

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL V

REKURSIF



Disusun Oleh :

ALTHAF TEGAR SOFYAN / 2311102217

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

5.1 Definisi 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 berikut.

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

5.2 Komponen Rekursif

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

II. GUIDED

1. Buatlah sebuah program baris bilangan n hingga 1

Source Code

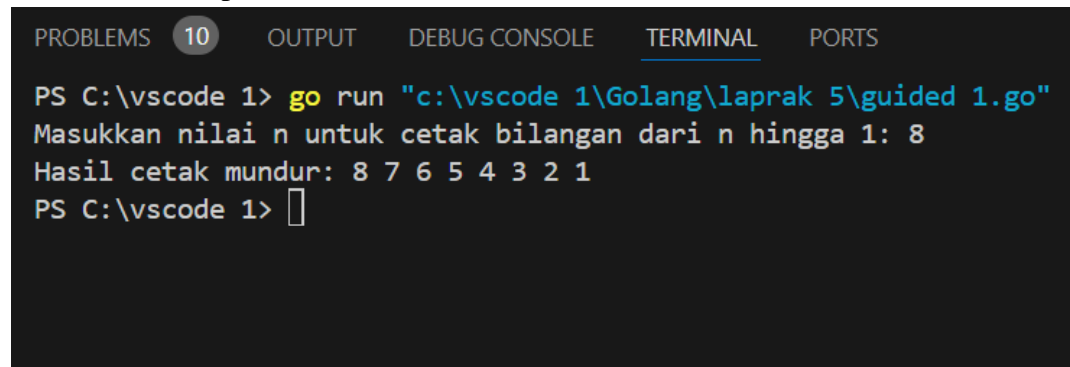
```
package main

import "fmt"

//Fungsi untuk mencetak bilangan dari n hingga 1
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)
}
```

Screenshot Output



```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\vscode 1> go run "c:\vscode 1\Golang\laprak 5\guided 1.go"
Masukkan nilai n untuk cetak bilangan dari n hingga 1: 8
Hasil cetak mundur: 8 7 6 5 4 3 2 1
PS C:\vscode 1> 
```

Deskripsi Program: Program diatas menggunakan fungsi rekursif yang menerima satu parameter, yaitu n (bilangan bulat). Jika n sama dengan 1 program akan mencetak n dan mengembalikan fungsi, tapi jika n tidak sama dengan satu ,maka program akan mencetak n lalu memanggil fungsi dengan arugumen n-1. Setelah itu, program akan mencetak "Hasil cetak mundur: " diikuti dengan memanggil fungsi cetakMundur(n). Saat fungsi cetakMundur(n) dipanggil, program akan melakukan rekursi hingga n bernilai 1, lalu akan mencetak bilangan mulai dari n hingga 1 secara mundur.

2. Buatlah sebuah program untuk menghitung penjumlahan 1 hingga n
- Source Code

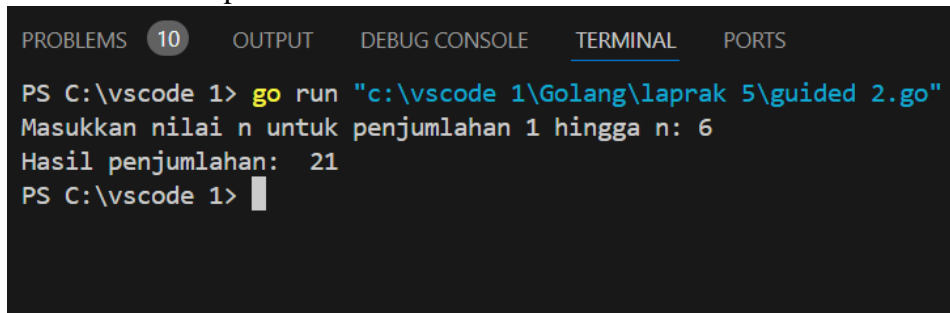
```
package main

import "fmt"

//Fungsi untuk menghitung penjumlahan 1 hingga n
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



