

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI  
OBJEK**

**MODUL V  
REKURSIF**



**Oleh :**

NAMA : FAISAL KHOIRUDDIN

NIM : 2311102046

Kelas : IF-11-02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2024**

## I. DASAR TEORI

### 1. Pengantar 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**.

```
1 package main
2 import "fmt"
3 func main(){
4     cetak(5)
5 }
6 func cetak(x int){
7     fmt.Println(x)
8     cetak(x+1)
9 }

D:\DEV\DEMO>go build contoh.go
D:\DEV\DEMO>contoh.exe
5
6
7
8
9
10
11
12
13
...
```

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.

```

1 procedure cetak(in x:integer)
2 algoritma
3   if x == 10 then
4     output(x)
5   else
6     output(x)
7     cetak(x+1)
8   endif
9 endprocedure

```

Apabila diperhatikan pada baris ke-3 di Program di atas, kita telah menambahkan **base-case** seperti penjelasan sebelumnya. Selanjutnya pada bagian aksi dari else di baris ke-6 dan ke-7 kita namakan **recursive-case** atau kasus pemanggilan dirinya sendiri tersebut terjadi. Kondisi dari **recursive-case** ini adalah negasi dari kondisi **base-case** atau ketika nilai  $x \neq 10$ .

```

1 package main
2 import "fmt"
3 func main(){
4   cetak(5)
5 }
6 func cetak(x int){
7   if x == 10 {
8     fmt.Println(x)
9   }else{
10    fmt.Println(x)
11    cetak(x+1)
12  }
13 }

```

```

D:\DEV\DEMO>go build contoh.go
D:\DEV\DEMO>contoh.exe

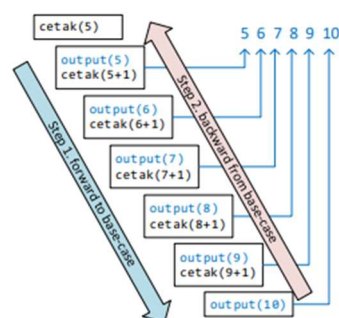
```

```

5
6
7
8
9
10

```

Apabila program di atas ini dijalankan maka akan tampil angka 5 6 7 8 9 10. Terlihat bahwa proses rekursif berhasil dihentikan ketika  $x == 10$ .



Gambar 1 merupakan ilustrasi forward dan backward saat rekursif

Pada Gambar 2 memperlihatkan 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. Artinya 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). Perhatikan modifikasi program di atas dengan menukar posisi baris 10 dan 11, mengakibatkan ketika program dijalankan maka akan menampilkan 10 9 8 7 6 5. Kenapa bisa demikian? Pahami proses **backward** pada Gambar 2

```
1 package main
2 import "fmt"
3 func main(){
4     cetak(5)
5 }
6 func cetak(x int){
7     if x == 10 {
8         fmt.Println(x)
9     }else{
10        cetak(x+1)
11        fmt.Println(x)
12    }
13 }
```

```
D:\DEV\DEMO>go build contoh.go
D:\DEV\DEMO>contoh.exe
10
9
8
7
6
5
```

#### Catatan:

- Teknik rekursif ini merupakan salah satu alternatif untuk mengganti struktur kontrol perulangan dengan memanfaatkan subprogram (bisa fungsi ataupun prosedur).
- Untuk menghentikan proses rekursif digunakan percabangan (if-then).
- **Base-case** adalah kondisi proses rekursif berhenti. **Base-case** merupakan hal terpenting dan pertama yang harus diketahui ketika akan membuat program rekursif. **Mustahil** membuat program

rekursif tanpa mengetahui **base-case** terlebih dahulu.

- **Recursive-case** adalah kondisi dimana proses pemanggilan dirinya sendiri dilakukan. Kondisi **recursive-case** adalah komplemen atau negasi dari **base-case**.
- Setiap algoritma rekursif selalu memiliki padanan dalam bentuk algoritma iteratif.

