

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**

**MODUL V
“REKURSIF”**



Oleh:

ZAHRINA ANTIKA MALAHATI

2311102109

IF 11 02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

I. DASAR TEORI

➤ Definisi

Rekursif dalam pemrograman adalah metode pengulangan yang menggunakan diri sendiri. Dalam fungsi rekursif, fungsi akan memanggil dirinya sendiri secara langsung atau tidak langsung hingga kondisi tertentu terpenuhi. Rekursif dapat digunakan untuk menyelesaikan masalah yang membutuhkan pemecahan berulang. Metode ini membantu memecahkan masalah menjadi kategori masalah yang lebih kecil. Contoh klasik fungsi rekursif adalah perhitungan faktorial dari suatu bilangan. Faktorial dari suatu bilangan adalah hasil perkalian angka tersebut dengan semua angka di bawahnya hingga dan termasuk 1. 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 dalam notasi bahasa Go :

<pre>func cetak(x int){ fmt.Println(x) cetak(x+1) }</pre>	
---	--

➤ Catatan :

- Teknik rekursif ini merupakan salah satu alternatif untuk mengganti struktur control perulangan dengan memanfaatkan subprogram (bisa fungsi ataupun procedure).
- 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 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

❖ Guided 1

Source code

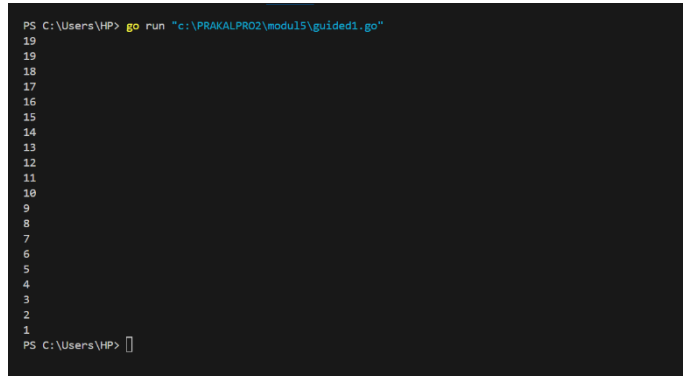
```
package main

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 {
        fmt.Println(bilangan)
        baris(bilangan - 1)
    }
}
```

Output



```
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\guided1.go"
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
PS C:\Users\HP>
```

Deskripsi

Program ini adalah sebuah program dengan menggunakan bahasa Go untuk mencetak angka dari input pengguna hingga 1 secara menurun. Program ini memanfaatkan konsep rekursi untuk mencapai tujuan tersebut. Rekursi adalah teknik pemrograman di mana sebuah fungsi memanggil dirinya sendiri. Dalam kasus ini, fungsi `baris` memanggil dirinya sendiri dengan nilai argumen yang lebih kecil hingga mencapai kondisi dasar (base case). Rekursi sangat berguna untuk menyelesaikan masalah yang memiliki struktur berulang atau hierarkis.

❖ Guided 2

Source code

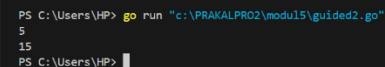
```
package main

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\HP> go run "c:\PRAKALPRO2\modul5\guided2.go"
5
15
PS C:\Users\HP> █
```

Deskripsi

Program ini adalah sebuah program dengan menggunakan bahasa Go untuk menghitung jumlah dari semua bilangan bulat dari 1 hingga n, di mana nilai n dimasukkan oleh pengguna. Program ini menggunakan konsep rekursif untuk mencapai tujuan ini. Rekursi adalah teknik pemrograman di mana sebuah fungsi memanggil dirinya sendiri. Dalam contoh ini, fungsi penjumlahan memanggil dirinya sendiri dengan argumen yang lebih kecil hingga mencapai kasus dasar (yaitu ketika n adalah 1). Rekursi sering digunakan untuk menyelesaikan masalah yang dapat dipecah menjadi submasalah yang lebih kecil dengan struktur yang sama.