```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\vscode 1> go run "c:\vscode 1\Golang\laprak 5\guided 2.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 6
Hasil penjumlahan: 21
PS C:\vscode 1> █
```

Deskripsi Program: Program diatas menggunakan fungsi rekursif yang menerima satu parameter, yaitu n (bilangan bulat). Jika n sama dengan 1, maka fungsi akan mengembalikan nilai 1. Tetapi, jika n tidak sama dengan 1, maka fungsi akan mengembalikan nilai n ditambah dengan hasil pemanggilan rekursif.

3. Buatlah program untuk mencari dua pangkat n atau 2^n

Source Code

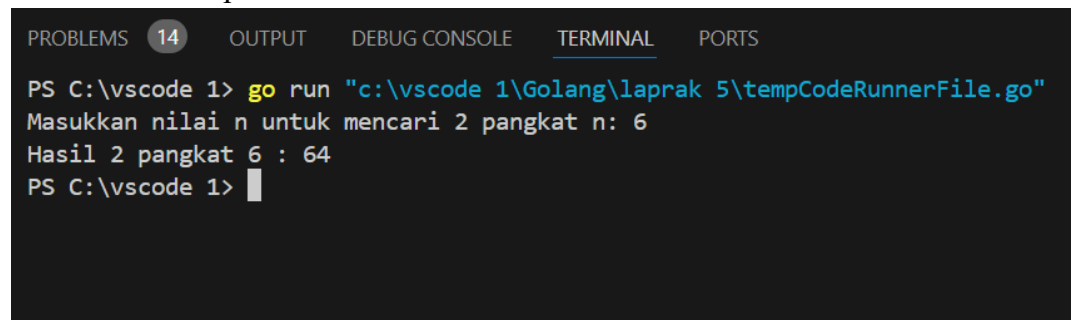
```
package main

import "fmt"

// Fungsi untuk mencari 2 pangkat n
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



```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\vscode 1> go run "c:\vscode 1\Golang\laprak 5\tempCodeRunnerFile.go"
Masukkan nilai n untuk mencari 2 pangkat n: 6
Hasil 2 pangkat 6 : 64
PS C:\vscode 1> █
```

Deskripsi Program: Program diatas menggunakan fungsi rekursif yang menerima satu parameter, yaitu n (bilangan bulat). Jika n sama dengan 0, maka fungsi akan mengembalikan nilai 1. Tetapi, jika n tidak sama dengan 0, maka fungsi akan mengembalikan nilai 2 dikali dengan hasil pemanggilan rekursif.

4. Buatlah program untuk mencari nilai faktorial atau n!

Source Code

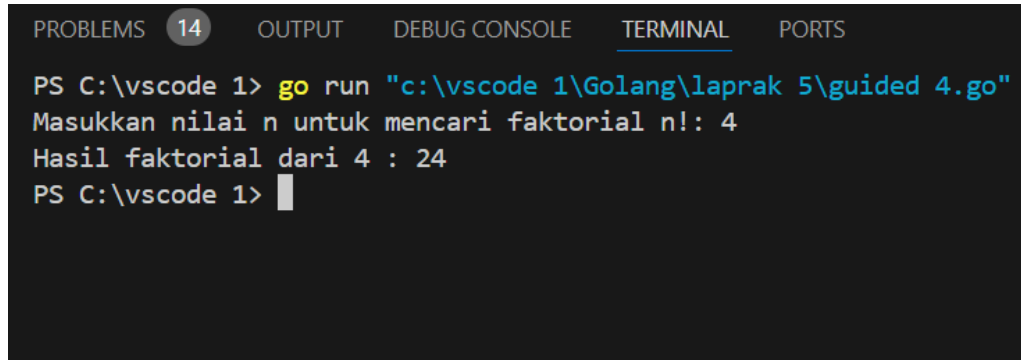
```
package main

import "fmt"

// Fungsi untuk menghitung faktorial n!
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))
}
```

Screenshot Output



The screenshot shows a terminal window with the following output:

```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\vscode 1> go run "c:\vscode 1\Golang\laprak 5\guided 4.go"
Masukkan nilai n untuk mencari faktorial n!: 4
Hasil faktorial dari 4 : 24
PS C:\vscode 1> 
```

Deskripsi Program: Program diatas menggunakan fungsi rekursif yang menerima satu parameter, yaitu n (bilangan bulat). Jika n sama dengan 0 atau 1, maka fungsi akan mengembalikan nilai 1. Tetapi, jika n tidak sama dengan 0 atau 1, maka fungsi akan mengembalikan nilai n dikali dengan hasil pemanggilan rekursif.

III. UNGUIDED

1. Buatlah sebuah program yang mengimplementasikan fungsi rekursif deret fibonanci dengan nilai suku ke-0 dan suku ke-1, dan nilai suku ke-n selanjutnya adalah hasil penjumlahan dua suku berikutnya. Secara umum dapat diformulasasikan $S_n = S_{n-1} + S_{n-2}$.

