

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VI
REKURSIF**



Disusun Oleh :

SYAHRUL ROMADHONI / 2311102261

S1 IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom.,M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Definisi Rekursif dalam Golang

Pengantar Rekursif dalam Golang

Rekursif adalah teknik pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan suatu masalah. Dalam bahasa pemrograman Go (Golang), rekursif sangat mendukung pembuatan algoritma yang lebih sederhana dan elegan, terutama untuk masalah yang dapat dipecahkan dengan cara yang berulang.

Konsep Dasar Rekursif

Definisi

Rekursif adalah proses di mana fungsi memanggil dirinya sendiri. Untuk mencegah terjadinya loop tak terbatas, setiap fungsi rekursif harus memiliki kondisi dasar (base case) yang menghentikan pemanggilan lebih lanjut.

Perbedaan dengan Iterasi

- **Rekursif:**
 - Memiliki variabel lokal baru untuk setiap panggilan.
 - Pemanggilan harus berada dalam satu kondisi.
 - Program sering kali lebih sederhana dan lebih mudah dibaca.
- **Iteratif:**
 - Tidak memiliki variabel lokal baru.
 - Pemanggilan tidak terikat pada kondisi tertentu.
 - Mungkin tidak sesederhana rekursif dalam beberapa kasus.

II. GUIDED

Soal Studi Case

1.

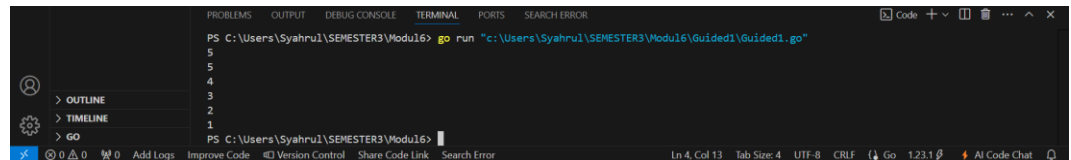
SOURCECODE

```
package main
import "fmt"

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

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

SCREENSHOOT OUTPUT



DESKRIPSI PROGRAM

Program di atas itu menerima input bilangan bulat positif nnn dari pengguna, lalu mencetak bilangan tersebut secara berurutan dari nnn hingga 1 menggunakan fungsi rekursif. Fungsi baris memeriksa apakah nilai bilangan sama dengan 1. Jika ya, maka akan mencetak angka 1 sebagai basis kasus dan mengakhiri rekursi. Jika bilangan lebih besar dari 1, program mencetak nilai bilangan saat ini dan kemudian memanggil kembali fungsi baris dengan argumen bilangan - 1, sehingga proses ini akan berlanjut hingga mencapai 1.

2.

SOURCECODE

```
package main

import "fmt"
```

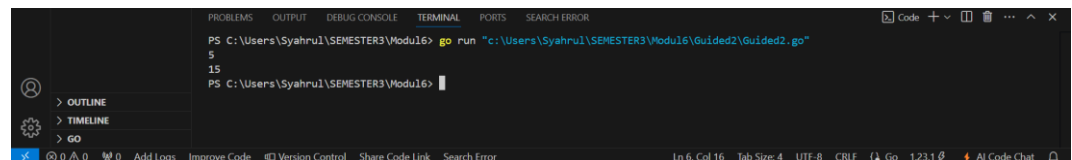
```

func penjumlahan(n int) int {
    if n == 1 {
        return 1
    } else {
        return n + penjumlahan(n-1)
    }
}

func main() {
    var n int
    fmt.Scan(&n)
    fmt.Println(penjumlahan(n))
}

```

SCREENSHOOT OUTPUT



DESKRIPSI PROGRAM

Program di atas adalah implementasi GoLang untuk menghitung jumlah total dari bilangan 1 hingga nnn menggunakan fungsi rekursif. Fungsi penjumlahan menerima parameter n dan memiliki basis kasus ketika n sama dengan 1, di mana ia mengembalikan 1. Jika n lebih besar dari 1, fungsi akan mengembalikan hasil penjumlahan dari n dan pemanggilan fungsi penjumlahan dengan parameter n-1, sehingga fungsi terus memanggil dirinya sendiri hingga mencapai nilai 1. Dalam fungsi main, pengguna diminta memasukkan nilai n, yang kemudian dihitung hasil penjumlahannya oleh fungsi penjumlahan, dan hasilnya dicetak ke layar.

3.

SOURCECODE