## 2. Contoh Program dengan menggunakan Rekursif

1. Membuat baris bilangan dari n hingga 1 Base-case: bilangan == 1

```
1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Scan(&n)
6     baris(n)
7 }
8
9 func baris(bilangan int){
10    if bilangan == 1 {
11        fmt.Println(1)
12    }else{
13        fmt.Println(bilangan)
14        baris(bilangan - 1)
15    }
16 }
```

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

```
1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Scan(&n)
6     fmt.Println(penjumlahan(n))
7 }
8
9 func penjumlahan(n int) int {
10    if n == 1 {
11        return 1
12    }else{
13        return n + penjumlahan(n-1)
14    }
15 }
```

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

```

1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Scan(&n)
6     fmt.Println(pangkat(n))
7 }
8
9 func pangkat(n int) int {
10     if n == 0 {
11         return 1
12     }else{
13         return 2 * pangkat(n-1)
14     }
15 }

```

4. Mencari nilai faktorial atau  $n!$  Base-case:  $n == 0$  atau  $n == 1$

```

1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Scan(&n)
6     fmt.Println(faktorial(n))
7 }
8
9 func faktorial(n int) int {
10     if n == 0 || n == 1 {
11         return 1
12     }else{
13         return n * faktorial(n-1)
14     }
15 }

```

## II. GUIDED

### 1. Guided 1

#### Source Code

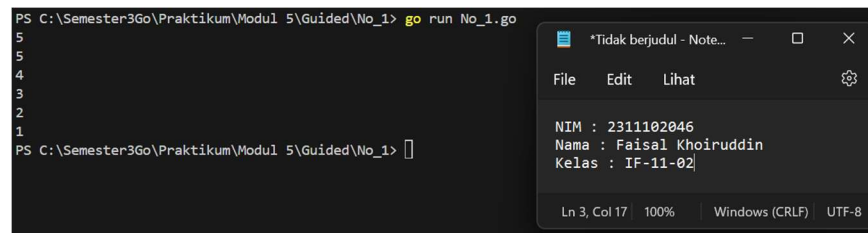
```
package main

import "fmt"

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

func baris(bilangan int) {
    if bilangan == 1 { // base case: Jika
bilangan sama dengan 1
        fmt.Println(1) // cetak angka 1
    } else { // jika bilangan lebih besar
dari 1
        fmt.Println(bilangan) // cetak
bilangan saat ini
        baris(bilangan - 1)    // panggil
fungsi baris dengan
    }
}
```

#### Screenshots Output



```
PS C:\Semester3Go\Praktikum\Modul 5\Guided\No_1> go run No_1.go
5
4
3
2
1
PS C:\Semester3Go\Praktikum\Modul 5\Guided\No_1>
```

\*Tidak berjudul - Note...  
File Edit Lihat  
NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-02  
Ln 3, Col 17 100% Windows (CRLF) UTF-8

Deskripsi:

Program tersebut merupakan program rekursif sederhana. Program tersebut. Pada program tersebut, jika user menginput angka 5, lalu program memanggil baris(5) dan mencetak angka 5, dilanjutkan memanggil baris(4). Pada baris(4) lalu mencetak angka 4 secara berulang sampai baris(1) kemudian mencetak angka 1 kemudian berhenti. Program tersebut menampilkan bilangan menurun yaitu 5, 4, 3, 2, 1.

- package main → paket utama program golang
- import "fmt" → mengimpor fmt
- func main() { → merupakan fungsi utama
- var n int → deklarasi variabel n dengan tipe data int
- fmt.Scan(&n) → membaca input pengguna
- baris(n) → memanggil fungsi rekursif 'baris'
- func baris(bilangan int) { → deklarasi fungsi baris dengan bilangan sebagai argument dengan tipe data integer
- if bilangan == 1 { → base case: Jika bilangan sama dengan 1
- fmt.Println(1) → cetak angka 1
- } else { → jika bilangan lebih besar dari 1
- fmt.Println(bilangan) → cetak bilangan saat ini
- baris(bilangan - 1) → baris(bilangan - 1)

## 2. Guided 2

Source Code



```

package main

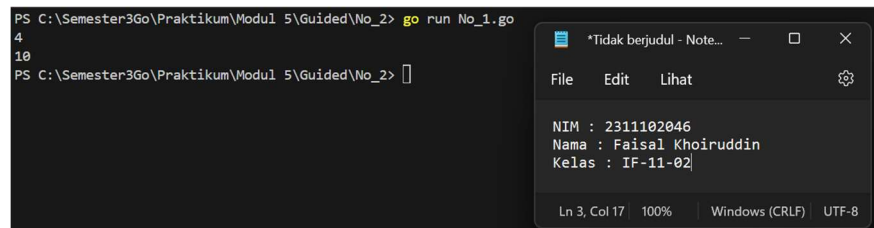
import "fmt"

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)
    }
}

