

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

**MODUL 5
REKURSIF**



Oleh:

Alberta Aurora Adiswari

2311102220

S1-IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Rekursif secara sederhana dapat diartikan sebagai cara menyelesaikan suatu masalah dengan 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**.

```
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 biasanya ditambahkan struktur control 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.

Catatan:

- Teknik rekursif ini merupakan salah satu alternatif untuk mengganti struktur control 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.

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. Guided 1

Source code

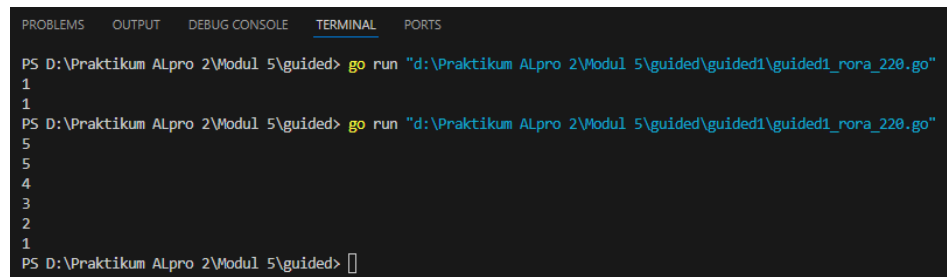
```
package main

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 { // Base case: Jika bilangan
sama dengan 1
        fmt.Println(1) // Cetak angka 1
    } else { // Jika bilangan lebih besar dari 1
        fmt.Println(bilangan) // cetak bilangan saat
ini
        baris(bilangan - 1) // Panggil fungsi
'baris' dengan bilangan
    }
}
```

Screenshoot program



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Praktikum ALpro 2\Modul 5\guided> go run "d:\Praktikum ALpro 2\Modul 5\guided\guided1\guided1_rora_220.go"
1
1
PS D:\Praktikum ALpro 2\Modul 5\guided> go run "d:\Praktikum ALpro 2\Modul 5\guided\guided1\guided1_rora_220.go"
5
5
4
3
2
1
PS D:\Praktikum ALpro 2\Modul 5\guided> []
```

Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go untuk mencetak angka secara menurun dari n hingga 1 menggunakan metode rekursif. Pengguna diminta menginputkan bilangan yang kemudian program mengeksekusi dengan memanggil fungsi `baris(bilangan)` untuk mulai mencetak. Fungsi `baris` memiliki logika rekursif yang mencetak bilangan saat ini dan kemudian memanggil dirinya sendiri dengan nilai bilangan yang dikurangi satu. Jika bilangan mencapai 1, fungsi mencetak angka 1 sebagai kondisi dasar (base case) dan menghentikan rekursi.

2. Guided 2

Source code

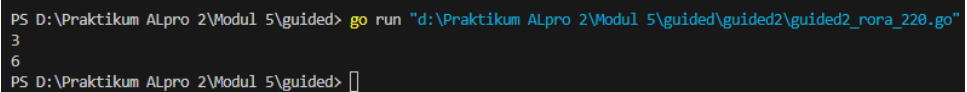
```
package main

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)
    }
}
```

Screenshoot program



```
PS D:\Praktikum ALpro 2\Modul 5\guided> go run "d:\Praktikum ALpro 2\Modul 5\guided\guided2\guided2_rora_220.go"
3
6
PS D:\Praktikum ALpro 2\Modul 5\guided> 
```

Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go untuk menghitung jumlah semua bilangan bulat positif dari 1 hingga n menggunakan metode rekursif. Pengguna diminta untuk memasukkan bilangan kemudian program memanggil fungsi penjumlahan dan bekerja secara rekursif, di mana jika n sama dengan 1 maka fungsi mengembalikan nilai 1. Jika n lebih besar dari 1 fungsi mengembalikan n ditambah dengan hasil pemanggilan dirinya sendiri dengan n yang dikurangi satu.

III. UNGUIDED

1. Unguided 1

Source code

```
package main

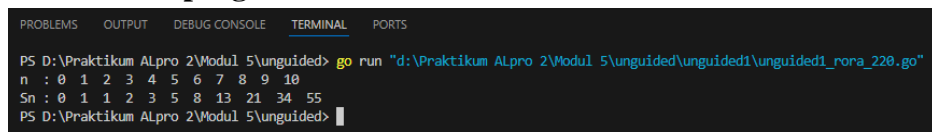
import "fmt"

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

func main() {
    fmt.Print("n : ")
    for i := 0; i <= 10; i++ {
        fmt.Printf("%d ", i)
    }
    fmt.Println()

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

Screenshoot program



