

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V
REKURSIF**



Disusun Oleh :

RAKHA YUDHISTIRA / 2311102010

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

1. Definisi Rekursif

Rekursi terjadi ketika suatu fungsi memanggil dirinya sendiri untuk menyelesaikan bagian dari tugas yang lebih besar. Fungsi rekursif biasanya memiliki dua bagian penting:

- **Basis Kasus:** Kondisi yang menghentikan rekursi. Ini mencegah fungsi dari pemanggilan tak terhingga dan memungkinkan proses untuk berakhir dengan hasil yang benar.
- **Kasus Rekursif:** Bagian di mana fungsi memanggil dirinya sendiri dengan argumen yang lebih kecil atau berbeda, bergerak menuju basis kasus.

2. Keuntungan

- Rekursi sering kali menghasilkan solusi yang lebih bersih dan lebih mudah dipahami untuk masalah yang kompleks, seperti pemrograman dinamis dan struktur data pohon.
- Mempudahkan pengkodean algoritma yang mengharuskan eksplorasi berbagai jalur atau pengulangan yang dalam.

3. Kekurangan

- Rekursif dapat mengakibatkan penggunaan memori yang lebih besar dibandingkan dengan iterasi, karena setiap pemanggilan fungsi memerlukan ruang di stack.
- Jika tidak dirancang dengan baik, rekursi dapat menyebabkan stack overflow jika terlalu banyak pemanggilan dilakukan sebelum mencapai basis kasus.

4. Contoh sederhana

Fungsi rekursif yang umum adalah perhitungan faktorial. Misalnya, faktorial dari n (ditulis $n!$) dapat dinyatakan sebagai:

- **Basis Kasus:** $0! = 1$
- **Kasus Rekursif:** $n! = n \times (n - 1)!$

II. GUIDED

1. Soal Studi Case

Sourcecode

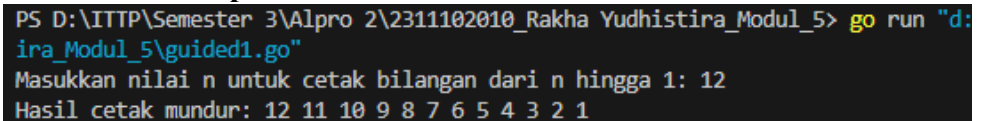
```
package main

import "fmt"

// Fungsi untuk mencetak bilangan dari n hingga 1
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



```
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_5> go run "d:
ira_Modul_5\guided1.go"
Masukkan nilai n untuk cetak bilangan dari n hingga 1: 12
Hasil cetak mundur: 12 11 10 9 8 7 6 5 4 3 2 1
```

Deskripsi Program

Kode di atas adalah program yang mencetak bilangan secara mundur dari nilai yang dimasukkan pengguna hingga 1. Fungsi cetakMundur menggunakan rekursi untuk mencetak setiap angka dari n ke 1. Jika nilai n mencapai 1, program mencetak angka tersebut dan menghentikan rekursi dengan return. Dalam fungsi main, program meminta pengguna memasukkan nilai n yang kemudian diproses oleh cetakMundur untuk mencetak hasilnya secara berurutan mundur.

2. Soal Studi Case

Sourcecode

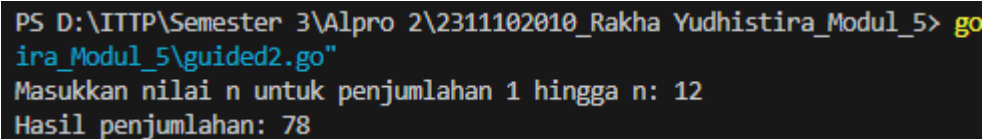
```
package main

import "fmt"

// Fungsi untuk menghitung penjumlahan 1 hingga n
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:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_5> go
ira_Modul_5\guided2.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 12
Hasil penjumlahan: 78
```

Deskripsi Program

Kode di atas adalah program yang menghitung penjumlahan dari 1 hingga nilai n yang diberikan pengguna. Fungsi `jumlahRekursi` menggunakan rekursi untuk menambahkan angka n dengan hasil dari `jumlahRekursi(n-1)` hingga mencapai dasar rekursi, yaitu saat n sama dengan 1. Pada titik tersebut, fungsi mengembalikan nilai 1. Dalam fungsi `main`, program meminta pengguna memasukkan nilai n, kemudian memanggil `jumlahRekursi(n)` untuk menghitung total penjumlahan dari 1 hingga n, dan menampilkan hasilnya di layar.

3. Soal Studi Case

Sourcecode

