

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6
REKURSIF**



Disusun Oleh :

PRIESTY AMEILIANA MAULIDAH / 2311102175

IF-11-05

Dosen Pengampu :

ARIF AMRULLOH, S.KOM.,M.KOM

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

1.) 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.

2 Komponen Rekursif

Algoritma rekursif terdiri dari dua komponen utama:

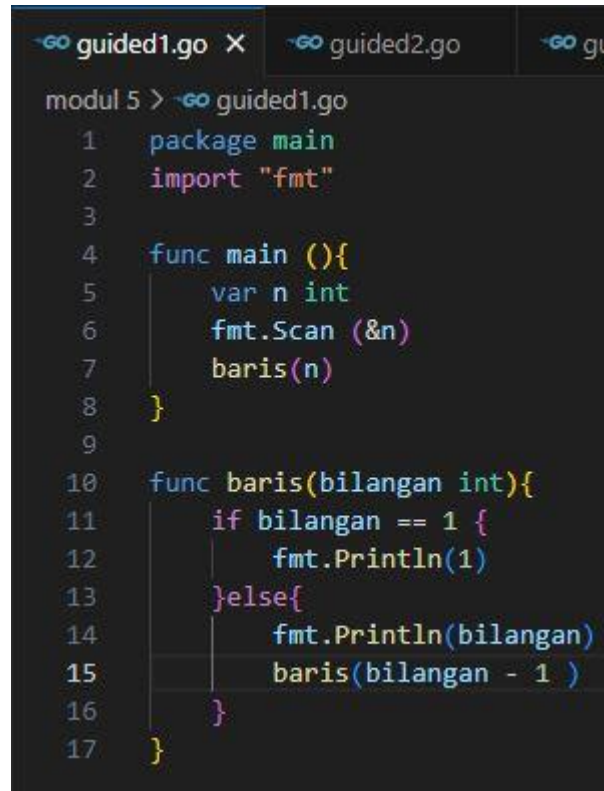
- Base-case (Basis), yaitu bagian untuk menghentikan proses rekursif dan menjadi komponen terpenting di dalam sebuah rekursif.
- Recursive-case, yaitu bagian pemanggilan subprogramnya

2.) GUIDED

Soal Studi Case

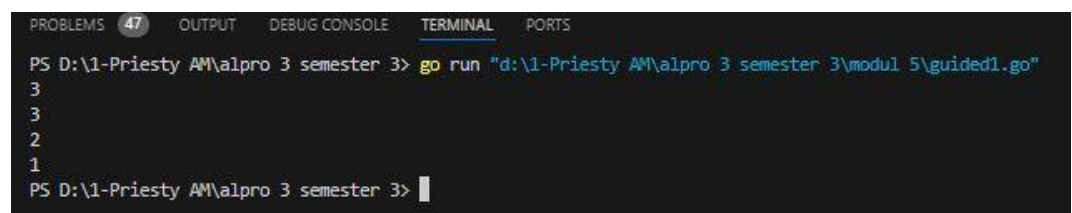
1Membuat baris bilangan dari n hingga 1 Base-case: bilangan == 1

Sourcecode



```
modul 5 > go guided1.go
1  package main
2  import "fmt"
3
4  func main(){
5      var n int
6      fmt.Scan(&n)
7      baris(n)
8  }
9
10 func baris(bilangan int){
11     if bilangan == 1 {
12         fmt.Println(1)
13     }else{
14         fmt.Println(bilangan)
15         baris(bilangan - 1 )
16     }
17 }
```

Screenshoot Output



```
PROBLEMS 47 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\guided1.go"
3
3
2
1
PS D:\1-Priesty AM\alpro 3 semester 3> |
```

Deskripsi Program

Deklarasi paket main dan mengimpor pustaka fmt, yang diperlukan untuk operasi input-output (1/0).pustaka fmt digunakan untuk membaca nilai dari pengguna dan menampilkan hasil ke layar.

Fungsi main

Fungsi utama main()berfungsi sebagai titik masuk program untuk menerima input dari pengguna. Pada awalnya,variable n dideklarasikan

dengan tipe int untuk menyimpan bilangan bulat yang akan dimasukkan oleh pengguna

1. Dengan `fmt.Scan(&n)`, program membaca input yang dimasukkan pengguna dan menyimpannya di variabel `n`.
2. Setelah nilai `n` diterima, program memanggil fungsi `baris()` dengan `n` sebagai argumen.

Fungsi Rekursif baris

Fungsi `baris()` menerima parameter bilangan bertipe int dan mencetak nilai tersebut. Fungsi ini memiliki dua komponen utama:

- **Basis kasus:** Jika nilai bilangan sama dengan 1, maka `fmt.Println(1)` akan mencetak angka 1 dan mengakhiri proses rekursi.
- **Proses rekursif:** Jika bilangan lebih besar dari 1, maka `fmt.Println(bilangan)` mencetak nilai bilangan saat ini, dan kemudian memanggil `baris(bilangan - 1)` untuk melanjutkan rekursi dengan nilai yang lebih kecil.

