

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK  
MODUL V  
“REKURSIF”**



**Oleh:**

**MUHAMMAD RAGIEL PRASTYO**

**2311102183**

**S1IF-11-02**

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2024**

## 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. Sebagai contoh perhatikan prosedur cetak berikut ini!

	Notasi Algoritma	Notasi dalam bahasa GO
1	procedure cetak(in x:integer)	func cetak(x int){
2	algoritma	fmt.Println(x)
3	output(x)	cetak(x+1)
4	cetak(x+1)	}
5	endprocedure	

Apabila diperhatikan subprogram cetak() di atas, terlihat pada baris ke-4 terdapat pemanggilan subprogram cetak() kembali. Misalnya apabila kita eksekusi perintah cetak(5) maka akan menampilkan angka 5 6 7 8 9...dst tanpa henti. Artinya setiap pemanggilan subprogram cetak() nilai x akan selalu bertambah 1 (increment by one) secara terus menerus tanpa henti.

Oleh karena itu bisanya ditambahkan struktur kontrol percabangan (if-then) untuk menghentikan proses rekursif ini. Kondisi ini disebut juga dengan base-case, artinya apabila kondisi base-case bernilai true maka proses rekursif akan berhenti. Sebagai contoh misalnya base case adalah ketika x bernilai 10 atau x < 10, maka tidak perlu dilakukan rekursif.

#### Catatan:

- Teknik rekursif ini 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:

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

## II. GUIDED

1. Membuat baris bilangan dari n hingga 1

Base-case: bilangan == 1

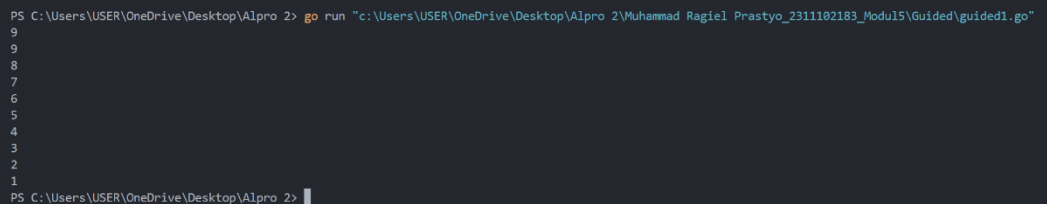
### Source Code

```
// MUHAMMAD RAGIEL PRASTYO
// 2311102183
package main
import "fmt"

func main() {
    var n int
    fmt.Scan(&n) // Membaca input pengguna
    baris(n) // Memanggil fungsi rekursif "baris"
}

func baris(bilangan int) {
    if bilangan == 1 {
        fmt.Println(1)
    } else {
        fmt.Println(bilangan)
        baris(bilangan - 1)
    }
}
```

### Screenshot Output



```
PS C:\Users\USER\OneDrive\Desktop\Alpro 2> go run "c:\Users\USER\OneDrive\Desktop\Alpro 2\Muhammad Ragiel Prastyo_2311102183_Modul5\Guided\guided1.go"
9
9
8
7
6
5
4
3
2
1
PS C:\Users\USER\OneDrive\Desktop\Alpro 2> █
```

### Penjelasan:

Program di atas akan menampilkan angka secara berurutan dari nilai yang dimasukkan hingga angka 1. Awalnya program mengambil input dari pengguna dan menyimpannya di variabel n. Selanjutnya, fungsi baris rekursif dipanggil dengan n sebagai parameter. Apabila bilangan memiliki nilai yang setara dengan 1, program akan mencetak angka 1. Jika nilai tersebut lebih besar,

program akan mencetak nilai tersebut dan melakukan panggilan rekursif dengan nilai yang lebih kecil. Proses terus berlangsung sampai mencetak angka 1.

## 2. Menghitung hasil penjumlahan 1 hingga n

Base-case:  $n == 1$

### Source Code

```
// MUHAMMAD RAGIEL PRASTYO
// 2311102183
package main
import "fmt"

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

func penjumlahan(n int) int {
    if n == 1 {
        return 1
    } else {
        return n + penjumlahan(n - 1)
    }
}
```

### Screenshot Output



```
PS C:\Users\USER\OneDrive\Desktop\Alpro 2> go run "c:\Users\USER\OneDrive\Desktop\Alpro 2\Muhammad Ragiel Prastyo_2311102183_Modul5\Guided\guided2.go"
83
3486
PS C:\Users\USER\OneDrive\Desktop\Alpro 2>
```

### Penjelasan:

Program di atas digunakan untuk menambahkan angka mulai dari 1 hingga nilai yang dimasukkan oleh pengguna. Program dimulai dengan mengambil input dan menyimpannya ke dalam variabel  $n$ . Setelah itu, program memanggil fungsi penjumlahan rekursif, yang melakukan perhitungan jumlah. Apabila nilai  $n$  adalah 1, maka fungsi akan menghasilkan nilai 1. Apabila nilai tersebut lebih tinggi, akan dikembalikan nilai  $n$  ditambahkan dengan jumlah dari  $n - 1$ . Dengan metode ini, program akan menjumlahkan semua angka dari 1 sampai  $n$ .

