

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V
REKURSIF**



**Disusun Oleh :
PETRA PRIADI S.P GINTING (2311102273)
IF-06**

**Dosen Pengampu :
ABEDNEGO DWI SEPTIADI**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

Apa itu Rekursif?

Rekursif adalah sebuah teknik pemrograman di mana sebuah fungsi memanggil dirinya sendiri. Ini seperti cermin yang memantulkan cahaya ke cermin lain, menciptakan efek berulang. Dalam konteks pemrograman, rekursif digunakan untuk memecahkan masalah yang kompleks menjadi masalah-masalah yang lebih kecil dan serupa.

Bagaimana Cara Kerja Rekursif?

1. **Kasus Dasar (Base Case):** Setiap fungsi rekursif harus memiliki setidaknya satu kondisi yang menghentikan pemanggilan berulang. Ini disebut kasus dasar. Jika kondisi ini terpenuhi, fungsi tidak akan memanggil dirinya sendiri lagi.
2. **Kasus Rekursif:** Ini adalah bagian di mana fungsi memanggil dirinya sendiri dengan input yang berbeda. Input ini biasanya lebih sederhana atau lebih dekat ke kasus dasar.

Mengapa Menggunakan Rekursif?

- **Memudahkan Pemahaman:** Untuk beberapa masalah, solusi rekursif lebih intuitif dan mudah dipahami dibandingkan solusi iteratif (menggunakan loop).
- **Struktur Data Rekursif:** Struktur data seperti tree dan linked list secara alami cocok untuk diproses secara rekursif.
- **Problem-Problem Tertentu:** Beberapa masalah, seperti traversal tree, pencarian biner, dan perhitungan faktorial, seringkali lebih mudah diselesaikan dengan rekursif.

Kapan Harus Menggunakan Rekursif?

- **Masalah Dapat Dipecah Menjadi Submasalah yang Lebih Kecil:** Jika masalah dapat dipecah menjadi submasalah yang lebih kecil dan identik, rekursif bisa menjadi pilihan yang baik.
- **Struktur Data Mendukung Rekursif:** Jika struktur data yang digunakan memiliki sifat rekursif (misalnya, tree), rekursif bisa sangat efektif.

- **Kode Lebih Ringkas:** Dalam beberapa kasus, solusi rekursif bisa menghasilkan kode yang lebih ringkas dan mudah dibaca.

Kekurangan Rekursif

- **Performa:** Rekursif bisa kurang efisien dibandingkan solusi iteratif karena overhead pemanggilan fungsi.
- **Stack Overflow:** Jika kasus dasar tidak tercapai atau tidak terdefinisi dengan baik, rekursif bisa menyebabkan stack overflow.
- **Sulit Dibaca:** Untuk programmer pemula, rekursif bisa sulit dipahami dan didebug.

Konsep Penting dalam Rekursif

- **Stack Frame:** Setiap kali sebuah fungsi dipanggil, sistem operasi akan mengalokasikan sebuah blok memori yang disebut stack frame. Stack frame ini digunakan untuk menyimpan informasi lokal fungsi, seperti variabel lokal dan alamat pengembalian.
- **Depth of Recursion:** Ini adalah jumlah maksimum tingkat pemanggilan fungsi rekursif.
- **Tail Recursion:** Ini adalah jenis rekursif di mana pemanggilan rekursif adalah hal terakhir yang dilakukan dalam fungsi. Beberapa compiler dapat mengoptimalkan tail recursion menjadi loop, sehingga menghindari masalah stack overflow.

Kesimpulan

Rekursif adalah alat yang kuat dalam pemrograman, tetapi harus digunakan dengan bijak. Pemahaman yang baik tentang kasus dasar, kasus rekursif, dan potensi masalah seperti stack overflow sangat penting. Meskipun tidak selalu lebih efisien dibandingkan solusi iteratif, rekursif sering kali memberikan cara yang elegan untuk menyelesaikan masalah tertentu.

