

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VI
REKURSIF**



Disusun Oleh :

Liya Khoirunnisa / 2311102124

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

Algoritma rekursif adalah sebuah fungsi, prosedur, metode yang memanggil dirinya sendiri dan bentuk dimana pemanggilan subrutin terdapat dalam body subrutin, seperti halnya struktur pengulangan seperti perulangan for, perulangan while, repeat...until dan lain-lain yang memiliki kondisi awal dan akhir pengeksekusian. Rekursif adalah sebuah teknik pemecahan masalah yang efisien untuk dipakai dalam definisi dan algoritma, yang mana dalam pemakaiannya hanya melakukan sekali pemanggilan. Tetapi akan menjadi sangat tidak efisien jika memanggil dua atau lebih pemanggilan karena akan diperoleh kompleksitas waktu yang eksponensial.

Algoritma rekursif memiliki kondisi yaitu base case dan end case, Base case menunjukkan batas bawah berapa kali algoritma rekursif akan memanggil dirinya sendiri. Serta end case adalah kondisi berhentinya atau jumlah maksimal inputan yang akan diproses. Selain syarat kondisi basis diatas subrutin call yang melibatkan parameter yang nilainya menuju kondisi terminal (recurrence) harus terpenuhi. Kita dapat mengenali apakah sebuah fungsi itu rekursif atau tidak dengan melihat di source code, jika ini sebuah fungsi atau prosedur yang memanggil dirinya sendiri maka ini adalah rekursif. Bentuk rekursif juga bertujuan untuk menyederhanakan penulisan program dan menggantikan bentuk iterasi.

Rekursif sangat berarti karena banyak algoritma perlu untuk diimplementasikan secara rekursif seperti kombinatorial dengan algoritma brute force, backtracking, fibonacci dan lain-lain. Selain itu dengan rekursif program akan lebih mudah untuk dilihat. Fungsi rekursif selalu memiliki kondisi yang menyatakan kapan fungsi tersebut berhenti. Kondisi ini harus dapat dibuktikan akan tercapai, karena jika tidak tercapai maka kita tidak akan dapat membuktikan bahwa fungsi akan berhenti, yang berarti algoritma tidak benar.

II. GUIDED

1. Membuat baris bilangan dari n hingga 1

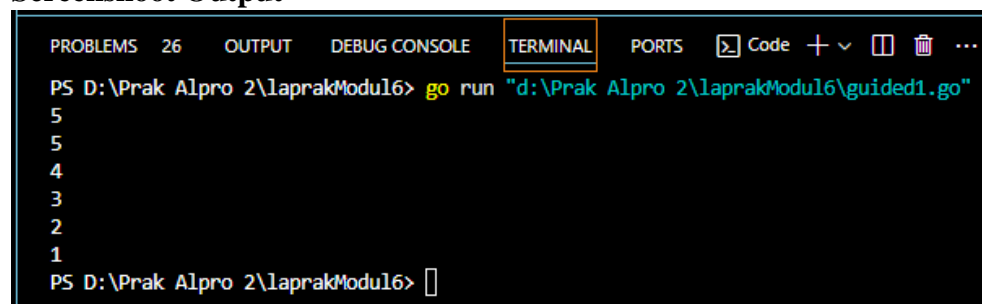
Source code

```
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



```
PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + v [] [] ...
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided1.go"
5
5
4
3
2
1
PS D:\Prak Alpro 2\laprakModul6> []
```

Deskripsi Program

Program di atas dibuat untuk mencetak deret angka dari bilangan yang diinputkan oleh pengguna hingga angka 1, menggunakan algoritma rekursif. Program diawali dengan memasukkan nilai int n. Kemudian memanggil fungsi baris dengan parameter n. Fungsi baris berguna untuk memeriksa apakah nilai bilangan sama dengan 1. Jika bilangan = 1, maka program mencetak angka 1. Namun jika bilangan bukan = 1, maka program akan mencetak bilangan saat ini, lalu memanggil kembali fungsi baris dengan nilai bilangan dikurangi 1. Proses akan berulang hingga mencapai 1. Output yang dihasilkan adalah deret angka mulai dari n hingga 1 urutan menurun

***Note:** Pada terminal terdapat 26 problem dikarenakan package dan func main digunakan juga di file lain

2. Menghitung hasil penjumlahan 1 hingga n

Source code

```
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



```
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided2.go"
2
3
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided2.go"
1
1
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided2.go"
2
3
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided2.go"
4
10
PS D:\Prak Alpro 2\laprakModul6> 
```

