

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

MODUL V

REKURSIF



Oleh:

MUHAMAD IHSAN

2311102077

IF - 11 – 02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

I. DASAR TEORI

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 (Telkom University by one) secara terus menerus tanpa henti.

```
1 package main
2 import "fmt"
3 func main(){
4     cetak(5)
5 }
6 func cetak(x int){
7     fmt.Println(x)
8     cetak(x+1)
9 }
```

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.

```
1 procedure cetak(in x:integer)
2   algoritma
3     if x == 10 then
4       output(x)
5     else
6       output(x)
7       cetak(x+1)
8     endif
9 endprocedure
```

Apabila diperhatikan pada baris ke-3 di Program di atas, kita telah menambahkan base-case seperti penjelasan sebelumnya. Selanjutnya pada bagian aksi dari else di baris ke-6 dan ke-7 kita namakan recursive-case atau kasus pemanggilan dirinya sendiri tersebut terjadi. Kondisi dari recursive-case ini adalah negasi dari kondisi base-case atau ketika nilai $x \neq 10$.

- 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

- Contoh Program dengan menggunakan Rekursif

- a) Membuat baris bilangan dari n hingga 1

Base-case: bilangan == 1

```
1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Scan(&n)
6     baris(n)
7 }
8
9 func baris(bilangan int){
10    if bilangan == 1 {
11        fmt.Println(1)
12    }else{
13        fmt.Println(bilangan)
14        baris(bilangan - 1)
15    }
16 }
```

- b) Menghitung hasil penjumlahan 1 hingga n

Base-case: n == 1

```
1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Scan(&n)
6     fmt.Println(penjumlahan(n))
7 }
8
9 func penjumlahan(n int) int {
10    if n == 1 {
11        return 1
12    }else{
13        return n + penjumlahan(n-1)
14    }
15 }
```

- c) Mencari dua pangkat n atau 2pangka

Base-case: n == 0

```

1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Scan(&n)
6     fmt.Println(pangkat(n))
7 }
8
9 func pangkat(n int) int {
10     if n == 0 {
11         return 1
12     }else{
13         return 2 * pangkat(n-1)
14     }
15 }

```

d) Mencari nilai faktorial atau $n!$

Base-case: $n == 0$ atau $n == 1$

```

1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Scan(&n)
6     fmt.Println(faktorial(n))
7 }
8
9 func faktorial(n int) int {
10     if n == 0 || n == 1 {
11         return 1
12     }else{
13         return n * faktorial(n-1)
14     }
15 }

```

II. GUIDED

Guided 1

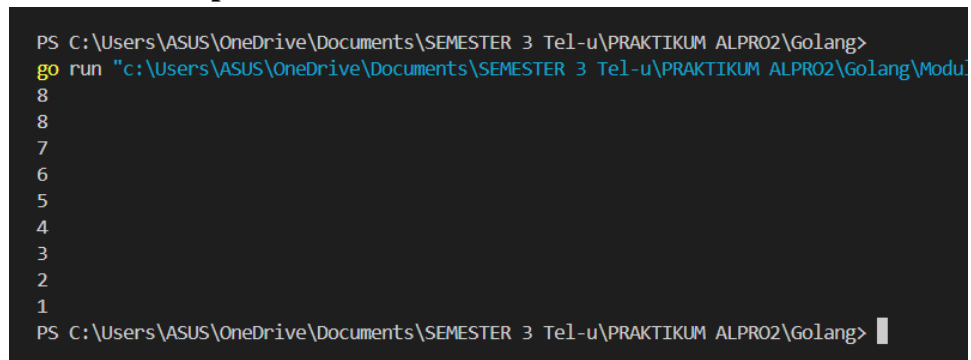
```
package main

import "fmt"

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

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

Screenshot output



```
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang\Modu
8
8
7
6
5
4
3
2
1
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
```

Deskripsi program

Program ini untuk mencetak bilangan secara menurun dari input pengguna hingga angka 1 menggunakan fungsi rekursif. Program dimulai dengan mendeklarasikan variabel `n` sebagai integer, di mana nilainya diambil dari input pengguna melalui `fmt.Scan(&n)`. Kemudian, fungsi `baris(n)` dipanggil untuk mencetak bilangan. Fungsi `baris` merupakan fungsi rekursif yang mencetak nilai bilangan dan terus memanggil dirinya sendiri dengan parameter `bilangan - 1` hingga mencapai angka 1. Jika bilangan sama dengan 1, program akan mencetak 1 dan menghentikan rekursi. Dengan cara ini, program mencetak urutan angka menurun dari `n` hingga 1.

Guided 2

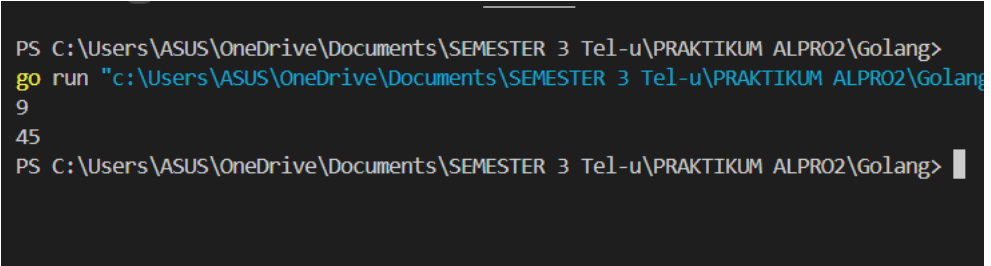
```
package main