Source Code

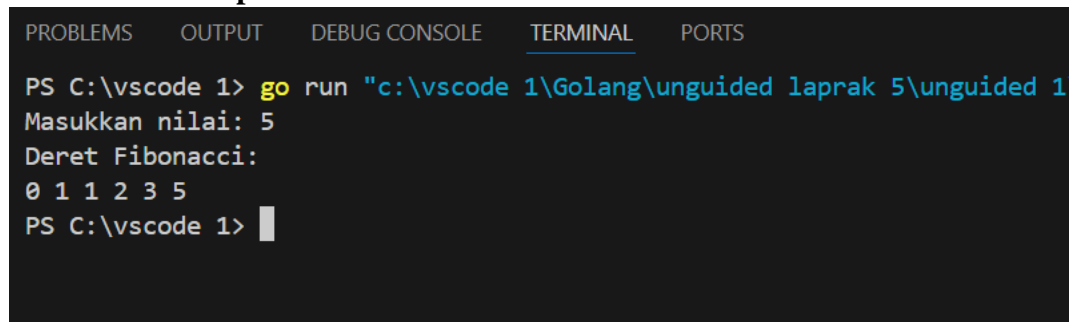
```
package main

import "fmt"

// Deret Fibonanci
func deret_fibonacci217(n217 int) int {
    if n217 == 0 {
        return 0
    } else if n217 == 1 {
        return 1
    } else {
        return deret_fibonacci217(n217-1) +
        deret_fibonacci217(n217-2)
    }
}

func main() {
    var n int
    fmt.Print("Masukkan nilai: ")
    fmt.Scanln(&n)
    fmt.Println("Deret Fibonacci:")
    for i := 0; i <= n; i++ {
        fmt.Print(deret_fibonacci217(i), " ")
    }
    fmt.Println()
}
```

Screenshoot Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 5\unguided 1"
Masukkan nilai: 5
Deret Fibonacci:
0 1 1 2 3 5
PS C:\vscode 1>
```

Deskripsi Program: Program diatas bertujuan untuk menghitung nilai dari deret fibonanci. Terdapat dua fungsi yaitu, fungsi deret_fibonanci217 dan fungsi main. Fungsi deret_fibonanci217 akan menerima sebuah bilangan bulat sebagai argument dan terdapat fungsi rekursif untuk menghitung nilai fibonanci. Jika n adalah 0, maka fungsi akan mengembalikan 0, jika 1 maka akan mengembalikan 1. Untuk nilai lainnya fungsi akan memanggil serta menghitung dua nilai fibonanci sebelumnya dan menjumlahkannya. Fungsi main disini meminta pengguna memasukkan nilai dan fungsi main akan mencetak deret fibonanci tersebut.

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

Source Code

```
package main

import "fmt"

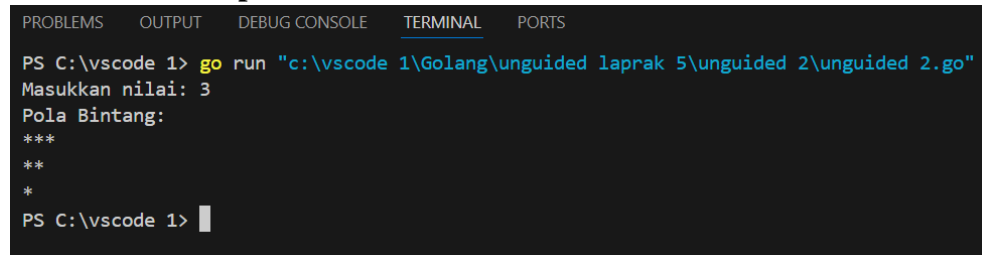
func printBintang2217(n int) {
    if n == 0 {
        return
    }

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

    printBintang2217(n - 1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai: ")
    fmt.Scanln(&n)
    fmt.Println("Pola Bintang:")
    printBintang2217(n)
}
```

Screenshoot Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 5\unguided 2\unguided 2.go"
Masukkan nilai: 3
Pola Bintang:
***
**
*
PS C:\vscode 1> █
```

Deskripsi Program: Program tersebut merupakan fungsi rekursif untuk mencetak baris Bintang sebanyak n. Dalam fungsi ini jika n adalah 0 maka fungsi akan keluar sebagai base case, jika n tidak 0 maka fungsi akan mencetak baris Bintang sebanyak n menggunakan loop for. Setelah mencetak baris Bintang, fungsi akan memanggil dirinya sendiri untuk mencetak baris Bintang selanjutnya dengan nilai n-1.