Deskripsi Program

Program di atas dibuat untuk menghitung penjumlahan bilangan bulat dari 1 hingga angka n yang dimasukkan pengguna menggunakan algoritma rekursif. Terdapat fungsi penjumlahan yang menerima input int n. Jika n bernilai = 1 maka fungsi akan mengembalikan nilai 1. Jika n bukan bernilai = 1, maka fungsi menambahkan n dengan pemanggilan fungsi penjumlahan (n-1) untuk melakukan penjumlahan secara rekursif hingga mencapai 1. Pada program utama, pengguna diminta untuk menginputkan

angka n , kemudian program memanggil fungsi penjumlahan untuk menghitung dan mencetak hasil.

***Note:** Pada terminal terdapat 22 problem dikarenakan package dan func main digunakan juga di file lain

3. Mencari dua pangkat n atau 2^n

Source code

```
package main
import "fmt"

// Fungsi rekursif untuk menghitung  $2^n$ 
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.Scan(&n)
    fmt.Println("Hasil dari 2 pangkat", n, "adalah: ",
    pangkat(n))
}
```

Screenshoot Output



```
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided3.go"
Masukkan nilai n: 1
Hasil dari 2 pangkat 1 adalah: 2
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided3.go"
Masukkan nilai n: 2
Hasil dari 2 pangkat 2 adalah: 4
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided3.go"
Masukkan nilai n: 4
Hasil dari 2 pangkat 4 adalah: 16
PS D:\Prak Alpro 2\laprakModul6>
```

Deskripsi Program

Program di atas dibuat untuk menghitung nilai 2^n menggunakan algoritma rekursif. Terdapat fungsi pangkat yang menerima input int n . Jika n bernilai $= 0$ maka fungsi akan mengembalikan nilai 1. Jika n bukan bernilai $= 0$, maka fungsi akan melakukan pemanggilan fungsi pangkat ($n-1$) dikali dengan 2 untuk melakukan perkalian secara rekursif hingga n

mencapai 0. Pada program utama, pengguna diminta untuk memasukkan angka n , kemudian program memanggil fungsi pangkat(n) untuk menghitung dan mencetak hasil.

***Note:** Pada terminal terdapat 22 problem dikarenakan package dan func main digunakan juga di file lain.

4. 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



```
PROBLEMS 22 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + v [] [] ...
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided4.go"
1
1
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided4.go"
2
2
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\guided4.go"
4
24
PS D:\Prak Alpro 2\laprakModul6> []
```

Deskripsi Program

Program di atas dibuat untuk menghitung nilai faktorial dari suatu bilangan bulat positif n menggunakan algoritma rekursif. Terdapat fungsi pangkat yang menerima input int n . Jika n bernilai $= 0$ atau $= 1$ maka fungsi akan mengembalikan nilai 1. Jika n bukan bernilai $= 0$ atau $= 1$, maka fungsi akan melakukan pemanggilan fungsi faktorial ($n-1$) dikali dengan n untuk melakukan perkalian secara rekursif hingga n mencapai 0 atau 1. Pada program utama, pengguna diminta untuk memasukkan angka

n, kemudian program memanggil fungsi faktorial(n) untuk menghitung dan mencetak hasil.

***Note:** Pada terminal terdapat 40 problem dikarenakan package dan func main digunakan juga di file lain

III. UNGUIDED

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 tersebut.

Source code

```
/* Liya Khoirunnisa - 2311102124 */
package main

import "fmt"

// Fungsi untuk menghitung suku Fibonacci ke-n
func fibonacci(n int) int {
    if n <= 1 {
        return n // jika n adalah 0 atau 1, kembalikan n
    }
    return fibonacci(n-1) + fibonacci(n-2) // algoritma
    rekursif
}

func main() {
    // deklarasi variabel
    var n int

    // Input jumlah suku fibonacci dari pengguna
    fmt.Print("Masukkan jumlah suku Fibonacci: ")
    fmt.Scan(&n)

    // Memastikan input positif
    if n < 0 {
        fmt.Println("Masukkan angka positif.")
        return
    }

    // Mencetak deret fibonacci
    fmt.Print("Deret Fibonacci: ")
    for i := 0; i < n; i++ {
        fmt.Print(fibonacci(i), " ")
    }
    fmt.Println()
}
```

