

LAPORAN PRAKTIKUM
Algoritma Pemrograman
MODUL V
REKURSIF



Oleh:

Ilhan Sahal Mansiz

2311102029

IF-11-02

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. DASAR TEORI

Rekursi adalah sebuah teknik pemrograman di mana sebuah fungsi memanggil dirinya sendiri secara langsung atau tidak langsung. Rekursi sering digunakan untuk menyelesaikan masalah yang dapat dipecah menjadi submasalah yang lebih kecil, di mana solusi dari submasalah tersebut membantu menyelesaikan masalah aslinya.

Fungsi Rekursif

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri dalam proses eksekusinya. Fungsi rekursif biasanya terdiri dari dua komponen utama:

1. **Kondisi dasar (base case):** Kondisi yang digunakan untuk menghentikan rekursi. Kondisi ini mencegah fungsi terus memanggil dirinya sendiri tanpa henti.
2. **Panggilan rekursif:** Panggilan kembali fungsi itu sendiri dengan parameter yang telah dimodifikasi, mendekati kondisi dasar.

```
package main
import "fmt"

// Fungsi rekursif untuk menghitung faktorial
func factorial(n int) int {
    if n == 0 {
        return 1 // Kondisi dasar
    }
    return n * factorial(n-1) // Panggilan rekursif
}

func main() {
    number := 5
    fmt.Println("Faktorial dari", number, "adalah",
        factorial(number))
}
```

Pada contoh di atas:

- Jika $n == 0$, fungsi akan berhenti dan mengembalikan nilai 1 sebagai kondisi dasar.
- Jika $n != 0$, fungsi akan memanggil dirinya sendiri dengan nilai $n-1$ hingga mencapai kondisi dasar.

Manfaat dan Kelemahan Rekursi

Manfaat utama dari penggunaan rekursi adalah penyederhanaan kode dalam menyelesaikan masalah yang kompleks atau berulang, seperti algoritma Divide and Conquer (misalnya, QuickSort dan MergeSort), dan traversal struktur data rekursif seperti pohon.

Namun, rekursi juga memiliki kelemahan, seperti:

- **Efisiensi:** Rekursi bisa lebih lambat dibandingkan dengan iterasi karena overhead pemanggilan fungsi berulang kali.
- **Resiko Stack Overflow:** Terlalu banyak panggilan rekursif bisa menyebabkan stack memory penuh, yang dikenal dengan istilah *stack overflow*.

II. GUIDED

1. Source Code :

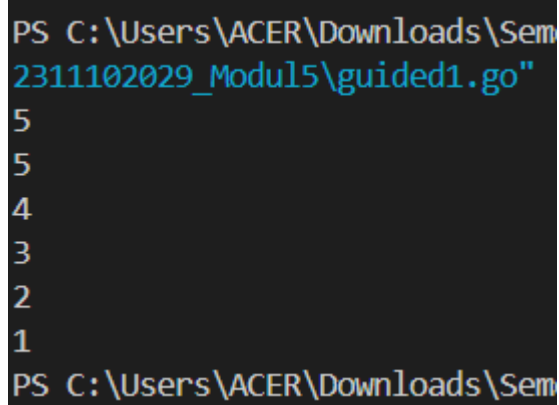
```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n) // Membaca input pengguna
    baris(n) // memanggil fungsi rekursif 'baris'
}

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

Screenshot Program :



```
PS C:\Users\ACER\Downloads\Sem2020\2311102029_Modul5\guided1.go"
5
4
3
2
1
PS C:\Users\ACER\Downloads\Sem2020\2311102029_Modul5\guided1.go"
```

Deskripsi Program :

Program ini membaca input bilangan bulat dari pengguna dan menggunakan fungsi rekursif bernama `baris` untuk mencetak bilangan secara menurun dari bilangan yang diinputkan hingga 1. Pada fungsi `baris`, jika bilangan sama dengan 1 (kondisi dasar), program mencetak angka 1 dan rekursi berhenti. Jika bilangan lebih besar dari 1, program mencetak nilai bilangan saat ini, kemudian memanggil fungsi `baris` dengan argumen bilangan dikurangi 1, sehingga proses berlanjut hingga mencapai kondisi dasar. Program ini menunjukkan contoh sederhana penggunaan rekursi untuk mencetak deret angka menurun.

2. Source Code :

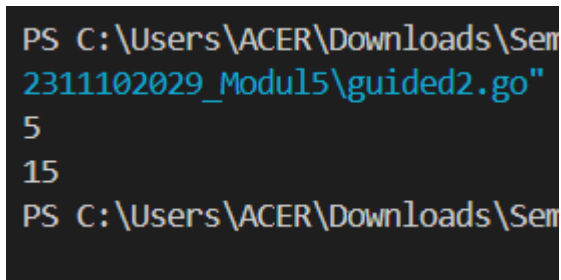
```
package main

