

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 6

REKURSIF



Disusun Oleh : Brian Farrel Evandhika

2311102037

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

6.1 Pengantar Rekursif

pada modul-modul sebelumnya sudah dijelaskan bahwa suatu subprogram baik fungsi atau prosedur

bisa memanggil subprogram lainnya. Hal ini tidak menutup kemungkinan bahwa subprogram yang

dipanggil adalah dirinya sendiri. Dalam pemrograman teknik ini dikenal dengan istilah rekursif.

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!

Berikut ini adalah cara penulisan deklarasi prosedur pada notasi Pseudocode dan GoLang.

Notasi Algoritma

```
procedure cetak(in x: integer)
```

```
algoritma
```

```
    output(x)
```

```
    cetak(x+1)
```

```
endprocedure
```

Notasi dalam bahasa Go

```
func cetak(x int) {
```

```
    fmt.Println(x)
```

```
    cetak(x + 1)
```

```
}
```

Apabila diperhatikan subprogram

cetak() di atas, terlihat pada baris ke-4 terdapat pemanggilan subprogram cetak() ke baris ke-5. Misalnya apabila kita eksekusi perintah cetak(5) maka akan menampilkan angka 5 6 7 8 9...dst tanpa henti. Artinya setiap pemanggilan subprogram cetak() nilai x akan selalu bertambah 1 (Increment by one) secara terus menerus tanpa henti

Notasi dalam bahasa Go

```
package main

import "fmt"

func main() {
    cetak(5)
}

func cetak(x int) {
    fmt.Println(x)
    cetak(x+1)
}
```

Oleh karena itu biasanya ditambahkan struktur kontrol percabangan (if-then) untuk menghentikan proses rekursif ini. Kondisi ini disebut juga dengan base-case, artinya apabila kondisi base-case bernilai true maka proses rekursif akan berhenti. Sebagai contoh misalnya base case adalah ketika x bernilai 10 atau $x == 10$, maka tidak perlu dilakukan rekursif

Notasi Algoritma

```
procedure cetak(in x:integer)

    algoritma
        if x == 10 then
            output(x)
        else
            output(x)
```

```
        cetak(x+1)

    endif

endprocedure
```

Apabila diperhatikan pada baris ke-3 di Program di atas, kita telah menambahkan base-case seperti penjelasan sebelumnya. Selanjutnya pada bagian atasi else di baris ke-5 dan ke-7 kita namakan recursive-case atau kasus pemanggilan dirinya sendiri tersebut terjadi. Kondisi dari recursive-case ini adalah negasi dari kondisi base-case atau ketika nilai $x \neq 10$.

Notasi dalam bahasa Go

```
package main

import "fmt"

func main() {

    cetak(5)

}

func cetak(x int) {

    if x == 10 {

        fmt.Println(x)

    } else {

        fmt.Println(x)

        cetak(x+1)

    }

}
```

Apabila program di atas ini dijalankan maka akan tampil angka 5 6 7 8 9 10. Terlihat bahwa proses

rekursif berhasil dihentikan ketika $x = 10$.

Catatan:

- Teknik rekursif ini merupakan salah satu alternatif untuk mengganti struktur kontrol perulangan dengan memanfaatkan subprogram (bisa fungsi ataupun prosedur).
- 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 dalam bentuk algoritma iteratif.

6.2 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.

6.3 Contoh Program dengan menggunakan Rekursif

a. Membuat baris bilangan dari n hingga 1

Base-case: `bilangan == 1`

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)  
    }  
}
```

b. Menghitung hasil penjumlahan 1 hingga n

Base-case: n 1

```
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)
    }
}

```

c. Mencari dua pangkat n atau 2^n

Base-case: $n == 0$

```
package main
```

```
import "fmt"
```

```

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

```

```

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

```

```
}
```

```
}
```

Mencari nilai faktorial atau $n!$

Base-case: $n == 0$ atau $n == 1$

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var n int
```

```
    fmt.Scan(&n)
```

```
    fmt.Println(faktorial(n))
```

```
}
```

```
func faktorial(n int) int {
```

```
    if n == 0 || n == 1 {
```

```
        return 1
```

```
    } else {
```

```
        return n * faktorial(n-1)
```

```
    }
```

```
}
```