Output

Jika pengguna memasukkan angka 3, program akan mencetak angka-angka secara berurutan dari 3 hingga 1

3

2

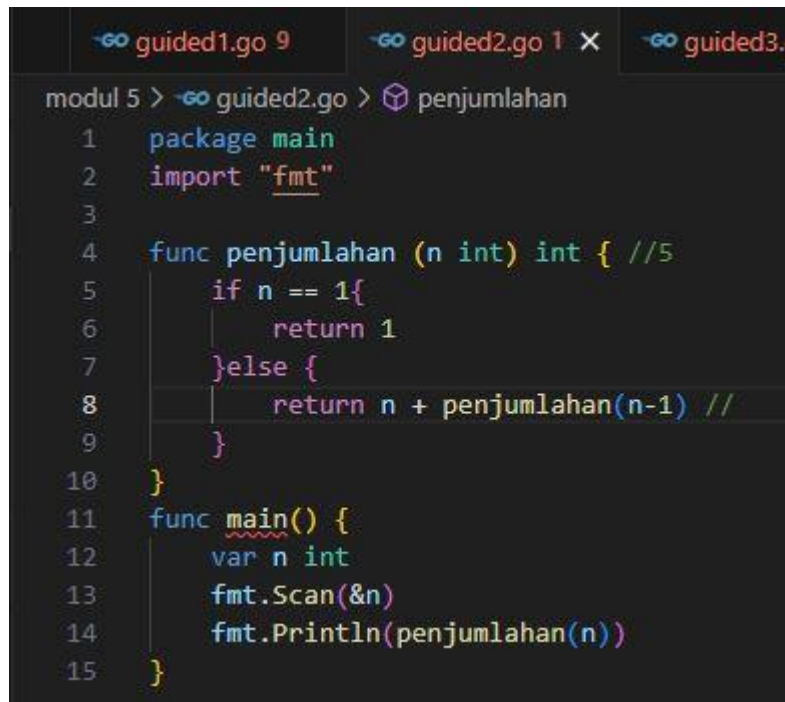
1

GUIDED

Soal Studi Case

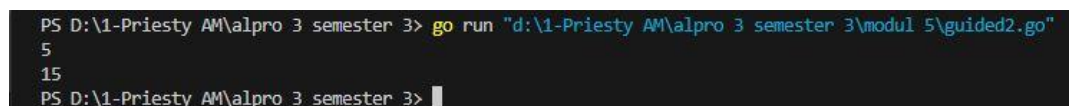
2. Menghitung hasil penjumlahan 1 hingga `n` Base-case: `n == 1`

Sourcecode



```
modul 5 > go guided2.go > penjumlahan
1 package main
2 import "fmt"
3
4 func penjumlahan (n int) int { //5
5     if n == 1{
6         return 1
7     }else {
8         return n + penjumlahan(n-1) //
9     }
10 }
11 func main() {
12     var n int
13     fmt.Scan(&n)
14     fmt.Println(penjumlahan(n))
15 }
```

Screenshoot Output



```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\guided2.go"
5
15
PS D:\1-Priesty AM\alpro 3 semester 3> |
```

Deskripsi Program

mendeklarasikan paket main dan mengimpor pustaka fmt, yang berfungsi untuk operasi input-output (I/O), seperti membaca nilai dari pengguna dan menampilkan hasil ke layar.

Fungsi penjumlahan

Fungsi penjumlahan(n int) int bertugas untuk menghitung jumlah deret bilangan dari 1 hingga n dengan pendekatan rekursif.

1. Parameter: Fungsi ini menerima parameter n bertipe int, yang merupakan batas akhir deret bilangan yang akan dijumlahkan.
2. Basis Kasus: Jika n bernilai 1, fungsi akan mengembalikan nilai 1, karena jumlah dari deret yang hanya terdiri dari satu angka yaitu 1.
3. Proses Rekursi: Jika n lebih besar dari 1, fungsi akan mengembalikan nilai n ditambah hasil pemanggilan fungsi penjumlahan(n-1). Ini membuat fungsi memanggil dirinya sendiri dengan nilai n-1, sehingga secara

bertahap, nilai-nilai sebelumnya ikut dijumlahkan satu per satu hingga mencapai basis kasus.

Fungsi main

Fungsi main() adalah fungsi utama program yang menerima input dari pengguna dan menampilkan hasil akhir dari proses penjumlahan.

1. Variabel n dideklarasikan dengan tipe int untuk menyimpan bilangan bulat yang dimasukkan.
2. Dengan `fmt.Scan(&n)`, program membaca input dari pengguna dan menyimpannya di variabel n.
3. Fungsi `penjumlahan(n)` kemudian dipanggil untuk menghitung hasil penjumlahan deret bilangan hingga n, dan hasilnya dicetak dengan `fmt.Println()`.

Contoh Cara Kerja Program dan Output

Jika pengguna memasukkan angka 5, proses akan berjalan sebagai berikut:

- `penjumlahan(5)` akan memanggil `5 + penjumlahan(4)`
- `penjumlahan(4)` memanggil `4 + penjumlahan(3)`
- `penjumlahan(3)` memanggil `3 + penjumlahan(2)`
- `penjumlahan(2)` memanggil `2 + penjumlahan(1)`
- `penjumlahan(1)` mengembalikan nilai 1 (basis kasus).