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)
    }
}
```

Screenshot Program :



```
PS C:\Users\ACER\Downloads\Sem2311102029_Modul5\guided2.go"
5
15
PS C:\Users\ACER\Downloads\Sem
```

Deskripsi Program :

Program ini menggunakan rekursi untuk menghitung penjumlahan deret bilangan dari 1 hingga bilangan `n` yang diinputkan oleh pengguna. Fungsi `penjumlahan` bekerja dengan cara memeriksa apakah nilai `n` sudah mencapai 1 (kondisi dasar). Jika ya, fungsi mengembalikan nilai 1, menghentikan rekursi. Jika tidak, fungsi menambahkan nilai `n` dengan hasil dari pemanggilan fungsi `penjumlahan` dengan argumen `n-1`. Proses ini berlanjut hingga rekursi mencapai kondisi dasar.

Hasil akhirnya adalah jumlah dari semua bilangan bulat mulai dari 1 hingga `n`, yang kemudian dicetak oleh fungsi `main`.

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 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

Source Code :

```
package main

import (
    "fmt"
)

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



```

func main() {

    var n int

    fmt.Println("Masukkan nilai n: ")

    fmt.Scan(&n)


    fmt.Println("Deret Fibonacci hingga suku ke-", n, ":")

    for i := 0; i <= n; i++ {

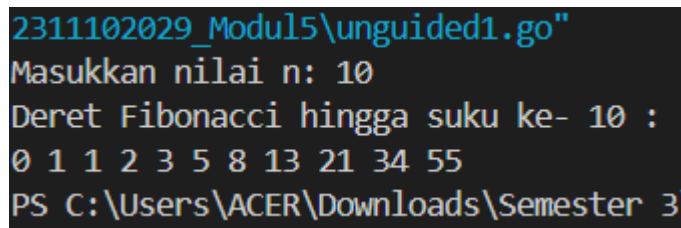
        fmt.Print(fibonacci(i), " ")

    }

}

```

Screenshot Program :



```

2311102029_Modul5\unguided1.go
Masukkan nilai n: 10
Deret Fibonacci hingga suku ke- 10 :
0 1 1 2 3 5 8 13 21 34 55
PS C:\Users\ACER\Downloads\Semester 3

```

Deskripsi Program :

Program ini menggunakan rekursi untuk menghitung dan mencetak deret Fibonacci hingga suku ke-`n`, di mana `n` adalah bilangan yang diinputkan oleh pengguna. Fungsi rekursif `fibonacci` menghitung nilai Fibonacci untuk bilangan tertentu dengan memeriksa dua kondisi dasar: jika `n` adalah 0, fungsi mengembalikan 0, dan jika `n` adalah 1, fungsi mengembalikan 1. Jika tidak, fungsi mengembalikan hasil penjumlahan dari dua nilai Fibonacci sebelumnya, yaitu `fibonacci(n-1)` dan `fibonacci(n-2)`. Pada fungsi `main`, setelah pengguna memasukkan nilai `n`, program mencetak deret Fibonacci mulai dari 0 hingga suku ke-`n` dengan menggunakan perulangan untuk memanggil fungsi `fibonacci` pada setiap nilai dari 0 hingga `n`.

- 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	* ** ***

Source Code :

```
package main

import (
    "fmt"
)

func printStars(n int) {
    if n > 0 {
        printStars(n - 1)
        for i := 0; i < n; i++ {
            fmt.Print("*")
        }
    }
}
```

```

    }

    fmt.Println()

}

}

```

```

func main() {

    var n int

    fmt.Print("Masukkan nilai N: ")

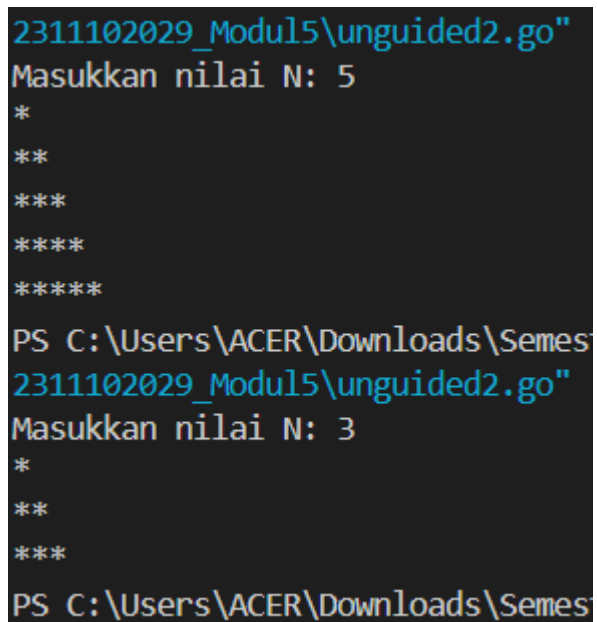
    fmt.Scan(&n)

    printStars(n)

}

