

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

MODUL V

REKURSIF



Oleh:

IKRAM IRIANSYAH

2311102184

IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

1. Definisi Procedure

Rekursif secara sederhana dapat diartikan sebagai cara menyelesaikan suatu masalah dengan cara menyelesaikan sub-masalah yang identik dari masalah utama. Contoh :

	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 dieksekusi perintah **cetak()** 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 }
```

Oleh karena itu, biasanya 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

```

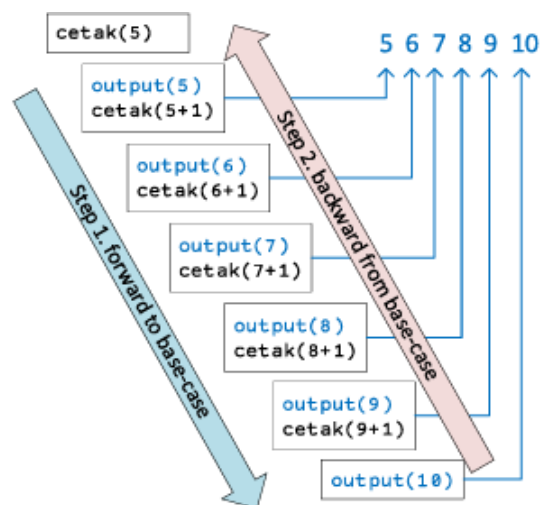
Apabila diperhatikan pada baris ke-3 di Program di atas, kita telah menambahkan **base-case** seperti penjelasan sebelumnya. Selanjutnya pada bagian aksi else di baris ke-6 dan ke-7 dinamakan **recursive-case** atau kasus pemanggilan dirinya sendiri tersebut terjadi. Kondisi dari **recursive-case** ini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

```

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 }

```

Apabila program di atas ini dijalankan maka akan tampil angka 5 6 7 8 9 10. Terlihat bahwa proses rekursif berhasil dihentikan ketika $x == 10$



Gambar 1. Ilustrasi proses forward dan backward pada saat rekursif.

Pada Gambar diatas meperlihatkan saat subprogram dipanggil secara rekursif, maka subprogram akan terus menerus melakukan pemanggilan (**forward**) hingga berhenti pada saat kondisi **base-case** akan terpenuhi atau **true**. Setelah itu terjadi proses **backward** atau kembali ke subprogram yang sebelumnya. Artinya setelah semua instruksi **cetak(10)** selesai dieksekusi, maka program akan kembali ke **cetak(9)** yang memanggil cetak(10) tersebut. begitu seterusnya hingga kembali ke cetak(5).

Perhatikan modifikasi program di atas dengan menukar posisi baris 10 dan 11, mengakibatkan ketika program dijalankan maka akan menampilkan 10 9 8 7 6 5, sesuai dengan proses **backward** pada gambar.

```
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        cetak(x+1)
11        fmt.Println(x)
12    }
13 }
```

Catatan :

- Teknik rekursif ini merupakan salah satu alternatif untuk mengganti struktur kontrol perulangan dengan memanfaatkan subprogram (bisa fungsi ataupun prosedur)
- Untuk mengidentifikasi 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 **base-case**.
- Setiap algoritma rekursif selalu memiliki padanan dalam bentuk algoritma iteratif.

2. Komponen Rekursif

Algoritma rekursif terdiri dari dua komponen utama:

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

II. GUIDED

GUIDED 1

SOURCE CODE

```
package main

import "fmt"

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

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

OUTPUT

```
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5\GUIDED\guided1.go"
8
8
7
6
5
4
3
2
1
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> |
```

Deskripsi Program

Program diatas merupakan implementasi penggunaan rekursif untuk menampilkan urutan bilangan. Program meminta user untuk menginputkan sebuah bilangan, lalu program akan memanggil fungsi rekursif yang akan menampilkan bilangan ke-n (pada program diatas menginputkan angka 8) hingga bilangan dibawahnya sampai mencapai angka base case yang telah ditentukan yaitu 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)

    }

}
```

OUTPUT

```
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> go run "c:\Users\Ikram\Documents\Folder Ikr
am\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5\GUIDED\guided2.go"
6
21
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> |
```

Deskripsi Program

Program diatas merupakan implementasi penggunaan rekursif untuk menghitung penjumlahan dari sebuah bilangan. Program akan meminta user untuk menginputkan sebuah angka lalu fungsi rekursif akan dipanggil. Program akan mengeksekusi angka yang diinputkan dengan menjumlahkannya dengan angka dibawahnya sampai pada angka base yang ditentukan yaitu angka 1. Untuk program diatas user menginputkan angka 6, program akan memproses $(6 + 5 + 4 + 3 + 2 + 1 = 21)$ melalui fungsi rekursif.

III. UNGUIDED

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.