```

package main
import "fmt"

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

func main() {
    var n int
    fmt.Println("Masukkan nilai n : ")
}

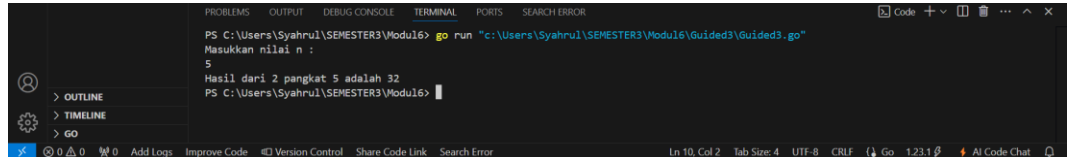
```

```

    fmt.Scan(&n)
    fmt.Println("Hasil dari 2 pangkat", n, "adalah", pangkat(n))
}

```

SCREENSHOOT OUTPUT



DESKRIPSI PROGRAM

Program ini menghitung nilai dari $2n^{2n}$ (dua pangkat nnn) menggunakan fungsi rekursif. Fungsi pangkat menerima parameter n dan mengembalikan hasil dari $2n^{2n}$ pangkat n . Basis kasus dalam fungsi ini adalah ketika n sama dengan 0, yang mengembalikan nilai 1 (karena $20=12^0=120=1$). Jika n lebih besar dari 0, fungsi akan memanggil dirinya sendiri dengan parameter $n-1$ dan mengalikannya dengan 2. Di dalam fungsi main, program meminta pengguna untuk memasukkan nilai n , kemudian memanggil fungsi pangkat untuk menghitung hasilnya, dan menampilkan hasil perhitungan ke layar.

4.

SOURCECODE

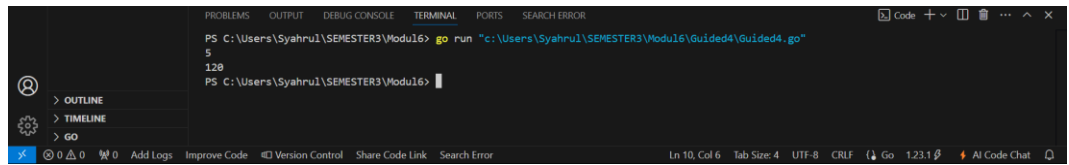
```

package main
import "fmt"

var n int
func faktorial (n int) int {
    if n == 0 || n == 1 {
        return 1
    } else {
        return n * faktorial(n-1)
    }
}
func main () {
    fmt. Scan(&n)
    fmt.Println(faktorial(n))
}

```

SCREENSHOOT OUTPUT



```
PS C:\Users\Syahrul\SEMESTER3\Modu16> go run "c:\Users\Syahrul\SEMESTER3\Modu16\Guided4\Guided4.go"
5
128
PS C:\Users\Syahrul\SEMESTER3\Modu16>
```

DESKRIPSI PROGRAM

Program di atas adalah program GoLang yang menghitung nilai faktorial dari suatu bilangan bulat positif yang diberikan pengguna menggunakan fungsi rekursif. Pertama, program mendeklarasikan variabel global `n` untuk menyimpan input. Fungsi faktorial kemudian didefinisikan untuk menghitung faktorial dari bilangan `n`. Fungsi ini memiliki kondisi dasar yang mengembalikan 1 jika `n` bernilai 0 atau 1, sesuai dengan definisi faktorial. Jika `n` lebih besar dari 1, fungsi akan memanggil dirinya sendiri dengan nilai `n-1` dan mengalikan hasilnya dengan `n`. Di dalam fungsi main, program menerima input pengguna melalui `fmt.Scan(&n)`, lalu memanggil fungsi faktorial dengan argumen `n`, dan mencetak hasilnya menggunakan `fmt.Println`.

III. UNGUIDED

Soal Studi Case

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

Sourcecode