Proses penjumlahan ini kemudian menghasilkan nilai:

$$5 + 4 + 3 + 2 + 1 = 15$$

Sehingga, ketika pengguna memasukkan nilai 5, output yang dihasilkan adalah:

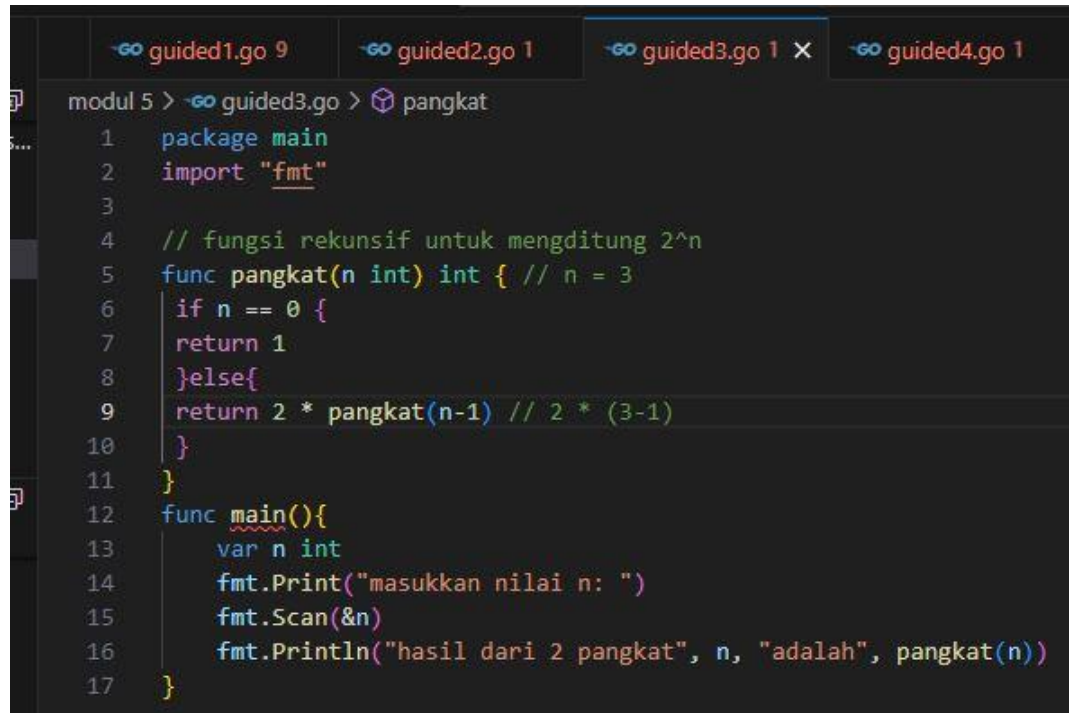
15

GUIDED

Soal Studi Case

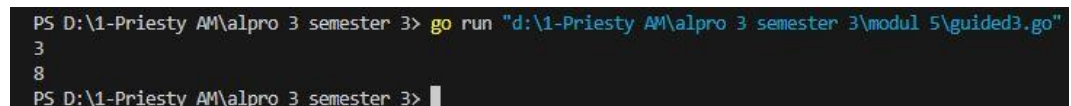
3. Mencari dua pangkat n atau 2(Base-case: `n == 0`

Sourcecode



```
modul 5 > go guided3.go > pangkat
1 package main
2 import "fmt"
3
4 // fungsi rekursif untuk menghitung 2^n
5 func pangkat(n int) int { // n = 3
6     if n == 0 {
7         return 1
8     }else{
9         return 2 * pangkat(n-1) // 2 * (3-1)
10    }
11 }
12 func main(){
13     var n int
14     fmt.Print("masukkan nilai n: ")
15     fmt.Scan(&n)
16     fmt.Println("hasil dari 2 pangkat", n, "adalah", pangkat(n))
17 }
```

Screenshoot Output



```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\guided3.go"
3
8
PS D:\1-Priesty AM\alpro 3 semester 3>
```

Deskripsi Program

mendeklarasikan paket main dan mengimpor pustaka fmt yang diperlukan untuk operasi input-output (I/O), seperti membaca input dari pengguna dan menampilkan hasil ke layar.

Fungsi Rekursif pangkat

Fungsi pangkat(n int) int bertugas menghitung nilai 2^n dengan pendekatan rekursif.

1. **Parameter:** Fungsi menerima parameter n bertipe int, yang menunjukkan pangkat yang akan digunakan dalam perhitungan 2^n .
2. **Basis Kasus:** Jika n bernilai 0, fungsi akan mengembalikan 1, karena $2^0 = 1$.

3. **Proses Rekursi:** Jika n lebih besar dari 0, fungsi mengembalikan hasil $2 * \text{pangkat}(n-1)$, yang mengalikan 2 dengan hasil $\text{pangkat}(n-1)$. Proses ini berlanjut hingga mencapai basis kasus.

Fungsi main

Fungsi `main()` adalah fungsi utama program yang berfungsi untuk menerima input dari pengguna dan menampilkan hasil perhitungan pangkat.

1. Variabel n dideklarasikan dengan tipe `int` untuk menyimpan pangkat yang akan digunakan.
2. Program menampilkan pesan untuk memasukkan nilai n , dan dengan `fmt.Scan(&n)`, program membaca nilai input dari pengguna.
3. Fungsi `pangkat(n)` kemudian dipanggil untuk menghitung nilai 2^n , dan hasilnya dicetak menggunakan `fmt.Println()`.