Screenshoot Output



```
PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [] ... ^

PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided1.go"
Masukkan jumlah suku Fibonacci: 1
Deret Fibonacci: 0
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided1.go"
Masukkan jumlah suku Fibonacci: 2
Deret Fibonacci: 0 1
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided1.go"
Masukkan jumlah suku Fibonacci: 4
Deret Fibonacci: 0 1 1 2
PS D:\Prak Alpro 2\laprakModul6> 
```

Deskripsi Program

Program di atas dibuat untuk mencetak deret fibonacci hingga jumlah suku yang diminta oleh pengguna. Untuk menghitung nilai suku ke-n dalam deret menggunakan algoritma rekursif. Terdapat fungsi fibonacci yang menerima input int n. Jika n bernilai ≤ 1 maka fungsi akan mengembalikan nilai n. Jika n bukan bernilai ≤ 1 , maka fungsi akan menjumlahkan dua suku sebelumnya. Pada program utama, pengguna diminta untuk menginputkan jumlah suku fibonacci. Jika pengguna memasukkan angka negatif maka program akan memberikan pesan untuk memasukkan angka positif. Jika inputan valid (positif) maka program akan memanggil fungsi fibonacci berulang dari suku ke-0 hingga suku ke-(n-1). Kemudian program akan mencetak deret fibonacci.

***Note:** Pada terminal terdapat 18 problem dikarenakan package dan func main digunakan juga di file lain

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

Source code

```
/* Liya Khoirunnisa - 2311102124 */
package main

import (
    "fmt"
)

// Fungsi untuk mencetak pola bintang
func cetakBintang(n int, i int) {
    if i > n {
        return
    }

    // Perulangan mencetak bintang sebanyak i
```

```

for j := 0; j < i; j++ {
    fmt.Print("* ")
}
fmt.Println()

// Panggil fungsi rekursif untuk baris berikutnya
cetakBintang(n, i+1)
}

func main() {
    // Deklarasi variabel
    var n int

    // Input n
    fmt.Print("Masukkan n: ")
    fmt.Scan(&n)

    // Cetak pola bintang
    fmt.Println("Pola bintang:")
    cetakBintang(n, 1)
}

```

Screenshot Output

```
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided2.go"
Masukkan n: 1
Pola bintang:
*

PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided2.go"
Masukkan n: 2
Pola bintang:
* *

PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided2.go"
Masukkan n: 4
Pola bintang:
* *
* * *
* * * *
```

Deskripsi Program

Program di atas dibuat untuk mencetak pola bintang segitiga. Program menggunakan algoritma rekursif. Program diawali dengan meminta input berupa jumlah baris pola bintang yang diinginkan. Fungsi cetakbintang akan menjalankan pencetakan bintang dengan menerima input int n. Jika n bernilai ≤ 1 maka fungsi akan mengembalikan nilai n. Fungsi akan mencetak bintang sesuai dengan nilai yang diminta, kemudian mencetak baris berikutnya dengan memanggil dirinya sendiri secara rekursif dengan nilai i yang bertambah 1. Proses akan berhenti ketika nilai i lebih besar

dari n. Lalu program akan mencetak pola bintang dari baris pertama dan bertambah satu bintang hingga mencapai jumlah sesuai inputan.

***Note:** Pada terminal terdapat 18 problem dikarenakan package dan func main digunakan juga di file lain

3. Buatlah program yang mengimplementasikan rekursif untuk menampilkan faktor bilangan dari suatu N, atau bilangan apa saja yang habis membagi N. Masukkan terdiri dari sebuah bilangan bulat positif N. Keluaran terdiri dari barisan bilangan yang menjadi faktor dari N (terurut dari 1 hingga N).

Source code

```
/* Liya Khoirunnisa - 2311102124 */

package main

import (
    "fmt"
)

// Fungsi untuk menampilkan faktor dari N
func faktor(n int, i int) {
    // Jika i lebih besar dari n, maka keluar dari fungsi
    if i > n {
        return
    }

    // Jika n habis dibagi i, maka cetak faktor
    if n%i == 0 {
        fmt.Print(i)
    }

    // Memanggil fungsi rekursif
    faktor(n, i+1)
}

func main() {
    // Deklarasi variabel
    var N int

    // Input n
    fmt.Print("Masukkan bilangan n: ")
    fmt.Scan(&N)

    // Jika n negatif
    if N <= 0 {
        fmt.Println("Silakan masukkan bilangan bulat positif.")
        return
    }

    // Mencetak faktor ke layar
```