```
PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided1\unguided1_rora_228.go"
n : 0 1 2 3 4 5 6 7 8 9 10
Sn : 0 1 1 2 3 5 8 13 21 34 55
PS D:\Praktikum ALpro 2\Modul 5\unguided> |
```

Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go untuk menghitung dan mencetak deret Fibonacci untuk nilai n dari 0 hingga 10. Program menggunakan metode rekursif untuk menghitung angka Fibonacci, dan fungsi main() yang bertanggung jawab untuk menampilkan hasil. Dalam program pertama-tama mencetak header “n:” diikuti oleh angka dari 0 hingga 10. Kemudian program mencetak header “Sn:” dan memanggil fungsi fibonacci untuk setiap angka dari 0 hingga 10 dan menampilkan hasil outputannya dalam deret.

2. Unguided 2

Source code

```
package main
```

```

import (
    "fmt"
)

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

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

func main() {
    var N int

    fmt.Print("Masukkan nilai N: ")
    fmt.Scan(&N)

    cetakPola(N)
}

```

Screenshoot program

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided2\unguided2_rora_220.go"
Masukkan nilai N: 5
*
**
***
****
*****
PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided2\unguided2_rora_220.go"
Masukkan nilai N: 1
*
PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided2\unguided2_rora_220.go"
Masukkan nilai N: 3
*
**
***
PS D:\Praktikum ALpro 2\Modul 5\unguided> 

```

Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go untuk mencetak pola bintang berbentuk segitiga, dengan jumlah baris yang telah diinputkan oleh pengguna. Terdapat dua fungsi rekursif yaitu cetakBintang(n int) yang mencetak Bintang sebanyak n kali, dan cetakPola(n int) yang mengatur pola yang akan dicetak.

3. Unguided 3

Source code

```
package main

import (
    "fmt"
)

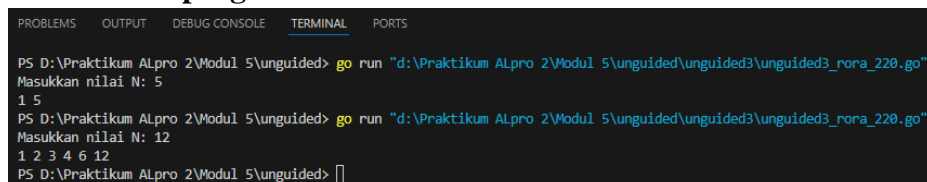
func cetakFaktor(N, i int) {
    if i > N {
        return
    }
    if N%i == 0 {
        fmt.Print(i, " ")
    }
    cetakFaktor(N, i+1)
}

func main() {
    var N int

    fmt.Print("Masukkan nilai N: ")
    fmt.Scan(&N)

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

Screenshoot program



```
PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided3\unguided3_rora_220.go"
Masukkan nilai N: 5
1 5
PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided3\unguided3_rora_220.go"
Masukkan nilai N: 12
1 2 3 4 6 12
PS D:\Praktikum ALpro 2\Modul 5\unguided> 
```

Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go untuk mencetak semua factor dari bilangan yang diinputkan. Program menggunakan fungsi rekursif cetakFaktor(N, i) yang menerima dua parameter: N, yang merupakan bilangan yang ingin dicari faktornya, dan I yang merupakan bilangan yang sedang diuji apakah merupakan factor dari N. Pengguna diminta untuk menginputkan nilai N kemudian program mengeksekusi dengan memanggil cetakFaktor(N, 1) untuk memulai pencarian factor.

4. Unguided 4

Source code

```
package main
```



```
import "fmt"

func printPattern(N int) {
    if N == 0 {
        return
    }
    fmt.Printf("%d ", N)
    if N > 1 {
        printPattern(N - 1)
        fmt.Printf("%d ", N)
    }
}

func main() {
    var N int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)

    printPattern(N)
}
```

Screenshot program

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided4\unguided4_rona_228.go"
Masukkan bilangan bulat positif N: 5
5 4 3 2 1 2 3 4 5
PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided4\unguided4_rona_228.go"
Masukkan bilangan bulat positif N: 9
9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS D:\Praktikum ALpro 2\Modul 5\unguided> 
```

Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go untuk mencetak pola angka yang menurun dari N hingga 1, kemudian Kembali naik dari 1 ke N, dengan menggunakan fungsi rekursif printPattern(N). Pengguna diminta untuk menginputkan bilangan bulat positif N. Fungsi printPattern akan mencetak nilai N saat ini, apabila N lebih besar dari 1, fungsi akan memanggil dirinya sendiri dengan nilai N-1 dan mencetak angka yang lebih kecil hingga mencapai angka 1. Setelah mencetak angka 1 fungsi melanjutkan mencetak angka yang sudah diproses secara menaik.