```

Screenshot Program :



```

2311102029_Modul5\unguided2.go"
Masukkan nilai N: 5
*
**
***
****
*****

PS C:\Users\ACER\Downloads\Semes
2311102029_Modul5\unguided2.go"
Masukkan nilai N: 3
*
**
***

PS C:\Users\ACER\Downloads\Semes

```

Deskripsi Program :

Program ini menggunakan rekursi untuk mencetak pola bintang (*) bertingkat dari 1 hingga `n` baris, di mana `n` adalah bilangan yang diinputkan oleh pengguna. Fungsi rekursif `printStars` akan mencetak deretan bintang sesuai dengan nilai `n` dalam urutan bertingkat dari yang terkecil hingga terbesar. Jika nilai `n` lebih besar dari 0, fungsi akan memanggil dirinya sendiri dengan `n-1` terlebih dahulu, sehingga memastikan bahwa baris-baris dengan jumlah bintang yang lebih kecil dicetak terlebih dahulu. Setelah rekursi selesai untuk nilai `n`, deretan bintang dengan panjang `n` akan dicetak pada baris berikutnya. Program ini mengilustrasikan pola rekursi "mendalam" yang mengerjakan tugas setelah pemanggilan rekursif.

- 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

Source Code :

```
package main

import (
    "fmt"
)

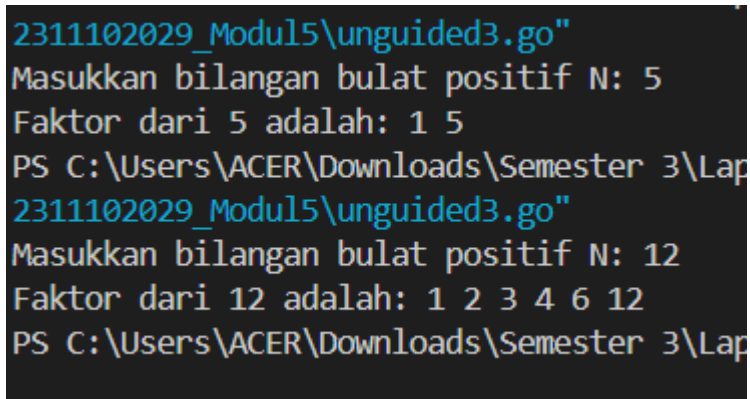
func findFactors(n int, i int) {
    if i > n {
        return
    }
    if n % i == 0 {
        fmt.Printf("%d ", i)
    }
    findFactors(n, i + 1)
}
```

```

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&n)
    fmt.Printf("Faktor dari %d adalah: ", n)
    findFactors(n, 1)
}

```

Screenshot Program :



```

2311102029_Modul5\unguided3.go"
Masukkan bilangan bulat positif N: 5
Faktor dari 5 adalah: 1 5
PS C:\Users\ACER\Downloads\Semester 3\Lap
2311102029_Modul5\unguided3.go"
Masukkan bilangan bulat positif N: 12
Faktor dari 12 adalah: 1 2 3 4 6 12
PS C:\Users\ACER\Downloads\Semester 3\Lap

```

Deskripsi Program :

Program ini menggunakan rekursi untuk menemukan dan mencetak semua faktor dari bilangan bulat positif `n`, yang diinputkan oleh pengguna. Fungsi `findFactors` bekerja dengan memeriksa setiap bilangan dari 1 hingga `n` (parameter `i`) untuk melihat apakah bilangan tersebut merupakan faktor dari `n`. Jika `n` habis dibagi oleh `i`, maka `i` dicetak sebagai faktor. Fungsi kemudian memanggil dirinya sendiri dengan nilai `i+1` hingga mencapai nilai yang lebih besar dari `n`, yang akan menghentikan proses rekursi. Program ini mencetak semua faktor dari bilangan `n` secara berurutan, dimulai dari 1 hingga `n`.