GUIDED

Soal Studi Case

Buatlah fungsi untuk mencetak bilangan dari n hingga 1

Sourcecode

```
package main

import "fmt"

// fungsi untuk mencetak bilangan dari n hingga 1
func cetakMundur(n int) {
    if n <= 0 {
        return
    }

    fmt.Println(n)
    cetakMundur(n - 1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk mencetak bilangan dari n
hingga 1: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil cetak mundur:")
    cetakMundur(n)
}
```

Screenshot Program

```
-Module PSReadLine'.

PS C:\Users\MSI GAMING> go run "c:\Users\MSI GAMING\Documents\TELKOM UNIVERS
ITY\SEMESTER 3\PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 2\Pertemuan Ke 6\Guided-1
.go"
Masukkan nilai n untuk mencetak bilangan dari n hingga 1: 10
Hasil cetak mundur:
10
9
8
7
6
5
4
3
2
1
PS C:\Users\MSI GAMING> 
```

Deskripsi Program

Program ini dirancang untuk mencetak bilangan bulat secara menurun dari nilai yang diinputkan pengguna hingga mencapai 1. Proses ini memanfaatkan konsep rekursif, di mana fungsi `cetakMundur` terus memanggil dirinya sendiri dengan nilai yang semakin kecil hingga mencapai kondisi dasar. Setiap pemanggilan fungsi akan mencetak bilangan saat itu dan melanjutkan proses rekursif. Dengan demikian, program akan menghasilkan output berupa urutan bilangan menurun mulai dari nilai input hingga 1.

Soal Studi Case

Buatlah fungsi untuk menghitung penjumlahan dari 1 hingga n

Sourcecode

```
package main

import "fmt"

// fungsi untuk menghitung penjumlahan dari 1 hingga n
func jumlahRekursif(n int) int {
    if n == 1 {
        return 1
    }
    return n + jumlahRekursif(n-1)
}

func main() {
    var n int
    fmt.Print("Masukan nilai n untuk cetak bilangan dari 1 hingga n: ")
    fmt.Scan(&n)
    fmt.Println("Hasil penjumlahan: ", jumlahRekursif(n))
}
```

Screenshot Program

```
-Module PSReadLine'.

PS C:\Users\MSI GAMING> go run "c:\Users\MSI GAMING\Documents\TELKOM UNIVERS
ITY\SEMESTER 3\PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 2\Pertemuan Ke 6\Guided-2
.go"
Masukan nilai n untuk penjumlahan dari 1 hingga n: 10
Hasil penjumlahan: 55
PS C:\Users\MSI GAMING> |
```

Deskripsi Program

Program ini dirancang untuk menghitung jumlah bilangan bulat dari 1 hingga n dengan menggunakan konsep rekursif. Fungsi `jumlahRekursif` akan terus memanggil dirinya sendiri dengan nilai n yang semakin kecil hingga mencapai kondisi dasar (ketika n sama dengan 1). Setiap pemanggilan fungsi akan menambahkan nilai n saat itu dengan hasil pemanggilan fungsi sebelumnya. Dengan demikian, program akan menghitung jumlah total dari 1 hingga n secara rekursif dan menampilkan hasilnya.

Soal Studi Case

Buatlah Fungsi rekursif untuk menghitung 2^n

Sourcecode

```
package main

import "fmt"

// Fungsi rekursif untuk menghitung 2^n
func pangkatDua(n int) int {
    if n == 0 {
        return 1
    }
    return 2 * pangkatDua(n-1)
}

func main() {
    var n int
    fmt.Println("Masukan nilai n untuk mencari 2 pangkat n: ")
    fmt.Scan(&n)
    fmt.Println("Hasil 2 pangkat", n, ":", pangkatDua(n))
}
```

Screenshot Program

```
-Module PSReadLine'.

PS C:\Users\MSI GAMING> go run "c:\Users\MSI GAMING\Documents\TELKOM UNIVERS
ITY\SEMESTER 3\PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 2\Pertemuan Ke 6\tempCode
RunnerFile.go"
Masukan nilai n untuk mencari 2 pangkat n:
3
Hasil 2 pangkat 3 : 8
PS C:\Users\MSI GAMING> |
```

