

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V
REKURSIF**



Disusun Oleh :

Didik Setiawan/2311102030

IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi,S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Rekursif dalam bahasa pemrograman Golang adalah konsep pemanggilan fungsi di dalam fungsi itu sendiri. Rekursif sering digunakan dalam pemrograman untuk menyelesaikan masalah yang dapat dipecah menjadi sub-masalah yang lebih kecil dengan pola yang sama, seperti penghitungan faktorial, pencarian elemen dalam struktur data yang berulang, dan sebagainya.

II. GUIDED

1. Soal Studi Case

Sourcecode

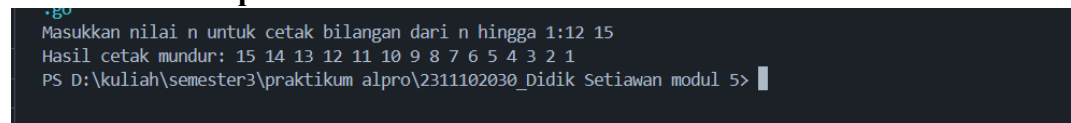
```
package main

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("Hasil cetak mundur: ")
    cetakMundur(n)
}
```

Screenshoot Output



```
Masukkan nilai n untuk cetak bilangan dari n hingga 1:15
Hasil cetak mundur: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5>
```

Deskripsi Program

Program ini meminta pengguna memasukkan nilai integer positif n, lalu menggunakan fungsi rekursif bernama cetakMundur untuk mencetak setiap bilangan dari n hingga 1 secara terpisah

2. Soal Studi Case

Sourcecode

```
package main

import "fmt"

func jumlahRekursi(n int) int {
```

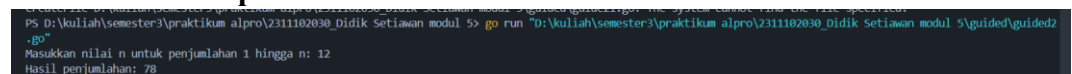
```

        if n == 1 {
            return 1
        }
        return n + jumlahRekursi(n-1)
    }

    func main() {
        var n int
        fmt.Print("Masukkan nilai n untuk penjumlahan 1
        hingga n: ")
        fmt.Scanln(&n)
        fmt.Println("Hasil penjumlahan:", jumlahRekursi(n))
    }

```

Screenshoot Output



```

PS D:\kullah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5> go run "D:\kullah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5\guided\guided2.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 12
Hasil penjumlahan: 78

```

Deskripsi Program

meminta pengguna memasukkan nilai integer positif n, lalu memanggil fungsi jumlahRekursi untuk menghitung total penjumlahan. Fungsi jumlahRekursi bekerja secara rekursif: jika n sama dengan 1, ia mengembalikan nilai 1 sebagai basis rekursi.

3. Soal Studi Case

Sourcecode

```

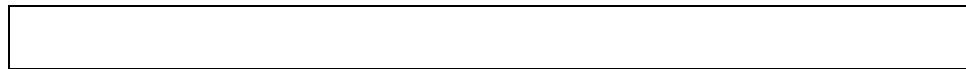
package main

import "fmt"

func pangkatDua(n int) int {
    if n == 0 {
        return 1
    }
    return 2 * pangkatDua(n-1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk mencari 2 pangkat
    n: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil 2 pangkat", n, ":",
    pangkatDua(n))
}

```



Screenshot Output

```
Hasil pangkat 2 : 4
PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5> go run "D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5\guided\guided3.go"
Masukkan nilai n untuk mencari 2 pangkat n: 5
Hasil 2 pangkat 5 : 32
```

Deskripsi Program

fungsi pangkatDua dipanggil untuk melakukan perhitungan. Fungsi ini bekerja berdasarkan prinsip rekursi, di mana ia terus memanggil dirinya sendiri sambil mengurangi nilai n satu per satu hingga mencapai nilai dasar

4. Soal Studi Case

Sourcecode

```
package main

import "fmt"

func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    return n * faktorial(n-1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk mencari faktorial n!: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil faktorial dari", n, ":", faktorial(n))
}
```

Screenshot Output

```
.go
Masukkan nilai n untuk mencari faktorial n!: 5
Hasil faktorial dari 5 : 120
PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5>
```

Deskripsi Program

Fungsi faktorial menggunakan rekursi untuk perhitungan: jika $n=0$ atau $n=1$, fungsi mengembalikan 1, karena secara matematis $0!$ dan $1!$ sama dengan 1. Untuk nilai n yang lebih besar dari 1, fungsi mengalikan n dengan hasil pemanggilan rekursif faktorial($n-1$), secara bertahap mengurangi nilai n hingga mencapai kondisi dasar

III. UNGUIDED

1. 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(term int) int {
    if term <= 1 {
        return term
    }
    return fibonacci(term-1) + fibonacci(term-2)
}

func main() {
    numTerms := 10
    fmt.Printf("Deret Fibonacci hingga suku ke-%d:\n", numTerms)
    for index := 0; index <= numTerms; index++ {
        fmt.Printf("Suku ke-%d: %d\n", index, fibonacci(index))
    }
}
```

Screenshot Output

