

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

MODUL VI

REKURSIV



Oleh:

AHMADAN SYARIDIN

2311102038

S1-IF11-02

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

1. Dasar Teori

Rekuraif dapat diartikan sebagai cara penyelesaian suatu masalah dengan cara menyelesaikan sub-masalah yang identic dari masalah utama. Contohnya

	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	

Subprogram cetak() di atas, terlihat 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 pemanggilan subprogram cetak() nilai x akan selalu bertambah 1 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 }
```

Biasa ditambahkan struktur control percabangan (if-then) untuk menghentikan proses rekursif ini. Kondisi disebut dengan base-case, jika kondisi base-case bernilai true maka proses rekurif akan berhenti.

Catatan :

- Teknik rekursif ini merupakan salah satu alternatif untuk mengganti struktur control perulangan dengan memanfaatkan subprogram(fungsi dan procedure)
- Untuk menghentikan proses rekurtif digunakan pecabangan (if-then)
- Base-case merupakan kondisi proses rekurstif berhenti. Base-casea merupakan hal terpendi dan pertama yang harus diketahui ketika akan membuat program rekurtif. Mustahil membuat program rekurtif tanpa mengetahui base-case terlebih dahulu
- Recursive-case 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), merupakan bagian untuk menghentikan prosesn rekursif dan menjadi komponen terpenting di Dalam sebuah rekursif.
- Recussive-case yaitu bagian pengmanggilan subprogram.

Contoh program menggunakan rekursif

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 }
```

2.GUIDED

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{
        fmt.Println(1)
    } else {
        fmt.Println(bilangan)

        baris(bilangan -1)
    }
}
```

Screenshot Program

```
PS C:\semester 3\2311102038_Ahamdan Syaridin_modul5> go run "c:\semester 3\2311102038_Ahamdan Syaridin_
21
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
PS C:\semester 3\2311102038_Ahamdan Syaridin_modul5>
```

Deskripsi Program

Program dibuat untuk Melakukan cetak bilangan, kemudian memanggil dirinya sendiri dengan nilai dikurangi satu. Proses akan terus berjalan hingga bilangan mencapai 1(SATU), dimana itu akan berhenti. Maka hasilnya bilangan yang dicetak dari n hingga menurun ke 1

Guided 2

Source code

```
package main

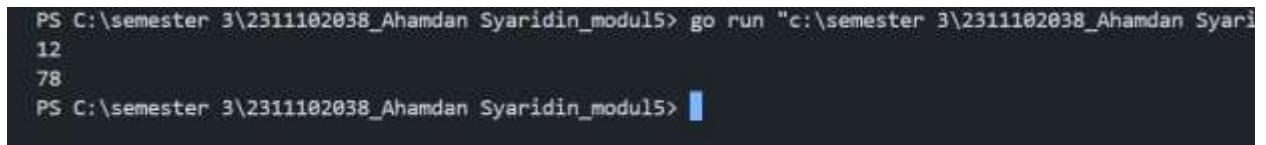
import "fmt"

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)
    }
}
```

Screenshot



```
PS C:\semester 3\2311102038_Ahamdan Syaridin_modul5> go run "c:\semester 3\2311102038_Ahamdan Syaridin_modul5\main.go"
12
78
PS C:\semester 3\2311102038_Ahamdan Syaridin_modul5>
```

Deskripsi

program menghitung penjumlahan deret bilangan dari 1 hingga n menggunakan rekursi, di mana fungsi Penjumlahan akan memanggil dirinya sendiri untuk menambahkan n dengan hasil penjumlahan dari (n-1) hingga mencapai 1.

3.UNGUIDED

Unguided 1

Source Code

```
package main

import "fmt"

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

    a, b := 0, 1
    for i := 2; i <= n; i++ {
        a, b = b, a+b
    }

    return b
}

func main() {
    fmt.Println("Deret Fibonacci hingga suku ke-10:")

    for i := 0; i <= 10; i++ {
        fmt.Printf("Suku ke-%d: %d\n", i, fibonacci(i))
    }
}
```

Screenshot Program

```
PS C:\semester 3\2311102038_Ahanda Syaridin_modul5> go run "c:\semester 3\2311102038_Ahanda Syaridin_modul5\unguided1.go"
Deret Fibonacci hingga suku ke-10:
Suku ke-0: 0
Suku ke-1: 1
Suku ke-2: 1
Suku ke-3: 2
Suku ke-4: 3
Suku ke-5: 5
Suku ke-6: 8
Suku ke-7: 13
Suku ke-8: 21
Suku ke-9: 34
Suku ke-10: 55
PS C:\semester 3\2311102038_Ahanda Syaridin_modul5> █
```

Deskripsi Program

Program go menghitung dan mencetak deret Fibonacci hingga suku ke-10. Fibonacci adalah deret di mana setiap angkanya merupakan penjumlahan dari dua angka sebelumnya, dimulai dengan 0 dan 1. Fungsi Fibonacci menggunakan pendekatan rekursif untuk menghitung suku ke-n Dalam deret

Unguided 2

Source Code