- 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

Source Code :

```
package main

import (
    "fmt"
)

func printSequence(N, current int) {
    fmt.Printf("%d ", current)
    if current > 1 {
        printSequence(N, current - 1)
        fmt.Printf("%d ", current)
    }
}

func main() {
```

```

var N int

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

fmt.Scan(&N)

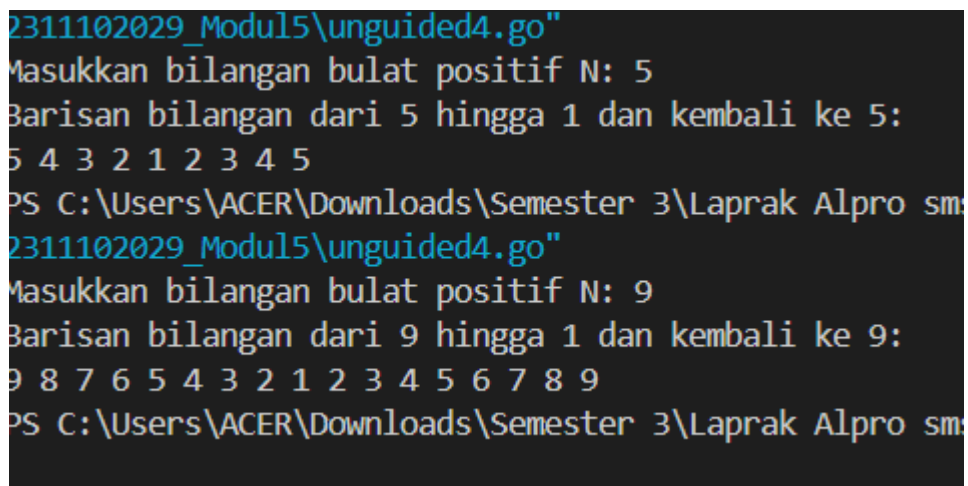
fmt.Printf("Barisan bilangan dari %d hingga 1 dan kembali ke %d:\n",
N, N)

printSequence(N, N)

fmt.Println()
}

```

Screenshot Program :



```

2311102029_Modul5\unguided4.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
PS C:\Users\ACER\Downloads\Semester 3\Laprak Alpro sm
2311102029_Modul5\unguided4.go
Masukkan bilangan bulat positif N: 9
Barisan bilangan dari 9 hingga 1 dan kembali ke 9:
9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS C:\Users\ACER\Downloads\Semester 3\Laprak Alpro sm

```

Deskripsi Program :

Program ini menggunakan rekursi untuk mencetak barisan bilangan dari `N` hingga 1, kemudian kembali lagi ke `N`, di mana `N` adalah bilangan bulat positif yang diinputkan oleh pengguna. Fungsi rekursif `printSequence` mencetak nilai `current`, kemudian memanggil dirinya sendiri dengan `current - 1` jika nilai `current` lebih besar dari 1. Setelah mencapai 1, proses rekursi berhenti, lalu dalam perjalanan kembali, fungsi mencetak lagi nilai `current` untuk menghasilkan efek "bolak-balik" dari `N` ke 1 dan kembali ke `N`. Program ini mencetak barisan tersebut dalam satu baris, dengan angka dipisahkan oleh spasi.

- 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

Source Code :

```
package main

import (
    "fmt"
)

func printOdds(current, N int) {
    if current > N {
        return
    }
    if current % 2 != 0 {
        fmt.Printf("%d ", current)
    }
    printOdds(current + 1, N)
}
```

```

func main() {
    var N int

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

    fmt.Scan(&N)

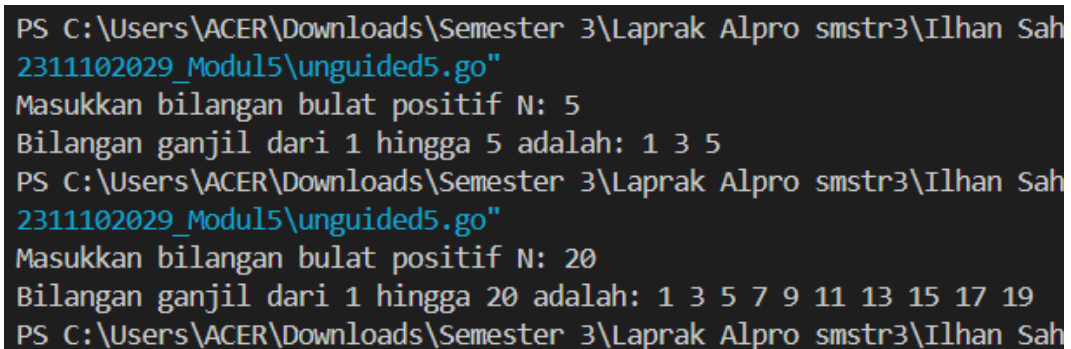
    fmt.Printf("Bilangan ganjil dari 1 hingga %d adalah: ", N)

    printOdds(1, N)

    fmt.Println()
}

