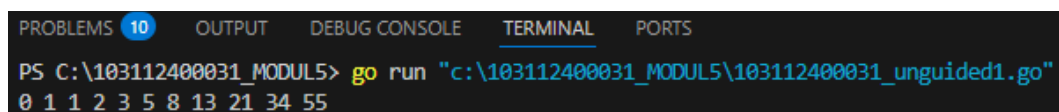
package main

import "fmt"

func fibonacci(n int) int {
    if n <= 1 {
        return n
    }
    return fibonacci(n-1) + fibonacci(n-2)
}

func main() {
    for i := 0; i <= 10; i++ {
        fmt.Printf("%v ", fibonacci(i))
    }
    fmt.Println()
}
```

Output :



The screenshot shows a Go IDE interface with tabs for PROBLEMS (10), OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, displaying the command `go run "c:\103112400031_MODULE5\103112400031_unguided1.go"` and its output: `0 1 1 2 3 5 8 13 21 34 55`.

Penjelasan : Program ini bertujuan untuk menghitung deret Fibonacci menggunakan metode rekursif. Jika n kurang dari atau sama dengan 1, fungsi langsung mengembalikan n . Jika $n > 1$, fungsi memanggil dirinya sendiri ($\text{fibonacci}(n-1) + \text{fibonacci}(n-2)$) untuk menghitung nilai Fibonacci secara rekursif. Program ini menggunakan loop untuk mencetak deret Fibonacci dari $n = 0$ hingga $n = 10$ dengan memanggil fungsi $\text{fibonacci}(n)$. Output berupa satu baris yang berupa deret Fibonacci dari 0 hingga 10.

2.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import "fmt"

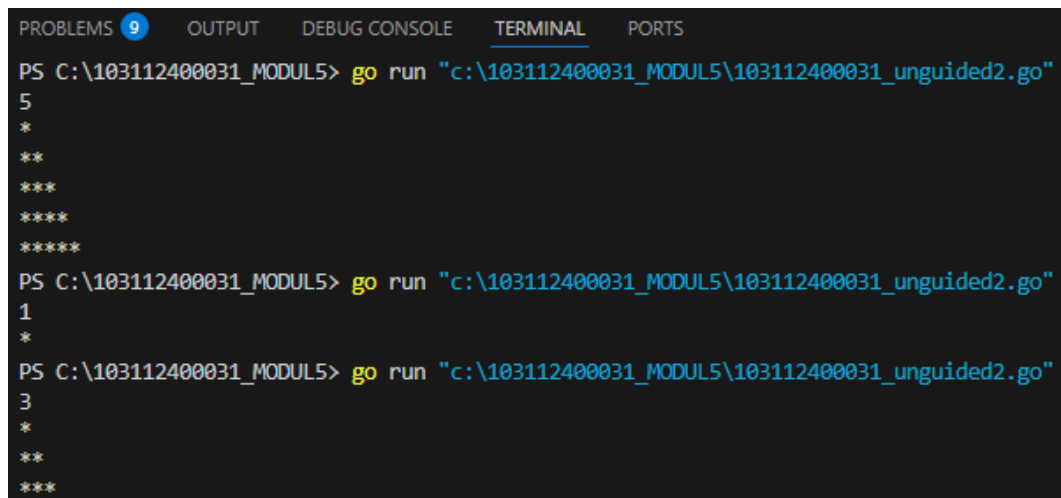
func cetakBaris(n int) {
    if n == 0 {
        return
    }
    fmt.Print("*")
    cetakBaris(n-1)
}

func cetakBintang(n int) {
    if n == 0 {
        return
    }
    cetakBintang(n-1)
    cetakBaris(n)
    fmt.Println()
}

func main() {
    var n int
    fmt.Scan(&n)

    cetakBintang(n)
}
```


Output :



```
PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL5> go run "c:\103112400031_MODUL5\103112400031_unguided2.go"
5
*
**
***
****
*****
PS C:\103112400031_MODUL5> go run "c:\103112400031_MODUL5\103112400031_unguided2.go"
1
*
PS C:\103112400031_MODUL5> go run "c:\103112400031_MODUL5\103112400031_unguided2.go"
3
*
**
***
```

Penjelasan : Program ini bertujuan untuk mencetak segitiga bintang menggunakan metode rekursif. Program ini akan meminta input berupa bilangan bulat n yang menyatakan jumlah baris. Fungsi `cetakBaris(n int)` untuk mencetak n bintang dalam satu baris menggunakan rekursi. Fungsi `cetakBintang(n int)` memanggil `cetakBintang(n-1)` untuk mencetak baris sebelumnya, lalu mencetak n bintang di baris saat ini. Output berupa pola segitiga bintang dimana setiap baris akan bertambah satu bintang.

3.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

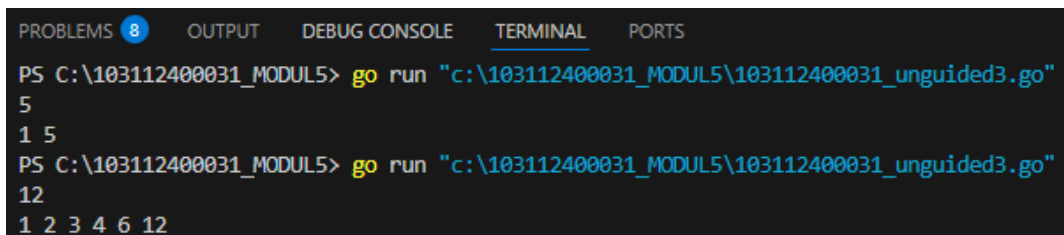
import "fmt"

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

func main() {
    var n int
    fmt.Scan(&n)

    cetakFaktor(n, 1)
    fmt.Println()
}
```

Output :



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL5> go run "c:\103112400031_MODUL5\103112400031_unguided3.go"
5
1 5
PS C:\103112400031_MODUL5> go run "c:\103112400031_MODUL5\103112400031_unguided3.go"
12
1 2 3 4 6 12
```

Penjelasan : Program ini bertujuan untuk menampilkan faktor dari suatu bilangan n dengan metode rekursif. Program ini akan meminta input berupa bilangan bulat n. Jika $i > n$, rekursi berhenti. Jika $n \% i == 0$, maka i adalah faktor dari n, lalu akan dicetak. Panggilan rekursif faktor(n, i+1) untuk memeriksa faktor berikutnya. Output berupa faktor faktor dari bilangan n.

IV. KESIMPULAN

Berdasarkan laporan ini, dapat disimpulkan jika soal soal tersebut menggunakan struktur fungsi rekursif. Struktur ini digunakan agar program lebih terstruktur, mudah dibaca, dan mudah dipahami. Beberapa dari program juga memuat struktur perulangan for ataupun memuat struktur percabangan if else di dalam struktur fungsinya.

- a.) Program iteratif menggunakan metode iterative untuk menghitung pangkat dan faktorial
- b.) Program rekursif menggunakan metode rekursif untuk menghitung pangkat dan faktorial
- c.) Program cetak deret menggunakan metode rekursif untuk mencetak deret Fibonacci
- d.) Program segitiga bintang menggunakan metode rekursif untuk mencetak bintang di setiap barisnya
- e.) Program cetak faktor menggunakan metode rekursif untuk menentukan faktor faktor dari bilangan yang diinput

REFERENSI

Telkom University. (2025). *Modul Praktikum Algoritma dan Pemrograman 2*.