Recursive, Number Theory, Sorting & Searching

* Required

Email *

m.rezahidayat.rh@gmail.com

dari fungsi cek bilangan prima, apa yang menyebabkan fungsi berikut jauh lebih efisien dibanding algoritma pengecekan bilangan prima umumnya.. *

```
func checkPrime(number int) bool {
 if number < 2 {
   return false
 }
 sqrtNumber := int(math.Sqrt(float64(number)))
 for i := 2; i <= sqrtNumber; i++ {
   if number % i == 0 {
     return false
   }
 }
 return true
}
```

- ○ if number < 2
- ◉ math.Sqrt
- ○ number % i
- ○ i++

Algoritma berikut merupakan algoritma pencarian jenis apa *

```
for i := 0; i < len(elements); i++ {
  if elements[i] == x {
    return i
  }
}
return -1
```

- ( ) Binary Search
- (•) Linier search - O(n)
- ( ) Builtins Search
- ( ) Buble Search

Yang benar mengenai fungsi recursive adalah, kecuali *

- ( ) memerlukan break poin untuk perulangan
- ( ) harus memiliki return value
- (•) Fungsi yang memanggil dirinya sendiri
- ( ) banyak masalah dapat diselesaikan dengan mudah dan code yang sedikit

Algoritma berikut merupakan algoritma sorting jenis apa *

```
count := make([]int, k + 1)
for i := 0; i < len(elements); i++ {
  count[elements[i]]++
}
 counter := 0
for i := 0; i < k + 1; i++ {
  for j := 0; j < count[i]; j++ {
    elements[counter] = i
    counter += 1
  }
}

return elements
```

- (•) Counting sort - O(n + k)
- ( ) Buble Sort
- ( ) Selection sort - O(n^2)
- ( ) Merge sort - O (log n)

ketikan dipanggil factorial(4) makan return value yang diberikan adalah *

```
func factorial(n int) int {
 if n == 1 {
   return 1
 } else {
   return n * factorial(n - 1)
 }
}
```

- ○ 24
- ◉ 120
- ○ 10
- ○ 15

Submit Clear form