



# Soal Recursive, Number Theory, Sorting & Searching

📅 Date	March 1, 2023 → March 5, 2023
👤 Assign	Empty
▼ Status	Empty
▼ Minggu	Minggu-3
☰ Tipe	Soal Section

## ▼ Praktikum Recursive, Number Theory, Sorting & Searching

### Overview

peserta mampu membuat program dengan menerapkan recursive, number theory, sorting, searching, dan regex untuk menyelesaikan masalah.

### ▼ Soal Prioritas 1 (80)

- Dalam matematika, bilangan Fibonacci adalah barisan yang didefinisikan secara rekursif sebagai berikut:

Penjelasan: barisan ini berawal dari 0 dan 1, kemudian angka berikutnya didapat dengan cara menambahkan kedua bilangan yang berurutan sebelumnya. Dengan aturan ini, maka barisan bilangan Fibonacci yang pertama adalah:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946.....

```
package main
import "fmt"
```

```
func fibonacci(n int) int {
```

```
func fibonacci(number int) int {
    // your code here
}

func main() {
    fmt.Println(fibonacci(8)) // 0
    fmt.Println(fibonacci(2)) // 1
    fmt.Println(fibonacci(9)) // 34
    fmt.Println(fibonacci(10)) // 55
    fmt.Println(fibonacci(12)) // 144
}
```

- Buatlah program di Golang yang dapat mengurutkan barang berdasarkan jumlah kemunculannya. Jika ada barang yang duplicate kamu hanya perlu memunculkan sekali, namun kamu perlu menampilkan total kemunculan barang tersebut!

Sample Test Cases  
 Input: ["js", "js", "golang", "ruby", "ruby", "js", "js"]  
 Output: golang->1 ruby->2 js->4

```
package main

import "fmt"

type pair struct {
    name string
    count int
}

func MostAppearItem(items []string) []pair {
    // your code here
}

func main() {
    fmt.Println(MostAppearItem([]string{"js", "js", "golang", "ruby", "ruby", "js", "js"}))
    // golang->1 ruby->2 js->4
    fmt.Println(MostAppearItem([]string{"A", "B", "B", "C", "A", "A", "B", "A", "D", "D"}))
    // C->1 D->2 B->3 A->4
    fmt.Println(MostAppearItem([]string{"football", "basketball", "tenis"}))
    // football->1 basketball->1 tenis->1
}
```

- Dalam matematika, bilangan prima adalah bilangan asli yang lebih besar dari angka 1, yang faktor pembaginya adalah 1 dan bilangan itu sendiri. Angka 2 dan 3 adalah bilangan prima. Angka 4 bukan bilangan prima karena 4 bisa dibagi 2. Sepuluh deret bilangan prima yang pertama adalah 2, 3, 5, 7, 11, 13, 17, 19, 23 dan 29. Buatlah sebuah fungsi bernama getPrime yang menampilkan bilangan prima sesuai dengan deret urutannya.

Sample Test Cases

Input: 1

Output: 2

Input: 5

Output: 11

```
package main

import (
    "fmt"
)

func primeX(number int) int {
    // your code here
}

func main() {
    fmt.Println(primeX(1)) // 2
    fmt.Println(primeX(5)) // 11
    fmt.Println(primeX(8)) // 19
    fmt.Println(primeX(9)) // 23
    fmt.Println(primeX(10)) // 29
}
```

## ▼ Soal Prioritas 2 (20)

- Buatlah program playingDomino yang menerima 2 parameter array; parameter pertama merupakan kartu domino yang ada di tangan, • Parameter kedua merupakan kartu yang sedang ada di deck. Jika ada kartu yang disarankan maka output: [x,y], jika tidak ada kartu yang sesuai maka keluarkan: 'tutup kartu'.

```
package main

import (
    "fmt"
)

func playingDomino(cards [][]int, deck []int) interface{} {
    // your code here
}

func main() {
    fmt.Println(playingDomino([][]int{{6, 5}, {3, 4}, {2, 1}, {3, 3}}, []int{4, 3}))
    // [3, 4]
    fmt.Println(playingDomino([][]int{{6, 5}, {3, 3}, {3, 4}, {2, 1}}, []int{3, 6}))
    // [6 5]
    fmt.Println(playingDomino([][]int{{6, 6}, {2, 4}, {3, 6}}, []int{5, 1}))
    // "tutup kartu"
}
```

## ▼ Soal Eksplorasi (20)

1. Pada problem ini kamu harus menemukan total maksimum jumlah bilangan dari deret sebuah integer secara berurutan.

Sample Test Case

Input: [-2, 1, -3, 4, -1, 2, 1, -5, 4]

Output: 6

Penjelasan: 6 adalah hasil penambahan dari deret 4, -1, 2, 1

Sample Test Case

Input: [-2, -5, 6, -2, -3, 1, 5, -6]

Output: 7

Penjelasan: 7 adalah hasil penambahan dari deret 6, -2, -3, 1, 5

```
package main

import fmt

func MaxSequence(arr []int) {
    // your code here
}

func main() {
    fmt.Println(MaxSequence([]int{-2, 1, -3, 4, -1, 2, 1, -5, 4})) // 6
    fmt.Println(MaxSequence([]int{-2, -5, 6, -2, -3, 1, 5, -6})) // 7
    fmt.Println(MaxSequence([]int{-2, -3, 4, -1, -2, 1, 5, -3})) // 7
    fmt.Println(MaxSequence([]int{-2, -5, 6, -2, -3, 1, 5, -6})) // 8
}
```

```

fmt.Println(MaxSequence([]int{-2, -5, 6, 2, -3, 1, 6, -6})) // 12
}

```

2. Kamu diminta untuk membeli sebuah barang, dan tantangan kali ini kamu harus bisa membeli barang dengan jumlah maksimum dengan uang yang kamu miliki.

Program ini menerima `money` sebagai parameter pertama dan yang kedua berupa list harga produk yang dapat dibeli. Kamu harus menampilkan nilai jumlah barang yang bisa dibeli.

#### Sample Test Case

Input: `money = 50000`, `productPrice = [25000, 25000, 10000, 14000]`

Output: 3

Input: `money = 50000`, `productPrice = 15000, 10000, 12000, 5000, 3000]`

Output: 5

```

package main


import "fmt"

func MaximumBuyProduct(money int, productPrice []int) {
    // your code here
}

func main() {
    MaximumBuyProduct(50000, []int{25000, 25000, 10000, 14000}) // 3
    MaximumBuyProduct(30000, []int{15000, 10000, 12000, 5000, 3000}) // 4
    MaximumBuyProduct(10000, []int{2000, 3000, 1000, 2000, 10000}) // 4
    MaximumBuyProduct(4000, []int{7500, 3000, 2500, 2000}) // 1
    MaximumBuyProduct(0, []int{10000, 30000}) // 0
}

```

## Note

- simpan project kalian ke dalam github yang telah kalian buat. jangan lupa untuk screen shoot dan membuat review terkait materi yang kalian pelajari sekarang
- Standart penilaian :  Standard penilaian