3. Buatlah program yang mengimplementasikan rekursif untuk menampilkan faktor bilangan dari suatu N, atau bilangan yang apasaja habis membagi N. Masukkan terdiri dari sebuah bilangan bulat positif N dan keluaran terdiri dari barisan bilangan yang menjadi faktor dari N.

Source Code

```
package main

import "fmt"

func detectFactors17(n217 int, i2217 int, factors []int) []int {
    if i2217 > n217 {
        return factors
    }

    if n217%i2217 == 0 {
        factors = append(factors, i2217)
    }

    return detectFactors17(n217, i2217+1, factors)
}

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

    factors := detectFactors17(n, 1, []int{})
    fmt.Println("Faktor bilangan:", factors)
}
```

Screenshoot Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 5\unguided 3
Masukkan bilangan: 12
Faktor bilangan: [1 2 3 4 6 12]
PS C:\vscode 1>
```

Deskripsi Program: Program ini adalah fungsi rekursif yang akan mencari factor factor dari bilangan n. Dalam fungsi ini jika I lebih besar dari n, maka fungsi akan mengembalikan slice factor. Tetapi, jika n habis dibagi dengan I, maka I akan ditambahkan ke dalam slice factor. Setelah itu fungsi akan memanggil dirinya sendiri dengan i+1 sebagai argument untuk mencari factor selanjutnya. Kemudian fungsi main akan mencetak factor factor bilangan.

4. Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu. Dengan masukkan terdiri dari sebuah bilangan bulat positif N dan dengan keluaran terdiri dari barisan bilangan N hingga 1 dan kembali ke N.

Source Code

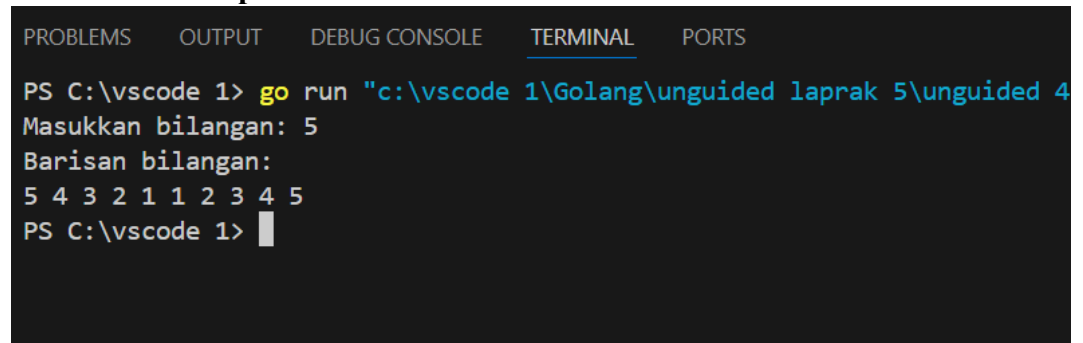
```
package main

import "fmt"

func SequencePrint_2217(n17 int) {
    if n17 == 0 {
        return
    }
    fmt.Print(n17, " ")
    SequencePrint_2217(n17 - 1)
    fmt.Print(n17+1-1, " ")
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&n)
    fmt.Println("Barisan bilangan:")
    SequencePrint_2217(n)
    fmt.Println()
}
```

Screenshoot Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 5\unguided 4"
Masukkan bilangan: 5
Barisan bilangan:
5 4 3 2 1 1 2 3 4 5
PS C:\vscode 1> █
```

Deskripsi Program: Program ini adalah fungsi rekursif yang akan mencetak baris bilangan. Dalam fungsi ini jika n adalah 0, maka fungsi akan keluar sebagai base case. Tetapi, jika n tidak 0, maka fungsi akan mencetak n diikuti dengan pemanggilan dirinya sendiri dengan n-1, kemudian fungsi akan mencetak n lagi dengan memanggil rekursfi. Terakhir fungsi main akan memanggil fungsi sequenceprint_2217 untuk mencetak barisan bilangan.

5. Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil. Dengan masukkan terdiri dari sebuah bilangan bulat positif N dan keluaran terdiri dari barisan bilangan ganjil dari 1 hingga N.