Contoh Cara Kerja Program dan Output

Jika pengguna memasukkan nilai 3, proses perhitungan akan berjalan sebagai berikut:

- `pangkat(3)` akan memanggil $2 * \text{pangkat}(2)$
- `pangkat(2)` akan memanggil $2 * \text{pangkat}(1)$
- `pangkat(1)` akan memanggil $2 * \text{pangkat}(0)$
- `pangkat(0)` akan mengembalikan nilai 1 (basis kasus).

Selanjutnya, hasil-hasil perkalian dikembalikan secara berurutan sebagai berikut:

$$2 * 1 = 2 \quad (\text{dari pangkat}(1))$$

$$2 * 2 = 4 \quad (\text{dari pangkat}(2))$$

$$2 * 4 = 8 \quad (\text{dari pangkat}(3))$$

Sehingga, ketika pengguna memasukkan nilai 3, output yang dihasilkan adalah:

masukkan nilai n : 3

hasil dari 2 pangkat 3 adalah 8

GUIDED

Soal Studi Case

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

Sourcecode



```
modul 5 > go guided4.go >  faktorial
1  package main
2  import "fmt"
3
4  var n int
5  func faktorial (n int) int {
6      if n == 0 || n == 1 { // n = 3
7          return 1
8      } else {
9          return n * faktorial(n-1) // 3 * 2 * 1 = 6
10     }
11 }
12
13 func main () {
14     fmt.Scan(&n)
15     fmt.Println(faktorial(n))
16 }
```

Screenshoot Output



```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\guided4.go"
3
6
```

Deskripsi Program

mendeklarasikan paket main dan mengimpor pustaka fmt, yang digunakan untuk operasi input-output (I/O), seperti membaca input dari pengguna dan menampilkan hasil perhitungan.

Deklarasi Variabel n

Variabel n dideklarasikan secara global dengan tipe int, sehingga nilainya dapat diakses dari semua fungsi di dalam paket ini.

Fungsi Rekursif faktorial

Fungsi faktorial(`n int`) `int` bertugas menghitung nilai faktorial dari bilangan `n` menggunakan pendekatan rekursif.

1. Parameter: Fungsi ini menerima parameter `n`, yang merupakan bilangan yang akan dihitung faktorialnya.
2. Basis Kasus: Jika nilai `n` adalah 0 atau 1, fungsi akan mengembalikan 1, karena faktorial dari 0 dan 1 adalah 1.
3. Proses Rekursi: Jika `n` lebih dari 1, fungsi akan mengembalikan hasil `n * faktorial(n-1)`, yang mengalikan `n` dengan hasil pemanggilan fungsi faktorial(`n-1`). Proses ini berlanjut hingga mencapai basis kasus.

Fungsi main

Fungsi `main()` adalah fungsi utama program yang menerima input dari pengguna dan menampilkan hasil perhitungan faktorial.

1. Program meminta pengguna untuk memasukkan nilai `n`, dan `fmt.Scan(&n)` digunakan untuk membaca input dari pengguna dan menyimpannya di variabel `n`.
2. Fungsi faktorial(`n`) kemudian dipanggil untuk menghitung nilai faktorial dari `n`, dan hasilnya dicetak menggunakan `fmt.Println()`.

Contoh Cara Kerja Program dan Output

Jika pengguna memasukkan nilai 3, proses perhitungan faktorial akan berjalan sebagai berikut:

- faktorial(3) akan memanggil 3 * faktorial(2)
- faktorial(2) akan memanggil 2 * faktorial(1)
- faktorial(1) mengembalikan nilai 1 (basis kasus).

Proses perhitungan ini menghasilkan:

$$3 * 2 * 1 = 6$$

Sehingga, ketika pengguna memasukkan nilai 3, output yang dihasilkan adalah:

3.) UNGUIDED

Modul 6

Soal Studi Case

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 $sn = sn-1 + sn-2$. 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
sn	0	1	1	2	3	4	8	13	21	34	55

Sourcecode

```
modul 5 > guided1.go 9    guided2.go 1    guided3.go 1    guided4.go 1
1  package main
2
3  import (
4      "fmt"
5  )
6
7  func fibonacci(n int) int {
8      if n <= 0 {
9          return 0
10     } else if n == 1 {
11         return 1
12     }
13     return fibonacci(n-1) + fibonacci(n-2)
14 }
15
16 func main() {
17     for i := 0; i <= 10; i++ {
18         fmt.Printf("Fibonacci(%d) = %d\n", i, fibonacci(i))
19     }
20 }
```

Screenshoot Output

```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\unguided1.go"
Fibonacci(0) = 0
Fibonacci(1) = 1
Fibonacci(2) = 1
Fibonacci(3) = 2
Fibonacci(4) = 3
Fibonacci(5) = 5
Fibonacci(6) = 8
Fibonacci(7) = 13
Fibonacci(8) = 21
Fibonacci(9) = 34
Fibonacci(10) = 55
```

Deskripsi Program

Deklarasi Paket dan Import

Program dimulai dengan mendeklarasikan paket main dan mengimpor pustaka fmt, yang digunakan untuk operasi input-output (I/O), khususnya untuk menampilkan hasil perhitungan deret Fibonacci ke layar.