```

### Screenshots Output



The screenshot shows a terminal window with the following content:

```

PS C:\Semester3Go\Praktikum\Modul 5\Guided\No_2> go run No_1.go
4
10
PS C:\Semester3Go\Praktikum\Modul 5\Guided\No_2>

```

In the background, a Notepad window titled '\*Tidak berjudul - Note...' is open, displaying the following text:

```

NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-02

```

The status bar at the bottom of the Notepad window indicates 'Ln 3, Col 17', '100%', 'Windows (CRLF)', and 'UTF-8'.

### Deskripsi:

Program tersebut merupakan program rekursif sederhana. Program tersebut Pada program tersebut, jika nilai  $n$  lebih besar dari 1, maka fungsi mengembalikan nilai  $n$  ditambah hasil dari penjumlahan( $n-1$ ). Pada fungsi ini terus memanggil dirinya sendiri dengan nilai  $n$  yang berkurang 1 hingga  $n$  sama dengan 1 maka rekursi akan berhenti.

Program tersebut menampilkan hasil yaitu angka 10 yang berasal dari  $4 + 3 + 2 + 1$ .

- `package main` ➔ paket utama program golang
- `import "fmt"` ➔ mengimpor `fmt`
- `func main() {` ➔ merupakan fungsi utama
- `var n int` ➔ deklarasi variabel `n` dengan tipe data integer
- `fmt.Scan(&n)` ➔ membaca input pengguna
- `fmt.Println(penjumlahan(n))` ➔ menampilkan (`penjumlahan(n)`)
- `func penjumlahan(n int) int {` ➔ deklarasi fungsi `penjumlahan` dengan parameter `n` dengan tipe data integer
- `if n == 1 {` ➔ percabangan jika `if` pada variabel `n` sama dengan 1
- `return 1` ➔ maka mengembalikan 1
- `} else {` ➔ kalau tidak
- `return n + penjumlahan(n-1)` ➔ mengembalikan `n + penjumlahan(n-1)`

### 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"
    "strings"
)

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

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

```

fibs := make([]int, n+1)
for i := 0; i <= n; i++ {
    fibs[i] = fibonacci(i)
}

colWidth := 6
tableWidth := (n + 2) * colWidth
line := strings.Repeat("-", tableWidth)

fmt.Println(line)
fmt.Printf("| %-3s ", "n ")
for i := 0; i <= n; i++ {
    fmt.Printf("| %-3d ", i)
}
fmt.Println("|")
fmt.Println(line)
fmt.Printf("| %-3s ", "Sn")
for i := 0; i <= n; i++ {
    fmt.Printf("| %-3d ", fibs[i])
}
fmt.Println("|")
fmt.Println(line)
}

```

### Screenshots Output

```

PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_1> go run No_1.go
Masukkan angka: 10
-----
| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
-----
| Sn | 0 | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
-----
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_1>

```

\*Tidak berjudul - Not...

File Edit Lihat

NIM : 2311102046  
 Nama : Faisal Khoiruddin  
 Kelas : IF-11-02

Ln 3, Col 17 100% Windows (CRLF) UTF-8

Deskripsi:

Program tersebut merupakan program fungsi rekursif pada deret Fibonacci. Program tersebut membaca angka dari pengguna, menghitung deret Fibonacci hingga angka tersebut. Program tersebut menampilkan deret Fibonacci.

- `package main` → paket utama program golang
- `import` → mengimport
- `"fmt"` → mengimpor `fmt`
- `"strings"` → mengimpor `fmt`
- `func fibonacci(n int) int {` → fungsi Fibonacci dengan parameter `n` dengan tipe data integer
- `if n <= 1 {` → percabangan if jika `n` kurang sama dengan 1
- `return n` → mengembalikan `n`
- `return fibonacci(n-1) + fibonacci(n-2)` → mengembalikan `fibonacci(n-1) + fibonacci(n-2)`
- `func main() {` → merupakan fungsi utama
- `var n int` → deklarasi `v` dengan tipe data integer
- `fmt.Print("Masukkan angka: ")` → menampilkan statement untuk memasukkan angka
- `fmt.Scan(&n)` → membaca input pengguna
- `fibs := make([]int, n+1)` → membuat slice untuk menyimpan bilangan fibonacci
- `for i := 0; i <= n; i++ {` → perulangan untuk menghitung setiap bilangan fibonacci hingga `n`
- `fibs[i] = fibonacci(i)` → `fibs[i] = fibonacci(i)`
- `colWidth := 6` → mengatur lebar kolom tabel