### III. UNGUIDED

- 1) Deret fibonacci adalah sebuah deret dengan nilai suku ke-0 dan ke-1 adalah 0 dan 1, dan nilai suku ke-n selanjutnya adalah hasil penjumlahan dua suku sebelumnya. Secara umum dapat diformulasikan  $S_n = S_{n-1} + S_{n-2}$  Berikut ini adalah contoh nilai deret fibonacci hingga suku ke-10. Buatlah program yang mengimplementasikan fungsi rekursif pada deret fibonacci tersebut.

#### Source Code

```
// MUHAMMAD RAGIEL PRASTYO
// 2311102183
package main
import "fmt"

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

func main() {
    var n int
    fmt.Print("Masukkan nilai n: ")
    fmt.Scan(&n)
    fmt.Printf("Deret Fibonacci(%d)adalah;\n", n)
    for i := 0; i <= n; i++ {
        fmt.Printf("Fibonacci(%d) -> %d\n", i, fibonacci(i))
    }
}
```

#### Screenshot Output

```
PS C:\Users\USER\OneDrive\Desktop\Alpro 2> go run "c:\Users\USER\OneDrive\Desktop\Alpro 2\Muhammad Ragiel Prastyo_2311102183_Modul5\Unguided\unguided1.go"
Masukkan nilai n: 10
Deret Fibonacci(10)adalah;
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
PS C:\Users\USER\OneDrive\Desktop\Alpro 2> |
```

### Penjelasan:

Program di atas menghasilkan deret Fibonacci hingga angka yang dimasukkan oleh pengguna. Program mempunyai fungsi fibonacci yang melakukan perhitungan angka Fibonacci secara berulang-ulang. Apabila nilai n kurang dari 1 atau sama dengan 1, fungsi akan mengembalikan nilai n. Jika angka tersebut lebih tinggi, maka akan menambahkan hasil dari dua panggilan sebelumnya. Di bagian utama program, pengguna diminta untuk memasukkan nilai n. Lalu, program akan mencetak deret Fibonacci dari 0 hingga n dengan menggunakan loop dan memanggil fungsi fibonacci untuk setiap angka, kemudian menampilkan hasilnya.

- 2) Buatlah sebuah program yang digunakan untuk menampilkan pola bintang berikut ini dengan menggunakan fungsi rekursif. N adalah masukan dari user.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	* ** *** **** *****
2	1	*
3	3	* ** ***

### Source Code

```
// MUHAMMAD RAGIEL PRASTYO
// 2311102183
package main
import (
    "fmt"
)

func cetakBintang(n int) {
    if n <= 0 {
        return
    }
}
```

```

    for i := 0; i < n; i++ {
        fmt.Print("*")
    }
    fmt.Println()
    cetakBintang(n - 1)
}

func tampilkanPola(n int) {
    if n <= 0 {
        fmt.Println("Masukan harus lebih besar dari 0")
        return
    }
    cetakBintang(n)
}

func main() {
    var n int

    fmt.Print("Masukkan angka: ")
    fmt.Scan(&n)

    fmt.Printf("\nPola bintang untuk N = %d:\n", n)
    tampilkanPola(n)
}

```

### Screenshot Output

```

PS C:\Users\USER\OneDrive\Desktop\Alpro 2> go run "c:\Users\USER\OneDrive\Desktop\Alpro 2\Muhammad Ragiel Prastyo_2311102183_Modul15\Unguided\unguided2.go"
Masukkan angka: 5

Pola bintang untuk N = 5:
*****
****
***
**
*
PS C:\Users\USER\OneDrive\Desktop\Alpro 2>

```

### Penjelasan:

Pada program di atas, fungsi `cetakBintang` mencetak jumlah bintang sebanyak `n`, lalu mengurangi jumlah bintang secara rekursif hingga `n` mencapai 0. Fungsi `tampilkanPola` memastikan bahwa input `n` valid (lebih dari 0), dan kemudian memulai pencetakan bintang. Fungsi utama menampilkan pola bintang yang sesuai dengan bilangan `n` dan meminta input pengguna.