III. UNGUIDED

❖ Unguided 1

Soal

- 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.Scan(&n)

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

// Zahrina Antika Malahati_2311102109
```

Output

```
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided1.go"
Masukkan nilai n: 10
Fibonacci ke-0: 0
Fibonacci ke-1: 1
Fibonacci ke-2: 1
Fibonacci ke-3: 2
Fibonacci ke-4: 3
Fibonacci ke-5: 5
Fibonacci ke-6: 8
Fibonacci ke-7: 13
Fibonacci ke-8: 21
Fibonacci ke-9: 34
Fibonacci ke-10: 55
PS C:\Users\HP> █
```

Deskripsi

Program ini adalah program dengan bahasa Go yang bekerja dengan cara memecah masalah besar (menghitung Fibonacci ke-n) menjadi masalah-masalah yang lebih kecil (menghitung Fibonacci ke-n-1 dan Fibonacci ke-n-2). Proses ini terus berulang hingga mencapai kasus dasar ($n \leq 1$).

Cara kerja program ini yaitu,

- Program meminta user memasukkan nilai n.
- Fungsi main akan memanggil fungsi fibonacci secara berulang untuk menghitung nilai Fibonacci dari 0 hingga n.
- Fungsi fibonacci akan melakukan rekursi hingga mencapai kondisi dasar ($n \leq 1$).
- Hasil perhitungan akan dicetak ke layar.

❖ Unguided 2

Soal

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

func main() {
    var n int
    fmt.Print("Masukkan jumlah baris: ")
}
```

```

    fmt.Scan(&n)

    printStars(n, 1)
}

// Zahrina Antika Malahati_2311102109

```

Output

```

PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided2.go"
Masukkan jumlah baris: 5
*
**
***
****
*****

PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided2.go"
Masukkan jumlah baris: 1
*

PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided2.go"
Masukkan jumlah baris: 3
*
**
***

PS C:\Users\HP>

```

Deskripsi

Program ini adalah program dengan bahasa Go yang menggunakan rekursi untuk membangun pola bintang secara bertahap. Setiap kali fungsi `printStars` dipanggil, ia mencetak satu baris bintang dan kemudian memanggil dirinya sendiri untuk mencetak baris berikutnya yang lebih panjang. Proses ini berulang hingga semua baris telah dicetak. Cara kerja dari program ini yaitu,

- Program meminta user memasukkan jumlah baris `n`.
- Fungsi `printStars` dipanggil dengan nilai awal `n` dan `1`.
- Fungsi ini akan terus memanggil dirinya sendiri dengan nilai `n` yang semakin kecil dan baris yang semakin besar hingga `n` menjadi `0`.
- Setiap kali fungsi dipanggil, sejumlah bintang sesuai dengan nilai baris akan dicetak.

❖ Unguided 3

Soal

- 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 isFaktor(bilangan, faktor int) bool {
    return bilangan%faktor == 0
}

func cetakFaktor(bilangan, faktor int) {
    if faktor > bilangan {
        return
    }
    if isFaktor(bilangan, faktor) {
        fmt.Print(faktor, " ")
    }
    cetakFaktor(bilangan, faktor+1)
}

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

    fmt.Printf("Faktor dari %d adalah: ", n)
    cetakFaktor(n, 1)
    fmt.Println()
}

// Zahrina Antika Malahati_2311102109
```

Output

```
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided3.go"
Masukkan bilangan: 5
Faktor dari 5 adalah: 1 5
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided3.go"
Masukkan bilangan: 12
Faktor dari 12 adalah: 1 2 3 4 6 12
PS C:\Users\HP>
```

Deskripsi

Program ini adalah program dengan menggunakan bahasa Go yang dirancang untuk mencari dan mencetak semua faktor dari sebuah bilangan bulat positif yang diberikan oleh user. Faktor dari suatu bilangan adalah bilangan bulat yang dapat membagi habis bilangan tersebut tanpa sisa. Cara kerja program ini yaitu,

- Program meminta user memasukkan sebuah bilangan.
- Fungsi cetakFaktor dipanggil dengan bilangan yang dimasukkan dan nilai awal faktor adalah 1.
- Fungsi ini akan terus memanggil dirinya sendiri dengan nilai faktor yang semakin besar hingga faktor lebih besar dari bilangan.

- Setiap kali fungsi dipanggil, akan dicek apakah faktor adalah faktor dari bilangan. Jika iya, maka faktor tersebut akan dicetak.

❖ Unguided 4

Soal

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"

// Fungsi rekursif untuk menampilkan angka dari N hingga 1
func printDescending(n int) {
    if n > 0 {
        fmt.Printf("%d ", n)
        printDescending(n - 1)
    }
}

// Fungsi rekursif untuk menampilkan angka dari 1 hingga N
func printAscending(n, current int) {
    if current <= n {
        fmt.Printf("%d ", current)
        printAscending(n, current+1)
    }
}

// Fungsi utama yang menggabungkan kedua pola
func printPattern(n int) {
    printDescending(n)
    printAscending(n, 2)
    fmt.Println()
}

func main() {
    var n int

    // Contoh input
    fmt.Print("Masukkan angka: ")
    fmt.Scanln(&n)

    // Cetak pola bilangan sesuai dengan input
    printPattern(n)
}
```

```
// Zahrina Antika Malahati_2311102109
```