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\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang
9
45
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
```

Deskripsi program

Program di atas program yang menghitung penjumlahan deret dari 1 hingga angka n yang dimasukkan oleh pengguna. Program menggunakan fungsi rekursif penjumlahan, di mana jika nilai n adalah 1, maka fungsi mengembalikan nilai 1 sebagai kondisi dasar. Jika n lebih besar dari 1, fungsi akan menambahkan nilai n dengan hasil pemanggilan fungsi penjumlahan untuk n-1, sehingga terjadi proses rekursi. Pada fungsi main, input dari pengguna diterima menggunakan `fmt.Scan`, dan hasil penjumlahan ditampilkan ke layar menggunakan `fmt.Println`. Program ini berguna untuk menghitung total penjumlahan deret bilangan secara rekursif.

III. UNGUIDED

Soal Latihan Modul 5

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.

n	0	1	2	3	4	5	6	7	8	9	10
S_n	0	1	1	2	3	5	8	13	21	34	55

Source code

```
package main

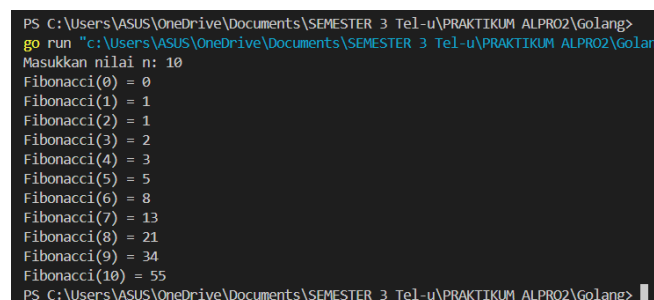
import "fmt"

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.Scanln(&n)

    for i := 0; i <= n; i++ {
        fmt.Printf("Fibonacci(%d) = %d\n", i, fibonacci(i))
    }
}
```

Screenshot output



```
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang\
Masukkan nilai n: 10
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\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
```

Deskripsi

Program ini untuk menghitung dan mencetak deret Fibonacci hingga nilai ke-n menggunakan fungsi rekursif. Fungsi fibonacci(n int) int menghitung nilai Fibonacci ke-n, mengembalikan n jika n kurang dari atau sama dengan 1, dan menjumlahkan hasil panggilan rekursif untuk n-1 dan n-2 jika lebih dari 1. Dalam fungsi main(), program meminta pengguna memasukkan nilai n, lalu mencetak deret Fibonacci dari 0 hingga n menggunakan loop for. Sebagai contoh, jika pengguna memasukkan n = 5, program akan mencetak angka Fibonacci: 0, 1, 1, 2, 3, dan 5.

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

```
package main

import "fmt"

func bintang(baris, barisSekarang, b int) {
    if barisSekarang > baris {
        return
    }
    if b < barisSekarang {
        fmt.Print("*")
    }
    bintang(baris, barisSekarang+1, b)
}
```



```

        bintang(baris, barisSekarang, b+1)
    } else {
        fmt.Println()
        bintang(baris, barisSekarang+1, 0)
    }
}

func main() {
    var baris int
    fmt.Scan(&baris)
    bintang(baris, 1, 0)
}

```

Screenshot output

```

PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang> go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang\
5
*
**
***
****
*****
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang> go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang\
1
*
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>

```

Deskripsi

Program ini mencetak pola bintang berbentuk segitiga sesuai jumlah baris yang diinputkan pengguna. Fungsi `bintang(baris, barisSekarang, b int)` adalah fungsi rekursif yang mencetak bintang, di mana `baris` adalah total baris, `barisSekarang` menunjukkan baris yang sedang diproses, dan `b` adalah jumlah bintang yang dicetak dalam baris itu. Jika `barisSekarang` lebih besar dari `baris`, fungsi akan berhenti; jika tidak, program mencetak bintang hingga jumlahnya sesuai dengan nomor baris dan melanjutkan ke baris berikutnya. Di dalam fungsi `main()`, pengguna diminta memasukkan jumlah baris, dan fungsi `bintang` dipanggil. Contohnya, jika pengguna memasukkan 3, program akan mencetak segitiga bintang dengan 3 baris: `*`, `**`, dan `***`.

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