II. GUIDED

1. Sourcecode

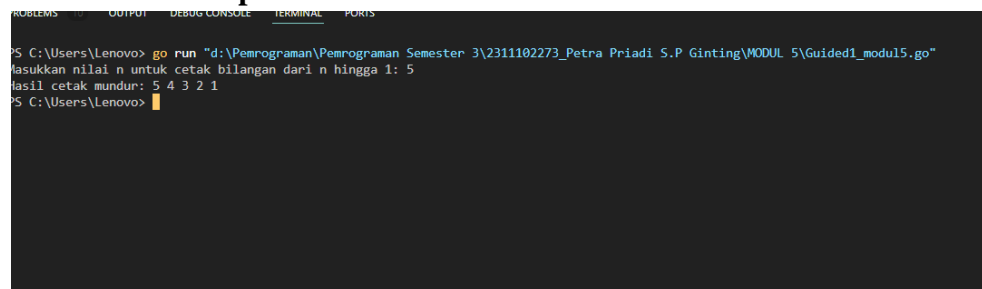
```
package main

import "fmt"

//fungsi untuk mencetak bilangan dari n hingga 1
func cetakMundur(n int) {
    if n == 1 {
        fmt.Println(n)
        return
    }
    fmt.Print(n, " ")
    cetakMundur(n - 1)
}

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

Screenshoot Output



```
PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\2311102273_Petra Priadi S.P Ginting\MODUL 5\Guided1_modul5.go"
Masukkan nilai n untuk cetak bilangan dari n hingga 1: 5
Hasil cetak mundur: 5 4 3 2 1
PS C:\Users\Lenovo>
```

Deskripsi Program

Fungsi cetakMundur yang menerima satu argumen berupa bilangan bulat n . Fungsi ini bekerja secara rekursif, artinya fungsi memanggil dirinya sendiri di dalam tubuhnya. Ketika fungsi dipanggil, ia akan mencetak nilai n saat ini, kemudian memanggil dirinya sendiri dengan nilai $n-1$. Proses ini berulang terus hingga nilai n mencapai 1. Kondisi $\text{if } n == 1$ berfungsi sebagai basis rekursi, yaitu titik berhenti dari pemanggilan fungsi secara berulang.

2. Sourcecode

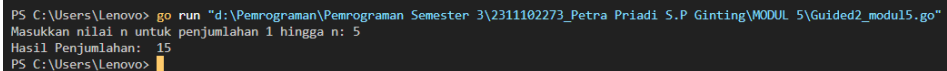
```
package main

import "fmt"

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

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk penjumlahan 1
    hingga n: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil Penjumlahan: ",
    jumlahRekursif(n))
}
```

Screenshoot Output



```
PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\2311102273_Petra Priadi S.P Ginting\MODUL 5\Guided2_modul5.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 5
Hasil Penjumlahan: 15
PS C:\Users\Lenovo>
```

Deskripsi Program

Fungsi jumlahRekursif yang menerima input berupa bilangan bulat n. Fungsi ini bekerja dengan cara memanggil dirinya sendiri dengan nilai n-1 hingga mencapai kondisi dasar ketika n bernilai 1. Pada setiap pemanggilan, nilai n saat itu akan ditambahkan ke hasil pemanggilan fungsi sebelumnya. Proses ini berulang hingga mencapai nilai 1, lalu semua hasil penjumlahan akan dikembalikan. Dalam fungsi main, program meminta pengguna untuk memasukkan nilai n, kemudian memanggil fungsi jumlahRekursif dan mencetak hasilnya.

3. Sourcecode

```
package main

import "fmt"

func pangkatDua(n int) int {

    if n == 0 {

        return 1

    }

    return 2 * pangkatDua(n-1)

}

func main() {

    var n int

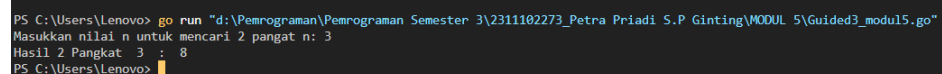
    fmt.Print("Masukkan nilai n untuk mencari 2 pangkat  
n: ")

    fmt.Scanln(&n)

    fmt.Println("Hasil 2 Pangkat ", n, " : ",  
pangkatDua(n))

}
```

Screenshoot Output



```
PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\2311102273_Petra Priadi S.P Ginting\MODUL 5\Guided3_modul5.go"
Masukkan nilai n untuk mencari 2 pangkat n: 3
Hasil 2 Pangkat 3 : 8
PS C:\Users\Lenovo>
```

Deskripsi Program

Fungsi pangkatDua menerima input berupa bilangan bulat n dan secara berulang mengalikan 2 dengan hasil pemanggilan fungsi itu sendiri dengan nilai n-1. Proses ini berulang hingga n mencapai 0, di mana fungsi akan mengembalikan nilai 1. Dalam fungsi main, program meminta pengguna untuk memasukkan nilai n, kemudian memanggil fungsi pangkatDua dan mencetak hasilnya.

4. Sourcecode

