

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V
REKURSIF**



Disusun Oleh :

Naya Putwi Setiasih / 2311102155

S1 11 IF - 05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

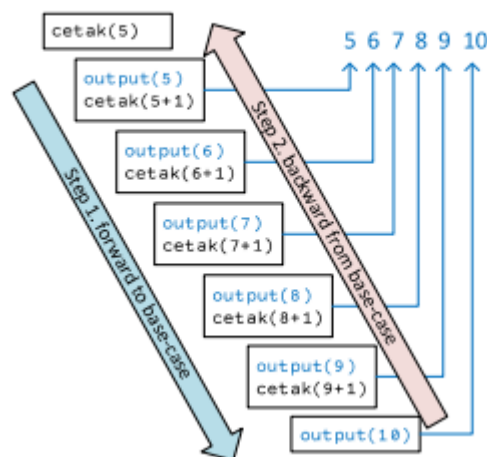
2024

I. DASAR TEORI

Rekursif adalah suatu konsep dalam matematika dan ilmu komputer yang memungkinkan suatu fungsi atau prosedur untuk memanggil dirinya sendiri secara terus – menerus. Dalam rekursif, fungsi atau prosedur tersebut dapat memanggil dirinya sendiri dengan parameter yang berbeda – beda, sehingga dapat menyelesaikan masalah yang kompleks dengan cara yang lebih efektif. Rekursif juga digunakan dalam berbagai aplikasi, seperti perhitungan factorial, perhitungan deret, dan perhitungan polynomial.

	Notasi Algoritma	Notasi dalam bahasa GO
1	procedure cetak(in x:integer)	func cetak(x int){
2	algoritma	fmt.Println(x)
3	output(x)	cetak(x+1)
4	cetak(x+1)	}
5	endprocedure	

Apabila diperhatikan subprogram cetak() di atas, terlihat pada baris ke – 4 terdapat pemanggilan subprogram cetak () kembali. Apabila kita eksekusi perintah cetak (5) maka akan menampilkan angka 5 6 7 8 9 .. dst tanpa henti. Setiap pemanggilan subprogram cetak () nilai x akan selalu bertambah 1 secara terus menerus tanpa henti.



Saat subprogram dipanggil secara rekursif, maka subprogram akan terus melakukan pemanggilan (forward) hingga berhenti pada saat kondisi base – case terpenuhi atau true. Setelah itu akan terjadi proses backward atau kembali ke

subprogram yang sebelumnya. Setelah semua instruksi cetak (10) selesai dieksekusi, maka program akan kembali ke cetak (9) yang memanggil cetak (10) tersebut. Begitu seterusnya hingga kembali ke cetak (5)

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.

II. GUIDED

1.

Soal Studi Case

Membuat baris bilangan dari n hingga 1

Base case : bilangan == 1

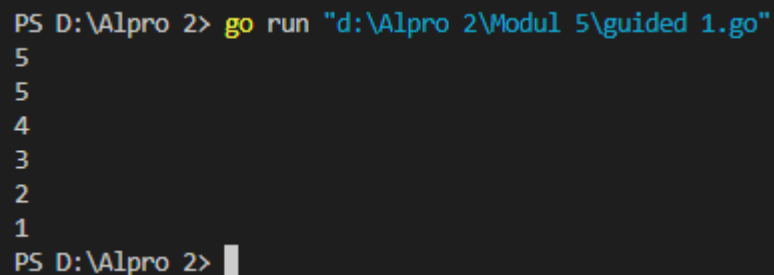
Sourcecode

```
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



