

**LAPORAN PRAKTIKUM PEMROGRAMAN
BERORIENTASI OBJEK**

MODUL V

REKURSIF



Oleh:

NAMA : ARVAN MURBIYANTO

NIM : 2311102074

KELAS : IF-11-02

S1 TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

I. DASAR TEORI

Rekursif adalah teknik di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan masalah. Fungsi rekursif biasanya terdiri dari dua bagian utama: basis kasus (base case), yang menentukan kapan fungsi harus berhenti memanggil dirinya sendiri, dan rekursi (recursive case), yang memecah masalah menjadi sub-masalah yang lebih kecil dan memanggil fungsi tersebut kembali. Rekursi sangat berguna untuk masalah yang memiliki sifat berulang atau dapat dipecah menjadi versi yang lebih sederhana dari masalah yang sama, seperti perhitungan faktorial, bilangan Fibonacci, atau traversing struktur data seperti pohon atau graf.

	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	

Catatan:

1. **Rekursi** adalah salah satu alternatif untuk menggantikan struktur perulangan dengan memanfaatkan subprogram, seperti fungsi atau prosedur.
2. Untuk menghentikan rekursi, diperlukan struktur percabangan (if-then) yang menentukan kapan rekursi harus berhenti.
3. **Base-case** adalah kondisi yang menghentikan proses rekursif. Ini merupakan komponen paling penting dalam rekursi dan harus ditentukan terlebih dahulu sebelum membuat program rekursif. Tanpa base-case, rekursi tidak akan bisa berjalan dengan benar.
4. **Recursive-case** adalah kondisi di mana fungsi memanggil dirinya sendiri untuk menyelesaikan masalah. Recursive-case merupakan kebalikan dari base-case.
5. Setiap algoritma rekursif dapat diubah menjadi bentuk iteratif yang setara.

Komponen Rekursif yang Dimodifikasi:

Algoritma rekursif terdiri dari dua komponen utama:

- **Base-case (Basis):** Kondisi yang menghentikan proses rekursif dan merupakan elemen krusial dalam rekursi.
- **Recursive-case:** Kondisi di mana fungsi memanggil dirinya sendiri untuk melanjutkan pemecahan masalah.

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

SOAL

NO 1.

Source code:

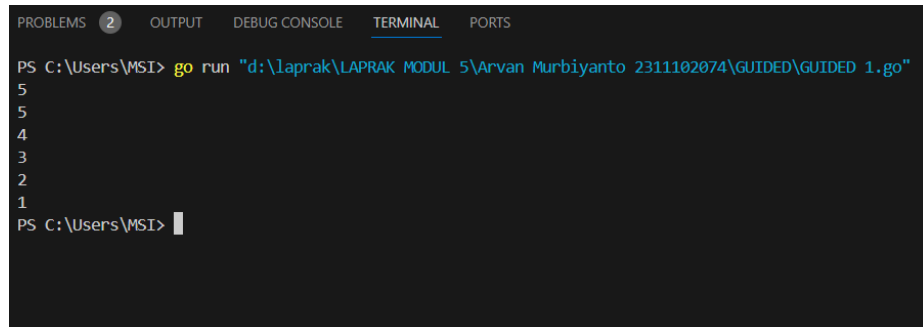
```
package main

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)
    }
}
```

Output :