```
package main

import "fmt"

// Fungsi untuk mencari 2 pangkal n
```

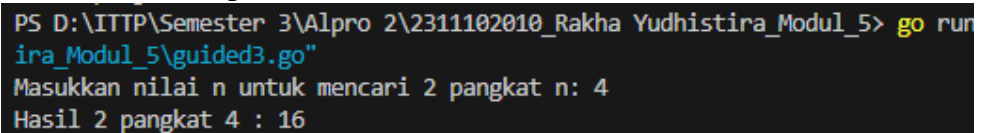
```

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))
}

```

Screenshoot Output



```

PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_5> go run
ira_Modul_5\guided3.go
Masukkan nilai n untuk mencari 2 pangkat n: 4
Hasil 2 pangkat 4 : 16

```

Deskripsi Program

Kode di atas adalah program yang menghitung nilai 2 pangkat n menggunakan rekursi. Fungsi pangkatDua menerima parameter n dan menghitung hasil 2^n secara rekursif. Jika n bernilai 0, fungsi mengembalikan 1 sebagai basis rekursi (karena $2^0 = 1$). Jika tidak, fungsi mengalikan 2 dengan hasil dari pangkatDua(n-1), yang secara bertahap mengurangi nilai n hingga mencapai 0. Pada fungsi main, program meminta pengguna memasukkan nilai n, memanggil pangkatDua(n) untuk menghitung hasil, lalu menampilkan hasilnya sebagai 2 pangkat n.

4. Soal Studi Case

Sourcecode

```

package main

import "fmt"

// Fungsi untuk menghitung faktorial n!
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

```

PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_5> go run
ira_Modul_5\guided4.go
Masukkan nilai n untuk mencari faktorial n!: 5
Hasil faktorial dari 5 : 120

```

Deskripsi Program

Kode di atas adalah program yang menghitung nilai faktorial dari suatu bilangan n menggunakan rekursi. Fungsi faktorial menerima parameter n dan mengembalikan hasil perkalian faktorial $n!$. Jika n bernilai 0 atau 1, fungsi mengembalikan 1 sebagai dasar rekursi (karena $0!$ dan $1!$ sama dengan 1). Untuk nilai n lainnya, fungsi mengalikan n dengan hasil dari faktorial($n-1$), yang terus memanggil dirinya sendiri hingga mencapai nilai dasar rekursi. Pada fungsi main, program meminta pengguna memasukkan nilai n , lalu memanggil faktorial(n) untuk menghitung faktorial dan menampilkan hasilnya di layar.

III. UNGUIDED

1. Soal 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

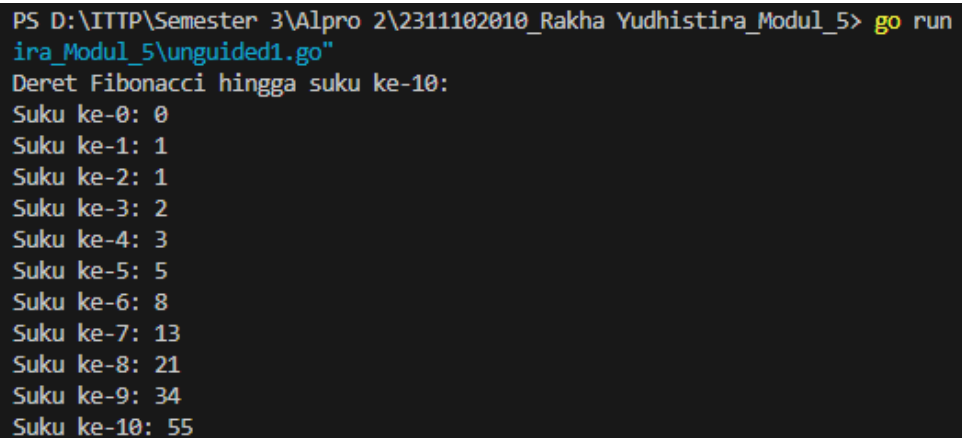
import (
    "fmt"
)

```

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

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

Screenshoot Output