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

OUTPUT

```
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5\UNGUIDED\unguided1.go"
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\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5>
```

Deskripsi Program :

Program ini digunakan untuk menghitung dan menampilkan deret Fibonacci hingga angka ke-10. Fungsi fibonacci bekerja secara rekursif untuk menghitung nilai Fibonacci dari suatu bilangan n. Jika n adalah 0 atau 1, fungsi akan mengembalikan nilai n. Jika lebih besar dari 1, fungsi menambahkan hasil dari dua pemanggilan sebelumnya (fibonacci(n-1) + fibonacci(n-2)). Program utama menampilkan deret Fibonacci dari 0 hingga 10.

UNGUIDED 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 cetakSegitiga(n int) {
    if n == 0 {
        return
    }
    cetakSegitiga(n - 1)
    for i := 0; i < n; i++ {
        fmt.Print("* ")
    }
    fmt.Println()
}

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

OUTPUT

```
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> go run "c:\Users\Ikram\Documents\Folder Ikr
am\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5\UNGUIDED\unguided2.go"
Masukkan Angka : 7
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> |
```

Deskripsi Program :

Program ini digunakan untuk mencetak pola segitiga bintang dengan tinggi yang ditentukan user. Program meminta user memasukkan angka, yang digunakan untuk menentukan tinggi segitiga. Fungsi cetakSegitiga bekerja secara rekursif untuk mencetak setiap baris segitiga dengan jumlah bintang yang berkurang setiap kali pemanggilan. Setiap kali fungsi dipanggil, pola bintang bertambah hingga mencapai tinggi yang dimasukkan.

UNGUIDED 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 faktorisasi(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
```

```

        fmt.Print(i, " ")
    }
    faktorisasi(n, i+1)
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif : ")
    fmt.Scan(&n)
    fmt.Print("Faktor dari ", n, " adalah : ")
    faktorisasi(n, 1)
    fmt.Println()
}

```

OUTPUT

```

PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5\UNGUIDED\unguided3.go"
Masukkan bilangan bulat positif : 12
Faktor dari 12 adalah : 1 2 3 4 6 12
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5>

```

Deskripsi Program :

Program ini digunakan untuk mencari dan menampilkan semua faktor dari bilangan yang diberikan user. Program meminta user memasukkan bilangan bulat positif, lalu fungsi faktorisasi memeriksa secara rekursif apakah setiap angka dari 1 hingga n adalah faktor dari bilangan tersebut. Jika n habis dibagi oleh i, maka i akan dicetak sebagai faktor dari n.

UNGUIDED 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 main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scanln(&n)
    printRecursive(n)
}

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

    fmt.Print(n, " ")

    printRecursive(n - 1)

    fmt.Print(n, " ")
}
```

OUTPUT

```
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5\UNGUIDED\unguided4.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 C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> |
```

Deskripsi Program :

Program ini adalah program yang meminta input berupa bilangan bulat positif n dan menggunakan fungsi rekursif untuk mencetak urutan angka dari n hingga 1, lalu kembali dari 1 ke n . Program ini menampilkan pola simetris yang dimulai dari n ke 1 dan kembali ke n .

UNGUIDED 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 printGanjil(n int, current int) {
    if current > n {
        return
    }
    if current%2 != 0 {
        fmt.Print(current, " ")
    }
    printGanjil(n, current+1)
}

func main() {
```

```

    var n int
    fmt.Print("Masukkan bilangan bulat positif N : ")
    fmt.Scan(&n)
    fmt.Print("Bilangan Ganjil dari 1 - ", n, " adalah : ")
    printGanjil(n, 1)
    fmt.Println()
}

```

OUTPUT

```

PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5\UNGUIDED\unguided5.go"
Masukkan bilangan bulat positif N : 20
Bilangan Ganjil dari 1 - 20 adalah : 1 3 5 7 9 11 13 15 17 19
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5>

```

Deskripsi Program :

Program ini digunakan untuk menampilkan bilangan ganjil dari 1 hingga bilangan yang diberikan oleh user. Program meminta user memasukkan bilangan bulat positif N, kemudian menggunakan fungsi rekursif printGanjil untuk memeriksa setiap angka dari 1 hingga N. Jika suatu angka ganjil, angka tersebut akan dicetak.

UNGUIDED 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 asterisk "*", 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(x int, y int) int {
    if y == 0 {
        return 1
    }
    return x * pangkat(x, y-1)
}

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

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

OUTPUT

```
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5\UNGUIDED\unguided6.go"
Masukkan bilangan bulat x dan y : 5 3
Hasil Pangkat : 125
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 5> |
```

Deskripsi Program :

Program ini digunakan untuk menghitung hasil pemangkatan dari dua bilangan bulat yang diberikan user. Program meminta user memasukkan dua bilangan bulat x dan y, kemudian menggunakan fungsi rekursif pangkat untuk menghitung hasil x pangkat y. Jika y sama dengan 0, fungsi mengembalikan nilai 1. jika tidak, fungsi mengalikan x dengan hasil pemanggilan fungsi dengan eksponen y yang berkurang 1.