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

Screenshoot Output

Deskripsi Program

Program di atas dibuat untuk menampilkan faktor sebuah bilangan bulat positif yang diinputkan pengguna. Algoritma yang digunakan menggunakan algoritma rekursif untuk mengecek apakah suatu angka merupakan faktor dari bilangan yang diberikan. Fungsi faktor memeriksa, jika $i > n$ maka angka dicetak sebagai faktor. Fungsi akan dipanggil kembali dengan nilai i bertambah 1 sampai $i > n$ untuk menghentikan proses pencarian faktor. Jika pengguna memasukkan bilangan negatif atau nol, maka akan menampilkan pesan untuk memasukkan bilangan bulat positif. Kemudian hasil dicetak ke layar

***Note:** Pada terminal terdapat 18 problem dikarenakan package dan func main digunakan juga di file lain

4. Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu. Masukkan terdiri dari sebuah bilangan bulat positif N . Keluaran terdiri dari barisan bilangan dari N hingga 1 dan kembali ke N .

Source code

```
/* Liya Khoirunnisa - 2311102124 */
package main

import (
    "fmt"
)

// Fungsi untuk menampilkan bilangan dari N hingga 1
func cetakBilangan(n int) {
```

```

        if n < 1 {
            return
        }
        fmt.Print(n, " ") // Menampilkan bilangan N
        cetakBilangan(n - 1) // Panggil fungsi secara rekursif
        fmt.Print(n, " ") // Menampilkan bilangan N setelah
        kembali dari rekursif
    }

func main() {
    // Deklarasi variabel
    var N int

    // Input N dari pengguna
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)

    // Jika N <= 0
    if N <= 0 {
        fmt.Println("Silakan masukkan bilangan bulat
        positif.")
        return
    }

    // Jika N sudah sesuai maka tampilkan barisan bilangan
    fmt.Print("Barisan bilangan: ")
    cetakBilangan(N)
}

```

Screenshoot Output

```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [ ] [ ] ... ^
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided4.go"
Masukkan bilangan bulat positif N: 1
Barisan bilangan: 1 1
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided4.go"
Masukkan bilangan bulat positif N: 2
Barisan bilangan: 2 1 1 2
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided4.go"
Masukkan bilangan bulat positif N: 4
Barisan bilangan: 4 3 2 1 1 2 3 4
PS D:\Prak Alpro 2\laprakModul6>

```

Deskripsi Program

Program di atas dibuat untuk menampilkan baris bilangan dari N hingga 1 secara menurun, kemudian menampilkan bilangan dari 1 hingga N secara menaik menggunakan algoritma rekursif. Program dimulai dengan meminta input bilangan bulat positif. Jika bilangan $N \leq 0$, maka program akan meminta pengguna untuk menginputkan bilangan positif. Fungsi `cetakBilangan` ditampilkan secara berulang hingga mencapai $N < 1$. Ketika fungsi dipanggil lagi dengan nilai $N-1$ hingga nilai N mencapai 1. Setelah

mencapai 1, bilangan akan dicetak kembali. Hasil urutan bilangan akan ditampilkan dari N hingga 1 dan kembali dari 1 hingga N

***Note:** Pada terminal terdapat 18 problem dikarenakan package dan func main digunakan juga di file lain

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

```
/* Liya Khoirunnisa - 2311102124*/
package main

import (
    "fmt"
)

// Fungsi untuk menampilkan bilangan ganjil
func bilanganGanjil(n int) {
    // Jika n kurang dari 1, hentikan rekursif
    if n < 1 {
        return
    }
    // Panggil fungsi untuk menampilkan bilangan ganjil
    sebelumnya
    bilanganGanjil(n - 1)

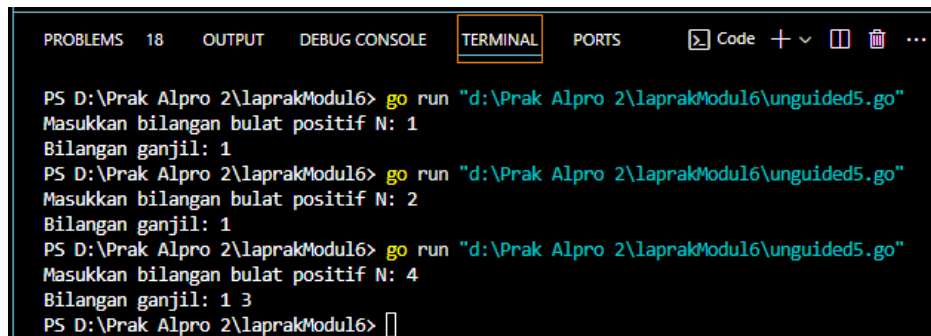
    // Jika n adalah bilangan ganjil, cetak n
    if n%2 != 0 {
        fmt.Print(n, " ")
    }
}

func main() {
    // Deklarasi variabel
    var N int

    // Input dari pengguna
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&N)

    // Memanggil fungsi untuk menampilkan bilangan ganjil
    fmt.Print("Bilangan ganjil: ")
    bilanganGanjil(N)
}
```