Source Code

```
package main

import "fmt"

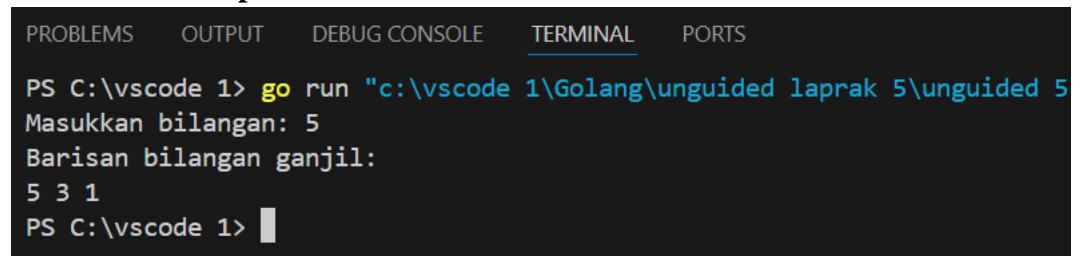
func OddSequence2217(n17 int) {
    if n17 < 1 {
        return
    }

    fmt.Print(n17, " ")
    OddSequence2217(n17 - 2)
}

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

    fmt.Println("Barisan bilangan ganjil:")
    OddSequence2217(n)
    fmt.Println()
}
```

Screenshot Output



```
PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 5\unguided 5
Masukkan bilangan: 5
Barisan bilangan ganjil:
5 3 1
PS C:\vscode 1>
```

Deskripsi Program: Program ini adalah fungsi rekursif yang akan mencetak baris bilangan ganjil. Dalam fungsi ini jika $n < 1$, maka fungsi akan keluar sebagai base case. Jika $n > 1$, maka fungsi akan mencetak n diikuti dengan pemanggilan dirinya sendiri dengan $n-2$. Kemudian fungsi main akan memanggil fungsi oddsequence2217 untuk mencetak barisan bilangan ganjil.

6. Buatlah program yang mengimplementasikan rekursif untuk mencari hasil pangkat dari dua buah bilangan. Dengan masukkan terdiri dari bilangan bulat x dan y . Keluaran terdiri dari hasil x dipangkatkan. Catatan: diperbolehkan menggunakan asterik “*”, tapi dilarang menggunakan import “math”.

Source Code

```
package main
package main

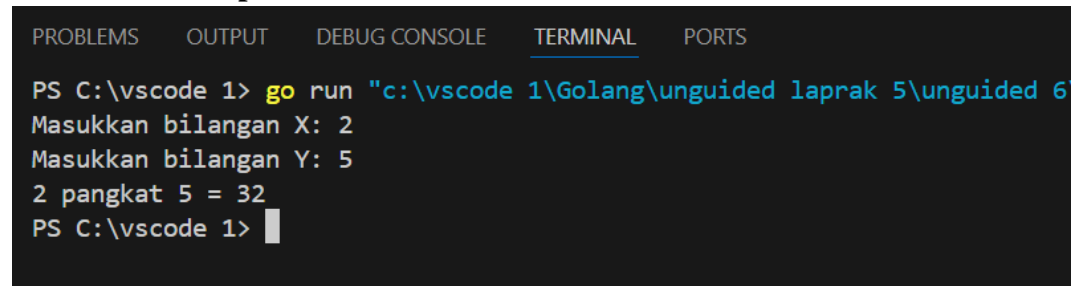
import "fmt"

func pangkat17(x22, y17 int) int {
    if y17 == 0 {
        return 1
    }
    return x22 * pangkat17(x22, y17-1)
}

func main() {
    var x2, y17 int
    fmt.Print("Masukkan bilangan X: ")
    fmt.Scanln(&x2)
    fmt.Print("Masukkan bilangan Y: ")
    fmt.Scanln(&y17)

    result := pangkat17(x2, y17)
    fmt.Printf("%d pangkat %d = %d\n", x2, y17,
result)
}
```

Screenshoot Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 5\unguided 6"
Masukkan bilangan X: 2
Masukkan bilangan Y: 5
2 pangkat 5 = 32
PS C:\vscode 1>
```

Deskripsi Program: Program ini adalah fungsi rekursif yang akan menghitung hasil pangkat x dengan pangkat y . Jika y adalah 0, maka fungsi akan mengembalikan 1 karena setiap bilangan dipangkatkan dg 0 hasilnya akan 1. Jika y tidak 0, maka fungsi akan mengembalikan hasil perkalian x dengan $y-1$ sebagai argument. Kemudian fungsi main akan memanggil fungsi pangkat17 untuk menghitung hasil pangkat dari x dengan pangkat y .