```
package main

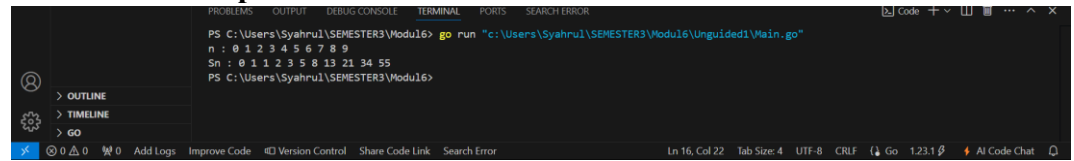
import "fmt"

func fibonacci(n int) int {
    if n == 0 {
        return 0
    } else if n == 1 {
        return 1
    } else {
        return fibonacci(n-1) + fibonacci(n-2)
    }
}

func main() {
    fmt.Println("n : ")
    for i := 0; i < 10; i++ {
        fmt.Print(i, " ")
    }

    fmt.Println("\nSn : ")
    for i := 0; i <= 10; i++ {
        fmt.Print(fibonacci(i), " ")
    }
}
```

Screenshoot Output



```
PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided1\Main.go"
n : 0 1 2 3 4 5 6 7 8 9
Sn : 0 1 1 2 3 5 8 13 21 34 55
PS C:\Users\Syahrul\SEMESTER3\Modul6>
```

Deskripsi Program

Program di atas adalah implementasi rekursif dari deret Fibonacci dalam bahasa Go. Program ini mendefinisikan fungsi `fibonacci(n int) int` yang menghitung nilai ke-`n` dalam deret Fibonacci. Fungsi ini memiliki basis kasus untuk nilai 0 dan 1, di mana `fibonacci(0)` mengembalikan 0 dan `fibonacci(1)` mengembalikan 1. Untuk nilai lainnya, fungsi memanggil dirinya sendiri dengan `n-1` dan `n-2`, lalu menjumlahkan hasilnya.

Pada fungsi `main`, program pertama-tama mencetak indeks deret Fibonacci dari 0 hingga 9 pada baris pertama. Kemudian, pada baris kedua, program mencetak nilai deret Fibonacci dari indeks 0 hingga 10 dengan memanggil fungsi `fibonacci`. Hasil akhirnya adalah dua baris: satu untuk indeks `n` dan satu lagi untuk nilai deret Fibonacci `Sn`.

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	* ** ***

Sourcecode

```
package main

import "fmt"

func printStars(n int) {
```



```

    if n > 0 {
        fmt.Print("*")
        printStars(n - 1)
    }
}

func printPattern(n int, current int) {
    if current > n {
        return
    }
    printStars(current)
    fmt.Println()
    printPattern(n, current + 1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai N untuk pola bintang: ")
    fmt.Scan(&n)

    printPattern(n, 1)
}

```

Screenshoot Output

```

PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided2\Main.go"
Masukkan nilai N untuk pola bintang: 5
*
**
***
****
*****

PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided2\Main.go"
Masukkan nilai N untuk pola bintang: 1
*

PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided2\Main.go"
Masukkan nilai N untuk pola bintang: 3
*
**
***

PS C:\Users\Syahrul\SEMESTER3\Modul6>

```

Deskripsi Program

Program ini menghasilkan pola bintang berbentuk segitiga bertingkat. Pengguna diminta untuk memasukkan nilai N sebagai jumlah baris pola yang diinginkan. Program menggunakan dua fungsi rekursif: `printStars` untuk mencetak sejumlah bintang di setiap baris, dan `printPattern` untuk mengatur jumlah bintang per baris dari 1 hingga N. Setiap baris bertambah satu bintang dari baris sebelumnya hingga mencapai baris ke-N. Hasilnya adalah pola bertahap dengan bintang yang meningkat di setiap baris.

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

Sourcecode

