

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V
REKURSF**



Disusun Oleh :

Andika Indra Prastawa / 2311102033

Kelas

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

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

	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. 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.

Oleh karena itu bisanya 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.

II. GUIDED 1

Sourcecode

```
package main

import "fmt"

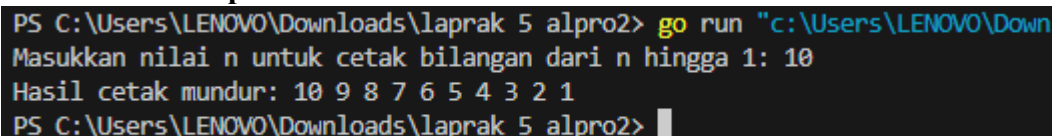
func cetakMundur(n int) {
    if n == 1 {
        fmt.Println(n)
        return
    }

    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\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Down
Masukkan nilai n untuk cetak bilangan dari n hingga 1: 10
Hasil cetak mundur: 10 9 8 7 6 5 4 3 2 1
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> █
```

Deskripsi Program

Program bertujuan untuk mencetak bilangan secara mundur dari angka yang dimasukkan oleh pengguna hingga 1. Di dalam fungsi main, program meminta pengguna untuk memasukkan nilai n, kemudian mencetak hasil dari fungsi cetakMundur. Fungsi cetakMundur menggunakan rekursi untuk menampilkan angka dari n hingga 1. Pada setiap langkah, fungsi ini akan mencetak nilai n saat ini dan kemudian memanggil dirinya sendiri dengan nilai n-1 hingga mencapai angka 1, yang kemudian dicetak tanpa spasi, dan menghentikan rekursi dengan perintah return.

2. Guided 2

Sourcecode

```
package main

import "fmt"

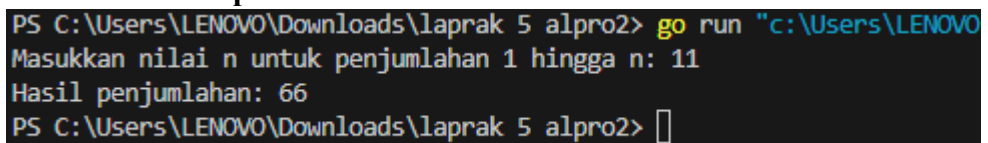
func jumlahRekursif(n int) int {
    if n == 1 {
        return 1
    }

    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))
}
```

Screenshot Output



```
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 alpro2\main.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 11
Hasil penjumlahan: 66
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> 
```

Deskripsi Program

Program berfungsi untuk menghitung jumlah bilangan dari 1 hingga n menggunakan rekursi. Di dalam fungsi main, program meminta pengguna untuk memasukkan nilai n, lalu mencetak hasil penjumlahan dari fungsi jumlahRekursif. Fungsi jumlahRekursif menggunakan konsep rekursi, di mana setiap pemanggilan fungsi akan menambahkan nilai n saat ini dengan

hasil pemanggilan fungsi berikutnya dengan nilai $n-1$, hingga mencapai kondisi dasar di mana n sama dengan 1. Pada titik ini, fungsi mengembalikan 1, dan rekursi berakhir.

3. Guided 3

Sourcecode

```
package main

import "fmt"

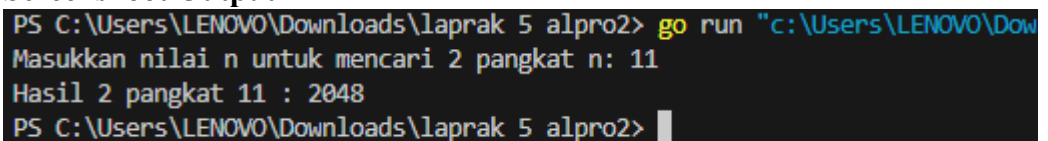
func pangkatDua(n int) int {
    if n == 0 {
        return 1
    }

    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\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 alpro2.go"
Masukkan nilai n untuk mencari 2 pangkat n: 11
Hasil 2 pangkat 11 : 2048
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> █
```

Deskripsi Program

Program ini menghitung hasil dari 2 pangkat n menggunakan rekursi. Pada fungsi main, program meminta pengguna untuk memasukkan nilai n , kemudian mencetak hasil perhitungan dari fungsi pangkatDua. Fungsi pangkatDua bekerja dengan pendekatan rekursif: jika n bernilai 0, fungsi mengembalikan nilai 1 sebagai kondisi dasar (karena 2 pangkat 0 adalah 1). Untuk nilai n yang lebih besar, fungsi ini mengalikan 2 dengan hasil

pemanggilan dirinya sendiri dengan nilai $n-1$, hingga rekursi mencapai kondisi dasar dan mengembalikan hasil perhitungan 2 pangkat n .