```
ded1.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 D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5> █
```

Deskripsi Program

Fungsi fibonacci menggunakan rekursi untuk menghitung nilai dari setiap suku dalam deret, di mana suku pertama dan kedua adalah 0 dan 1, sedangkan suku berikutnya adalah hasil penjumlahan dari dua suku sebelumnya. Pada fungsi main, variabel numTerms diset ke 10, kemudian perulangan for digunakan untuk mencetak setiap suku dalam deret Fibonacci dari suku ke-0 hingga ke-10.

2. 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 printStars(rows int) {
    if rows <= 0 {
        return
    }
    printStars(rows - 1)
    for col := 0; col < rows; col++ {
        fmt.Print("* ")
    }
    fmt.Println()
}

func main() {
    var numRows int
    fmt.Print("Masukkan jumlah baris: ")
    fmt.Scan(&numRows)

    printStars(numRows)
}
```



Screenshoot Output

```
PS D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5> go run "D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5\unguided\unguided2.go"
Masukkan jumlah baris: 5
*
*
*
*
*
*
*
PS D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5>
```

Deskripsi Program

Fungsi printStars menerima parameter rows yang menentukan jumlah baris bintang yang akan dicetak. Jika rows lebih kecil atau sama dengan 0, fungsi akan berhenti (return). Jika tidak, fungsi akan memanggil dirinya sendiri dengan nilai rows - 1 untuk mencetak baris sebelumnya terlebih dahulu.

3. Studi Case

-) 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 |

Sourcecode

```
package main

import (
    "fmt"
)

func printFactors(number, divisor int) {
    if divisor > number {
        return
    }

    if number%divisor == 0 {
        fmt.Printf("%d ", divisor)
    }

    printFactors(number, divisor+1)
}
```



```

func main() {
    var number int

    fmt.Print("Masukkan bilangan bulat positif: ")
    fmt.Scan(&number)

    fmt.Printf("Faktor-faktor dari %d adalah: ", number)
    printFactors(number, 1)
    fmt.Println()
}

```

Screenshoot Output

```

PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5> go run "D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5\unguided\unguided3.go"
Masukkan bilangan bulat positif: 12
Faktor-faktor dari 12 adalah: 1 2 3 4 6 12
PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5>

```

Deskripsi Program

Fungsi printFactors menerima dua parameter: number (bilangan yang ingin dicari faktornya) dan divisor (angka pembagi yang dimulai dari 1). Jika divisor lebih besar dari number, fungsi berhenti (return). Jika number dapat dibagi habis oleh divisor, maka divisor dicetak sebagai faktor dari number

4. Studi Case

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 |

Sourcecode

```

package main

import (
    "fmt"
)

func printSeries(number, current int) {
    if current == 1 {
        fmt.Printf("%d ", current)
    }
}

```

```

        return
    }

    fmt.Printf("%d ", current)
    printSeries(number, current-1)
    fmt.Printf("%d ", current)
}

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

    fmt.Printf("Barisan bilangan dari %d hingga 1 dan
kembali ke %d:\n", number, number)
    printSeries(number, number)
    fmt.Println()
}

```

Screenshoot Output

```

PS D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5> go run "D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5\unguided\unguided4.go"
Masukkan bilangan bulat positif: 5
Barisan bilangan dari 5 hingga 1 dan kembali ke 5:
5 4 3 2 1 2 3 4 5
PS D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5>

```

Deskripsi Program

Fungsi printSeries memiliki dua parameter: number (bilangan positif yang dimasukkan pengguna) dan current (bilangan saat ini yang sedang diproses). Jika current sama dengan 1, fungsi mencetak 1 dan berhenti

5. Studi Case

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 |

Sourcecode

```

package main

import (

```

```

    "fmt"
)

func printOddNumbers(limit, current int) {
    if current > limit {
        return
    }
    if current%2 != 0 {
        fmt.Printf("%d ", current)
    }
    printOddNumbers(limit, current+1)
}

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

    fmt.Printf("Bilangan ganjil dari 1 hingga %d
adalah:\n", limit)
    printOddNumbers(limit, 1)
    fmt.Println()
}

```

Screenshoot Output

```

Masukkan bilangan bulat positif: 5
Bilangan ganjil dari 1 hingga 5 adalah:
1 3 5
PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5>

```

Deskripsi Program

printOddNumbers memiliki dua parameter: limit, yang merupakan bilangan bulat positif yang ditentukan sebagai batas atas, dan current, yang dimulai dari 1 dan menunjukkan bilangan yang sedang diproses. Fungsi ini akan berhenti jika current melebihi limit.

6. Studi Case

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 |

Sourcecode

```
package main

import "fmt"

func power(base, exponent int) int {
    if exponent == 0 {
        return 1
    } else if exponent == 1 {
        return base
    } else {
        return base * power(base, exponent-1)
    }
}

func main() {
    var base, exponent int
    fmt.Print("Masukkan nilai basis (base): ")
    fmt.Scanln(&base)
    fmt.Print("Masukkan nilai eksponen (exponent): ")
    fmt.Scanln(&exponent)

    fmt.Printf("%d pangkat %d adalah %d\n", base,
    exponent, power(base, exponent))
}
```

Screenshoot Output

```
PS D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5> go run "D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5\unguided\unguided6.go"
Masukkan nilai basis (base): 5
Masukkan nilai eksponen (exponent): 3
5 pangkat 3 adalah 125
PS D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 5>
```

Deskripsi Program

Fungsi power memiliki dua parameter: base (bilangan yang ingin dipangkatkan) dan exponent (nilai pangkat). Jika exponent bernilai 0, fungsi akan mengembalikan 1, karena setiap bilangan berpangkat nol hasilnya adalah 1