```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 5\guided 1.go"
5
5
4
3
2
1
PS D:\Alpro 2> █
```

Deskripsi Program

Program ini ditulis dalam bahasa Go dan berfungsi untuk mencetak angka dari nilai yang dimasukkan hingga 1 secara berurutan. Program dimulai dengan mendeklarasikan fungsi “main”, Dimana pengguna diminta untuk memasukkan sebuah bilangan bulat “n”. Setelah mendapatkan input tersebut, program memanggil fungsi “baris” dengan argument “n”. Fungsi “baris” merupakan fungsi rekursif yang menerima satu parameter, yaitu “bilangan”. Di dalam fungsi ini, terdapat kondisi yang memeriksa apakah nilai “bilangan” sama dengan 1. Jika benar, maka program akan

mencetak angka 1. Namun, jika nilai “bilangan” lebih besar dari 1, program akan mencetak nilai “bilangan” saat ini dan kemudian memanggil dirinya sendiri dengan argument “bilangan – 1”. Proses ini berlanjut hingga nilai “bilangan” mencapai 1, sehingga hasil akhirnya adalah pencetakan angka dari “n” hingga 1 secara berurutan.

2. Menghitung hasil penjumlahan 1 hingga n
Base case : $n == 1$

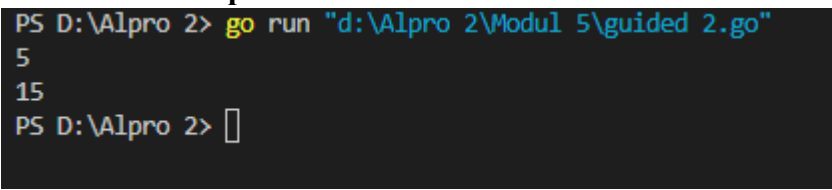
Sourcecode

```
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))
}
```

Screenshot Output



```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 5\guided 2.go"
5
15
PS D:\Alpro 2> 
```

Deskripsi Program

Program di atas adalah sebuah aplikasi sederhana dalam bahasa pemrograman Go yang bertujuan untuk menghitung jumlah dari semua bilangan bulat positif hingga bilangan yang diberikan oleh pengguna.

Ketika program dijalankan, pengguna diminta untuk memasukkan sebuah bilangan bulat positif “n”. Setelah itu, fungsi “penjumlahan” akan dipanggil dengan parameter “n”. Jika nilai “n” sama dengan 1, fungsi akan mengembalikan nilai 1. Namun, jika “n” lebih besar dari 1, fungsi akan mengembalikan jumlah dari “n” ditambah hasil pemanggilan fungsi “penjumlahan” dengan “n – 1”. Proses ini berlanjut hingga mencapai basis

rekursi, yaitu Ketika “n” sama dengan 1. Setelah semua panggilan rekursif selesai, hasil penjumlahan total akan dicetak ke layar.

3. Mencari dua pangkat n atau 2^n
Base case : $n == 0$

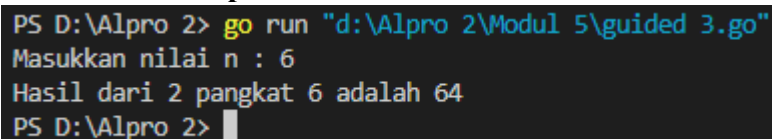
Sourcecode

```
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:\Alpro 2> go run "d:\Alpro 2\Modul 5\guided 3.go"
Masukkan nilai n : 6
Hasil dari 2 pangkat 6 adalah 64
PS D:\Alpro 2>
```

Deskripsi Program

Program di atas adalah sebuah aplikasi sederhana dalam bahasa pemrograman Go yang bertujuan untuk menghitung nilai dari 2^n menggunakan metode rekursif. Program ini terdiri dari sebuah fungsi bernama “pangkat”, yang menerima satu parameter integer “n”. Fungsi ini dirancang untuk mengembalikan hasil dari 2^n .

Ketika program dijalankan, pengguna akan diminta untuk memasukkan nilai integer “n”. Setelah nilai tersebut dimasukkan, fungsi “pangkat” akan dipanggil dengan parameter “n”. Jika nilai “n” sama dengan 0, fungsi akan mengembalikan 1, sesuai dengan sifat dasar eksponensial bahwa $2^0 = 1$. Namun, jika “n” lebih besar dari 0, fungsi akan mengembalikan hasil perkalian antara 2 dan pemanggilan rekursif dari fungsi “pangkat” dengan

parameter “ $n - 1$ ”. Proses ini berlanjut hingga mencapai kondisi dasar Ketika “ n ” menjadi 0. Setelah semua pemanggilan rekursif selesai, hasil akhir dari 2^n akan dicetak ke layar.

4. Mencari nilai faktorial atau $n!$

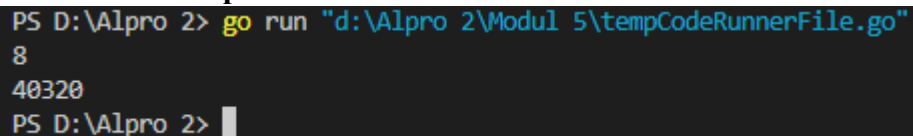
Base case : $n == 0$ atau $n == 1$

Sourcecode