4. Guided 4

Sourcecode

```
package main

import "fmt"

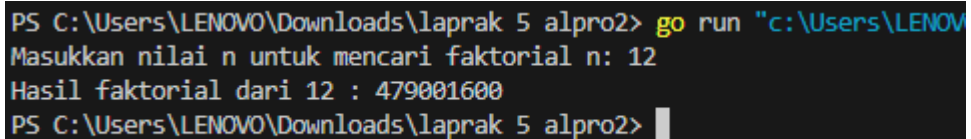
func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }

    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\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 alpro2\main.go"
Masukkan nilai n untuk mencari faktorial n: 12
Hasil faktorial dari 12 : 479001600
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> 
```

Deskripsi Program

Program ini bertujuan untuk menghitung nilai faktorial dari bilangan n menggunakan rekursi. Di dalam fungsi `main`, program meminta pengguna untuk memasukkan nilai n , lalu mencetak hasil perhitungan faktorial yang didapat dari fungsi `faktorial`. Fungsi `faktorial` menggunakan pendekatan rekursif dengan kondisi dasar saat n bernilai 0 atau 1, di mana fungsi mengembalikan nilai 1 (karena faktorial dari 0 dan 1 adalah 1). Untuk nilai

n yang lebih besar, fungsi ini mengalikan n dengan hasil pemanggilan dirinya sendiri dengan nilai $n-1$, dan proses ini berlanjut hingga mencapai kondisi dasar, memberikan hasil akhir faktorial dari n .

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 $S_2 = S_{n-1} + S_{n-2}$. Berikut ini adalah contoh nilai deret fibonacci hingga suku ke-10. Buatlah program yang mengimplementasikan fungsi rekursif pada deret fibonacci tersebut.

Sourcecode

```
package main

import "fmt"

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

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk menghitung deret\nFibonacci hingga suku ke-n: ")
    fmt.Scanln(&n)
    fmt.Println("Deret Fibonacci hingga suku ke-", n,
        ":")

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

Screenshoot Output


```

PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung deret Fibonacci hingga suku ke-n: 0
Deret Fibonacci hingga suku ke- 0 :
0
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung deret Fibonacci hingga suku ke-n: 1
Deret Fibonacci hingga suku ke- 1 :
0 1
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung deret Fibonacci hingga suku ke-n: 2
Deret Fibonacci hingga suku ke- 2 :
0 1 1
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung deret Fibonacci hingga suku ke-n: 3
Deret Fibonacci hingga suku ke- 3 :
0 1 1 2
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung deret Fibonacci hingga suku ke-n: 4
Deret Fibonacci hingga suku ke- 4 :
0 1 1 2 3
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung deret Fibonacci hingga suku ke-n: 5
Deret Fibonacci hingga suku ke- 5 :
0 1 1 2 3 5
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung deret Fibonacci hingga suku ke-n: 6
Deret Fibonacci hingga suku ke- 6 :
0 1 1 2 3 5 8
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung deret Fibonacci hingga suku ke-n: 7
Deret Fibonacci hingga suku ke- 7 :
0 1 1 2 3 5 8 13
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung deret Fibonacci hingga suku ke-n: 8
Deret Fibonacci hingga suku ke- 8 :
0 1 1 2 3 5 8 13 21
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung deret Fibonacci hingga suku ke-n: 9
Deret Fibonacci hingga suku ke- 9 :
0 1 1 2 3 5 8 13 21 34
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 a
Masukkan nilai n untuk menghitung 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\Downloads\laprak 5 alpro2> 

```

Deskripsi Program

Program menghitung dan menampilkan deret Fibonacci hingga suku ke-n yang diinputkan oleh pengguna. Dalam fungsi fibonacci, program menggunakan rekursi untuk menghitung nilai Fibonacci dari suatu bilangan n, di mana fibonacci(0) mengembalikan 0 dan fibonacci(1) mengembalikan 1. Nilai Fibonacci dari suku-suku berikutnya dihitung sebagai jumlah dari dua suku sebelumnya (fibonacci(n-1) + fibonacci(n-2)). Pada fungsi main,

program meminta pengguna memasukkan nilai n , kemudian menggunakan loop for untuk memanggil fungsi fibonacci dan mencetak setiap suku dalam deret Fibonacci mulai dari suku ke-0 hingga suku ke- n .

UNGUIDED 2

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"

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.Scanln(&n)

    fmt.Println("Pola Bintang:")
    printStars(n)
}
```

