

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V
REKURSIF**



Disusun Oleh :

Hamzah Ziyad Ibadurrohman / 2311102254

IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

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 56789...dst tanpa henti. Artinya setiap pemanggilan subprogram cetak() nilai x akan selalu bertambah 1 (*Increment 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 }
```

```
D:\DEV\DEMO>go build contoh.go
```

```
D:\DEV\DEMO>contoh.exe
```

```
5
6
7
8
9
10
11
12
13
...
```

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 != 10`.

```
1 package main
2 import "fmt"
3 func main(){
4     cetak(5)
5 }
6 func cetak(x int){
7     if x == 10 {
8         fmt.Println(x)
9     }else{
10        fmt.Println(x)
11        cetak(x+1)
12    }
13 }
```

```
D:\DEV\DEMO>go build contoh.go
D:\DEV\DEMO>contoh.exe
5
6
7
8
9
10
```

Apabila program di atas ini dijalankan maka akan tampil angka 5678910. Terlihat bahwa proses rekursif berhasil dihentikan Ketika `x == 10`.

II. GUIDED

1. Soal Studi Case

Buatlah sebuah program yang membuat baris bilangan dari n hingga 1.

base case : bilangan == 1

Sourcecode

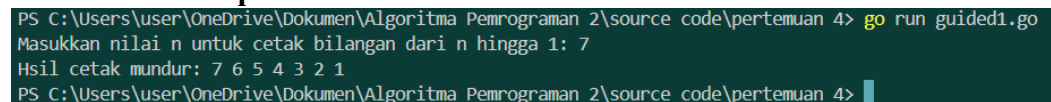
```
package main

import "fmt"

func cetakMundur(n int) {
    if n == 1 {
        fmt.Println(n)
        return
    }
    fmt.Print(n, " ")
    cetakMundur(n - 1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk cetak bilangan
dari n hingga 1: ")
    fmt.Scanln(&n)
    fmt.Print("Hsil cetak mundur: ")
    cetakMundur(n)
}
```

Screenshoot Output



```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> go run guided1.go
Masukkan nilai n untuk cetak bilangan dari n hingga 1: 7
Hsil cetak mundur: 7 6 5 4 3 2 1
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4>
```

Deskripsi Program

Ini adalah program yang membuat baris bilangan dari n hingga 1. Program ini memakai rekursif untuk membuat baris bilangan. Pada fungsi cetakmundur program menggunakan if statement jika $n == 1$ maka akan mencetak bilangan n. Kemudian diluar if statement program mencetak n. Kemudian memanggil dirinya sendiri sehingga terjadilah looping hingga if statement tersebut true. Pada fungsi utama ketika sudah menginput bilangan n maka program akan memanggil fungsi cetakmundur(n) dan hasil itu akan dicetak ke layar.

2. Soal Studi Case

Buat program yang Menghitung hasil penjumlahan 1 hingga n

Base case $n == 0$

Sourcecode

```
package main

import "fmt"

func jumlahreKursif(n int) int {
    if n == 1 {
        return 1
    }

    return n + jumlahreKursif(n-1)
}

func main() {
    var n int

    fmt.Print("Masukkan nilai n untuk penjumlahan 1 hingga n: ")

    fmt.Scanln(&n)

    fmt.Println("Hasil penjumlahan: ", jumlahreKursif(n))
}
```

Screenshot Output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> go run guided2.go
Masukkan nilai n untuk penjumlahan 1 hingga n: 7
Hasil penjumlahan: 28
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> █
```

Deskripsi Program

Program ini dibuat untuk membuat hitung hasil penjumlahan 1 hingga n dengan base-case: $n == 1$. Sama seperti sebelumnya program ini memakai rekursif untuk membuat hasil penjumlahannya. Pada fungsi jumlahrekursif jika $n == 1$ maka akan mengembalikan nilai yaitu 1. Dalam fungsi itu juga ada perintah mengembalikan $n +$ memanggil fungsi itu sendiri lagi dengan parameter $(n - 1)$. Ketika di fungsi main nilai n itu sudah diinput. Maka program akan memanggil fungsi jumlahrekursif dan menampilkannya ke layar.