```
package main

import "fmt"

func faktor(n int) {
    for i:=1; i <= n; i++ {
        if n % i == 0 {
            fmt.Print(i," ")
        }
    }
}

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

Screenshoot Output

```
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Te1-u\PRAKTIKUM ALPRO2\Gol
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Te1-u\PRAKTIKUM ALPRO
5
1 5
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Te1-u\PRAKTIKUM ALPRO2\Gol
```

Deskripsi

Program ini mencetak faktor dari bilangan bulat positif yang diinputkan pengguna. Dimulai dengan mendeklarasikan package main dan mengimpor paket fmt, program memiliki fungsi faktor(n int) yang menerima parameter n. Di dalam fungsi ini, loop iterasi dari 1 hingga n memeriksa setiap angka untuk melihat apakah itu faktor dari n dengan menggunakan kondisi $n \% i == 0$. Jika ya, angka tersebut dicetak. Dalam fungsi main(), program meminta pengguna memasukkan nilai n, lalu memanggil fungsi faktor(n) untuk mencetak semua faktornya. Sebagai contoh, jika pengguna memasukkan 12, outputnya akan menjadi 1 2 3 4 6 12.

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

```
package main

import "fmt"

func loop(n int) {
    for i:=n; i >= 1; i-- {
        fmt.Print(i, " ")
        if i == 1 {
            for i = 2; i <= n; i++ {
                fmt.Print(i, " ")
            }
            break
        }
    }
}

func main() {
    var n int
```

```

    fmt.Scan(&n)
    loop(n)
}

```

Screenshot output

```

PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\Pr
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\Pr
5
5 4 3 2 1 2 3 4 5
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\Pr

```

Deskripsi

Program ini mencetak urutan angka dari n hingga 1, lalu melanjutkan dari 2 hingga n. Program dimulai dengan mendeklarasikan package main dan mengimpor paket fmt. Fungsi loop(n int) menerima parameter n, di mana loop pertama mencetak angka dari n ke 1. Setelah mencapai 1, program beralih ke loop kedua yang mencetak angka dari 2 hingga n. Dalam fungsi main(), pengguna diminta untuk memasukkan nilai n, dan fungsi loop(n) dipanggil untuk mengeksekusi logika tersebut. Misalnya, jika pengguna memasukkan 5, outputnya akan menjadi 5 4 3 2 1 2 3 4 5.

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

```

package main

import "fmt"

func ganjil(a int) {
    for u:=0; u <= a; u++ {

```

```

    if u % 2 == 1 {
        fmt.Print(u, " ")
    }
}
}
func main() {
    var a int
    fmt.Scan(&a)
    ganjil(a)
}

```

Screenshot output

```

5
1 3 5
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang> go
20
1 3 5 7 9 11 13 15 17 19
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>

```

Deskripsi

Program ini mencetak semua bilangan ganjil dari 1 hingga angka yang diinputkan pengguna. Program dimulai dengan mendeklarasikan package main dan mengimpor paket fmt. Fungsi ganjil(a int) menerima parameter a dan menggunakan loop untuk iterasi dari 0 hingga a. Di dalam loop, angka diperiksa apakah ganjil dengan kondisi $u \% 2 == 1$, dan jika ya, angka tersebut dicetak. Dalam fungsi main(), pengguna diminta memasukkan nilai a, dan fungsi ganjil(a) dipanggil untuk mencetak bilangan ganjil. Sebagai contoh, jika pengguna memasukkan 5, outputnya akan menjadi 1 3 5.

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

```
package main

import "fmt"

func pangkat(n,m int) {
    val :=n
    for i:=1; i <m; i++ {
        val = val * n
    }
    fmt.Print(val)
}
func main() {
    var n,m int
    fmt.Scan(&n,&m)
    pangkat(n,m)
}
```

Screenshoot output

```
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golan
2 2
4
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang> go
"
5 3
125
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang> █
```

Deskripsi

Program ini menghitung n^m pangkat m . Pengguna memasukkan nilai n (basis) dan m (eksponen). Fungsi `pangkat(n, m int)` mengalikan n sebanyak $m-1$ kali dan mencetak hasilnya. Misalnya, jika pengguna memasukkan 2 dan 3, program akan mencetak 8, yang merupakan 2^3 .