Output

```
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided4.go"
Masukkan angka: 5
5 4 3 2 1 2 3 4 5
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided4.go"
Masukkan angka: 9
9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS C:\Users\HP>
```

Deskripsi

Program ini adalah program dengan bahasa Go yang dirancang untuk mencetak sebuah pola bilangan yang dimulai dari angka N , berkurang hingga angka 1, kemudian kembali meningkat sampai N . Pola ini dihasilkan menggunakan dua fungsi rekursif. Input program berupa satu bilangan bulat positif N , dan output adalah deret bilangan yang mencerminkan pola tersebut. Setiap fungsi memanggil dirinya sendiri dengan parameter yang berbeda (biasanya parameter dikurangi 1 atau ditambah 1) untuk mencapai kondisi dasar. Untuk `printDescending`, kondisi dasar adalah ketika `nnn` menjadi 0, sementara untuk `printAscending`, kondisi dasar adalah ketika angka saat ini lebih besar dari N .

❖ Unguided 5

Soal

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"
)

// Fungsi rekursif untuk menampilkan bilangan ganjil
func tampilkanGanjil(n int) {
    if n <= 0 {
        return
    }
}
```

```

    tampilkanGanjil(n - 2)
    fmt.Printf("%d ", n)
}

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

    // Menyesuaikan agar N hanya bilangan ganjil
    if N%2 == 0 {
        N = N - 1
    }

    fmt.Printf("Bilangan ganjil dari 1 hingga %d: ", N)
    tampilkanGanjil(N)
    fmt.Println()
}

// Zahrina Antika Malahati_2311102109

```

Output

```

PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided5.go"
Masukkan bilangan bulat positif N: 5
Bilangan ganjil dari 1 hingga 5: 1 3 5
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided5.go"
Masukkan bilangan bulat positif N: 20
Bilangan ganjil dari 1 hingga 19: 1 3 5 7 9 11 13 15 17 19
PS C:\Users\HP>

```

Deskripsi

Program ini adalah program dengan bahasa Go yang dirancang untuk menampilkan deret bilangan ganjil dari 1 hingga N secara rekursif, di mana N adalah bilangan bulat positif yang diinputkan oleh user. Cara kerja dari program ini yaitu,

- Program meminta user untuk memasukkan bilangan bulat positif N .
- Jika N genap, maka N diubah menjadi bilangan ganjil terdekat.
- Fungsi `tampilkanGanjil` dipanggil dengan nilai N .
- Fungsi ini akan terus memanggil dirinya sendiri dengan nilai n yang dikurangi 2 hingga n menjadi kurang dari atau sama dengan 0.
- Setelah setiap panggilan rekursif kembali, bilangan n saat itu akan dicetak.

❖ Unguided 6

Soal

- 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(x, y int) int {
    if y == 0 {
        return 1
    } else if y > 0 {
        return x * pangkat(x, y-1)
    } else {
        return 1 / pangkat(x, -y)
    }
}

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

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

// Zahrina Antika Malahati_2311102109
```

Output

```
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided6.go"
Masukkan bilangan x: 2 2
Masukkan bilangan y (pangkat): 2 pangkat 2 adalah 4
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul5\unguided6.go"
Masukkan bilangan x: 5 3
Masukkan bilangan y (pangkat): 5 pangkat 3 adalah 125
PS C:\Users\HP> █
```

Deskripsi

Program ini adalah program dengan bahasa Go yang dirancang untuk menghitung hasil pangkat dari dua buah bilangan bulat secara rekursif. Dengan kata lain, program ini akan menghitung nilai x pangkat y . Program ini memberikan contoh sederhana namun efektif tentang bagaimana menggunakan rekursi untuk menghitung pangkat dalam bahasa Go. Cara kerja dari program ini yaitu, fungsi pangkat bekerja secara rekursif dengan memecah masalah menjadi submasalah yang lebih kecil. Misalnya, untuk menghitung 2^3 , fungsi akan memecahnya menjadi $2 * 2^2$, kemudian 2^2 akan dipecah lagi menjadi $2 * 2^1$, dan seterusnya hingga mencapai kondisi dasar $y == 0$.