- `tableWidth := (n + 2) * colWidth` → lebar table berdasarkan jumlah kolom
- `line := strings.Repeat("-", tableWidth)` → membuat garis horizontal untuk tabel
- `fmt.Println(line)` → menampilkan garis horizontal ke- 1
- `fmt.Printf("| %-3s ", "n ")` → mencetak header kolom n
- `for i := 0; i <= n; i++ {` → perulangan untuk mencetak angka 0 hingga n sebagai header kolom
- `fmt.Printf("| %-3d ", i)` → mencetak ("| %-3d ", i)
- `fmt.Println("|")` → menampilkan garis horizontal kolom
- `fmt.Println(line)` → mencetak garis horizontal kedua
- `fmt.Printf("| %-3s ", "Sn")` → mencetak header kolom Sn untuk bilangan fibonacci
- `for i := 0; i <= n; i++ {` → perulangan untuk mencetak setiap bilangan Fibonacci yang dihitung
- `fmt.Printf("| %-3d ", fibs[i])` → menampilkan ("| %-3d ", fibs[i])
- `fmt.Println("|")` → menampilkan garis horizontal
- `fmt.Println(line)` → menampilkan garis horizontal

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

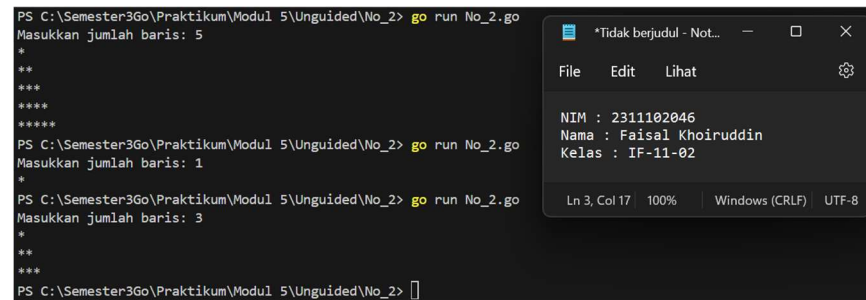
```
package main

import "fmt"

func cetakBintang(n int) {
    if n <= 0 {
        return
    }
    cetakBintang(n - 1)
    for i := 0; i < n; i++ {
        fmt.Print("*")
    }
    fmt.Println()
}

func main() {
    var n int
    fmt.Print("Masukkan jumlah baris: ")
    fmt.Scanln(&n)
    cetakBintang(n)
}
```

## Screenshots Output



```
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_2> go run No_2.go
Masukkan jumlah baris: 5
*
**
***
****
*****
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_2> go run No_2.go
Masukkan jumlah baris: 1
*
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_2> go run No_2.go
Masukkan jumlah baris: 3
*
**
***
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_2>
```

\*Tidak berjudul - Not...

File Edit Lihat

NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-02

Ln 3, Col 17 100% Windows (CRLF) UTF-8

## Deskripsi:

Program tersebut merupakan program untuk menampilkan pola bintang berikut menggunakan fungsi rekursif. Program tersebut meminta pengguna memasukkan jumlah baris dari pola Bintang tersebut. Program tersebut menampilkan pola Bintang sesuai jumlah baris yang diinputkan.

