

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VI
REKURSIF**



Disusun Oleh :

Siti Madina Halim Siregar / 2311102243

S1IF-11-05

Dosen Pengampu :

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Algoritma rekursif adalah jenis algoritma yang cukup menarik karena memungkinkan sebuah fungsi atau prosedur untuk memanggil dirinya sendiri dengan input yang semakin menyempit. Dalam bahasa yang lebih sederhana, algoritma ini adalah cara untuk memecahkan masalah dengan memecahkannya menjadi beberapa versi yang lebih kecil dari masalah itu sendiri.

- Algoritma rekursif memiliki beberapa tujuan, antara lain:
 - Dapat digunakan untuk memecahkan masalah yang kompleks menjadi sub masalah yang lebih sederhana.
 - Algoritma ini sering digunakan dalam implementasi struktur data seperti pohon, graf, atau daftar terhubung.
 - Serta dapat digunakan untuk melakukan pengulangan secara efisien.

- Jenis Algoritma Rekursif

a. Factorial

Algoritma factorial digunakan untuk menghitung faktorial dari suatu bilangan. Faktorial dari suatu bilangan n (ditulis $n!$) adalah hasil perkalian semua bilangan bulat positif dari 1 hingga n .

b. Tower of Hanoi

Algoritma Tower of Hanoi digunakan untuk memindahkan tumpukan cakram dari satu tiang ke tiang lainnya, dengan aturan bahwa hanya satu cakram yang dapat dipindahkan pada satu waktu dan cakram yang lebih besar tidak boleh ditempatkan di atas yang lebih kecil.

c. DFS of Graph

Algoritma Depth-First Search (DFS) digunakan untuk melakukan pencarian pada struktur data graf secara rekursif. Algoritma ini mengunjungi semua simpul dalam graf dengan mengikuti jalur secara mendalam sebelum kembali.

d. Eksponensial

Algoritma eksponensial digunakan untuk menghitung hasil dari suatu operasi eksponensial dengan memanggil diri sendiri.

Pengertian Golang

Golang adalah bahasa pemrograman open-source yang memiliki sintaksis sederhana namun kuat, memungkinkan pengembang untuk menulis kode dengan cepat dan efisien. Bahasa ini menggunakan tipe data statis dan menghasilkan kode biner yang dikompilasi, sehingga dapat berjalan dengan cepat dan efisien.