```
package main

import "fmt"

func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }

    return n * faktorial(n-1)
}

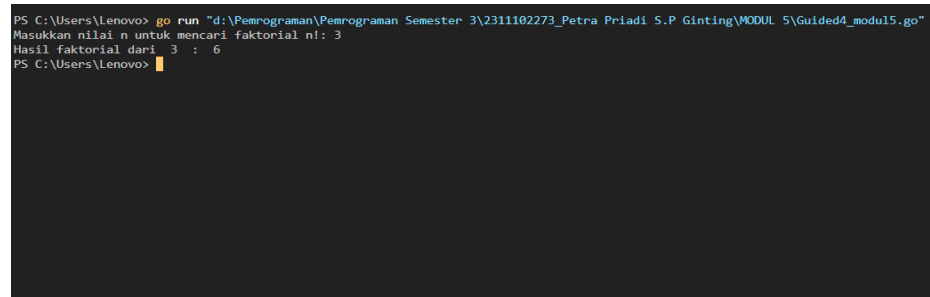
func main() {
    var n int

    fmt.Print("Masukkan nilai n untuk mencari faktorial
n!: ")

    fmt.Scanln(&n)

    fmt.Println("Hasil faktorial dari ", n, " : ",
faktorial(n))
}
```

Screenshoot Output



```
PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\2311102273_Petra Priadi S.P Ginting\MODUL 5\Guided4_modul5.go"
Masukkan nilai n untuk mencari faktorial n!: 3
Hasil faktorial dari 3 : 6
PS C:\Users\Lenovo>
```

Deskripsi Program

Fungsi faktorial yang menerima input berupa bilangan bulat n. Fungsi ini bekerja dengan cara memanggil dirinya sendiri dengan nilai n-1 hingga mencapai kondisi dasar ketika n bernilai 0 atau 1. Pada setiap pemanggilan, nilai n saat itu akan dikalikan dengan hasil pemanggilan fungsi sebelumnya. Proses ini berulang hingga mencapai nilai 0 atau 1, lalu semua hasil perkalian akan dikembalikan.

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(x int) int {
    if x == 0 {
        return 0
    } else if x == 1 {
        return 1
    } else {
        return Fibonacci(x-1) + Fibonacci(x-2)
    }
}

func CetakFibonacci(x int) {
    if x >= 0 {
        CetakFibonacci(x - 1)
        fmt.Println(Fibonacci(x))
    }
}

func main() {
    var x int
    fmt.Print("Masukkan angka untuk menampilkan deret\nFibonacci hingga suku ke-n: ")
    fmt.Scanln(&x)

    fmt.Printf("Deret Fibonacci hingga suku ke-%d:\n",
x)
    CetakFibonacci(x)
}
```


Screenshoot Output

```
PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\2311102273_Petra Priadi S.P Ginting\MODUL 5\Unguided1_modul5.go"
Masukkan angka untuk menampilkan deret Fibonacci hingga suku ke-n: 10
Deret Fibonacci hingga suku ke-10:
0
1
1
2
3
5
8
13
21
34
55
PS C:\Users\Lenovo>
```

Deskripsi Program

Fungsi Fibonacci(x) secara rekursif menghitung nilai suku ke-x dari deret Fibonacci. Jika x adalah 0, maka nilainya adalah 0. Jika x adalah 1, maka nilainya adalah 1. Untuk nilai x lainnya, nilai suku ke-x adalah penjumlahan dari suku ke-(x-1) dan suku ke-(x-2).

Fungsi CetakFibonacci(x) digunakan untuk mencetak deret Fibonacci secara terbalik. Fungsi ini juga bersifat rekursif, dimana ia akan memanggil dirinya sendiri dengan nilai x yang dikurangi 1, kemudian mencetak nilai Fibonacci dari x. Dengan cara ini, deret Fibonacci akan dicetak dari suku terakhir hingga suku pertama.