5. Unguided 5

Source code

```
package main

import "fmt"

func printOdd(N int) {
    if N <= 0 {
```

```

        return
    }
    if N%2 == 0 {
        printOdd(N - 1)
    } else {
        printOdd(N - 2)
        fmt.Printf("%d ", N)
    }
}

func main() {
    var N int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)

    printOdd(N)
}

```

Screenshoot program

```

PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided5_rora_220.go"
Masukkan bilangan bulat positif N: 5
1 3 5
PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided5_rora_220.go"
Masukkan bilangan bulat positif N: 20
1 3 5 7 9 11 13 15 17 19
PS D:\Praktikum ALpro 2\Modul 5\unguided> 

```

Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go untuk mencetak bilangan ganjil dari 1 hingga N, yang dimana N merupakan bilangan bulat positif yang diinputkan oleh pengguna. Program menggunakan fungsi rekursif printOdd(N) yang pertama mengecek apakah bilangan N adalah genap atau ganjil. Apabila N genap, fungsi memanggil dirinya sendiri dengan N-1 agar dapat menemukan bilangan ganjil terdekat dibawahnya. Namun jika N ganjil, fungsi memanggil dirinya sendiri dengan N-2 untuk mencetak bilangan ganjil yang lebih kecil terlebih dahulu kemudian mencetak N. Proses berlanjut hingga mencapai bilangan kurang dari atau sama dengan 0, dan menghentikan rekursi. Dan semua bilangan yang dioutputkan dicetak dalam urutan menaik.

6. Unguided 6

Source code

```

package main

import "fmt"

func power(x, y int) int {
    if y == 0 {

```

```

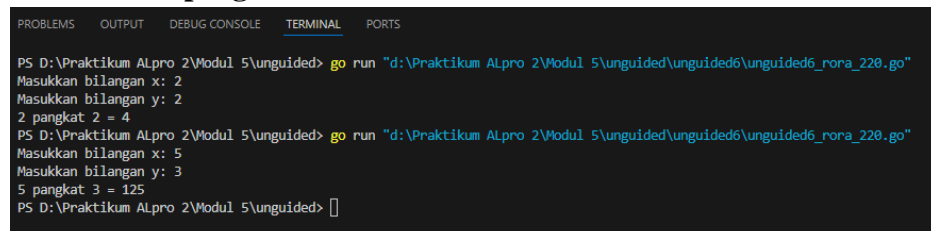
        return 1
    }
    return x * power(x, y-1)
}

func main() {
    var x, y int
    fmt.Print("Masukkan bilangan x: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan bilangan y: ")
    fmt.Scan(&y)

    result := power(x, y)
    fmt.Printf("%d pangkat %d = %d\n", x, y, result)
}

```

Screenshot program



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided6\unguided6_rora_228.go"
Masukkan bilangan x: 2
Masukkan bilangan y: 2
2 pangkat 2 = 4
PS D:\Praktikum ALpro 2\Modul 5\unguided> go run "d:\Praktikum ALpro 2\Modul 5\unguided\unguided6\unguided6_rora_228.go"
Masukkan bilangan x: 5
Masukkan bilangan y: 3
5 pangkat 3 = 125
PS D:\Praktikum ALpro 2\Modul 5\unguided> 

```

Deskripsi Program

Pada program ini menggunakan Bahasa pemrograman Go untuk menghitung hasil perpangkatan bilangan dengan menggunakan metode rekursi. Program terdiri dari dua fungsi yakni power dan main. Fungsi power (x,y int) bertujuan untuk menghitung nilai x pangkat y. apabila y 0, maka fungsi mengembalikan 1, sesuai dengan aturan matematika bahwa bilangan berapapun yang dipangkat 0 hasilnya adalah 1. Namun jika y lebih dari 0, fungsi mengalikan x dengan hasil dari power(x, y-1) hingga mencapai nilai basis.

Pengguna diminta menginputkan dua bilangan yaitu x dan y, yang mana bilangan x sebagai bilangan yang dipangkatkan dan y sebagai eksponen atau bilangan pangkat. Setelah inputan diterima program akan menjalankan fungsi power dan outputannya akan ditampilkan ke layar menggunakan fmt.Printf.

Dimana outputan pada gambar tersebut pengguna menginputkan nilai 2 untuk x dan 2 untuk y, sehingga program mencetak hasil 2 pangkat 2 adalah 4. Proses rekursi menghitung nilai tersebut dengan mengalikan 2 sebanyak dua kali, dan menghasilkan nilai 4.