Fungsi Rekursif fibonacci

Fungsi fibonacci(n int) int bertugas menghitung angka ke-n dalam deret Fibonacci menggunakan pendekatan rekursif.

1. **Parameter:** Fungsi ini menerima parameter n, yaitu indeks angka dalam deret Fibonacci yang ingin dihitung.
2. **Basis Kasus:**
 - o Jika $n \leq 0$, fungsi mengembalikan 0.
 - o Jika $n == 1$, fungsi mengembalikan 1.

Ini sesuai dengan aturan deret Fibonacci, di mana deret dimulai dengan angka 0 dan 1.

3. **Proses Rekursi:** Jika n lebih besar dari 1, fungsi mengembalikan hasil fibonacci(n-1) + fibonacci(n-2), yang merupakan penjumlahan dari dua angka Fibonacci sebelumnya. Proses ini berlanjut secara rekursif hingga mencapai basis kasus.

Fungsi main

Fungsi main() adalah fungsi utama program yang menjalankan perhitungan deret Fibonacci dan mencetak hasilnya.

1. Program menggunakan loop for $i := 0; i \leq 10; i++$ untuk mengulang perhitungan Fibonacci dari indeks 0 hingga 10.
2. Dalam setiap iterasi, fibonacci(i) dipanggil untuk menghitung angka Fibonacci ke-i, dan hasilnya dicetak dengan fmt.Printf() dalam format Fibonacci(i) = hasil.

Contoh Cara Kerja Program dan Output

Pada setiap iterasi dari $i = 0$ hingga $i = 10$, nilai Fibonacci dihitung dan ditampilkan. Berikut adalah urutan hasil perhitungan yang akan dicetak oleh program:

```
Fibonacci(0) = 0
Fibonacci(1) = 1
Fibonacci(2) = 1
```

Fibonacci(3) = 2
 Fibonacci(4) = 3
 Fibonacci(5) = 5
 Fibonacci(6) = 8
 Fibonacci(7) = 13
 Fibonacci(8) = 21
 Fibonacci(9) = 34
 Fibonacci(10) = 55

UNGUIDED

Soal Studi Case

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	masukkan	Keluaran
1	5	* ** *** **** *****
2	1	*
3	3	* ** ***

Sourcecode

```
modul 5 > -go guided1.go 9    -go guided2.go 1    -go guided3.go 1

modul 5 > -go unguided2.go > ...
1  package main
2
3  import (
4      "fmt"
5  )
6
7  func printStars(n int) {
8      if n == 0 {
9          return
10     }
11     printStars(n - 1)
12     for i := 0; i < n; i++ {
13         fmt.Print("*")
14     }
15     fmt.Println()
16 }
17
18 func main() {
19     var n int
20     fmt.Print("Masukkan nilai N: ")
21     fmt.Scan(&n)
22     printStars(n)
23 }
24
```

Screenshoot Output

```
lpro 3 semester 3\modul 5\unguided2.go"
Masukkan nilai N: 5
*
**
***
****
*****
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\unguided2.go"
Masukkan nilai N: 1
*
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\unguided2.go"
Masukkan nilai N: 3
*
**
***
```

Deskripsi Program

Deklarasi Paket dan Import

Program dimulai dengan mendeklarasikan paket main dan mengimpor pustaka fmt, yang digunakan untuk operasi input-output (I/O), khususnya untuk membaca input dari pengguna dan mencetak pola bintang ke layar.

Fungsi Rekursif printStars

Fungsi printStars(n int) bertugas mencetak pola bintang secara rekursif hingga baris ke-n.

1. **Basis Kasus:** Jika n bernilai 0, fungsi berhenti dan tidak melakukan apa-apa (return). Ini adalah kondisi yang menghentikan proses rekursi.
2. **Proses Rekursi:** Jika n lebih besar dari 0, fungsi akan terlebih dahulu memanggil dirinya sendiri dengan nilai n-1. Setelah pemanggilan rekursif selesai, program kemudian mencetak n bintang di baris tersebut menggunakan loop for yang berjalan n kali untuk mencetak * secara horizontal.

Dengan cara ini, fungsi mencetak baris dengan jumlah bintang bertambah satu pada setiap level rekursi, mulai dari 1 hingga n.

Fungsi main

Fungsi main() adalah fungsi utama program yang menerima input dari pengguna dan memulai pencetakan pola bintang.

1. Program meminta pengguna untuk memasukkan nilai n, yang merupakan jumlah baris bintang yang ingin ditampilkan.
2. printStars(n) kemudian dipanggil untuk mencetak pola bintang dengan jumlah baris sesuai nilai n.