2. Studi Case

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 cetakBaris(nomorBintang int) {
    if nomorBintang == 0 {
        return
    }
    fmt.Print("*")
    cetakBaris(nomorBintang - 1)
}

func cetakBintang(x int, current int) {
    if current > x {
        return
    }

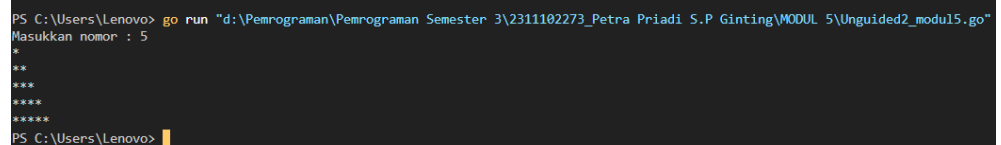
    cetakBaris(current)
    fmt.Println()

    cetakBintang(x, current+1)
}

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

    cetakBintang(x, 1)
}
```

Screenshoot Output



```
PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\2311102273_Petra Priadi S.P Ginting\MODUL 5\Unguided2_modul5.go"
Masukkan nomor : 5
*
*
*
*
*
PS C:\Users\Lenovo>
```

Deskripsi Program

Fungsi cetakBaris bertanggung berfungsi untuk mencetak satu baris bintang dengan jumlah yang ditentukan oleh parameter nomorBintang. Fungsi ini bekerja secara rekursif, di mana setiap pemanggilan akan mencetak satu bintang dan kemudian memanggil dirinya sendiri dengan nilai nomorBintang yang dikurangi satu, sampai nilai nomorBintang mencapai 0.

Fungsi cetakBintang berfungsi untuk mencetak keseluruhan segitiga. Fungsi ini menerima dua parameter: x yang menentukan jumlah baris maksimal dan current yang melacak baris yang sedang dicetak. Fungsi ini juga bekerja secara rekursif. Setiap pemanggilan akan mencetak satu baris bintang menggunakan fungsi cetakBaris, kemudian mencetak baris baru, dan memanggil dirinya sendiri dengan nilai current yang ditambah satu. Proses ini berulang sampai nilai current melebihi nilai x.

3. 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).

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 faktorBilangan(x, M int) {
    if M == x {
        fmt.Print(M)
    } else {
        if x%M == 0 {
            fmt.Print(M, " ")
            faktorBilangan(x, M+1)
        } else {
            faktorBilangan(x, M+1)
        }
    }
}
```

```

    }

}

func main() {
    var x int
    fmt.Print("Masukkan angka : ")
    fmt.Scan(&x)
    faktorBilangan(x, 1)
}

```

Screenshoot Output

```

PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\2311102273_Petra Priadi S.P Ginting\MODUL 5\Unguided3_modul5.go"
Masukkan angka : 12
1 2 3 4 6 12
PS C:\Users\Lenovo>

```

Deskripsi Program

Program ini bekerja dengan cara membandingkan bilangan input dengan bilangan-bilangan mulai dari 1 hingga bilangan input itu sendiri. Jika bilangan input habis dibagi dengan bilangan pembanding, maka bilangan pembanding tersebut dianggap sebagai faktor dan akan dicetak. Proses pembandingan ini dilakukan secara berulang menggunakan konsep rekursi, di mana fungsi faktorBilangan memanggil dirinya sendiri dengan nilai pembanding yang terus bertambah hingga mencapai bilangan input. Dengan demikian, program ini akan menghasilkan daftar lengkap semua faktor dari bilangan yang diberikan.

4. 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.

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

func cetakNaik(n, current int) {
    if current > n {
        return
    }
    fmt.Print(current, " ")
    cetakNaik(n, current+1)
}

func cetakKeseluruhan(n int) {
    if n <= 0 {
        fmt.Println("Masukan harus berupa bilangan
bulat positif!")
        return
    }
    cetakTurun(n)
    cetakNaik(n, 2)
    fmt.Println()
}

func main() {
    var n int

    for {
        fmt.Print("Masukkan angka,ketik 0 kalau
selsai: ")
        _, err := fmt.Scan(&n)

        if err != nil {
            fmt.Println("angka kurang pas,coba lagi
masukkan yang lain.")
            var dump string
            fmt.Scanln(&dump)
        }
    }
}
```