3. Soal Studi Case

Mencari dua pangkat n atau 2^n .

Base case $== 0$

Sourcecode

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("Masukkan nilai n untuk mencari 2 pangkat
n: ")

    fmt.Scan(&n)

    fmt.Println("Hasil pangkat", n, ":", pangkat(n))
}
```

```
func pangkat(n int) int {  
    if n == 0 {  
        return 1  
    } else {  
        return 2 * pangkat(n-1)  
    }  
}
```

Screenshot Output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> go run guided3.go  
Masukkan nilai n untuk mencari 2 pangkat n: 6  
Hasil pangkat 6 : 64  
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4>
```

Deskripsi Program

Program ini dirancang untuk mencari hasil dari 2 pangkat atau 2^n . Dan seperti sebelumnya, program ini menggunakan rekursif. Pada fungsi pangkat, program memakai if statement untuk menjalankan base case apabila kondisi base case terpenuhi. Base case di fungsi ini jika $n == 0$ maka akan mengembalikan 1. Apabila base case tidak terpenuhi, maka kondisi selain itu akan jalan yaitu akan mengembalikan 2 kali fungsi itu sendiri dengan parameternya $n - 1$. Itu akan terus mengulang dirinya sendiri secara terus menerus hingga kondisi base case terpenuhi. Di fungsi utama, setelah input n program akan mencetak hasil fungsi pangkat ke layer.

4. Soal Studi case

Mencari nilai faktorial atau $n!$ Base case $n == 0$ atau $n == 1$

Sourcecode

```
package main  
  
import "fmt"
```

```

func main() {

    var n int

    fmt.Print("Masukkan nilai n untuk mencari faktorial n!: ")

    fmt.Scan(&n)

    fmt.Println("Hasil faktorial dari", n, ":", faktorial(n))

}

func faktorial(n int) int {

    if n == 0 || n == 1 {

        return 1

    }

    return n * faktorial(n-1)

}

```

Screenshot output

```

PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> go run guided4.go
Masukkan nilai n untuk mencari faktorial n!: 5
Hasil faktorial dari 5 : 120
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4>

```

Deskripsi Program

Program ini dibuat untuk menghitung faktorial dari bilangan n yang dimasukkan oleh pengguna menggunakan fungsi rekursif. Di dalam fungsi main, program meminta pengguna memasukkan nilai n, lalu memanggil fungsi faktorial untuk menghitung faktorial dari nilai tersebut dan menampilkan hasilnya. Fungsi faktorial menggunakan rekursif. jika n bernilai 0 atau 1, fungsi akan mengembalikan 1 sebagai hasil (base case). Jika n lebih besar dari 1, fungsi akan mengalikan n dengan faktorial(n-1), dan ini berlanjut hingga nilai n mencapai 1.

III. UNGUIDED

1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

Soal Studi Case

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

Sourcecode

```
package main

import "fmt"

func fibonacci(n int) int {
    var hasil int
    if n == 0 {
        return 0
    } else if n == 1 {
        return 1
    } else {
        hasil = fibonacci(n-1) + fibonacci(n-2)
        return hasil
    }
}

func main() {
    var n, hsl int
    fmt.Print("masukkan nilai suku nya: ")
    fmt.Scan(&n)
    hsl = fibonacci(n)
    fmt.Println("Hasilnya adalah: ", hsl)
}
```