```
package main

import "fmt"

func printPattern(n int) {
    for i := 1; i <= n; i++ {
        for j := 0; j < i; j++ {
            fmt.Print("*")
```



```

    }

    fmt.Println()
}

}

func main() {
    var n int

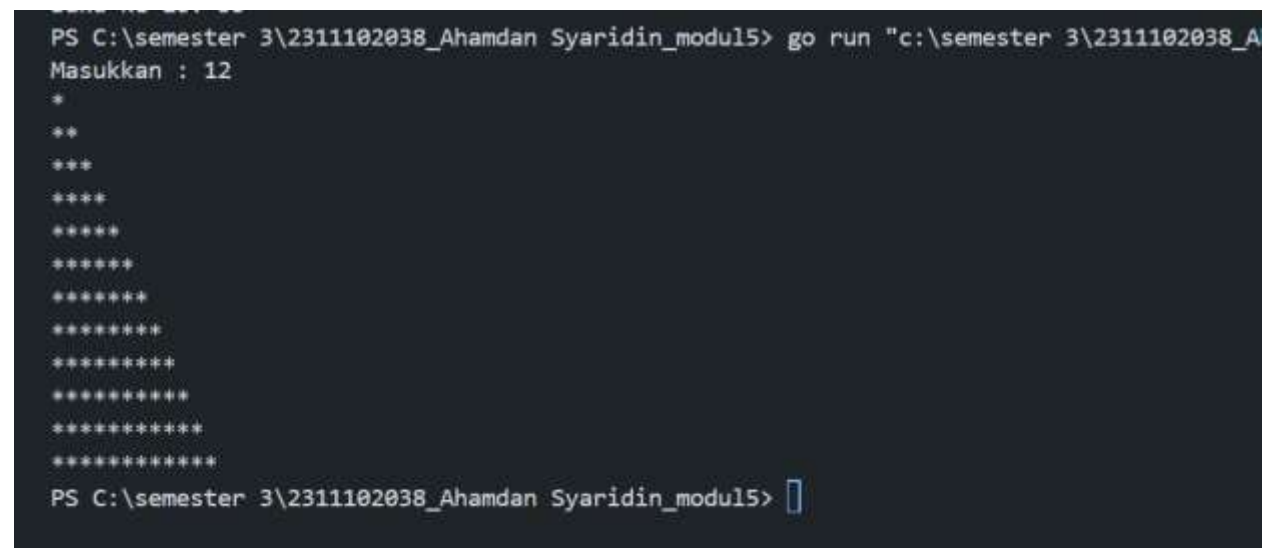
    fmt.Print("Masukkan : ")

    fmt.Scan(&n)

    printPattern(n)
}

```

Screenshot Program



```

PS C:\semester 3\2311102038_Ahamdan Syaridin_modul5> go run "c:\semester 3\2311102038_A
Masukkan : 12
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
PS C:\semester 3\2311102038_Ahamdan Syaridin_modul5>

```

Deskripsi Program

Kode di atas berfungsi untuk mencetak pola segitiga berbentuk bintang dengan pendekatan rekursif. Pola ini dimulai dari satu bintang di baris pertama, lalu bertambah satu bintang di setiap baris berikutnya hingga mencapai jumlah yang diinginkan.

Unguided 3

Source code

```
package main

import "fmt"

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

func main() {
    var n int

    fmt.Print("Masukkan : ")

    fmt.Scan(&n)

    fmt.Print("Faktor dari ", n, " adalah: ")

    printFactors(n)

    fmt.Println()
}
```

Screenshot Program



```
PS C:\semester 3\2311102038_Ahandan Syaridin_modul5> go run "c:\semester 3\2311102038_Ahandan Syaridin_modul5\unguided3.go"
Masukkan : 12
Faktor dari 12 adalah: 1 2 3 4 6 12
PS C:\semester 3\2311102038_Ahandan Syaridin_modul5>
```

Program go digunakan untuk mencetak semua faktor dari sebuah bilangan dengan menggunakan pendekatan rekursif. Faktor dari suatu bilangan merupakan angka- angka yang dapat membagi bilangan tersebut secara sempurna.

Unguided 4

Source Code