Screenshoot Output

```

PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO
Masukkan nilai N: 11
Pola Bintang:
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> 

```

Deskripsi Program

Program menampilkan pola bintang segitiga yang berurutan dari satu baris hingga baris dengan jumlah bintang terbanyak sesuai dengan input n yang dimasukkan oleh pengguna. Di dalam fungsi main, program meminta pengguna untuk memasukkan nilai n, kemudian mencetak pola bintang menggunakan fungsi printStars. Fungsi printStars adalah fungsi rekursif yang pertama-tama memanggil dirinya sendiri dengan n-1 hingga n mencapai 0, sehingga menunda pencetakan bintang hingga rekursi selesai. Setelah itu, untuk setiap pemanggilan dalam urutan balik (dari 1 hingga n), fungsi ini mencetak sejumlah bintang yang sesuai dengan nilai n saat ini, menghasilkan pola segitiga bintang yang terbalik.

UNGUIDED 3

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).

Sourcecode

```

package main

import "fmt"

func printFactors(n, i int) {
    if i > n {

```

```

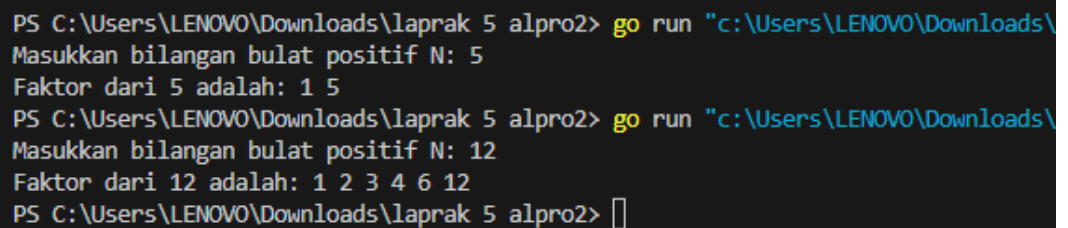
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    printFactors(n, i+1)
}

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

    fmt.Print("Faktor dari ", n, " adalah: ")
    printFactors(n, 1)
    fmt.Println()
}

```

Screenshoot Output



```

PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\
Masukkan bilangan bulat positif N: 5
Faktor dari 5 adalah: 1 5
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\
Masukkan bilangan bulat positif N: 12
Faktor dari 12 adalah: 1 2 3 4 6 12
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> 

```

Deskripsi Program

Program berfungsi untuk menampilkan semua faktor dari bilangan bulat positif n yang dimasukkan oleh pengguna. Dalam fungsi `main`, program meminta pengguna untuk memasukkan nilai n , kemudian memanggil fungsi `printFactors` untuk mencetak faktor-faktor dari n . Fungsi `printFactors` adalah fungsi rekursif yang menerima dua parameter: n (bilangan yang akan dicari faktornya) dan i (sebagai pembagi yang dimulai dari 1). Pada setiap pemanggilan, fungsi ini mengecek apakah n habis dibagi oleh i . Jika iya, maka i dicetak sebagai faktor. Setelah itu, fungsi memanggil dirinya sendiri dengan $i+1$, hingga i lebih besar dari n , yang menghentikan rekursi.

UNGUIDED 4

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"

func printDescending(n int) {
    if n < 1 {
        return
    }
    fmt.Print(n, " ")
    printDescending(n - 1)
}

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

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

    fmt.Println("Barisan bilangan dari", n, "hingga 1
dan kembali ke", n, ":")
    printDescending(n)
    printAscending(n, 1)
    fmt.Println()
}
```

Screenshoot Output