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbiyanto 2311102074\GUIDED\GUIDED 1.go"
5
5
4
3
2
1
PS C:\Users\MSI> |
```

Deskripsi :

Program ini meminta input berupa bilangan bulat n dari pengguna, kemudian mencetak bilangan tersebut secara berurutan menurun hingga 1 menggunakan fungsi rekursif bernama `baris`. Fungsi `baris(bilangan int)` mencetak nilai dari bilangan, lalu memanggil dirinya sendiri dengan mengurangi nilai bilangan sebesar 1. Rekursi akan berhenti ketika bilangan mencapai 1, di mana angka 1 dicetak terakhir, menandakan akhir proses. Program ini menunjukkan cara kerja rekursi dalam mencetak bilangan secara menurun

NO 2.

Source code:

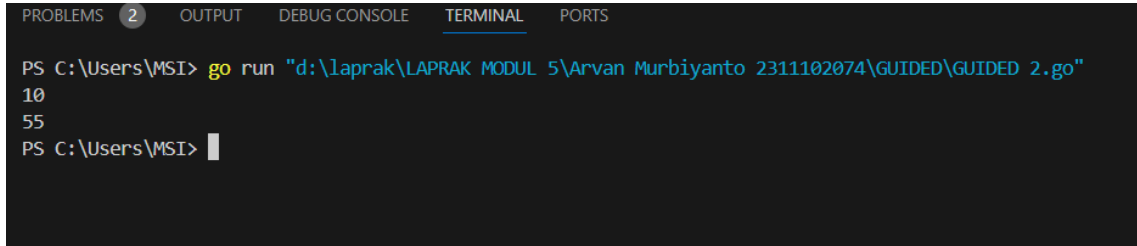
```
package main

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)
    }
}
```

Output :



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbiyanto 2311102074\GUIDED\GUIDED 2.go"
10
55
PS C:\Users\MSI>
```

Deskripsi :

Program ini menghitung total penjumlahan dari 1 hingga n menggunakan rekursi. Setelah pengguna memasukkan nilai n , fungsi rekursif `penjumlahan(n)` akan menambahkan nilai n dengan hasil penjumlahan dari $n-1$, dan proses ini berlanjut hingga mencapai nilai 1 sebagai kondisi dasar. Hasil akhir penjumlahan kemudian ditampilkan sebagai output.

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

SOAL 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.

Source code :

```
unguided (1).go X GUIDED 1.go 1 GUIDED 2.go 1
D: > laprak > LAPRAK MODUL 5 > Arvan Murbianto 2311102074 > UNGUIDED > -go unguided (1).go > baris
1 package main
2
3 import "fmt"
4
5 func baris(bilangan int) {
6     if bilangan == 1 {
7         fmt.Println(1)
8     } else {
9         fmt.Println(bilangan)
10        baris(bilangan - 1)
11    }
12 }
13
14 func fibonacci(n int) int {
15     if n <= 1 {
16         return n
17     }
18     return fibonacci(n-1) + fibonacci(n-2)
19 }
20
21 func main() {
22     var n int
23     fmt.Print("Masukkan nilai n: ")
24     fmt.Scan(&n)
25
26     fmt.Println("Urutan baris:")
27     baris(n)
28
29     fmt.Println("\nUrutan Fibonacci:")
30     for i := 0; i <= n; i++ {
31         fmt.Printf("Fibonacci ke-%d: %d\n", i, fibonacci(i))
32     }
33 }
34
```

Output :

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbianto 2311102074\UNGUIDED\unguided (1).go"
Masukkan nilai n: 10
Urutan baris:
10
9
8
7
6
5
4
3
2
1

Urutan Fibonacci:
Fibonacci ke-0: 0
Fibonacci ke-1: 1
Fibonacci ke-2: 1
Fibonacci ke-3: 2
Fibonacci ke-4: 3
Fibonacci ke-5: 5
Fibonacci ke-6: 8
Fibonacci ke-7: 13
Fibonacci ke-8: 21
Fibonacci ke-9: 34
Fibonacci ke-10: 55
PS C:\Users\MSI>
```

Deskripsi :

Program di atas menerima input dari pengguna berupa bilangan bulat n dan menampilkan deret Fibonacci hingga elemen ke- n . Fungsi `fibonacciSeries(n)` bertugas untuk mencetak setiap angka dalam deret Fibonacci dari 0 hingga n . Untuk menghitung nilai Fibonacci, fungsi ini memanggil fungsi rekursif `fibonacci(n)`, yang menggunakan rumus dasar: jika n adalah 0 atau 1, maka hasilnya adalah n . Untuk nilai yang lebih besar, hasilnya dihitung sebagai jumlah dari dua angka Fibonacci sebelumnya, yaitu `fibonacci(n-1) + fibonacci(n-2)`.

SOAL 2

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

Source code :