```
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



```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 5\tempCodeRunnerFile.go"
8
40320
PS D:\Alpro 2> 
```

Deskripsi Program

Program di atas adalah sebuah aplikasi sederhana dalam bahasa pemrograman Go yang bertujuan untuk menghitung nilai factorial dari sebuah bilangan bulat non – negative yang dimasukkan oleh pengguna. Factorial dari suatu bilangan n , yang dilambangkan dengan $n!$, didefinisikan sebagai hasil kali semua bilangan bulat positif dari 1 hingga n .

Program ini menggunakan fungsi rekursif bernama “factorial” untuk melakukan perhitungan. Ketika program dijalankan, pengguna diminta untuk memasukkan nilai integer n . setelah nilai tersebut dimasukkan, fungsi “factorial” akan dipanggil dengan parameter n . jika n sama dengan 0 atau 1, fungsi akan mengembalikan nilai 1, karena $0!$ Dan $1!$ Keduanya didefinisikan sebagai 1. Jika n lebih besar dari 1, fungsi akan

mengembalikan hasil kali antara n dan pemanggilan rekursif dari fungsi “factorial” dengan parameter $n - 1$. Proses ini berlanjut hingga mencapai kondisi dasar. Setelah semua panggilan rekursif selesai, hasil akhir factorial dari n akan dicetak ke layar.

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

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: ")
    fmt.Scan(&n)

    fmt.Println("n\tSn")
    for i := 0; i <= n; i++ {
        fmt.Printf("%d\t%d\n", i, fibonacci(i))
    }
}
```

Screenshoot Output

```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 5\unguided 1.go"
Masukkan nilai n: 6
n      Sn
0      0
1      1
2      1
3      2
4      3
5      5
6      8
PS D:\Alpro 2> █
```

Deskripsi Program

Program di atas adalah aplikasi yang ditulis dalam bahasa pemrograman Go untuk menghitung dan menampilkan deret Fibonacci hingga bilangan bulat positif yang dimasukkan oleh pengguna. Deret Fibonacci adalah urutan angka di mana setiap angka merupakan jumlah dari dua angka sebelumnya, dimulai dengan 0 dan 1.

Program ini terdiri dari fungsi rekursif bernama “Fibonacci”, yang menerima satu parameter integer “n”. Jika “n” sama dengan 0, fungsi mengembalikan 0, dan jika “n” sama dengan 1, fungsi mengembalikan 1. Untuk nilai “n” yang lebih besar dari 1, fungsi akan menghitung nilai Fibonacci dengan memanggil dirinya sendiri untuk “n – 1” dan “n – 2”, menjumlahkan hasilnya.

Saat program dijalankan, pengguna diminta untuk memasukkan nilai integer “n”. Setelah nilai tersebut dimasukkan, program akan mencetak header “n” dan “Sn” sebagai label kolom. Kemudian, dalam loop yang berjalan dari 0 hingga “n”, program akan memanggil fungsi “fibonacci” untuk setiap nilai “i” dan mencetak hasilnya dalam format label.

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

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi rekursif untuk mencetak bintang
func cetakBintang(n int, i int) {
    if i > n {
```

```

        return // Kondisi dasar: jika i lebih besar dari
n, hentikan rekursi
    }

    // Mencetak bintang untuk baris ke-i
    for j := 0; j < i; j++ {
        fmt.Print("* ")
    }
    fmt.Println() // Pindah ke baris berikutnya

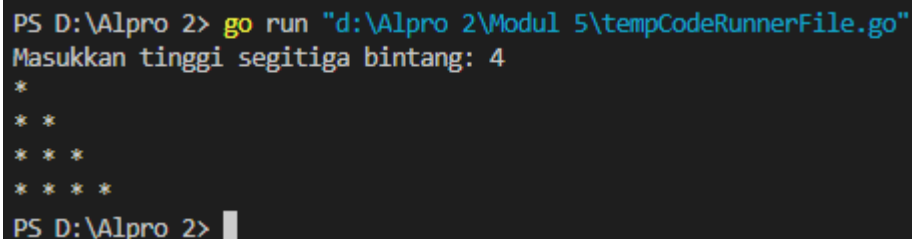
    // Panggil fungsi untuk baris berikutnya
    cetakBintang(n, i+1)
}

func main() {
    var n int
    fmt.Print("Masukkan tinggi segitiga bintang: ")
    fmt.Scan(&n)

    cetakBintang(n, 1) // Memulai pencetakan dari baris
    pertama
}