- package main ➔ paket utama pada program golang
- import "fmt" ➔ import fmt
- func cetakBintang (n int) { ➔ deklarasi fungsi cetak bintang dengan paramater n bertipe data integer
- if n <= 0 { ➔ percabangan if jika n kurang sama dengan 0
- return ➔ return
- cetakBintang(n - 1) ➔ cetakBintang(n - 1)
- for i := 0; i < n; i++ { ➔ perulangan for untuk i := 0; i < n; i++(increment)
- fmt.Print("\*\*") ➔ mencetak bintang
- fmt.Println() ➔ mencetak baris baru
- func main() { ➔ merupakan fungsi utama
- var n int ➔ deklarasi variabel n dengan tipe data integer



- `fmt.Print("Masukkan jumlah baris: ")` → menampilkan statement untuk memasukkan jumlah baris
- `fmt.Scanln(&n)` → membaca input pengguna dan menyimpan di variabel `n`
- `cetakBintang(n)` → memanggil fungsi `cetakBintang` dengan parameter `n`

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 faktor(n, i int) {
    if i > n {
        return
    }
}
```

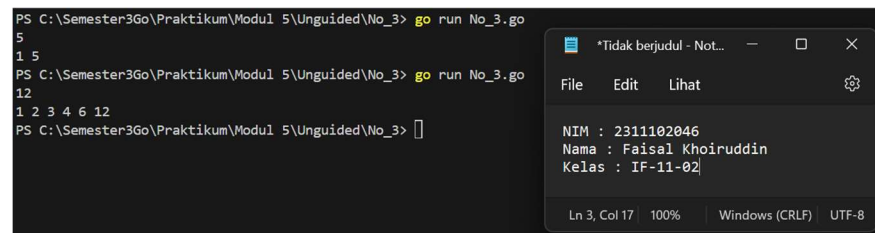
```

        if n%i == 0 {
            fmt.Printf("%d ", i)
        }
        faktor(n, i+1)
    }

func main() {
    var n int
    fmt.Scanln(&n)
    faktor(n, 1)
}

```

### Screenshots Output



```

PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_3> go run No_3.go
5
1 5
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_3> go run No_3.go
12
1 2 3 4 6 12
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_3>

```

Notepad window content:

```

NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-02

```

### Deskripsi:

Program tersebut merupakan program menampilkan faktor bilangan dari suatu N, atau bilangan yang apa saja yang habis membagi N. Program tersebut meminta pengguna menginput suatu bilangan dan menyimpan input ke variabel n. Program tersebut menampilkan faktor bilangan atau bilangan yang habis membagi N

- package main → paket utama pada program golang
- import "fmt" → import fmt
- func faktor(n, i int) { → deklarasi fungsi faktor dengan parameter n dan i dengan tipe data integer
- if i > n { → percabangan jika i lebih dari 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 descending(n int) {
    if n < 1 {
        return
    }
    fmt.Printf("%d ", n)
    descending(n - 1)
}

func ascending(n int, current int) {
    if current > n {
        return
    }
    fmt.Printf("%d ", current)
    ascending(n, current+1)
}
```

```
func main() {
    var n int
    fmt.Scanln(&n)
    descending(n)
    ascending(n, 2)
}
```

### Screenshots Output

```
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_4> go run No_4.go
5
5 4 3 2 1 2 3 4 5
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_4> go run No_4.go
9
9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_4>
```

### Deskripsi:

Program tersebut merupakan program menampilkan barisan bilangan tertentu. Program tersebut meminta pengguna menginput suatu bilangan positif. Program tersebut menampilkan barisan bilangan  $n$  menuju 1 hingga Kembali ke  $n$ . Pada output tersebut, user menginputkan angka 5 dan 9, sehingga menampilkan 5 4 3 2 1 2 3 4 5 dan 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9

- package main ➔ paket utama pada program golang
- import "fmt" ➔ import fmt
- func descending(n int){ ➔ deklarasi fungsi descending dengan parameter bertipe data integer
- if  $n < 1$  { ➔ percabangan if jika  $n$  lebih dari 1
- return ➔ rerurn
- fmt.Printf("%d ", n) ➔ mencetak nilai  $n$

- `descending(n - 1)` → memanggil fungsi `descending` dengan nilai `n-1`
- `func ascending(n int, current int){` → fungsi `ascending` dengan parameter `n` dan `current` bertipe data integer
- `if current > n {` → percabangan `if` jika `current` lebih besar dari `n`
- `return` □ `return`
- `fmt.Printf("%d ", current)` → mencetak nilai `current`
- `ascending(n, current + 1)` → memanggil fungsi `ascending` dengan nilai `current + 1`
- `func main() {` → merupakan fungsi utama
- `var n int =` deklarasi variabel `n` dengan tipe data integer
- `fmt.Scanln(&n)` → membaca input pengguna dan menyimpan di variabel `n`
- `descending(n)` → memanggil fungsi `descending` dengan parameter `n`
- `ascending(n, 2)` → memanggil fungsi `ascending` dengan parameter `n` dimulai dari 2