Contoh Cara Kerja Program dan Output

Jika pengguna memasukkan nilai n = 5, fungsi akan bekerja sebagai berikut:

1. printStars(5) akan memanggil printStars(4), yang memanggil printStars(3), dan seterusnya hingga mencapai printStars(0), yang menghentikan rekursi.
2. Setelah mencapai basis kasus, setiap pemanggilan fungsi rekursif akan mencetak * sebanyak nilai n yang sesuai dengan level pemanggilan, sehingga membentuk pola bintang berikut:

Masukkan nilai N: 5

*

**

Masukkan nilai N: 1

*

Masukkan nilai N: 3

*

**

UNGUIDED

Soal Studi Case

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	masukkan	Keluaran
1	5	1 5
2	12	1 2 3 4 6 12

Sourcecode


```
modul 5 > -go guided1.go 9 -go guided2.go 1 -go guided3.go 1 -go guided4.g
...
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var N int
9     fmt.Print("Masukkan bilangan bulat positif N: ")
10    fmt.Scan(&N)
11    fmt.Print("Faktor dari ", N, ": ")
12    printFactors(N, 1)
13 }
14
15 func printFactors(N, i int) {
16     if i > N {
17         return
18     }
19     if N%i == 0 {
20         fmt.Print(i, " ")
21     }
22     printFactors(N, i+1)
23 }
```

Screenshoot Output

```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\unguided3.go"
Masukkan bilangan bulat positif N: 5
Faktor dari 5: 1 5
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\unguided3.go"
Masukkan bilangan bulat positif N: 12
Faktor dari 12: 1 2 3 4 6 12
PS D:\1-Priesty AM\alpro 3 semester 3> |
```

Deskripsi Program

Deklarasi Paket dan Import

Program dimulai dengan mendeklarasikan paket main dan mengimpor pustaka fmt, yang digunakan untuk operasi input-output (I/O) seperti membaca input dari pengguna dan mencetak hasil perhitungan ke layar.

Fungsi main

Fungsi `main()` adalah fungsi utama yang bertugas untuk:

1. Menerima Input: Program meminta pengguna memasukkan bilangan bulat positif `N`.
2. Menampilkan Informasi Awal: Program menampilkan pesan yang menunjukkan bahwa hasil berikutnya adalah faktor dari `N`.
3. Memanggil Fungsi `printFactors`: Fungsi `printFactors(N, 1)` dipanggil untuk memulai pencarian dan pencetakan faktor-faktor dari `N`, mulai dari angka 1.

Fungsi Rekursif `printFactors`

Fungsi `printFactors(N, i int)` bertugas untuk mencari dan mencetak faktor dari bilangan `N` menggunakan pendekatan rekursif.

- Parameter:
 - `N` adalah bilangan yang akan dicari faktornya.
 - `i` adalah variabel counter, memulai dari 1 dan bertambah satu setiap kali fungsi dipanggil kembali.
- Basis Kasus: Jika `i` melebihi `N`, fungsi berhenti (return) karena tidak ada faktor yang lebih besar dari `N` itu sendiri.
- Proses Rekursi:
 - Fungsi memeriksa apakah `i` adalah faktor dari `N` dengan mengecek apakah `N % i == 0`. Jika benar, maka `i` dicetak sebagai faktor.
 - Fungsi kemudian memanggil dirinya sendiri dengan `printFactors(N, i+1)` untuk memeriksa faktor berikutnya.

Contoh Cara Kerja Program dan Output

Jika pengguna memasukkan `N = 5`, prosesnya akan sebagai berikut:

1. `printFactors(5, 1)` memeriksa apakah 1 adalah faktor dari 5. Karena `5 % 1 == 0`, 1 dicetak.
2. `printFactors(5, 2)` hingga `printFactors(5, 4)` tidak mencetak apa-apa karena 5 tidak habis dibagi 2, 3, atau 4.
3. `printFactors(5, 5)` mencetak 5 karena 5 adalah faktor dari dirinya sendiri.

Output:

Masukkan bilangan bulat positif N: 5

Faktor dari 5: 1 5

Jika pengguna memasukkan N = 12, maka output akan menunjukkan faktor-faktor dari 12:

Output:

Masukkan bilangan bulat positif N: 12

Faktor dari 12: 1 2 3 4 6 12

UNGUIDED

Soal Studi Case

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	Masukkan	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

```
modul 5 > go unguided4.go > ...
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func printSequence(n int) {
8     if n == 0 {
9         return
10    }
11    fmt.Print(n, " ")
12    printSequence(n - 1)
13    fmt.Print(n, " ")
14 }
15
16 func main() {
17     var N int
18     fmt.Print("Masukkan bilangan bulat positif N: ")
19     fmt.Scan(&N)
20     printSequence(N)
21     fmt.Println()
22 }
23
```

Screenshoot Output

```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\unguided4.go"
Masukkan bilangan bulat positif N: 5
5 4 3 2 1 1 2 3 4 5
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\unguided4.go"
Masukkan bilangan bulat positif N: 9
9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9
PS D:\1-Priesty AM\alpro 3 semester 3> |
```

Deskripsi Program

Deklarasi Paket dan Import

Program dimulai dengan mendeklarasikan paket main dan mengimpor pustaka fmt, yang digunakan untuk operasi input-output (I/O), seperti membaca input dari pengguna dan mencetak hasil ke layar.

Fungsi Rekursif printSequence

Fungsi `printSequence(n int)` bertugas untuk mencetak urutan angka dari `n` hingga 1, dan setelah itu, mencetak angka dari 1 hingga `n` kembali.