```

1 package main
2 import "fmt"
3
4 func printbintangStr(n int) string {
5     if n == 1 {
6         return ""
7     } else {
8         return printbintangStr(n-1) + "\n" + cetakBintang(n)
9     }
10 }
11 func cetakBintang(n int) string {
12     bintang := ""
13     for i := 0; i < n; i++ {
14         bintang += " "
15     }
16     return bintang
17 }
18 func printbintangDirect(n int) {
19     if n == 1 {
20         fmt.Println("")
21         return
22     }
23     printbintangDirect(n - 1)
24     for i := 0; i < n; i++ {
25         fmt.Print(" ")
26     }
27     fmt.Println()
28 }
29 func main() {
30     var n int
31     fmt.Print("Masukkan nilai N: ")
32     fmt.Scan(&n)
33     fmt.Println("\nHasil menggunakan string:")
34     fmt.Println(printbintangStr(n))
35     fmt.Println("\nHasil mencetak langsung:")
36     printbintangDirect(n)
37 }
```

Output :

```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbiyanto 2311102074\UNGUIDED\unguided (2).go"
Masukkan nilai N: 5

Hasil menggunakan string:
*
**
***
****
*****

Hasil mencetak langsung:
*
**
***
****
*****
PS C:\Users\MSI>
```

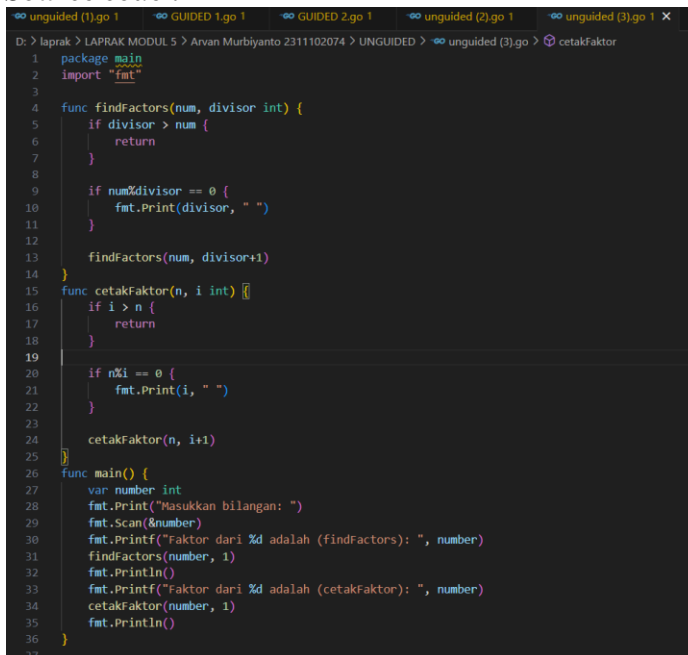
Deskripsi :

Program di atas dirancang untuk mencetak pola bintang secara bertingkat berdasarkan input dari pengguna. Pengguna diminta untuk memasukkan nilai n , dan kemudian program menggunakan fungsi rekursif `printbintang(n)` untuk menghasilkan pola bintang. Fungsi `printbintang(n)` memanggil dirinya sendiri untuk mencetak baris bintang sebelumnya, sebelum menambahkan baris bintang dengan panjang yang sesuai dengan nilai n . Proses ini berulang hingga mencapai kondisi dasar, yaitu ketika n sama dengan 1, di mana fungsi tersebut akan mengembalikan satu bintang

SOAL 3

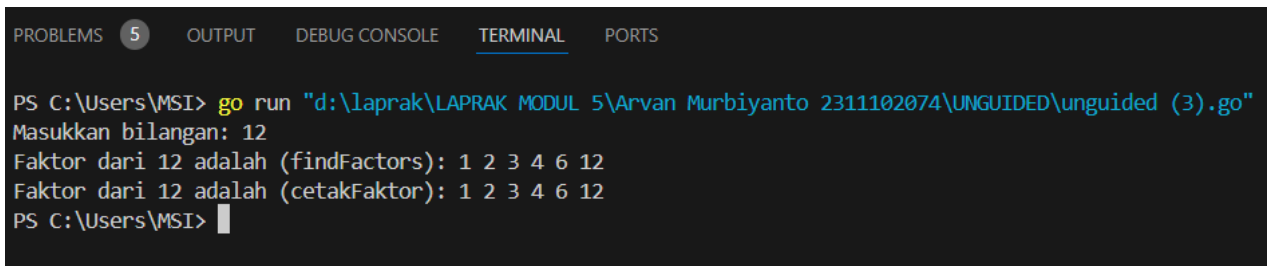
Buatlah program yang mengimplementasikan rekursif untuk menampilkan faktor bilangan dari suatu N , atau bilangan yang apa saja yang habis membagi N . Masukan terdiri dari sebuah bilangan bulat positif N . Keluaran terdiri dari barisan bilangan yang menjadi faktor dari N (terurut dari 1 hingga N ya).

Source code :