Screenshoot Output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> go run unguided1.go
masukkan nilai suku nya: 8
Hasilnya adalah: 21
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4>
```

Deskripsi Program

Program ini dibuat untuk menghitung suku n dari deret fibonacci dengan menggunakan rekursif. Pada fungsi `fibonacci(n int) int` ini adalah rekursif dari program ini. Base case ini adalah $n == 0$ dan $n == 1$. Jika kondisi yang pertama terpenuhi maka akan mengembalikan 0. Dan jika kondisi kedua terpenuhi maka akan mengembalikan 1. N disini adalah nilai suku. Lalu selain dari kondisi kedua tersebut, ada kondisi yang berbeda, jika $n > 1$ maka fungsi akan mengalikan n dengan faktorial($n-1$), dan proses ini berlanjut hingga nilai n mencapai 1. Di fungsi utama program meminta untuk input n . Ketika sudah input n maka program akan mencetak hasil fungsi ke layar.

2. Soal Studi Case

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

Sourcecode

```
package main

import "fmt"

func main() {
    var c int
    var b string

    fmt.Print("Masukkan jumlah baris pola: ")
    fmt.Scan(&c)

    pola_bintang(1, c, b)
}

func pola_bintang(i, c int, b string) {
```

```

        if i <= c {

            b = "*" + b

            fmt.Println(b)

            pola_bintang(i+1, c, b)

        }

    }
}

```

Screenshot output

```

PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> go run unguided2.go
Masukkan jumlah baris pola: 9
*
**
***
****
*****
******
*******
*****
****
***
**
*
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4>

```

Deskripsi Output

Ini adalah Program yang dibuat untuk mencetak pola bintang dengan menggunakan fungsi rekursif. Fungsi rekursif tersebut adalah pola_bintang. Base case dari ini adalah $i \leq c$. Selama kondisi ini terpenuhi maka program akan mencetak "*" baris sesuai nilai yang diinput hingga kondisi itu false. Didalam fungsi itu juga terdapat perintah untuk mencetak ke dalam layar pada setiap fungsi itu terpanggil. Lalu pada fungsi main program akan meminta untuk input n. Setelah input n program akan memanggil fungsi pola_bintang.

3. Soal Studi Case

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

Sourcecode

```
package main

import "fmt"

func faktor_bilang(n, i int) {

    if i <= n {

        if n%i == 0 {

            fmt.Print(i, " ")

        }

        faktor_bilang(n, i+1)

    }

}

func main() {

    var n int

    fmt.Print("Masukkan bilangan: ")

    fmt.Scan(&n)

    faktor_bilang(n, 1)

}
```

Screenshot Output

```
Masukkan bilangan: 9
1 3 9
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4>
```

Deskripsi Program

Ini adalah program yang dibuat untuk menghitung angka apa saja yang dapat dibagi habis dari n dengan menggunakan fungsi rekursif. Program ini menggunakan fungsi rekursif. Fungsi rekursif itu sendiri adalah faktor_bilang. Base case dari ini adalah jika $i \leq n$. Selama itu true maka program menjalankan if statement lagi yaitu jika n dibagi $i == 0$. Jika kondisi ini terpenuhi maka program akan menampilkan i yang bisa membagi habis n sekaligus dicetak ke layar. Diluar if statement yang dalam terdapat fungsi itu sendiri dengan parameter tambahan n , i ditambah 1. Lalu ke fungsi utama, program akan meminta nilai n untuk diinput. Setelah diinput nilai n program akan memanggil prosedur faktor_bilang.

4. Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu.

Sourcecode

```
package main

import "fmt"

func tampil(n int) {
    if n > 0 {
        fmt.Print(n, " ")
        tampil(n - 1)
    }
}

func main() {
    var n int
    fmt.Print("Masukkan bilangannya: ")
    fmt.Scan(&n)
    tampil(n)
    fmt.Println()
}
```

Screenshot Output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> go run unguided4.go
Masukkan bilangannya: 8
8 7 6 5 4 3 2 1
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4>
```

Deskripsi Program

Ini adalah program yang dibuat untuk menghitung baris n sampai 1 dengan menggunakan fungsi rekursif. Fungsi rekursi di program ini adalah func tampil. Dalam prosedur itu base case nya adalah $n > 0$. Selama kondisi ini terpenuhi prosedur ini akan mencetak nilai n dan memanggil fungsi dirinya dengan parameter pengurangan yaitu $n - 1$. Jika kondisinya sudah false maka prosedur tersebut berhenti. Lalu di fungsi main program meminta untuk input nilai n. Sesudah input program akan memanggil prosedur tampil dengan parameter n.

5. Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil.

Sourcecode

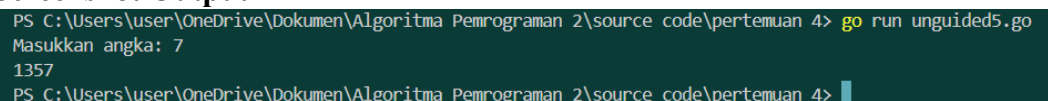
```
package main

import "fmt"

func bilanganganjil(n int) {
    if n == 1 {
        fmt.Print(1)
    } else if n%2 == 0 {
        bilanganganjil(n - 1)
    } else {
        bilanganganjil(n - 2)
        fmt.Print(n)
    }
}

func main() {
    var n int
    fmt.Print("Masukkan angka: ")
    fmt.Scan(&n)
    bilanganganjil(n)
}
```

Screenshot Output



```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> go run unguided5.go
Masukkan angka: 7
1357
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4>
```

Deskripsi Output

Program ini dibuat untuk menghitung baris yang ganjil dari 1 hingga nilai n dengan menggunakan fungsi rekursif. Fungsi rekursif di program ini adalah func bilanganganjil. Base case ini adalah $n == 1$. Jika kondisi ini true maka akan mengakhiri proses rekursif. Kondisi

berikutnya adalah $n\%2 == 0$ (genap) maka dia akan memanggil dirinya sendiri dengan parameter n dikurangi 1. Lalu kondisi berikutnya n adalah ganjil & lebih besar dari 1 maka dia akan memanggil dirinya lagi dengan parameter $n-2$. Setelah rekursif selesai program mencetak n sehingga bilangannya dicetak dari urutan terkecil hingga terbesar. Selanjutnya di fungsi main program meminta input n kepada pengguna. Setelah input program akan memanggil prosedur bilanganganjil dengan parameter n .

6. Soal Studi Case

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

Sourcecode

```
package main

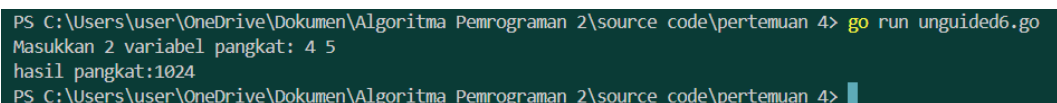
import "fmt"

func pangkatkan(x, y int) int {
    if y == 0 {
        return 1
    } else {
        return x * pangkatkan(x, y-1)
    }
}

func main() {
    var x, y, hasilnya int
    fmt.Print("Masukkan 2 variabel pangkat: ")

    fmt.Scan(&x, &y)
    hasilnya = pangkatkan(x, y)
    fmt.Print("hasil pangkat:", hasilnya)
}
```

Screenshot Output



```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4> go run unguided6.go
Masukkan 2 variabel pangkat: 4 5
hasil pangkat:1024
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 4>
```

Deskripsi Program

Program ini dibuat untuk mencari hasil pangkat dari dua buah bilangan dengan menggunakan fungsi rekursif. Fungsi rekursif pada program ini adalah func pangkatkan. Base case dari rekursif ini adalah $y == 0$. Jika ini true maka return 1, jika false maka akan

mengembalikan $x * \text{fungsi dirinya sendiri dengan parameter}(x, y - 1)$. Ini akan berjalan terus menerus hingga kondisi true. Lalu di fungsi main program akan meminta input 2 bilangan. Sesudah input program akan memanggil pangkatkan dengan cara memanggil hasil yang deklarasinya itu : $\text{hasil} = \text{pangkatkan}(x, y)$.