```
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_5> go run
ira_Modul_5\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
```

Deskripsi Program

Kode di atas adalah program yang menghitung deret Fibonacci hingga suku ke-n menggunakan rekursi. Fungsi fibonacci menerima parameter n dan mengembalikan nilai Fibonacci dari suku tersebut. Jika n bernilai 0 atau 1, fungsi mengembalikan n sebagai dasar rekursi (karena suku pertama dan kedua dari deret Fibonacci adalah 0 dan 1). Untuk nilai n lainnya, fungsi menjumlahkan hasil dari fibonacci(n-1) dan fibonacci(n-2), yang menghasilkan nilai Fibonacci dari suku ke-n. Dalam fungsi main, variabel n diatur ke 10, lalu program mencetak setiap suku dari deret Fibonacci hingga suku ke-10 menggunakan loop for.

2. Soal Studi Case

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	* ** ***

Sourcecode

```
package main
import "fmt"

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

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

    fmt.Print("\n")
    cetakbintang(n)
}
```


Screenshoot Output

```
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_5> go run
empCodeRunnerFile.go"
Masukkan bilangan n: 12
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Deskripsi Program

Kode di atas adalah program yang mencetak pola segitiga bintang dengan jumlah baris sesuai dengan nilai n yang diberikan oleh pengguna. Fungsi cetakbintang menggunakan rekursi untuk mencetak baris-baris bintang secara bertahap dari 1 hingga n bintang per baris. Jika n kurang dari atau sama dengan 0, fungsi akan berhenti. Untuk setiap panggilan rekursif, fungsi cetakbintang mengurangi n dengan 1 hingga mencapai dasar rekursi. Setelah kembali dari rekursi, loop for mencetak n bintang dalam satu baris. Dalam fungsi main, program meminta pengguna untuk memasukkan nilai n dan kemudian memanggil cetakbintang(n) untuk mencetak pola segitiga bintang sesuai dengan input.

3. Soal 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 findFactors(N, pembagi int) {

    if pembagi > N {
        return
    }

    if N%pembagi == 0 {
        fmt.Printf("%d ", pembagi)
    }

    findFactors(N, pembagi+1)
}

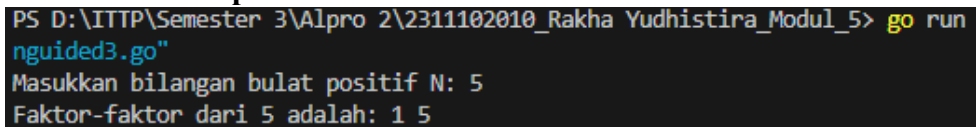
func main() {
    var N int

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

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

```

Screenshoot Output



```

PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_5> go run
nguided3.go
Masukkan bilangan bulat positif N: 5
Faktor-faktor dari 5 adalah: 1 5

```

Deskripsi Program

Kode di atas adalah program yang mencari dan mencetak faktor-faktor dari suatu bilangan bulat positif N menggunakan rekursi. Fungsi `findFactors` menerima parameter N dan divisor, di mana divisor bertindak sebagai pembagi yang diperiksa secara bertahap dari 1 hingga N. Jika divisor melebihi N, fungsi berhenti. Setiap kali N habis dibagi divisor, program mencetak divisor sebagai salah satu faktor N. Fungsi ini kemudian memanggil dirinya sendiri dengan divisor bertambah 1 untuk memeriksa faktor berikutnya. Dalam fungsi `main`, program meminta pengguna memasukkan nilai N dan kemudian memanggil `findFactors(N, 1)` untuk mencetak semua faktor dari N.

4. Soal 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(N, current int) {
    if current == 1 {
        fmt.Printf("%d ", current)
        return
    }

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

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

    fmt.Printf("Barisan bilangan dari %d hingga 1 dan\n", N, N)
    printSeries(N, N)
    fmt.Println()
}
```