II. GUIDED

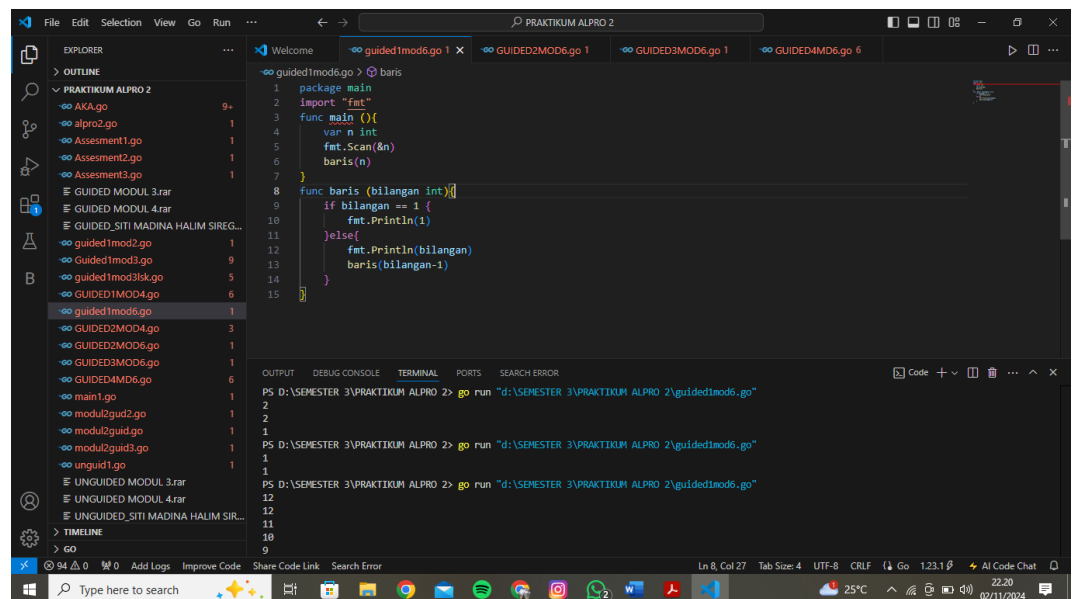
Soal Studi Case

Membuat baris bilangan dari n hingga 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 membaca bilangan bulat n dari pengguna.
- jika bilangan sama dengan 1, program mencetak 1.
- jika tidak, program mencetak nilai bilangan saat ini dan memanggil dirinya sendiri dengan nilai bilangan - 1.
- terus dipanggil secara rekursif hingga mencapai 1, mencetak semua bilangan dari n hingga 1.

Soal Studi Case

Menghitung hasil penjumlahan 1 hingga n

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

The screenshot shows a Go IDE with a project named "PRAKTIKUM ALPRO 2". The Explorer panel on the left lists various files, including "GUIDED2MOD6.go" which is currently selected. The main editor displays the code for "GUIDED2MOD6.go", which defines a recursive function "penjumlahan" and a "main" function. The "main" function reads an integer "n" from the user and prints the result of "penjumlahan(n)". The output console at the bottom shows the command "go run \"d:\SEMESTER 3\PRAKTIKUM ALPRO 2\GUIDED2MOD6.go\"" being executed, and the output "15" is displayed, indicating that the input was 5.

```
1 package main
2 import "fmt"
3
4 func penjumlahan(n int) int{
5     if n == 1{
6         return 1
7     }else{
8         return n + penjumlahan (n-1)
9     }
10 }
11 func main(){
12     var n int
13     fmt.Scan(&n)
14     fmt.Println(penjumlahan(n))
15 }
```

OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMESTER 3\PRAKTIKUM ALPRO 2\GUIDED2MOD6.go"

PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMESTER 3\PRAKTIKUM ALPRO 2\GUIDED2MOD6.go"

PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2> 15

Deskripsi Program

- Jika n sama dengan 1, fungsi mengembalikan 1.
- Jika tidak, fungsi mengembalikan nilai n ditambah hasil pemanggilan penjumlahan(n-1), sehingga menghitung jumlah secara rekursif.
- Membaca bilangan bulat n dari input pengguna.
- Memanggil fungsi penjumlahan(n) dan mencetak hasilnya.
- Contoh, jika pengguna memasukkan 5, outputnya akan menjadi 15 (1 + 2 + 3 + 4 + 5).

Soal Studi Case

Mencari 2 pangkat n atau 2^n

Source Code

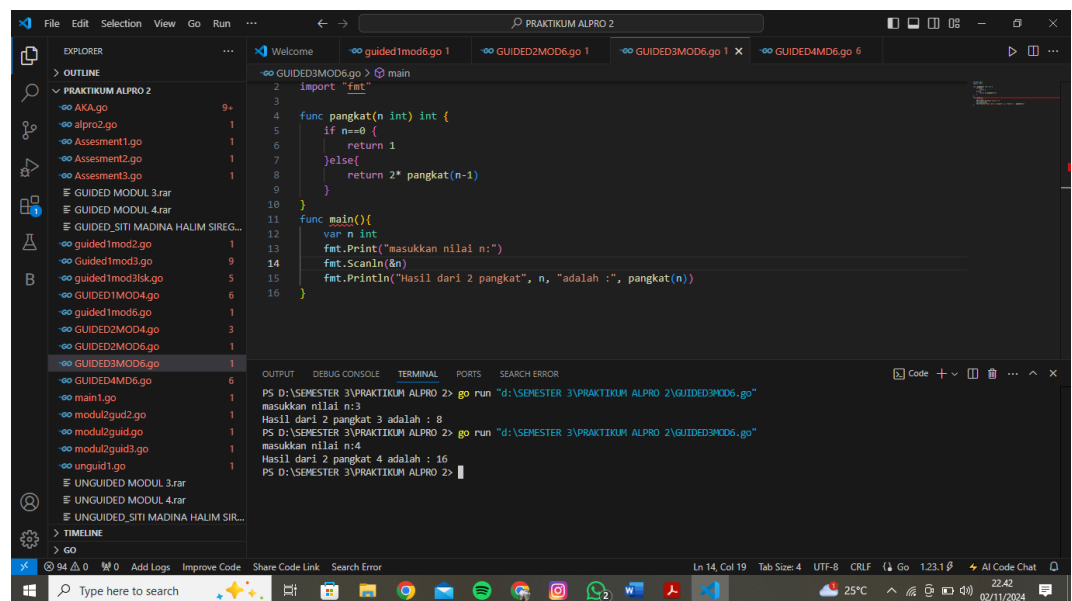
```
package main

import "fmt"

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

func main(){
    var n int
    fmt.Print("masukkan nilai n:")
    fmt.Scanln(&n)
    fmt.Println("Hasil dari 2 pangkat", n, "adalah :", pangkat(n))
}
```