Deskripsi Program

Program ini dirancang untuk menghitung nilai 2 pangkat n secara rekursif, di mana n adalah bilangan bulat non-negatif yang diinputkan oleh pengguna. Fungsi `pangkatDua` merupakan inti dari program ini. Fungsi ini akan terus memanggil dirinya sendiri dengan nilai n yang semakin kecil hingga mencapai 0 (base case). Pada setiap pemanggilan, hasil akan dikalikan dengan 2. Konsep rekursif ini memungkinkan kita untuk memecahkan masalah yang lebih besar menjadi submasalah yang lebih kecil secara berulang. Program ini memberikan contoh sederhana namun efektif tentang bagaimana rekursi dapat diterapkan dalam pemrograman Go.

Soal Studi Case

Buatlah fungsi untuk menghitung faktorial dari n!

Sourcecode

```
package main

import "fmt"

// Fungsi untuk menghitung faktorial dari n!
func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    return n * faktorial(n-1)
}

func main() {
    var n int
```

```
fmt.Println("Masukan nilai n untuk mencari faktorial n!: ")
fmt.Scan(&n)
fmt.Println("Hasil faktorial dari", n, ":", faktorial(n))
}
```

Screenshot Program

```
-Module PSReadLine'.

PS C:\Users\MSI GAMING> go run "c:\Users\MSI GAMING\Documents\TELKOM UNIVERS
ITY\SEMESTER 3\PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 2\Pertemuan Ke 6\Guided-4
.go"
Masukan nilai n untuk mencari faktorial n!:
10
Hasil faktorial dari 10 : 3628800
PS C:\Users\MSI GAMING> |
```

Deskripsi Program

Program ini dirancang untuk menghitung faktorial dari suatu bilangan bulat yang diinputkan oleh pengguna. Fungsi faktorial menggunakan konsep rekursif untuk menghitung faktorial. Jika nilai n adalah 0 atau 1, maka fungsi akan langsung mengembalikan nilai 1 (karena faktorial 0 dan 1 adalah 1). Jika tidak, fungsi akan mengembalikan hasil perkalian antara n dengan hasil rekursif dari $n-1$. Proses rekursif ini akan terus berulang hingga mencapai kondisi dasar (ketika n sama dengan 0 atau 1). Dengan demikian, program akan menghitung faktorial dari bilangan yang diinputkan secara efisien dan menampilkan hasilnya.

II. 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.

Source Code

```
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
package main

import "fmt"

// Fungsi rekursif untuk menghitung Fibonacci
func fibonacci(n int) int {
    // Basis kasus untuk suku ke-0 dan ke-1
    if n == 0 {
        return 0
    } else if n == 1 {
        return 1
    }
    // Rekurens untuk menghitung suku ke-n
    return fibonacci(n-1) + fibonacci(n-2)
}

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

Screenshot Program

```
PS C:\Users\MSI GAMING> go run "c:\Users\MSI GAMING\Documents\TELKOM UNIVERS  
ITY\SEMESTER 3\PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 2\Pertemuan Ke 6\Unguided  
-1.go"  
S_0 = 0  
S_1 = 1  
S_2 = 1  
S_3 = 2  
S_4 = 3  
S_5 = 5  
S_6 = 8  
S_7 = 13  
S_8 = 21  
S_9 = 34  
S_10 = 55  
PS C:\Users\MSI GAMING>
```

Deskripsi Program

Program ini dirancang untuk menghitung dan menampilkan deret Fibonacci hingga suku ke-10 menggunakan konsep rekursif. **Fungsi fibonacci bekerja** dengan cara membagi masalah menjadi submasalah yang lebih kecil. Fungsi ini memiliki dua kasus dasar: jika n adalah 0, maka hasilnya adalah 0; jika n adalah 1, maka hasilnya adalah 1. Untuk nilai n lainnya, fungsi akan menghitung nilai Fibonacci dengan menjumlahkan hasil rekursif dari $n-1$ dan $n-2$. Dalam fungsi main, program akan mengulang perhitungan Fibonacci dari suku ke-0 hingga ke-10 dan mencetak hasilnya ke layar. Dengan demikian, program ini memberikan contoh implementasi rekursif untuk menghitung deret bilangan Fibonacci.

Soal Studi Case 2

Buatlah sebuah program yang digunakan untuk menampilkan pola bintang berikut ini dengan

menggunakan fungsi rekursif. N adalah masukan dari user.

Source Code