Screenshoot Output

```
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_5> go run
empCodeRunnerFile.go
Masukkan bilangan bulat positif N: 5
Barisan bilangan dari 5 hingga 1 dan kembali ke 5:
5 4 3 2 1 2 3 4 5
```

Deskripsi Program

Kode di atas adalah program yang mencetak sebuah barisan bilangan dari N hingga 1, kemudian kembali lagi ke N, menggunakan rekursi. Fungsi `printSeries` menerima dua parameter: N (nilai awal yang dimasukkan pengguna) dan `current` (bilangan yang sedang diproses). Jika `current` bernilai 1, fungsi mencetak 1 dan berhenti. Jika tidak, fungsi mencetak `current`, kemudian memanggil dirinya sendiri dengan `current-1`, yang mencetak bilangan sebelumnya hingga mencapai 1. Setelah kembali dari rekursi, fungsi mencetak `current` lagi, menghasilkan pola simetris dari N ke 1 dan kembali ke N. Di `main`, program meminta pengguna memasukkan nilai N dan memanggil `printSeries(N, N)` untuk mencetak barisan tersebut.

5. Soal 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(N, current int) {
    if current > N {
        return
    }
    if current%2 != 0 {
        fmt.Printf("%d ", current)
    }
    printOddNumbers(N, current+1)
}
func main() {
    var N int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)

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

Screenshoot Output

```
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_5> go run  
empCodeRunnerFile.go  
Masukkan bilangan bulat positif N: 5  
Bilangan ganjil dari 1 hingga 5 adalah:  
1 3 5
```

Deskripsi Program

Program di atas merupakan program yang mencetak bilangan ganjil dari 1 hingga nilai N yang diinputkan oleh pengguna. Program ini menggunakan fungsi rekursif `printOddNumbers` untuk memeriksa setiap bilangan dari 1 hingga N. Jika bilangan tersebut ganjil, program akan mencetaknya. Fungsi utama `main` meminta pengguna untuk memasukkan bilangan bulat positif N, kemudian memanggil fungsi `printOddNumbers` mulai dari angka 1 untuk menampilkan semua bilangan ganjil dari 1 hingga N.

6. Soal 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:

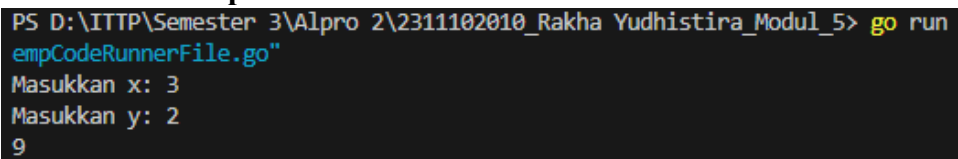
No	Masukan	Keluaran
1	2 2	4
2	5 3	125

Sourcecode

```
package main  
import "fmt"  
  
func pangkat(x, y int) int {  
    if y == 0 {  
        return 1  
    } else if y == 1 {  
        return x  
    } else {  
        return x * pangkat(x, y-1)  
    }  
}
```

```
func main() {  
    var x, y int  
    fmt.Print("Masukkan x: ")  
    fmt.Scanln(&x)  
    fmt.Print("Masukkan y: ")  
    fmt.Scanln(&y)  
  
    fmt.Println(pangkat(x, y))  
}
```

Screenshoot Output



```
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_5> go run  
empCodeRunnerFile.go"  
Masukkan x: 3  
Masukkan y: 2  
9
```

Deskripsi Program

Program di atas adalah program dalam bahasa Go yang menghitung hasil perpangkatan dari dua bilangan bulat, x dan y, yang diinputkan oleh pengguna. Fungsi rekursif pangkat menerima dua parameter, x sebagai basis dan y sebagai eksponen. Jika eksponen y bernilai 0, fungsi mengembalikan 1 (karena bilangan apa pun dipangkatkan 0 adalah 1); jika y bernilai 1, fungsi mengembalikan x. Jika tidak, fungsi mengalikan x dengan hasil pemanggilan fungsi pangkat(x, y-1) secara rekursif. Hasil perpangkatan kemudian ditampilkan pada layar.