```
package main

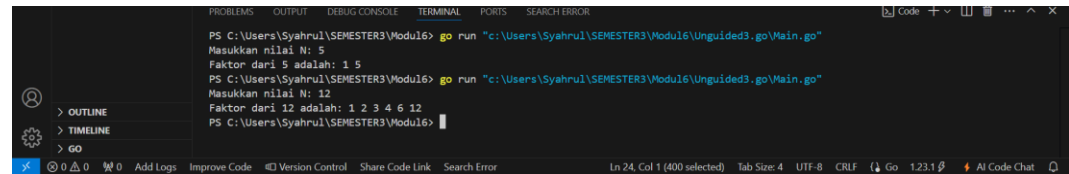
import "fmt"

func printFactors(n, divisor int) {
    if divisor > n {
        return
    }
    if n % divisor == 0 {
        fmt.Print(divisor, " ")
    }
    printFactors(n, divisor + 1)
}

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

    fmt.Print("Faktor dari ", n, " adalah: ")
    printFactors(n, 1)
    fmt.Println()
}
```

Screenshoot



```
PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided3.go\Main.go"
Masukkan nilai N: 5
Faktor dari 5 adalah: 1 5
PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided3.go\Main.go"
Masukkan nilai N: 12
Faktor dari 12 adalah: 1 2 3 4 6 12
PS C:\Users\Syahrul\SEMESTER3\Modul6>
```

Deskripsi Program

Program di atas berfungsi untuk menampilkan faktor-faktor dari suatu bilangan integer positif yang diinputkan oleh pengguna. Program meminta pengguna memasukkan sebuah nilai N, kemudian menggunakan fungsi `printFactors` untuk mencari dan mencetak semua faktor dari N secara berurutan mulai dari 1. Fungsi `printFactors` bekerja secara rekursif dengan memeriksa apakah nilai N habis dibagi oleh setiap angka dari 1 hingga N. Jika habis dibagi, angka tersebut dicetak sebagai faktor.

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

Sourcecode

```
package main

import "fmt"

func printNumbers(n, current int, descending bool) {
    fmt.Print(current, " ")

    if descending {
        if current == 1 {
            descending = false
        } else {
            printNumbers(n, current-1, true)
        }
    }
}
```

```

        if !descending && current < n {
            printNumbers(n, current+1, false)
        }
    }

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

        fmt.Print("Keluaran: ")
        printNumbers(n, n, true)
        fmt.Println()
    }

```

Screenshoot

```

PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided4.go\Main.go"
Masukkan nilai N: 5
Keluaran: 5 4 3 2 1 2 3 4 5
PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided4.go\Main.go"
Masukkan nilai N: 9
Keluaran: 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS C:\Users\Syahrul\SEMESTER3\Modul6>

```

Deskripsi Program

Program di atas adalah program dalam bahasa Go yang mencetak deret angka dari nilai n (yang dimasukkan pengguna) turun ke 1, lalu naik kembali ke nilai n . Program ini meminta pengguna untuk memasukkan nilai integer n , lalu mencetak hasilnya sesuai pola tersebut. Fungsi utama `printNumbers` menggunakan rekursi untuk mencetak angka secara berurutan. Saat $current$ sama dengan 1, arah pencetakan beralih dari menurun menjadi naik hingga mencapai nilai n kembali.

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

Sourcecode

```
package main
```

```

import "fmt"

func printOddNumbers(n, current int) {
    if current > n {
        return
    }
    fmt.Print(current, " ")
    printOddNumbers(n, current + 2)
}

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

    fmt.Print("Keluaran: ")
    printOddNumbers(n, 1)
    fmt.Println()
}

```

Screenshoot