```
1 package main
2 import "fmt"
3
4 func findFactors(num, divisor int) {
5     if divisor > num {
6         return
7     }
8
9     if num%divisor == 0 {
10        fmt.Print(divisor, " ")
11    }
12
13    findFactors(num, divisor+1)
14}
15
16 func cetakFaktor(n, i int) {
17     if i > n {
18         return
19     }
20
21     if n%i == 0 {
22         fmt.Print(i, " ")
23     }
24
25     cetakFaktor(n, i+1)
26}
27
28 func main() {
29     var number int
30     fmt.Print("Masukkan bilangan: ")
31     fmt.Scan(&number)
32     fmt.Printf("Faktor dari %d adalah (findFactors): ", number)
33     findFactors(number, 1)
34     fmt.Println()
35     fmt.Printf("Faktor dari %d adalah (cetakFaktor): ", number)
36     cetakFaktor(number, 1)
37     fmt.Println()
38}
```

Output :



```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbiyanto 2311102074\UNGUIDED\unguided (3).go"
Masukkan bilangan: 12
Faktor dari 12 adalah (findFactors): 1 2 3 4 6 12
Faktor dari 12 adalah (cetakFaktor): 1 2 3 4 6 12
PS C:\Users\MSI>
```


Deskripsi :

Program di atas dirancang untuk mencari dan mencetak faktor-faktor dari bilangan bulat n yang dimasukkan oleh pengguna. Fungsi utama `main` menerima input n dan memanggil fungsi rekursif `cetakFaktor`, yang berfungsi untuk mencari faktor-faktor dari n . Proses pencarian faktor dilakukan secara berulang, dimulai dari 1 hingga n . Pada setiap iterasi, program memeriksa apakah bilangan i adalah faktor dari n dengan menggunakan kondisi $n \% i == 0$. Jika kondisi ini terpenuhi, maka i dicetak sebagai faktor dari n . Proses ini berlanjut hingga semua faktor dari 1 hingga n ditemukan.

SOAL 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

Source code :