Screenshoot Output



Deskripsi Program

- Menerima parameter n dan mengembalikan nilai 2^n
- Jika n sama dengan 0, fungsi mengembalikan 1 (karena $2^0=1$)
- Jika tidak, fungsi mengembalikan 2 dikalikan dengan hasil pemanggilan pangkat($n-1$), sehingga menghitung nilai pangkat secara rekursif.
- Meminta pengguna untuk memasukkan nilai n .
- Mencetak hasil dari fungsi pangkat(n).
- Contoh, jika pengguna memasukkan 3, outputnya akan menjadi 8 (karena $2^3=8$)

Soal Studi Case

Mencari nilai faktorial atau $n!$

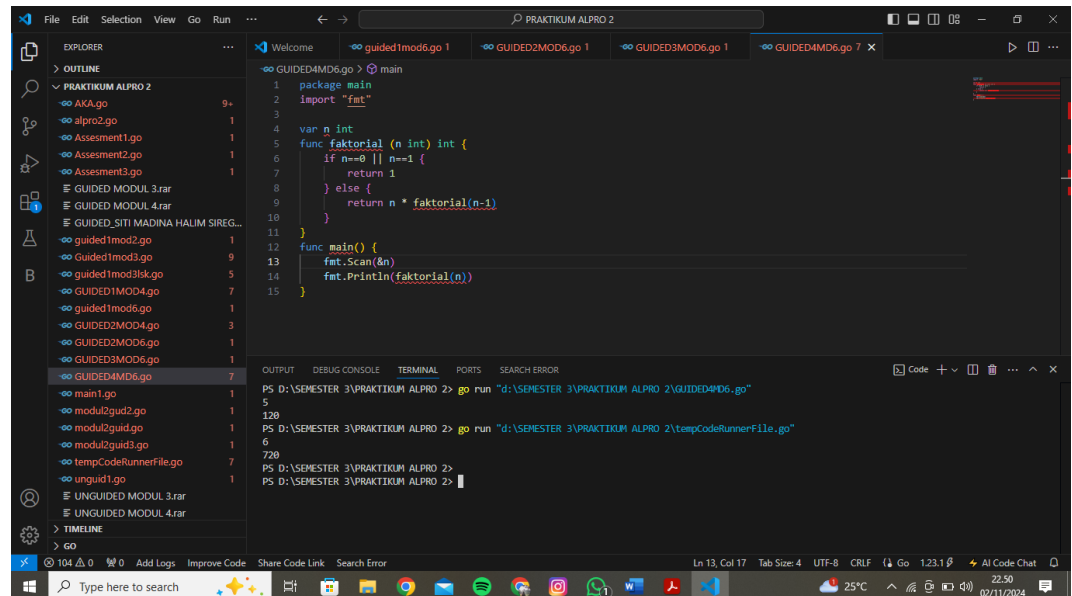
Source Code

```
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



The screenshot shows a Go IDE with a file explorer on the left, a code editor in the center, and an output console at the bottom. The code editor displays a Go program that calculates the factorial of a number n using a recursive function. The output console shows the program being run, the input 5 , and the resulting output 120 .