- **Parameter:** Fungsi ini menerima parameter `n`, yang merupakan bilangan bulat positif yang akan diproses.
- **Basis Kasus:** Jika nilai `n` adalah 0, fungsi akan berhenti dan tidak melakukan apa-apa (`return`). Ini adalah kondisi yang menghentikan proses rekursi.
- **Proses Rekursi:**
 1. Fungsi mencetak nilai `n` diikuti dengan spasi.
 2. Fungsi memanggil dirinya sendiri dengan `printSequence(n - 1)`, yang mencetak urutan angka yang lebih kecil.
 3. Setelah kembali dari pemanggilan rekursif, fungsi mencetak nilai `n` lagi diikuti dengan spasi. Ini menciptakan pola di mana angka dari `n` hingga 1 dicetak terlebih dahulu, dan setelah mencapai 0, angka dari 1 hingga `n` dicetak kembali.

Fungsi main

Fungsi `main()` adalah fungsi utama yang mengendalikan eksekusi program:

1. Program meminta pengguna memasukkan nilai bilangan bulat positif `N`.
2. Nilai `N` dibaca menggunakan `fmt.Scan(&N)`.
3. Fungsi `printSequence(N)` dipanggil untuk mencetak urutan angka sesuai dengan nilai yang dimasukkan.
4. Setelah selesai, `fmt.Println()` digunakan untuk mencetak newline agar output lebih rapi.

Contoh Cara Kerja Program dan Output

Jika pengguna memasukkan `N = 5`, prosesnya adalah sebagai berikut:

1. `printSequence(5)` mencetak 5 dan kemudian memanggil `printSequence(4)`.
2. Proses berulang hingga `printSequence(1)` yang mencetak 1, dan akhirnya `printSequence(0)` yang menghentikan rekursi.

3. Setelah mencapai 0, proses kembali mencetak angka dari 1 hingga 5.

Output:

Masukkan bilangan bulat positif N: 5

5 4 3 2 1 1 2 3 4 5

Jika pengguna memasukkan N = 9, maka outputnya akan menjadi:

Output:

Masukkan bilangan bulat positif N: 9

9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9

UNGUIDED

Soal Studi Case

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	Masukkan	keluaran
1	5	1 3 5
2	20	1 3 5 7 9 11 13 15 17 19

Sourcecode

```
guided4.go 1  GO unguided1.go 1  GO unguided2.go modul 5 1  GO unguided3.go

modul 5 > GO unguided5.go > ...
1 package main
2
3 import (
4     "fmt"
5 )
6
7 // Fungsi rekursif untuk mencetak bilangan ganjil
8 func printOddNumbers(n int, current int) {
9     if current > n {
10         return
11     }
12     if current%2 != 0 {
13         fmt.Print(current, " ")
14     }
15     printOddNumbers(n, current+1)
16 }
17
18 func main() {
19     var N int
20     fmt.Print("Masukkan bilangan bulat positif N: ")
21     fmt.Scan(&N)
22
23     fmt.Println("Barisan bilangan ganjil dari 1 hingga", N, ":")
24     printOddNumbers(N, 1)
25 }
```

Screenshoot Output

```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\guided5.go"
Masukkan bilangan bulat positif N: 5
Barisan bilangan ganjil dari 1 hingga 5 :
1 3 5
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\guided5.go"
Masukkan bilangan bulat positif N: 20
Barisan bilangan ganjil dari 1 hingga 20 :
1 3 5 7 9 11 13 15 17 19
PS D:\1-Priesty AM\alpro 3 semester 3> 
```

Deskripsi Program

Deklarasi Paket dan Import

Program dimulai dengan mendeklarasikan paket main dan mengimpor pustaka fmt, yang digunakan untuk operasi input-output (I/O), seperti membaca input dari pengguna dan mencetak hasil ke layar.

Fungsi Rekursif printOddNumbers

Fungsi `printOddNumbers(n int, current int)` bertugas untuk mencetak bilangan ganjil dari 1 hingga `n`.

- **Parameter:**
 - `n`: Bilangan bulat positif yang menentukan batas atas.
 - `current`: Angka saat ini yang sedang diperiksa, dimulai dari 1.
- **Basis Kasus:**
 - Jika `current` lebih besar dari `n`, fungsi akan berhenti (`return`), menandakan bahwa semua angka hingga `n` telah diperiksa.
- **Proses Rekursi:**
 1. Fungsi memeriksa apakah `current` adalah bilangan ganjil dengan menggunakan kondisi `if current%2 != 0`. Jika `true`, angka tersebut dicetak.
 2. Fungsi kemudian memanggil dirinya sendiri dengan `printOddNumbers(n, current+1)`, yang akan memeriksa angka berikutnya.

Fungsi main

Fungsi `main()` adalah fungsi utama yang mengontrol alur program:

1. Program meminta pengguna untuk memasukkan nilai bilangan bulat positif `N`.
2. Nilai `N` dibaca menggunakan `fmt.Scan(&N)`.
3. Program mencetak pesan yang menunjukkan bahwa berikutnya adalah urutan bilangan ganjil dari 1 hingga `N`.
4. Fungsi `printOddNumbers(N, 1)` dipanggil untuk mencetak bilangan ganjil sesuai dengan nilai yang dimasukkan.