```
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
package main

import (
    "fmt"
)

// Fungsi rekursif untuk mencetak satu baris bintang
func printStars(n int) {
    if n == 0 {
        return
    }
    fmt.Print("*")
    printStars(n - 1)
}

// Fungsi rekursif untuk mencetak pola bintang bertingkat
func printPattern(n, current int) {
    if current > n {
        return
    }
    // Cetak bintang sejumlah nilai 'current'
    printStars(current)
    fmt.Println() // Pindah ke baris berikutnya
    // Rekursi untuk baris berikutnya
    printPattern(n, current + 1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai N: ")
    fmt.Scan(&n)
    printPattern(n, 1)
}
```


Screenshot Program

```
PS C:\Users\MSI GAMING> go run "c:\Users\MSI GAMING\Documents\TELKOM UNIVERS  
ITY\SEMESTER 3\PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 2\Pertemuan Ke 6\tempCode  
RunnerFile.go"  
Masukkan nilai N: 10  
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
PS C:\Users\MSI GAMING> |
```

Deskripsi Program

Program ini dirancang untuk mencetak pola segitiga siku-siku yang terdiri dari bintang-bintang secara rekursif. Program ini menerima input berupa bilangan bulat N yang menentukan jumlah baris pada segitiga. Menggunakan dua fungsi rekursif, `printStars` dan `printPattern`, program ini secara bertahap membangun pola segitiga. Fungsi `printStars` mencetak sejumlah bintang tertentu pada satu baris, sedangkan fungsi `printPattern` mengatur jumlah bintang pada setiap baris dan mengulang proses mencetak baris hingga mencapai jumlah baris yang ditentukan oleh N. Dengan demikian, program ini memberikan contoh menarik tentang bagaimana rekursif dapat digunakan untuk menghasilkan pola grafis yang dinamis.

Soal Studi Case 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).

Source Code

```
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
package main

import "fmt"

// Fungsi rekursif untuk menemukan faktor
func findFactors(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    findFactors(n, i+1)
}

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

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

Screenshot Program

```
-Module PSReadLine'.

PS C:\Users\MSI GAMING> go run "c:\Users\MSI GAMING\Documents\TELKOM UNIVERS
ITY\SEMESTER 3\PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 2\Pertemuan Ke 6\tempCode
RunnerFile.go"
Masukkan bilangan positif: 10
Faktor dari 10: 1 2 5 10
PS C:\Users\MSI GAMING> 
```

Deskripsi Program

Program ini dirancang untuk menemukan dan mencetak semua faktor dari sebuah bilangan bulat positif yang diinputkan oleh pengguna. Fungsi `findFactors` menggunakan pendekatan rekursif untuk memeriksa setiap bilangan bulat mulai dari 1 hingga bilangan yang diinputkan. Jika bilangan tersebut merupakan faktor dari bilangan yang diinputkan, maka bilangan tersebut akan dicetak. Fungsi ini akan terus memanggil dirinya sendiri dengan nilai `i` yang bertambah satu hingga mencapai batas atas yaitu bilangan yang diinputkan. Dengan demikian, program ini akan secara sistematis mencari dan menampilkan semua faktor dari bilangan yang diberikan.

Soal Studi Case 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`.

Source Code

```
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
package main

import "fmt"

// Fungsi rekursif untuk mencetak barisan menurun dari N ke 1
func printDown(n int) {
    if n < 1 {
        return
    }
    fmt.Print(n, " ")
    printDown(n - 1)
}

// Fungsi rekursif untuk mencetak barisan naik dari 1 ke N
func printUp(n, current int) {
    if current > n {
        return
    }
    fmt.Print(current, " ")
    printUp(n, current+1)
}
```

```

}

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

    fmt.Print("Hasil: ")
    printDown(n)
    printUp(n, 2) // Mulai dari 2 agar tidak mencetak 1 dua kali
    fmt.Println()
}