```

Screenshoot Output



```

PS D:\Alpro 2> go run "d:\Alpro 2\Modul 5\tempCodeRunnerFile.go"
Masukkan tinggi segitiga bintang: 4
*
* *
* * *
* * * *
PS D:\Alpro 2>

```

Deskripsi Program

Program di atas adalah aplikasi sederhana yang ditulis dalam bahasa pemrograman Go, yang berfungsi untuk menampilkan pola bintang berbentuk segitiga berdasarkan tinggi yang dimasukkan oleh pengguna. Program ini menggunakan fungsi rekursif untuk mencetak bintang secara bertahap.

Ketika program dijalankan, pengguna diminta untuk memasukkan tinggi segitiga Bintang yang diinginkan. Setelah nilai tersebut dimasukkan, fungsi “cetak Bintang” dipanggil dengan parameter tinggi segitiga “n” dan nomor baris awal “1”, yang dimulai dari 1. Fungsi ini memiliki dua kondisi : jika nomor baris “i” lebih besar dari “n”, fungsi akan menghentikan eksekusi. Jika tidak, fungsi akan mencetak “1” Bintang pada baris tersebut, diikuti dengan pemanggilan rekursif dari dirinya sendiri dengan parameter

“i + 1”, untuk mencetak baris berikutnya. Proses ini berlanjut hingga semua baris dari 1 hingga “n” dicetak.

3. Buatlah program yang mengimplementasikan rekursif untuk menampilkan factor 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 factor dari N (terurut dari 1 hingga N ya)

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi rekursif untuk mencetak faktor
func cetakFaktor(n int, i int) {
    if i > n {
        return // Kondisi dasar: jika i lebih besar dari
n, hentikan rekursi
    }

    // Cek apakah i adalah faktor dari n
    if n%i == 0 {
        fmt.Print(i, " ") // Cetak faktor
    }

    // Panggil fungsi untuk angka berikutnya
    cetakFaktor(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)
    cetakFaktor(n, 1) // Memulai pencetakan dari 1
    fmt.Println() // Pindah ke baris baru setelah
selesai
}
```

Screenshoot Output