```

PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downr
Masukkan bilangan bulat positif N: 5
Barisan bilangan dari 5 hingga 1 dan kembali ke 5 :
5 4 3 2 1 1 2 3 4 5
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downr
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 1 2 3 4 5 6 7 8 9
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> 

```

Deskripsi Program

Program menampilkan bilangan bulat positif dari n hingga 1 secara menurun, kemudian kembali dari 1 hingga n secara menaik. Dalam fungsi main, program meminta pengguna untuk memasukkan bilangan bulat positif n , lalu mencetak barisan bilangan tersebut menggunakan dua fungsi rekursif: `printDescending` dan `printAscending`. Fungsi `printDescending` mencetak bilangan dari n hingga 1 secara menurun, dengan memanggil dirinya sendiri dengan nilai $n-1$ hingga n kurang dari 1, yang menghentikan rekursi. Setelah selesai, fungsi `printAscending` mencetak bilangan dari 1 hingga n secara menaik, dimulai dengan `current` yang awalnya bernilai 1 dan bertambah 1 di setiap panggilan rekursif hingga mencapai n . Hasilnya adalah deretan bilangan dari n ke 1 dan kembali ke n .

UNGUIDED 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"
)

func printtambahangka(n int, current int) {
    if current > n {
        return
    }
    if current%2 != 0 {

```

```

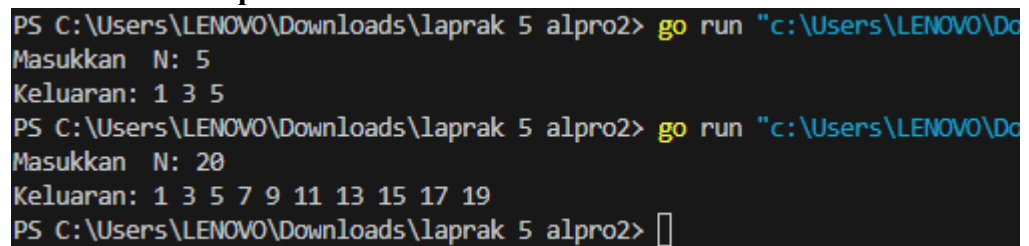
        fmt.Printf("%d ", current)
    }
    printtambahangka(n, current+1)
}

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

    fmt.Print("Keluaran: ")
    printtambahangka(n, 1)
    fmt.Println()
}

```

Screenshoot Output



```

PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 alpro2.go"
Masukkan N: 5
Keluaran: 1 3 5
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downloads\laprak 5 alpro2.go"
Masukkan N: 20
Keluaran: 1 3 5 7 9 11 13 15 17 19
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> 

```

Deskripsi Program

Program berfungsi untuk mencetak semua bilangan ganjil dari 1 hingga bilangan n yang dimasukkan oleh pengguna. Pada fungsi `main`, program meminta pengguna untuk memasukkan nilai n , lalu memanggil fungsi `printtambahangka` untuk mencetak bilangan ganjil dalam rentang tersebut. Fungsi `printtambahangka` adalah fungsi rekursif yang menerima dua parameter: n (batas atas) dan `current` (bilangan yang diperiksa, dimulai dari 1). Setiap kali fungsi dipanggil, ia memeriksa apakah `current` adalah bilangan ganjil. Jika iya, nilai `current` dicetak. Fungsi ini kemudian memanggil dirinya sendiri dengan nilai `current+1` hingga `current` melebihi n , yang menghentikan rekursi. Hasil akhir dari program ini adalah deretan bilangan ganjil dari 1 hingga n .

UNGUIDED 6

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".

Sourcecode

```
package main

import (
    "fmt"
)

func pangkat(basis int, eksponen int) int {

    if eksponen == 0 {
        return 1
    }

    return basis * pangkat(basis, eksponen-1)
}

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

    hasil := pangkat(x, y)
    fmt.Printf("Keluaran: %d\n", hasil)
}
```

Screenshot Output


```
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downl
Masukkan bilangan x: 2 2
Masukkan bilangan y: Keluaran: 4
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> go run "c:\Users\LENOVO\Downl
Masukkan bilangan x: 5 3
Masukkan bilangan y: Keluaran: 125
PS C:\Users\LENOVO\Downloads\laprak 5 alpro2> █
```

Deskripsi Program

Program ini menghitung hasil pangkat dari dua bilangan bulat, xxx sebagai basis dan yyy sebagai eksponen, menggunakan fungsi rekursif tanpa menggunakan pustaka matematika. Fungsi pangkat menerima dua parameter, basis dan eksponen. Jika eksponen bernilai 0, fungsi mengembalikan 1 karena setiap bilangan dipangkatkan 0 adalah 1 (ini adalah kasus dasar untuk menghentikan rekursi). Jika tidak, fungsi mengalikan basis dengan hasil dari pangkat(basis, eksponen-1), yang mengurangi nilai eksponen secara bertahap hingga mencapai 0. Dalam fungsi main, pengguna diminta memasukkan nilai x dan y, kemudian program memanggil fungsi pangkat dengan input tersebut dan mencetak hasilnya.