Screenshoot Output



```
PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [] ...

PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided5.go"
Masukkan bilangan bulat positif N: 1
Bilangan ganjil: 1
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided5.go"
Masukkan bilangan bulat positif N: 2
Bilangan ganjil: 1
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided5.go"
Masukkan bilangan bulat positif N: 4
Bilangan ganjil: 1 3
PS D:\Prak Alpro 2\laprakModul6> 
```

Deskripsi Program

Program di atas dibuat untuk menampilkan semua bilangan ganjil dari 1 hingga bilangan positif N menggunakan algoritma rekursif. Terdapat fungsi `bilanganGanjil` dengan menggunakan parameter `n`. Fungsi akan mengurangi nilai `n` sebesar 1 setiap pemanggilan, hingga mencapai 1. Jika `n` adalah ganjil dengan diperiksa `n%2 != 0` maka `n` akan dicetak ke layar. Proses akan berlanjut hingga semua bilangan ganjil dari 1 hingga `n` ditampilkan.

***Note:** Pada terminal terdapat 18 problem dikarenakan package dan func main digunakan juga di file lain

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

```
/* Liya Khoirunnisa - 2311102124 */
package main

import (
    "fmt"
)

// Fungsi untuk menghitung x dipangkatkan y
func pangkat(x int, y int) int {
    // Jika y = 0, maka hasil 1
    if y == 0 {
        return 1
    }
    // Rekursif
    return x * pangkat(x, y-1)
}

func main() {
    // Deklarasi variabel
```

```

var x, y int

// Input dari pengguna
fmt.Print("Masukkan bilangan bulat x: ")
fmt.Scan(&x)
fmt.Print("Masukkan bilangan bulat y: ")
fmt.Scan(&y)

// Menghitung hasil
hasil := pangkat(x, y)

// Menampilkan hasil
fmt.Printf("Hasil %d pangkat %d adalah %d\n", x, y,
hasil)
}

```

Screenshoot Output

```

PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided6.go"
Masukkan bilangan bulat x: 2
Masukkan bilangan bulat y: 1
Hasil 2 pangkat 1 adalah 2
PS D:\Prak Alpro 2\laprakModul6> go run "d:\Prak Alpro 2\laprakModul6\unguided6.go"
Masukkan bilangan bulat x: 2
Masukkan bilangan bulat y: 4
Hasil 2 pangkat 4 adalah 16
PS D:\Prak Alpro 2\laprakModul6>

```

Deskripsi Program

Program di atas dibuat untuk menghitung hasil perpangkatan dari dua bilangan bulat yang dimasukkan oleh pengguna menggunakan algoritma rekursif. Program menghitung dari inputan x dan y dari pengguna. Fungsi pangkat (x,y) memeriksa apakah nilai y adalah 0. Jika y adalah 0, maka fungsi akan mengembalikan nilai 1. Jika y bukan 0 maka fungsi akan memanggil dirinya sendiri dengan mengurangi nilai y sebesar 1 dan mengalikan hasil dengan x hingga y mencapai 0. Setelah itu, program akan menghitung hasil perpangkatan dengan memanggil fungsi pangkat dan hasil akan dicetak ke layar.

***Note:** Pada terminal terdapat 22 problem dikarenakan package dan func main digunakan juga di file lain

DAFTAR PUSTAKA

- [1] Putra, P. A., Shandi Dani, K., & Muhammad Aditya, F. Analisis Algoritma Nonrekursif pada Angka Fibonacci.
- [2] Putro, C. D., Kurniawati, A., & Angkoso, C. V. IMPLEMENTASI ALGORITMA RECURSIVE DFS UNTUK AGEN CERDAS PADA GAME EDUKASI MATEMATIKA DASAR BERBASIS ANDROID ACCELEROMETER. *SENASTIK* 2014, 219.