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 bil_ganjil(n, current int) {
    if current > n {
        return
    }
    if current%2 != 0 {
        fmt.Printf("%d ", current)
    }
    bil_ganjil(n, current+1)
}

func main() {
    var n int
    fmt.Scanln(&n)
    bil_ganjil(n, 1)
}
```

## Screenshots Output

```
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_5> go run No_5.go
5
1 3 5
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_5> go run No_5.go
20
1 3 5 7 9 11 13 15 17 19
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_5>
```

## Deskripsi:

Program tersebut merupakan program menampilkan barisan bilangan ganjil. Program tersebut meminta pengguna menginput suatu bilangan positif. Program tersebut menampilkan barisan bilangan ganjil dari 1 hingga n. Pada program tersebut menampilkan 1 3 5 untuk inputan 5 dan 1 3 5 7 9 11 13 15 17 19 untuk inputan 20.

- package main → paket utama pada program golang
- import "fmt" → import fmt
- func bil\_ganjil(n, current int) { → fungsi bil\_ganjil dengan parameter n, current dengan tipe data integer
- if current > n { → percabangan if jika current > n
- return → return
- if current%2 != 0 { → percabangan if untuk memeriksa current adalah bilangan ganjil
- fmt.Printf("%d ", current) → mencetak bilangan ganjil
- bil\_ganjil(n, current+1) → pemanggilan rekursif dengan menambah current
- func main() { → merupakan fungsi utama
- var n int → deklarasi variabel n dengan tipe data integer
- fmt.Scanln(&n) → menyimpan input pengguna di variabel n
- bil\_ganjil(n, 1) → memanggil fungsi bil\_ganjil dengan parameter n dan current mulai dari 1

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

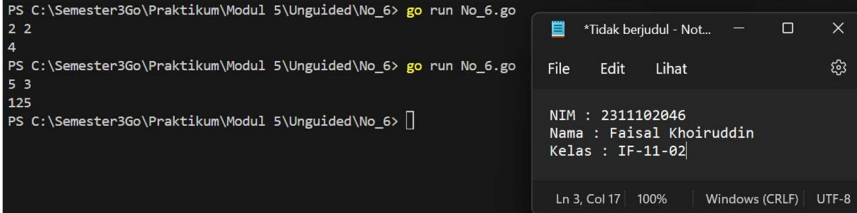
```
package main

import "fmt"

func pangkat(x, y int) int {
    if y == 0 {
        return 1
    }
    return x * pangkat(x, y-1)
}

func main() {
    var x, y int
    fmt.Scanf("%d %d", &x, &y)
    result := pangkat(x, y)
    fmt.Printf("%d\n", result)
}
```

### Screenshots Output



```
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_6> go run No_6.go
2 2
4
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_6> go run No_6.go
5 3
125
PS C:\Semester3Go\Praktikum\Modul 5\Unguided\No_6> 
```

\*Tidak berjudul - Notepad

File Edit Lihat

NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-02

Ln 3, Col 17 | 100% | Windows (CRLF) UTF-8



### Deskripsi:

Program tersebut merupakan program menampilkan hasil pangkat dari dua buah bilangan. Pengguna diminta memasukkan bilangan dimana bilangan pertama merupakan bilangan yang dipangkatkan dan bilangan kedua merupakan pangkatnya. Program tersebut menampilkan hasil bilangan x dipangkatkan y.

- `package main` → paket utama pada program golang
- `import "fmt"` → import fmt
- `func pangkat(x, y int) int {` → fungsi pangkat dengan parameter x dan y dengan tipe data integer
- `if y == 0 {` → percabangan if jika y sama dengan 0
- `return 1` → return 1
- `return x * pangkat(x, y-1)` → rekursi: x dikali hasil pangkat (x, y-1)
- `func main() {` → merupakan fungsi utama
- `var x, y int` → deklarasi variabel x dan y dengan tipe data integer
- `fmt.Scanf("%d %d", &x, &y)` → membaca input x dan y dari user dan menyimpan pada variabel x dan y
- `result := pangkat(x, y)` → memanggil fungsi pangkat x dan y
- `fmt.Printf("%d\n", result)` → menampilkan hasil