- 3) Buatlah program yang mengimplementasikan rekursif untuk menampilkan faktor bilangan dari suatu N, atau bilangan yang apa saja yang habis membagi N.

**Masukan** terdiri dari sebuah bilangan bulat positif N.

**Keluaran** terdiri dari barisan bilangan yang menjadi faktor dari N (terurut dari 1 hingga N ya).

**Contoh masukan dan keluaran:**

No	Masukan	Keluaran
1	5	1 5
2	12	1 2 3 4 6 12

#### *Source Code*

```
// MUHAMMAD RAGIEL PRASTYO
// 2311102183
package main
import (
    "fmt"
)

func findFactors(n int, current int) {
    if current > n {
        return
    }

    if n%current == 0 {
        fmt.Printf("%d ", current)
    }

    findFactors(n, current+1)
}

func main() {
    var n int

    fmt.Print("Masukkan bilangan positif: ")
    fmt.Scan(&n)

    if n <= 0 {
```

```
        fmt.Println("Mohon masukkan bilangan positif!")
        return
    }

    fmt.Printf("Faktor dari %d adalah: ", n)
    findFactors(n, 1)
    fmt.Println()
}
```

Screenshot Output

```
PS C:\Users\USER\OneDrive\Desktop\Alpro 2> go run "c:\Users\USER\OneDrive\Desktop\Alpro 2\Muhammad Ragiel Prastyo_2311102183_Modul5\Unguided\unguided3.go"
Masukkan bilangan positif: 12
Faktor dari 12 adalah: 1 2 3 4 6 12
PS C:\Users\USER\OneDrive\Desktop\Alpro 2> █
```

Penjelasan:

Dalam program di atas, fungsi rekursif findFactors digunakan untuk menemukan dan menampilkan semua faktor dari sebuah bilangan positif yang dimasukkan oleh pengguna. Fungsi ini memeriksa setiap bilangan dari 1 hingga bilangan tersebut apakah merupakan faktor (jika hasil bagi bilangan habis dibagi), dan jika ya, bilangan tersebut ditampilkan. Fungsi utama main program meminta input pengguna untuk memastikan bahwa bilangan yang dimasukkan oleh pengguna adalah positif, lalu memulai pencarian faktor dari 1.

- 4) Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu.

**Masukan** terdiri dari sebuah bilangan bulat positif N.

**Keluaran** terdiri dari barisan bilangan dari N hingga 1 dan kembali ke N.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	5 4 3 2 1 2 3 4 5
2	9	9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9

### Source Code

```
// MUHAMMAD RAGIEL PRASTYO
// 2311102183
package main
import (
    "fmt"
)

func cetakUrutan(n int, current int, turun bool) {
    fmt.Printf("%d ", current)

    if turun {
        if current > 1 {
            cetakUrutan(n, current-1, true)
        } else {
            cetakUrutan(n, current+1, false)
        }
    } else {
        if current < n {
            cetakUrutan(n, current+1, false)
        }
    }
}

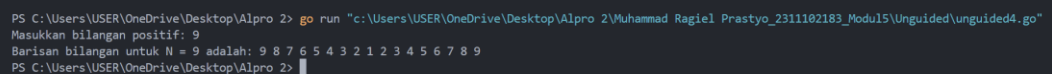
func main() {
    var n int

    fmt.Print("Masukkan bilangan positif: ")
    fmt.Scan(&n)

    if n <= 0 {
        fmt.Println("Mohon masukkan bilangan positif!")
        return
    }

    fmt.Printf("Barisan bilangan untuk N = %d adalah: ", n)
    cetakUrutan(n, n, true)
    fmt.Println()
}
```

### Screenshot Output



```
PS C:\Users\USER\OneDrive\Desktop\Alpro 2> go run "c:\Users\USER\OneDrive\Desktop\Alpro 2\Muhammad Ragiel Prastyo_2311102183_Modul5\Unguided\unguided4.go"
Masukkan bilangan positif: 9
Barisan bilangan untuk N = 9 adalah: 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS C:\Users\USER\OneDrive\Desktop\Alpro 2>
```

**Penjelasan:**

Program di atas mencetak bilangan dari n turun hingga 1, lalu kembali naik ke n menggunakan fungsi cetakUrutan. Saat 'turun' bernilai true, bilangan akan terus menurun sampai mencapai 1, kemudian naik lagi ketika 'turun' berubah menjadi false. Sebelum mencetak urutannya, program memeriksa bilangan positif yang dimasukkan pengguna dan memastikan bahwa angka itu benar.