```

Screenshot Program

```

-Module PSReadLine'.

PS C:\Users\MSI GAMING> go run "c:\Users\MSI GAMING\Documents\TELKOM UNIVERS
ITY\SEMESTER 3\PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 2\Pertemuan Ke 6\tempCode
RunnerFile.go"
Masukkan bilangan positif: 10
Hasil: 10 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 10
PS C:\Users\MSI GAMING>

```

Deskripsi Program

Program ini dirancang untuk mencetak dua jenis barisan bilangan secara rekursif, yaitu barisan menurun dari N ke 1 dan kemudian barisan naik dari 1 ke N. Fungsi `printDown` mencetak bilangan dari N hingga 1 dengan cara memanggil dirinya sendiri secara berulang dengan nilai yang semakin kecil. Fungsi `printUp` mencetak bilangan dari 1 hingga N dengan cara yang serupa, namun dimulai dari 2 untuk menghindari duplikasi angka 1. Pada fungsi `main`, pengguna diminta untuk memasukkan nilai N yang menentukan batas atas dan bawah dari barisan bilangan. Setelah itu, kedua fungsi rekursif dipanggil secara berurutan untuk menghasilkan output yang diinginkan.

Soal Studi Case 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.

Source Code

```
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
package main

import "fmt"

// Fungsi rekursif untuk mencetak bilangan ganjil dari 1 hingga N
func printOdd(n, current int) {
    if current > n {
        return
    }
    fmt.Print(current, " ")
    printOdd(n, current+2) // Melompat ke bilangan ganjil berikutnya
}

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

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

Deskripsi Program

Program ini dirancang untuk mencetak semua bilangan ganjil dari 1 hingga N secara rekursif. Fungsi `printOdd` menerima dua parameter: `n` sebagai batas atas dan `current` sebagai bilangan ganjil yang sedang dicetak. Fungsi ini akan terus memanggil dirinya sendiri dengan nilai `current` yang ditambah 2 (agar selalu mendapatkan bilangan ganjil berikutnya) hingga `current` melebihi `n`. Pada fungsi `main`, pengguna diminta untuk memasukkan nilai `N`, kemudian fungsi `printOdd` dipanggil untuk mencetak barisan bilangan ganjil sesuai dengan nilai `N` yang diinputkan. Dengan demikian, program ini memberikan cara yang efisien untuk menghasilkan barisan bilangan ganjil menggunakan konsep rekursif.

Soal Studi Case 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".

Source Code

```
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
package main

import (
    "fmt"
)

// Fungsi rekursif untuk menghitung x^y
func power(x, y int) int {
    if y == 0 {
        return 1
    }
    return x * power(x, y-1)
}

func main() {
    var x, y int

    // Meminta input dari user
    fmt.Print("Masukkan bilangan x: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan bilangan y: ")
    fmt.Scan(&y)

    // Menghitung hasil pangkat dan menampilkan
    result := power(x, y)
    fmt.Printf("Hasil %d pangkat %d adalah %d\n", x, y, result)
}
```

Screenshot Program

```
PS C:\Users\MSI GAMING> go run "c:\Users\MSI GAMING\Documents\TELKOM UNIVERS  
ITY\SEMESTER 3\PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 2\Pertemuan Ke 6\tempCode  
RunnerFile.go"  
Masukkan bilangan x: 10  
Masukkan bilangan y: 2  
Hasil 10 pangkat 2 adalah 100  
PS C:\Users\MSI GAMING> █
```

Deskripsi Program

Program ini dirancang untuk menghitung nilai pangkat (x^y) secara rekursif. Fungsi power menerima dua parameter, yaitu bilangan pokok x dan pangkat y . Fungsi ini bekerja dengan prinsip bahwa x^y sama dengan $x * x^{(y-1)}$. Jika y bernilai 0, maka hasilnya adalah 1. Dengan demikian, fungsi power akan terus memanggil dirinya sendiri dengan nilai y yang semakin kecil hingga mencapai kondisi dasar y sama dengan 0. Pada fungsi main, program meminta pengguna untuk memasukkan nilai x dan y , kemudian memanggil fungsi power untuk menghitung hasil pangkat dan menampilkannya ke layar.