```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 5\unguided 3.go"
Masukkan bilangan bulat positif N: 12
Faktor dari 12 adalah: 1 2 3 4 6 12
PS D:\Alpro 2> █
```

Deskripsi Program

Program di atas adalah aplikasi yang ditulis dalam bahasa pemrograman Go, yang bertujuan untuk menampilkan semua faktor dari suatu bilangan bulat positif “N” yang dimasukkan oleh pengguna. Program ini menggunakan pendekatan rekursif untuk memeriksa dan mencetak faktor-faktor tersebut.

Saat program dijalankan, pengguna diminta untuk memasukkan nilai integer positif “N”. Setelah nilai tersebut dimasukkan, fungsi “cetak factor” dipanggil dengan parameter “N” dan angka awal “i”, yang dimulai dari 1. Fungsi ini akan memeriksa apakah “i” adalah factor dari “N” dengan cara mengecek apakah sisa pembagian “n” dengan “i” sama dengan 0. Jika kondisi ini terpenuhi, maka “i” dicetak sebagai factor. Proses ini berlanjut dengan memanggil fungsi “cetak factor” secara rekursif dengan parameter “i + 1”, hingga nilai “i” melebihi “n”, pada titik mana fungsi akan berhenti.

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.

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi rekursif untuk mencetak bilangan dari n hingga
// 1, lalu kembali ke n
func printSequence(n int) {
    // Cetak bilangan saat ini
    fmt.Print(n, " ")

    // Kondisi basis untuk berhenti ketika mencapai 1
    if n > 1 {
        // Panggil rekursif menurun
        printSequence(n - 1)
        // Cetak saat kembali naik
        fmt.Print(n, " ")
    }
}
```

```

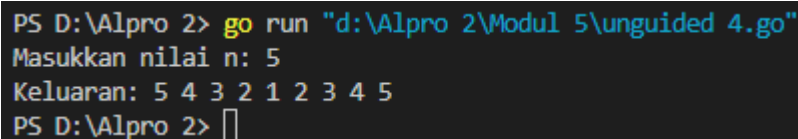
    }
}

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

    fmt.Print("Keluaran: ")
    printSequence(n)
    fmt.Println()
}

```

Screenshoot Output



```

PS D:\Alpro 2> go run "d:\Alpro 2\Modul 5\unguided 4.go"
Masukkan nilai n: 5
Keluaran: 5 4 3 2 1 2 3 4 5
PS D:\Alpro 2>

```

Deskripsi Program

Program di atas adalah aplikasi sederhana yang ditulis dalam bahasa pemrograman Go, yang bertugas untuk menampilkan barisan bilangan mulai dari n hingga 1, dan kemudian kembali ke nilai aslinya (n). Program ini menggunakan fungsi rekursif untuk mencetak urutan angka tersebut. Saat program dijalankan, pengguna diminta untuk memasukkan sebuah bilangan bulat positif n . Setelah nilai tersebut dimasukkan, fungsi `printSequence` dipanggil dengan parameter n . Fungsi ini pertama-tama mencetak nilai n , dan kemudian memeriksa apakah nilai tersebut lebih besar dari 1. Jika ya, fungsi akan memanggil dirinya sendiri dengan argumen yang lebih rendah ($n - 1$) untuk mencetak bilangan berikutnya dalam urutan menurun. Setelah mencapai kondisi dasar ketika n sama dengan 1, fungsi akan mulai kembali dan mencetak nilai n lagi untuk menyelesaikan urutan. Proses ini menghasilkan keluaran yang terstruktur, seperti contoh: jika pengguna memasukkan 5, maka output yang dihasilkan adalah 5 4 3 2 1 5.

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 .

Sourcecode

```
package main

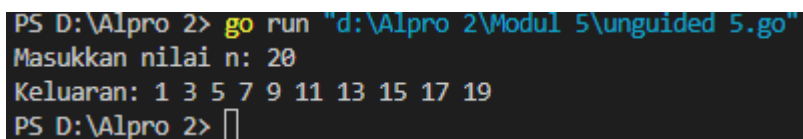
import (
    "fmt"
)

// Fungsi rekursif untuk mencetak bilangan ganjil dari 1
hingga N
func printOddNumbers(n, current int) {
    // Cetak bilangan ganjil saat ini jika tidak
    melebihi n
    if current <= n {
        fmt.Print(current, " ")
        // Panggil rekursif dengan bilangan ganjil
        berikutnya
        printOddNumbers(n, current + 2)
    }
}

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

    fmt.Print("Keluaran: ")
    printOddNumbers(n, 1) // Memulai dari 1 sebagai
    bilangan ganjil pertama
    fmt.Println()
}
```