```

        continue
    }

    if n == 0 {
        fmt.Println("Program selesai!")
        break
    }

    fmt.Printf("Barisan bilangan untuk N = %d:\n",
n)

    cetakKeseluruhan(n)
}
}

```

Screenshoot Output

```

PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\2311182273_Petra Priadi S.P Ginting\MODUL 5\Unguided4_modul5.go"
Masukkan angka,ketik 0 kalau selsai: 5
Barisan bilangan untuk N = 5:
5 4 3 2 1 2 3 4 5
Masukkan angka,ketik 0 kalau selsai: 

```

Deskripsi Program

Program ini menerima input berupa bilangan bulat positif dan akan mencetak angka dari bilangan input tersebut menurun hingga 1, lalu naik kembali hingga mencapai bilangan input awal. Logika ini diimplementasikan menggunakan rekursi, di mana fungsi cetakTurun mencetak bilangan secara menurun dan fungsi cetakNaik mencetak bilangan secara naik. Dengan memanggil kedua fungsi ini secara berurutan, program akan menghasilkan pola bilangan yang diinginkan.

5. 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.

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 cetakBilganjil(current, x int) {
    if current > x {
        return
    }

    if current%2 != 0 {
        fmt.Print(current, " ")
    }

    cetakBilganjil(current+1, x)
}

func cetakUrutanganjil(x int) {
    if x <= 0 {
        fmt.Println("Masukan harus positif")
        return
    }
    cetakBilganjil(1, x)
    fmt.Println()
}

func main() {
    var x int

    for {
        fmt.Print("Masukkan angka : ")
        _, err := fmt.Scan(&x)

        if err != nil {
            fmt.Println("angka gak pas,masukkan lagi
yang lain.")
            var dump string
            fmt.Scanln(&dump)
            continue
        }

        if x == 0 {
            fmt.Println("Program selesai!")
            break
        }
    }
}
```

```

        fmt.Printf("Bilangan ganjil dari 1 sampai
%d:\n", x)
        cetakUrutanganjil(x)
    }
}

```

Screenshoot Output

```

PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\2311102273_Petra Priadi S.P Ginting\MODUL 5\Unguided5_modul5.go"
Masukkan angka : 5
Bilangan ganjil dari 1 sampai 5:
1 3 5
Masukkan angka : 20
Bilangan ganjil dari 1 sampai 20:
1 3 5 7 9 11 13 15 17 19
Masukkan angka :

```

Deskripsi Program

Program ini bekerja dengan cara memanggil fungsi cetakBilganjil secara rekursif. Fungsi cetakBilganjil akan memeriksa apakah bilangan saat ini adalah ganjil. Jika ya, maka bilangan tersebut akan dicetak. Kemudian, fungsi akan memanggil dirinya sendiri untuk memeriksa bilangan berikutnya. Proses ini akan terus berulang hingga mencapai batas atas. Dengan demikian, program ini akan mencetak semua bilangan ganjil dalam rentang yang ditentukan.

6. 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".

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	2 2	4
2	5 3	125

Sourcecode

```
package main
import "fmt"

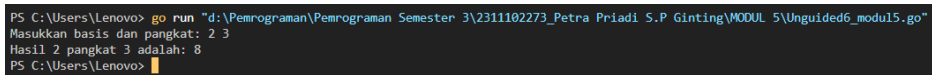
func pangkat(p, g int) int {
    if g == 0 {
        return 1
    }
    if g == 1 {
        return p
    }
    return p * pangkat(p, g-1)
}

func main() {
    var p, g int

    fmt.Print("Masukkan basis dan pangkat: ")
    _, err := fmt.Scan(&p, &g)

    if err != nil {
        fmt.Println("Input tidak valid!")
        return
    }
    hasil := pangkat(p, g)
    fmt.Printf("Hasil %d pangkat %d adalah: %d\n", p, g,
    hasil)
}
```

Screenshot Output



```
PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\2311102273_Petra Priadi S.P Ginting\MODUL 5\Unguided6_modul5.go"
Masukkan basis dan pangkat: 2 3
Hasil 2 pangkat 3 adalah: 8
PS C:\Users\Lenovo>
```

Deskripsi Program

Fungsi pangkat menerima dua input, yaitu basis dan pangkat. Logika kerjanya adalah dengan terus mengalikan basis dengan dirinya sendiri sebanyak nilai pangkat. Proses perhitungan ini dipecah menjadi masalah yang lebih kecil melalui pemanggilan fungsi pangkat secara berulang dengan nilai pangkat yang dikurangi satu. Jika nilai pangkat mencapai 0, maka fungsi akan mengembalikan 1 karena bilangan apa pun pangkat 0 adalah 1. Dengan demikian, fungsi pangkat akan menghitung hasil pangkat secara efisien melalui pendekatan rekursif.