```
1 package main
2 import "fmt"
3
4 var n int
5 func faktorial (n int) int {
6     if n==0 || n==1 {
7         return 1
8     } else {
9         return n * faktorial(n-1)
10    }
11 }
12 func main() {
13     fmt.Scan(&n)
14     fmt.Println(faktorial(n))
15 }
```

OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

```
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMESTER 3\PRAKTIKUM ALPRO 2\GUIDED4MD6.go"
5
120
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMESTER 3\PRAKTIKUM ALPRO 2\tempCodeRunnerFile.go"
6
720
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2>
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2>
```

Deskripsi Program

- Menerima parameter n dan mengembalikan nilai faktorial dari n .
- Jika n sama dengan 0 atau 1 , fungsi mengembalikan 1 (karena $0!=1$ dan $1!=1$).
- Jika tidak, fungsi mengembalikan n dikalikan dengan hasil pemanggilan $\text{faktorial}(n-1)$, sehingga menghitung faktorial secara rekursif.
- Membaca nilai n dari input pengguna.
- Mencetak hasil dari fungsi $\text{faktorial}(n)$.
- Contoh, jika pengguna memasukkan 5 , outputnya akan menjadi 120 (karena $5!=120$)

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"

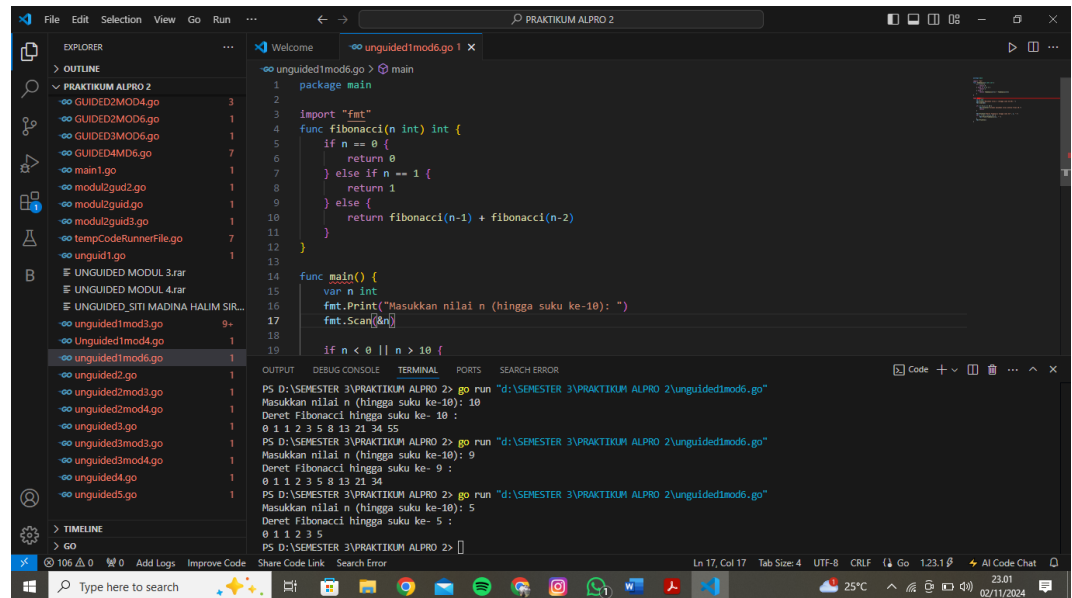
func fibonacci(n int) int {
    if n == 0 {
        return 0
    } else if n == 1 {
        return 1
    } else {
        return fibonacci(n-1) + fibonacci(n-2)
    }
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n (hingga suku ke-10): ")
    fmt.Scan(&n)

    if n < 0 || n > 10 {
        fmt.Println("Silakan masukkan nilai antara 0 dan 10.")
        return
    }

    fmt.Println("Deret Fibonacci hingga suku ke-", n, ":")
    for i := 0; i <= n; i++ {
        fmt.Print(fibonacci(i), " ")
    }
    fmt.Println()
}
```