Screenshoot Output



```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 5\unguided 5.go"
Masukkan nilai n: 20
Keluaran: 1 3 5 7 9 11 13 15 17 19
PS D:\Alpro 2> 
```

Deskripsi Program

Program di atas adalah aplikasi sederhana yang ditulis dalam bahasa pemrograman Go, yang bertujuan untuk mencetak semua bilangan ganjil dari 1 hingga N, di mana N adalah nilai yang dimasukkan oleh pengguna. Program ini menggunakan pendekatan rekursif untuk menghasilkan dan menampilkan bilangan ganjil tersebut. Saat program dijalankan, pengguna diminta untuk memasukkan sebuah bilangan bulat positif N. Setelah nilai tersebut dimasukkan, fungsi `printOddNumbers` dipanggil dengan parameter `n` dan `current`, yang dimulai dari 1 sebagai bilangan ganjil

pertama. Fungsi ini memeriksa apakah nilai current masih kurang dari atau sama dengan n. Jika ya, maka fungsi akan mencetak nilai current, diikuti dengan pemanggilan rekursif dari dirinya sendiri dengan parameter current + 2, yang akan menghasilkan bilangan ganjil berikutnya. Proses ini berlanjut hingga nilai current melebihi n, pada titik mana fungsi akan berhenti.

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.

Sourcecode

```
package main

import "fmt"

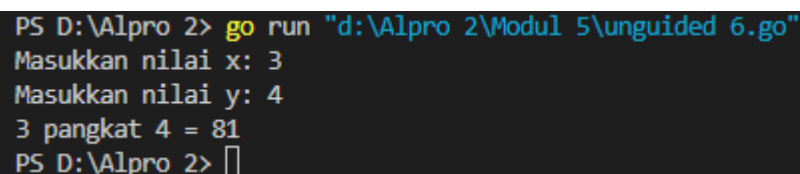
// Fungsi rekursif untuk menghitung x pangkat y
func power(x, y int) int {
    if y == 0 {
        return 1
    }
    return x * power(x, y-1)
}

func main() {
    var x, y int

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

    // Menampilkan hasil
    fmt.Printf("%d pangkat %d = %d\n", x, y, power(x,
y))
}
```

Screenshoot Output



```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 5\unguided 6.go"
Masukkan nilai x: 3
Masukkan nilai y: 4
3 pangkat 4 = 81
PS D:\Alpro 2>
```


Deskripsi Program

Program di atas adalah aplikasi sederhana yang ditulis dalam bahasa pemrograman Go, yang berfungsi untuk menghitung nilai eksponensial dari suatu bilangan x yang dipangkatkan dengan y , menggunakan pendekatan rekursif. Ketika program dijalankan, pengguna diminta untuk memasukkan dua bilangan bulat: x sebagai basis dan y sebagai eksponen. Setelah nilai-nilai tersebut dimasukkan, fungsi power dipanggil dengan parameter x dan y . Fungsi ini memiliki kondisi dasar yang menyatakan bahwa jika y sama dengan 0, maka hasilnya adalah 1, sesuai dengan sifat dasar eksponensial bahwa setiap bilangan pangkat nol adalah satu. Jika y lebih besar dari 0, fungsi akan mengembalikan hasil perkalian antara x dan pemanggilan rekursif dari fungsi power dengan parameter x dan $y-1$, yang secara bertahap mengurangi nilai eksponen hingga mencapai kondisi dasar. Setelah perhitungan selesai, program akan mencetak hasilnya dalam format yang jelas, menunjukkan hasil dari operasi eksponensial tersebut.

DAFTAR PUSTAKA

[1] Rahmawati, E., Medina, P., & Azhari, D. S. (2024). Rekursif Dalam Pemrograman Teori Dan Praktek. *Innovative: Journal Of Social Science Research*, 4(4), 5622-5630.

Asisten Praktikum, "Modul 5 Rekursif", Learning Management System, 2024.