```

Screenshot Program :



```

PS C:\Users\ACER\Downloads\Semester 3\Laprak Alpro smstr3\Ilhan Sah
2311102029_Modul5\unguided5.go"
Masukkan bilangan bulat positif N: 5
Bilangan ganjil dari 1 hingga 5 adalah: 1 3 5
PS C:\Users\ACER\Downloads\Semester 3\Laprak Alpro smstr3\Ilhan Sah
2311102029_Modul5\unguided5.go"
Masukkan bilangan bulat positif N: 20
Bilangan ganjil dari 1 hingga 20 adalah: 1 3 5 7 9 11 13 15 17 19
PS C:\Users\ACER\Downloads\Semester 3\Laprak Alpro smstr3\Ilhan Sah

```

Deskripsi Program :

Program ini menggunakan rekursi untuk mencetak semua bilangan ganjil dari 1 hingga `N`, di mana `N` adalah bilangan bulat positif yang diinputkan oleh pengguna. Fungsi `printOdds` memulai dengan bilangan `current`, yang awalnya bernilai 1, dan secara bertahap meningkatkannya satu per satu hingga mencapai `N`. Jika nilai `current` adalah bilangan ganjil (dengan memeriksa sisa hasil bagi 2), program mencetaknya. Fungsi ini terus memanggil dirinya sendiri dengan nilai `current + 1` sampai nilai `current` melebihi `N`, yang akan menghentikan rekursi.

Hasilnya adalah semua bilangan ganjil dari 1 hingga `N` dicetak dalam satu baris, dipisahkan oleh spasi.

- 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

Source Code :

```
package main

import (
    "fmt"
)

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

```

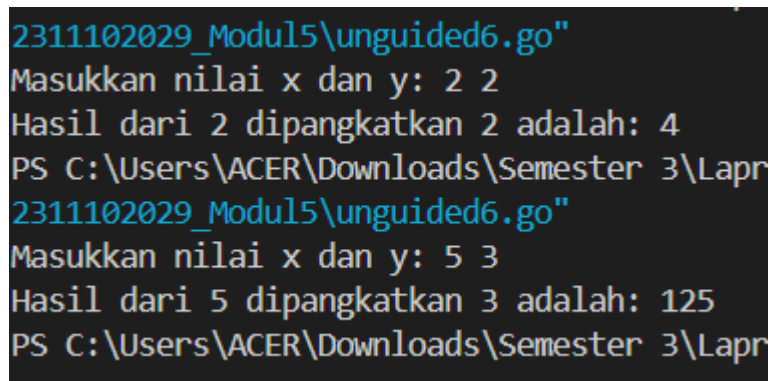
    return x * power(x, y-1)
}

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

    hasil := power(x, y)
    fmt.Printf("Hasil dari %d dipangkatkan %d adalah: %d\n", x, y, hasil)
}

```

Screenshot Program :



```

2311102029_Modul5\unguided6.go"
Masukkan nilai x dan y: 2 2
Hasil dari 2 dipangkatkan 2 adalah: 4
PS C:\Users\ACER\Downloads\Semester 3\Lapr
2311102029_Modul5\unguided6.go"
Masukkan nilai x dan y: 5 3
Hasil dari 5 dipangkatkan 3 adalah: 125
PS C:\Users\ACER\Downloads\Semester 3\Lapr

```

Deskripsi Program :

Program ini menggunakan rekursi untuk menghitung hasil perpangkatan dari bilangan `x` dengan pangkat `y`, di mana kedua nilai `x` dan `y` diinputkan oleh pengguna. Fungsi rekursif `power` bekerja dengan memeriksa kondisi dasar ketika

`y == 0`, yang mengembalikan nilai 1 karena setiap bilangan yang dipangkatkan 0 adalah 1. Jika `y` lebih besar dari 0, fungsi mengalikan `x` dengan hasil panggilan rekursif dari `power(x, y-1)` hingga pangkat `y` mencapai 0, sehingga menghasilkan hasil akhir `x` pangkat `y`. Program kemudian mencetak hasil perpangkatan tersebut dengan format yang sesuai.