Screenshoot Output



The screenshot shows a Go IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays a Go program for calculating Fibonacci numbers. The terminal shows the execution of the program with three test cases: n=10, n=9, and n=5. The output for each case shows the first n Fibonacci numbers.

```
1 package main
2
3 import "fmt"
4 func fibonacci(n int) int {
5     if n == 0 {
6         return 0
7     } else if n == 1 {
8         return 1
9     } else {
10        return fibonacci(n-1) + fibonacci(n-2)
11    }
12 }
13
14 func main() {
15     var n int
16     fmt.Print("Masukkan nilai n (hingga suku ke-10): ")
17     fmt.Scan(&n)
18
19     if n < 0 || n > 10 {
20         // Error handling code
21     }
22 }
```

Terminal Output:

```
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMESTER 3\PRAKTIKUM ALPRO 2\unguided1mod6.go"
Masukkan nilai n (hingga suku ke-10): 10
Deret Fibonacci hingga suku ke- 10 :
0 1 1 2 3 5 8 13 21 34 55
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMESTER 3\PRAKTIKUM ALPRO 2\unguided1mod6.go"
Masukkan nilai n (hingga suku ke-10): 9
Deret Fibonacci hingga suku ke- 9 :
0 1 1 2 3 5 8 13 21 34
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMESTER 3\PRAKTIKUM ALPRO 2\unguided1mod6.go"
Masukkan nilai n (hingga suku ke-10): 5
Deret Fibonacci hingga suku ke- 5 :
0 1 1 2 3 5
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2>
```

Deskripsi Program

- Menerima parameter n dan mengembalikan nilai suku Fibonacci ke-n.
- Jika n adalah 0, mengembalikan 0.
- Jika n adalah 1, mengembalikan 1.
- Jika n lebih dari 1, fungsi mengembalikan penjumlahan dari dua suku sebelumnya: $\text{fibonacci}(n-1) + \text{fibonacci}(n-2)$.
- Fungsi main:
- Membaca nilai n dari input pengguna, dengan batasan hingga 10.
- Jika n berada di luar rentang yang ditentukan, program akan mencetak pesan kesalahan.
- Jika valid, program mencetak deret Fibonacci dari suku 0 hingga suku ke-n.

2. Soal Studi Case

Buatlah sebuah program yang digunakan untuk menampilkan pola bintang berikut ini dengan menggunakan fungsi rekursif. N adalah masukan dari user.

No	Masukan	Keluaran
1	5	* ** *** **** *****
2	1	*
3	3	* ** ***

Sourcecode

```
package main

import "fmt"

func printStars(n int, current int) {
    if current > n {
        return
    }

    for i := 0; i < current; i++ {
        fmt.Print("*")
    }
    fmt.Println()
    printStars(n, current+1)
}

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

    printStars(n, 1)
}
```

Screenshoot Output

The screenshot shows a Go IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays a Go program with a recursive function `printStars` and a `main` function. The terminal shows the output of running the program, which prints stars in a recursive pattern for input values 5 and 3.

```
package main

import "fmt"

func printStars(n int, current int) {
    if current > n {
        return
    }

    for i := 0; i < current; i++ {
        fmt.Print("*")
    }
    fmt.Println()
    printStars(n, current+1)
}

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

OUTPUT: PS D:\SEMASTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMASTER 3\PRAKTIKUM ALPRO 2\unguided2mod6.go"
Masukkan nilai N: 5
*
**

PS D:\SEMASTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMASTER 3\PRAKTIKUM ALPRO 2\unguided2mod6.go"
Masukkan nilai N: 3
*
**

PS D:\SEMASTER 3\PRAKTIKUM ALPRO 2>

Deskripsi Program

- Fungsi `printStars`:
- Menerima dua parameter: `n` (jumlah maksimum bintang yang akan dicetak pada baris terakhir) dan `current` (jumlah bintang yang akan dicetak pada baris saat ini).
- Jika `current` lebih besar dari `n`, fungsi akan berhenti (basis rekursi).
- Mencetak `current` bintang.
- Setelah mencetak bintang, fungsi memanggil dirinya sendiri dengan `current` ditambah 1 untuk mencetak baris berikutnya.
- Membaca nilai `n` dari input pengguna.
- Memanggil fungsi `printStars` untuk memulai pencetakan bintang dari 1.

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).

No	Masukan	Keluaran
1	5	1 5
2	12	1 2 3 4 6 12

Sourcecode