```

PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided5.go\Main.go"
Masukkan nilai N: 5
Keluaran: 1 3 5
PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided5.go\Main.go"
Masukkan nilai N: 20
Keluaran: 1 3 5 7 9 11 13 15 17 19
PS C:\Users\Syahrul\SEMESTER3\Modul6>

```

Deskripsi Program

Program di atas adalah program Go yang mencetak bilangan ganjil dari 1 hingga nilai maksimum NNN yang ditentukan oleh pengguna. Program ini meminta pengguna untuk memasukkan nilai NNN, lalu memanggil fungsi rekursif `printOddNumbers`, yang mencetak bilangan ganjil secara berurutan mulai dari 1 hingga NNN. Fungsi `printOddNumbers` terus menambahkan 2 ke nilai saat ini (dengan nilai awal 1) sampai mencapai atau melebihi NNN. Jika kondisi ini terpenuhi, fungsi akan berhenti.

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

Sourcecode

```
package main

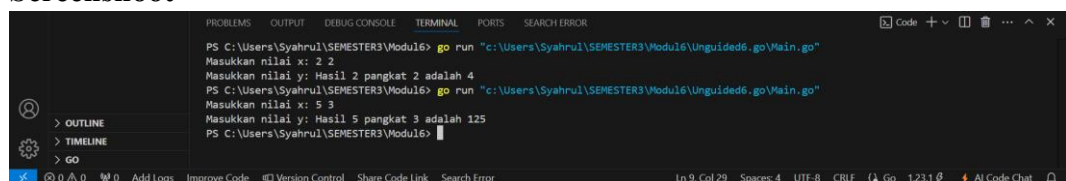
import "fmt"

func power(x, y int) int {
    if y == 0 {
        return 1
    }
    return x * power(x, y-1)
}

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

    fmt.Printf("Hasil %d pangkat %d adalah %d\n", x, y, power(x, y))
}
```

Screenshoot



```
PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided6.go\Main.go"
Masukkan nilai x: 2 2
Masukkan nilai y: Hasil 2 pangkat 2 adalah 4
PS C:\Users\Syahrul\SEMESTER3\Modul6> go run "c:\Users\Syahrul\SEMESTER3\Modul6\Unguided6.go\Main.go"
Masukkan nilai x: 5 3
Masukkan nilai y: Hasil 5 pangkat 3 adalah 125
PS C:\Users\Syahrul\SEMESTER3\Modul6>
```

Deskripsi Program

Program di atas adalah aplikasi sederhana dalam bahasa Go yang menghitung pangkat dari suatu bilangan. Fungsi `power(x, y int) int` adalah fungsi rekursif yang menghitung `xxx` pangkat `yyy`. Jika `yyy` adalah 0, fungsi mengembalikan 1, yang merupakan kasus dasar. Jika tidak, fungsi akan memanggil dirinya sendiri dengan nilai `yyy` dikurangi 1 hingga mencapai kasus dasar. Dalam fungsi `main`, program meminta pengguna memasukkan dua bilangan integer, `xxx` (basis) dan `yyy` (eksponen). Kemudian, program menampilkan hasil perhitungan pangkat tersebut dalam format yang rapi.

KESIMPULAN

Golang mendukung rekursi, yaitu kemampuan suatu fungsi untuk memanggil dirinya sendiri dalam penyelesaian masalah yang berulang. Rekursi pada Golang efektif untuk memecah masalah besar menjadi sub-masalah yang lebih kecil, seperti pada perhitungan pangkat, faktorial, atau deret Fibonacci. Namun, penggunaan rekursi di Golang perlu dipertimbangkan dengan bijak, karena dapat meningkatkan penggunaan memori dan risiko stack overflow jika kedalaman rekursi terlalu besar.

DAFTAR PUSTAKA

https://lmsspada.kemdikbud.go.id/pluginfile.php/658695/mod_resource/content/9/M6%20-%20Bahasa%20Pemrograman%20Java.pdf