```
unguided (4).go 2 X  unguided (5).go 3
D: > laprak > LAPRAK MODUL 5 > Arvan Murbiyanto 2311102074 > UNGUIDED > -go unguided (4).go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Print("Masukkan bilangan: ")
8     fmt.Scan(&n)
9
10    fmt.Printf("Barisan bilangan: ")
11    printSequence(n)
12    fmt.Println()
13 }
14
15 func printSequence(n int) {
16     if n == 1 {
17         fmt.Print(1, " ")
18     } else {
19         fmt.Print(n, " ")
20         printSequence(n - 1)
21         fmt.Print(n, " ")
22     }
23 }
24
```

Output :

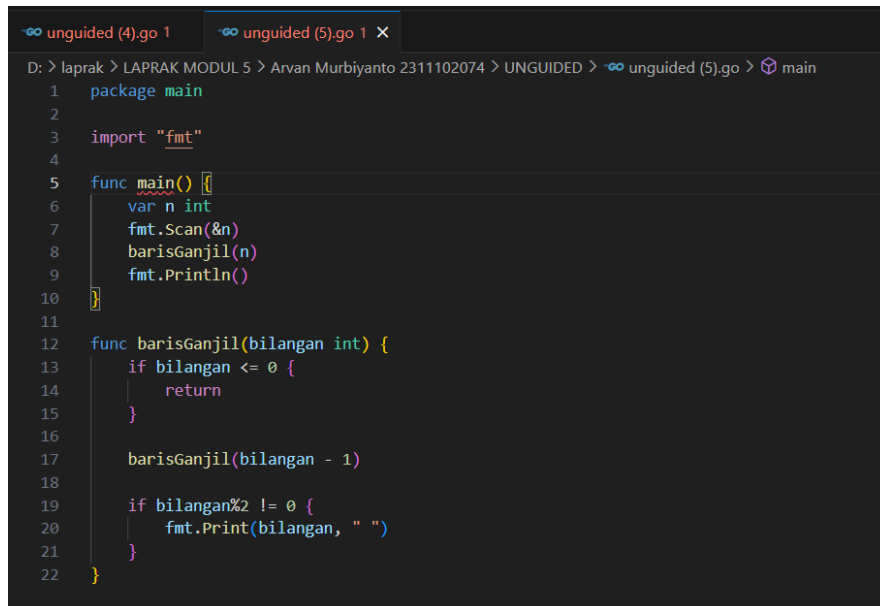
```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbiyanto 2311102074\UNGUIDED\unguided (4).go"
Masukkan bilangan: 5
Barisan bilangan: 5 4 3 2 1 2 3 4 5
PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbiyanto 2311102074\UNGUIDED\unguided (4).go"
Masukkan bilangan: 9
Barisan bilangan: 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS C:\Users\MSI>
```

Deskripsi :

Program di atas menggunakan rekursi untuk mencetak pola baris angka berdasarkan input dari pengguna. Fungsi `baris(bilangan int)` bertugas untuk mencetak angka secara berurutan dari nilai input hingga 1, kemudian kembali naik ke angka input tersebut. Ketika nilai mencapai 1, program mencetak angka 1 dan mulai melanjutkan pencetakan angka yang lebih besar hingga kembali ke angka awal yang dimasukkan oleh pengguna. Sebagai contoh, jika pengguna memasukkan angka 3, output yang dihasilkan akan menjadi: 3 2 1 2 3.

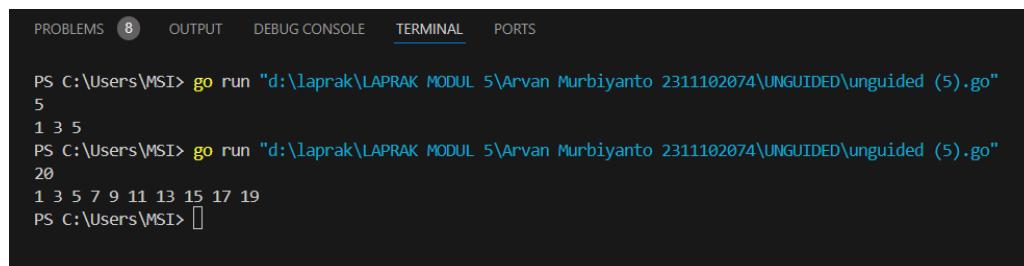
SOAL 5

Source code :



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Scan(&n)
8     barisGanjil(n)
9     fmt.Println()
10 }
11
12 func barisGanjil(bilangan int) {
13     if bilangan <= 0 {
14         return
15     }
16
17     barisGanjil(bilangan - 1)
18
19     if bilangan%2 != 0 {
20         fmt.Print(bilangan, " ")
21     }
22 }
```

Output :



```
PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbiyanto 2311102074\UNGUIDED\unguided (5).go"
5
1 3 5
PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbiyanto 2311102074\UNGUIDED\unguided (5).go"
20
1 3 5 7 9 11 13 15 17 19
PS C:\Users\MSI>
```