```
package main

import "fmt"

func printSequence(n int) {

    for i := n; i >= 1; i-- {

        fmt.Print(i, " ")

    }

    for i := 2; i <= n; i++ {

        fmt.Print(i, " ")

    }

}

func main() {

    var n int

    fmt.Print("Masukkan : ")

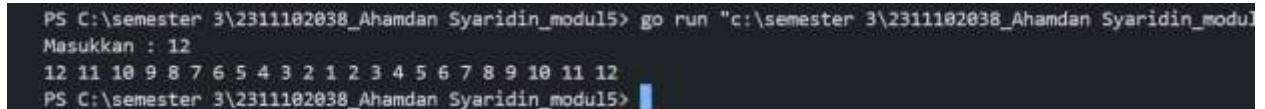
    fmt.Scan(&n)

    printSequence(n)
```

```
fmt.Println()

}
```

Screenshot program



```
PS C:\semester 3\2311102038_Ahamdan Syaridin_modul5> go run "c:\semester 3\2311102038_Ahamdan Syaridin_modul5\main.go"
Masukkan : 12
12 11 10 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 10 11 12
PS C:\semester 3\2311102038_Ahamdan Syaridin_modul5>
```

Deskripsi program

Program go diatas membuat urutan angka secara menurun dan kemudia kembali naik menggunakan pendekatan rekursif. Dimulai dengan angka n turun 1, lalu naik kembali ke n.

Dengan cara kerja fungsi printsequence(n, current int) bertugas mencetak urutan angka. Lalu current angka yang diproses. Jika mencapai 0, maka fungsi berhenti. Dan current dicetak sebelum Melakukan pemanggilan rekursif, lalu memanggil dirinya dengan current-1, setelah kembali dari rekursi jika bukan nilai awal maka current di cetak lagi untuk menciptakan urutan naik.

Unguided 5

Source code

```
package main

import "fmt"

func bilGanjil(n int) {
    for i := 1; i <= n; i += 2 {
        fmt.Print(i, " ")
    }
}
```

```

func main() {

    var n int

    fmt.Print("Masukkan: ")

    fmt.Scan(&n)

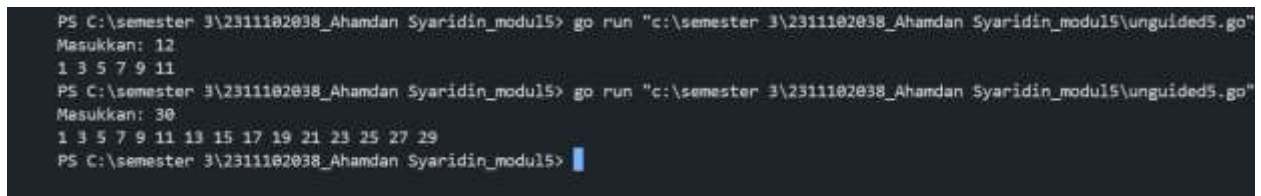
    bilGanjil(n)

    fmt.Println()

}

```

Screenshot program



```

PS C:\semester 3\2311102038_Ahmandan Syaridin_modul5> go run "c:\semester 3\2311102038_Ahmandan Syaridin_modul5\unguided5.go"
Masukkan: 12
1 3 5 7 9 11
PS C:\semester 3\2311102038_Ahmandan Syaridin_modul5> go run "c:\semester 3\2311102038_Ahmandan Syaridin_modul5\unguided5.go"
Masukkan: 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
PS C:\semester 3\2311102038_Ahmandan Syaridin_modul5>

```

Deskripsi program

Kode tersebut berfungsi untuk mencetak bilangan ganjil dari 1 hingga n, menggunakan pendekatan rekursif untuk memanggil fungsi berulang kali a menggunakan perulangan for agar lebih sederhana.

Unguided 6

Source code

```

package main

import "fmt"

func pangkat(x, y int) int {

```

```

    result := 1

    for i := 0; i < y; i++ {

        result *= x

    }

    return result
}

func main() {

    var x, y int

    fmt.Print("Masukkan x dan y: ")

    fmt.Scan(&x, &y)

    fmt.Println(pangkat(x, y))

}

```

Screenshot program



```

+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\semester 3\2311102038_Ahmandan Syaridin_modul5> go run "c:\semester 3\2311102038_Ahmandan Syaridin_modul5\unguided6.go"
Masukkan x dan y: 12 18
61917364224
PS C:\semester 3\2311102038_Ahmandan Syaridin_modul5>

```

Deskripsi program

Program go tersebut menghitung hasil perpangkatan x pangkat y menggunakan rekursi, di mana fungsi memanggil dirinya sendiri untuk mengalikan x secara berulang hingga y mencapai 0