```
package main

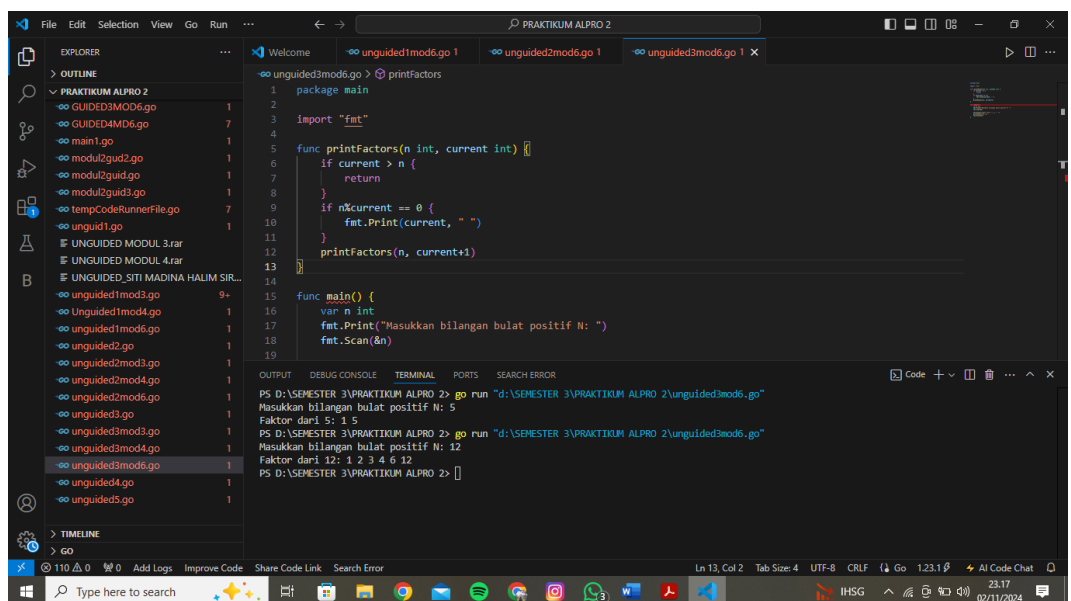
import "fmt"

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

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

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

Screenshoot Output



Deskripsi Program

- Fungsi printFactors:
- Menerima dua parameter: n (bilangan yang dicari faktornya) dan current (angka yang sedang diperiksa apakah merupakan faktor dari n).
- Jika current lebih besar dari n, fungsi akan berhenti (basis rekursi).
- Jika n dapat dibagi habis oleh current, maka current adalah faktor, dan program akan
- Fungsi main:
- Membaca nilai n dari input pengguna.
- Memanggil fungsi printFactors untuk mulai mencetak faktor dari 1.mencetaknya.
- Fungsi kemudian memanggil dirinya sendiri dengan current ditambah 1 untuk memeriksa angka berikutnya.

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.