Penjelasan :

Program di atas dirancang untuk mencetak bilangan ganjil dari 1 hingga n menggunakan pendekatan rekursif. Pertama, pengguna diminta untuk memasukkan nilai n. Fungsi `barisGanjil` kemudian dipanggil dengan n sebagai argumen. Di dalam fungsi tersebut, jika nilai bilangan kurang dari atau sama dengan 0, fungsi akan berhenti. Jika tidak, fungsi akan memanggil dirinya sendiri dengan argumen `bilangan - 1`, sehingga menciptakan tumpukan rekursi. Setelah kembali dari pemanggilan rekursif, program memeriksa apakah bilangan adalah bilangan ganjil dengan menggunakan kondisi `bilangan % 2 != 0`. Jika kondisi ini terpenuhi, bilangan tersebut akan dicetak ke layar. Dengan cara ini, bilangan ganjil dari 1 hingga n dicetak dalam urutan yang meningkat.

SOAL 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.

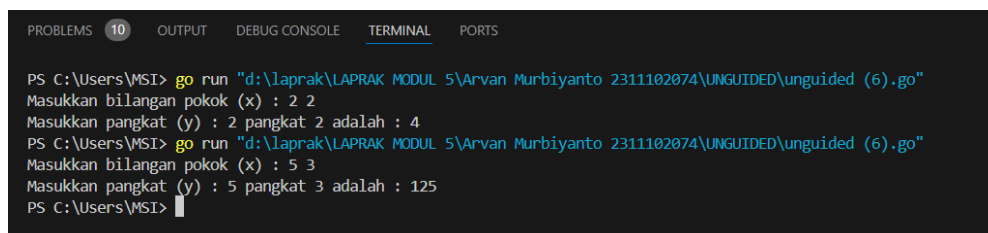
Catatan: diperbolehkan menggunakan asterik "*", tapi dilarang menggunakan import "math".

Source code :



```
1 package main
2
3 import "fmt"
4
5 func pangkat(x, y int) int {
6     if y == 0 {
7         return 1
8     } else {
9         return x * pangkat(x, y-1)
10    }
11 }
12
13 func main() {
14     var base, exponent int
15
16     fmt.Print("Masukkan bilangan pokok (x) : ")
17     fmt.Scan(&base)
18     fmt.Print("Masukkan pangkat (y) : ")
19     fmt.Scan(&exponent)
20
21     result := pangkat(base, exponent)
22     fmt.Printf("%d pangkat %d adalah : %d\n", base, exponent, result)
23 }
24
```

Output :



```
PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbiyanto 2311102074\UNGUIDED\unguided (6).go"
Masukkan bilangan pokok (x) : 2
Masukkan pangkat (y) : 2 pangkat 2 adalah : 4
PS C:\Users\MSI> go run "d:\laprak\LAPRAK MODUL 5\Arvan Murbiyanto 2311102074\UNGUIDED\unguided (6).go"
Masukkan bilangan pokok (x) : 5
Masukkan pangkat (y) : 3 pangkat 3 adalah : 125
PS C:\Users\MSI>
```

Penjelasan :

Program di atas adalah implementasi sederhana dari fungsi rekursif yang digunakan untuk menghitung hasil dari bilangan x dipangkatkan dengan y. Di dalam fungsi main, program meminta pengguna untuk memasukkan dua bilangan bulat, yaitu x dan y. Setelah menerima input tersebut, program memanggil fungsi pangkat, yang melakukan perhitungan pangkat secara rekursif. Jika nilai y sama dengan 0, fungsi akan mengembalikan 1, karena setiap bilangan yang dipangkatkan 0 adalah 1. Jika tidak, fungsi akan mengalikan x dengan hasil dari pemanggilan pangkat(x, y-1), yang terus berlanjut hingga mencapai kondisi dasar. Setelah proses rekursi selesai, hasil dari perhitungan pangkat tersebut akan dicetak ke layar.