- 5) Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil.

**Masukan** terdiri dari sebuah bilangan bulat positif N.

**Keluaran** terdiri dari barisan bilangan ganjil dari 1 hingga N.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	1 3 5
2	20	1 3 5 7 9 11 13 15 17 19

*Source Code*

```
// MUHAMMAD RAGIEL PRASTYO
// 2311102183
package main
import (
    "fmt"
)

func cetakBilanganGanjil(current int, n int) {
    if current > n {
        return
    }

    fmt.Printf("%d ", current)

    cetakBilanganGanjil(current + 2, n)
}

func main() {
    var n int

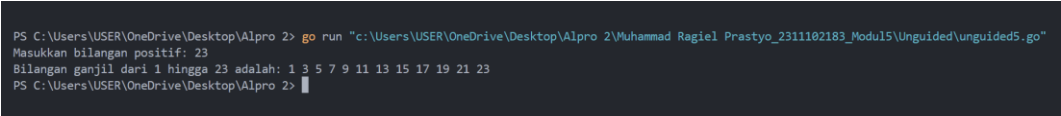
    fmt.Print("Masukkan bilangan positif: ")
```

```
fmt.Scan(&n)

if n <= 0 {
    fmt.Println("Mohon masukkan bilangan positif!")
    return
}

fmt.Printf("Bilangan ganjil dari 1 hingga %d adalah: ", n)
cetakBilanganGanjil(1, n)
fmt.Println()
}
```

Screenshot Output



Penjelasan:

Program di atas menggunakan fungsi cetakBilanganGanjil untuk mencetak bilangan ganjil secara rekursif mulai dari 1 hingga n yang dimasukkan oleh pengguna. Proses berhenti jika arus melebihi n. Fungsi utama main menerima input pengguna, memastikan bilangan positif dan mengaktifkan fungsi untuk mencetak bilangan ganjil.

- 6) Buatlah program yang mengimplementasikan rekursif untuk mencari hasil pangkat dari dua buah bilangan.

Masukan terdiri dari bilangan bulat x dan y.

Keluaran terdiri dari hasil x dipangkatkan y.

Catatan: diperbolehkan menggunakan asterik “\*”, tapi dilarang menggunakan import “math”.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	2 2	4
2	5 3	125

### Source Code

```
// MUHAMMAD RAGIEL PRASTYO
// 2311102183
package main
import (
    "fmt"
)

func pangkat(basis int, eksponen int) int {
    if eksponen == 0 {
        return 1
    }

    if eksponen == 1 {
        return basis
    }

    return basis * pangkat(basis, eksponen-1)
}

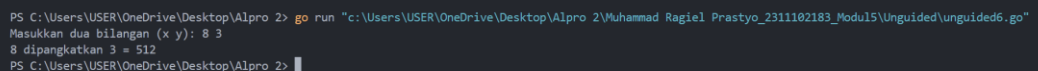
func main() {
    var x, y int

    fmt.Print("Masukkan dua bilangan (x y): ")
    fmt.Scan(&x, &y)

    if y < 0 {
        fmt.Println("Pangkat tidak boleh negatif!")
        return
    }

    hasil := pangkat(x, y)
    fmt.Printf("%d dipangkatkan %d = %d\n", x, y, hasil)
}
```

### Screenshot Output



```
PS C:\Users\USER\OneDrive\Desktop\Alpro 2> go run "c:\Users\USER\OneDrive\Desktop\Alpro 2\Muhammad Ragiel Prastyo_2311102183_Modul5\Unguided\unguided6.go"
Masukkan dua bilangan (x y): 8 3
8 dipangkatkan 3 = 512
PS C:\Users\USER\OneDrive\Desktop\Alpro 2> █
```

### Penjelasan:

Program di atas menggunakan rekursif untuk menghitung perpangkatan. Fungsi pangkat mengalikan base dengan hasil pemanggilan fungsi yang sama sambil mengurangi nilai exponent hingga nilai 0(kondisi dasar). Fungsi

mengembalikan 1 jika exponennya adalah 0. Program dalam fungsi main meminta bilangan pokok  $x$  dan pangkat  $y$ , dan kemudian menggunakan fungsi pangkat untuk menghitung hasilnya. Program memberi pesan kesalahan jika pangkat negatif. Perhitungan selesai dan hasilnya akan ditampilkan.