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

5

Sourcecode

```
package main

import "fmt"

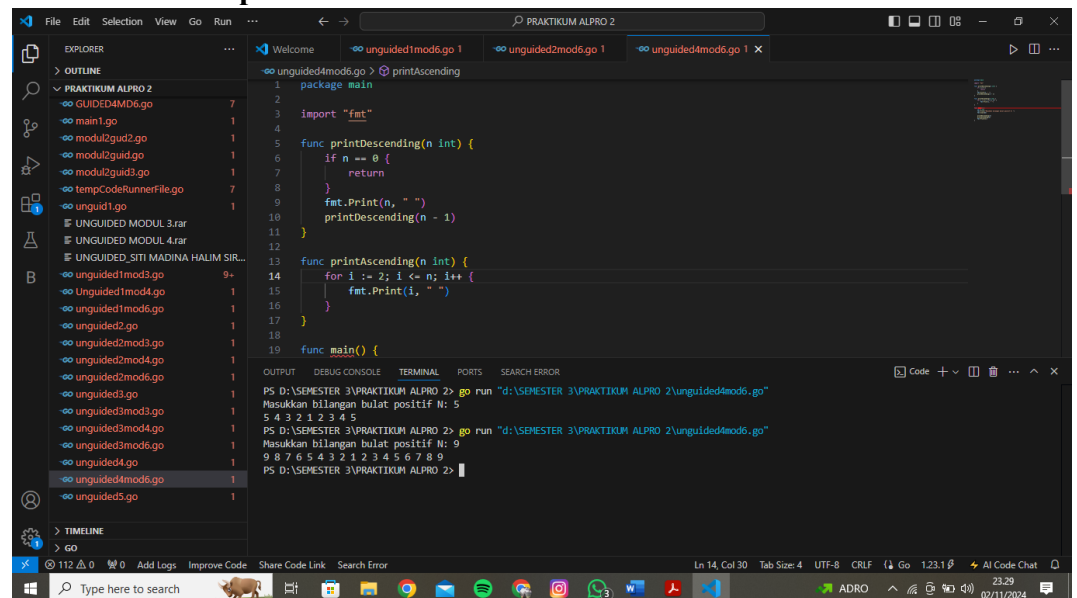
func printDescending(n int) {
    if n == 0 {
        return
    }
    fmt.Print(n, " ")
    printDescending(n - 1)
}

func printAscending(n int) {
    for i := 2; i <= n; i++ {
        fmt.Print(i, " ")
    }
}
```

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

    printDescending(n)
    printAscending(n)
    fmt.Println()
}
```

Screenshoot Output



Deskripsi Program

- Fungsi `printDescending`:
- Menerima parameter `n` dan mencetak bilangan dari `n` hingga 1 secara rekursif.
- Jika `n` sama dengan 0, fungsi berhenti (basis rekursi).
- Jika tidak, fungsi mencetak `n` dan memanggil dirinya sendiri dengan `n` dikurangi 1.
- Fungsi `printAscending`:
- Mencetak bilangan dari 2 hingga `n` menggunakan loop sederhana.
- Fungsi `main`:
- Membaca nilai `n` dari input pengguna.
- Memanggil fungsi `printDescending` untuk mencetak dari `N` hingga 1.
- Memanggil fungsi `printAscending` untuk mencetak dari 2 hingga `N`.

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.

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

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

    fmt.Print("Bilangan ganjil dari 1 hingga ", n, ": ")
    printOddNumbers(n, 1)
    fmt.Println()
}
```


Screenshoot Output

The screenshot shows a Go IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays a Go program with a recursive function `printOddNumbers` and a `main` function. The terminal shows the output of the program, which prints odd numbers from 1 to 5 and then from 1 to 20.

```
1 package main
2
3 import "fmt"
4
5 func printOddNumbers(n int, current int) {
6     if current > n {
7         return
8     }
9     if current%2 != 0 {
10        fmt.Print(current, " ")
11    }
12    printOddNumbers(n, current+1)
13 }
14
15 func main() {
16     var n int
17     fmt.Print("Masukkan bilangan bulat positif N: ")
18     fmt.Scan(&n)
19 }
```

OUTPUT

```
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMESTER 3\PRAKTIKUM ALPRO 2\unguided5mod6.go"
Masukkan bilangan bulat positif N: 5
Bilangan ganjil dari 1 hingga 5: 1 3 5
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2> go run "d:\SEMESTER 3\PRAKTIKUM ALPRO 2\unguided5mod6.go"
Masukkan bilangan bulat positif N: 20
Bilangan ganjil dari 1 hingga 20: 1 3 5 7 9 11 13 15 17 19
PS D:\SEMESTER 3\PRAKTIKUM ALPRO 2>
```

Deskripsi Program

- Fungsi `printOddNumbers`:
- Menerima dua parameter: `n` (batas atas) dan `current` (angka yang sedang diperiksa).
- Jika `current` lebih besar dari `n`, fungsi berhenti (basis rekursi).
- Jika `current` adalah bilangan ganjil (diperiksa dengan `current%2 != 0`), fungsi mencetak `current`.
- Fungsi kemudian memanggil dirinya sendiri dengan `current` ditambah 1 untuk memeriksa angka berikutnya.
- Fungsi `main`:
- Membaca nilai `n` dari input pengguna.
- Mencetak header dan memanggil fungsi `printOddNumbers` untuk mulai mencetak dari 1.