Contoh Cara Kerja Program dan Output

Jika pengguna memasukkan `N = 5`, prosesnya adalah sebagai berikut:

1. `printOddNumbers(5, 1)` memeriksa 1, yang merupakan bilangan ganjil, sehingga mencetak 1.

2. Fungsi kemudian memanggil `printOddNumbers(5, 2)`, yang tidak mencetak apa-apa karena 2 adalah bilangan genap.
3. Proses ini berlanjut hingga `printOddNumbers(5, 5)`, yang mencetak 5.

Output:

Masukkan bilangan bulat positif N: 5

Barisan bilangan ganjil dari 1 hingga 5:

1 3 5

Jika pengguna memasukkan N = 20, maka outputnya adalah:

Output:

Masukkan bilangan bulat positif N: 20

Barisan bilangan ganjil dari 1 hingga 20:

1 3 5 7 9 11 13 15 17 19

UNGUIDED

Soal Studi Case

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	Masukkan	keluaran
1	2 2	4
2	5 3	125

Sourcecode

```
guided4.go 1  ~GO unguided1.go 1  ~GO unguided2.go modul 5 1  ~GO unguided3.go 1  ~GO

modul 5 > ~GO unguided6.go > ...
1  package main
2
3  import (
4      "fmt"
5  )
6
7  // Fungsi rekursif untuk menghitung x^y
8  func power(x int, y int) int {
9      if y == 0 {
10         return 1 // Basis: x^0 = 1
11     }
12     return x * power(x, y-1) // Rekurens: x^y = x * x^(y-1)
13 }
14
15 func main() {
16     var x, y int
17     fmt.Print("Masukkan bilangan bulat x: ")
18     fmt.Scan(&x)
19     fmt.Print("Masukkan bilangan bulat y: ")
20     fmt.Scan(&y)
21
22     result := power(x, y)
23     fmt.Printf("Hasil dari %d dipangkatkan %d adalah %d\n", x, y, result)
24 }
25
```

Screenshoot Output

```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\unguided6.go"
Masukkan bilangan bulat x: 2
Masukkan bilangan bulat y: 2
Hasil dari 2 dipangkatkan 2 adalah 4
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 5\unguided6.go"
Masukkan bilangan bulat x: 5
Masukkan bilangan bulat y: 3
Hasil dari 5 dipangkatkan 3 adalah 125
PS D:\1-Priesty AM\alpro 3 semester 3> |
```

Deskripsi Program

□ Deklarasi Paket dan Import

Program dimulai dengan mendeklarasikan paket main dan mengimpor pustaka fmt, yang digunakan untuk melakukan operasi input-output (I/O), seperti membaca input dari pengguna dan mencetak hasil ke layar.

□ Fungsi Rekursif power

Fungsi power(x int, y int) int bertanggung jawab untuk menghitung nilai x^y dengan menggunakan rekursi.

- **Parameter:**

- x: bilangan bulat yang akan dipangkatkan.
- y: bilangan bulat yang merupakan eksponen.
- **Basis Kasus:** Jika yyy adalah 0, fungsi akan mengembalikan 1 (return 1). Ini mengikuti aturan matematis bahwa setiap bilangan yang dipangkatkan 0 adalah 1.
- **Proses Rekursi:** Jika yyy lebih besar dari 0, fungsi mengembalikan hasil dari xxx dikalikan dengan hasil pemanggilan fungsi power(x, y-1). Ini berarti bahwa $xy = x \times x(y-1)$ $x^y = x \times x^{(y-1)}$ $xy = x \times x(y-1)$, sehingga fungsi akan terus memanggil dirinya sendiri hingga mencapai basis kasus.

□ **Fungsi main**

Fungsi main() adalah titik awal eksekusi program:

- Program meminta pengguna untuk memasukkan nilai bilangan bulat xxx dan eksponen yyy.
- Nilai xxx dan yyy dibaca menggunakan `fmt.Scan(&x)` dan `fmt.Scan(&y)`.
- Fungsi `power(x, y)` kemudian dipanggil untuk menghitung hasil perpangkatan.
- Hasilnya dicetak ke layar menggunakan `fmt.Printf()` dalam format yang terstruktur.

□ **Contoh Cara Kerja Program dan Output**

- Jika pengguna memasukkan nilai $x=2$ $x=2$ dan $y=2$ $y=2$, program akan melakukan proses sebagai berikut:
 - Memanggil `power(2, 2)`, yang akan memanggil `power(2, 1)`, dan seterusnya hingga `power(2, 0)`, yang mengembalikan 1.
 - Proses kembali ke atas, menghitung 2×1 2×1 dan kemudian 2×2 2×2 , sehingga menghasilkan 4.
- Output yang dihasilkan:

Masukkan bilangan bulat x: 2

Masukkan bilangan bulat y: 2

Hasil dari 2 dipangkatkan 2 adalah 4

- Jika pengguna memasukkan $x=5$ dan $y=3$:
 - Proses yang sama akan terjadi, memanggil `power(5, 3)` hingga `power(5, 0)`, menghasilkan $5 \times 5 \times 5 = 125$ $\times 5 \times 5 = 125 \times 5 \times 5 = 125$.
- Output yang dihasilkan:

Masukkan bilangan bulat x: 5

Masukkan bilangan bulat y: 3

Hasil dari 5 dipangkatkan 3 adalah 125