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"`.

No	Masukan	Keluaran
1	2 2	4
2	5 3	125

Sourcecode

```
package main

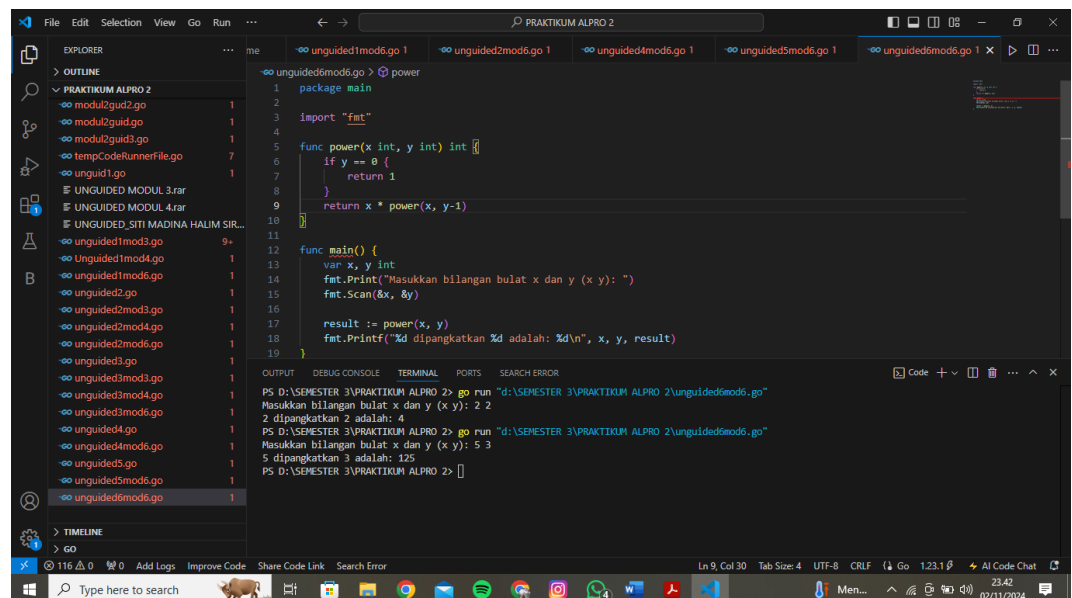
import "fmt"

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

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

    result := power(x, y)
    fmt.Printf("%d dipangkatkan %d adalah: %d\n", x, y, result)
}
```

Screenshoot Output



Deskripsi Program

- Fungsi power:
- Menerima dua parameter: x (bilangan yang akan dipangkatkan) dan y (pangkat).
- Jika y sama dengan 0, fungsi mengembalikan 1 (karena $x^0 = 1$)
- Jika tidak, fungsi mengembalikan hasil perkalian x dengan pemanggilan rekursif power(x, y-1), sehingga menghitung x^y
- Fungsi main:
- Membaca dua bilangan bulat x dan y dari input pengguna.
- Memanggil fungsi power untuk menghitung hasil x dipangkatkan y.
- Mencetak hasilnya.

Kesimpulan

Algoritma rekursif adalah jenis algoritma yang cukup menarik karena memungkinkan sebuah fungsi atau prosedur untuk memanggil dirinya sendiri dengan input yang semakin menyempit. Dalam bahasa yang lebih sederhana, algoritma ini adalah cara untuk memecahkan masalah dengan memecahkannya menjadi beberapa versi yang lebih kecil dari masalah itu sendiri.

Daftar Pustaka

<https://fikti.umsu.ac.id/algoritma-rekursif-pengertian-tujuan-dan-jenisnya/>