

LAPORAN PRAKTIKUM PENGEMBANGAN APLIKASI BERGERAK

Pengenalan Kotlin - Week 1



Disusun oleh:

Nama : Muhammad Ryan Fikri Fakhrezi

Nim : L0122114

Kelas : C

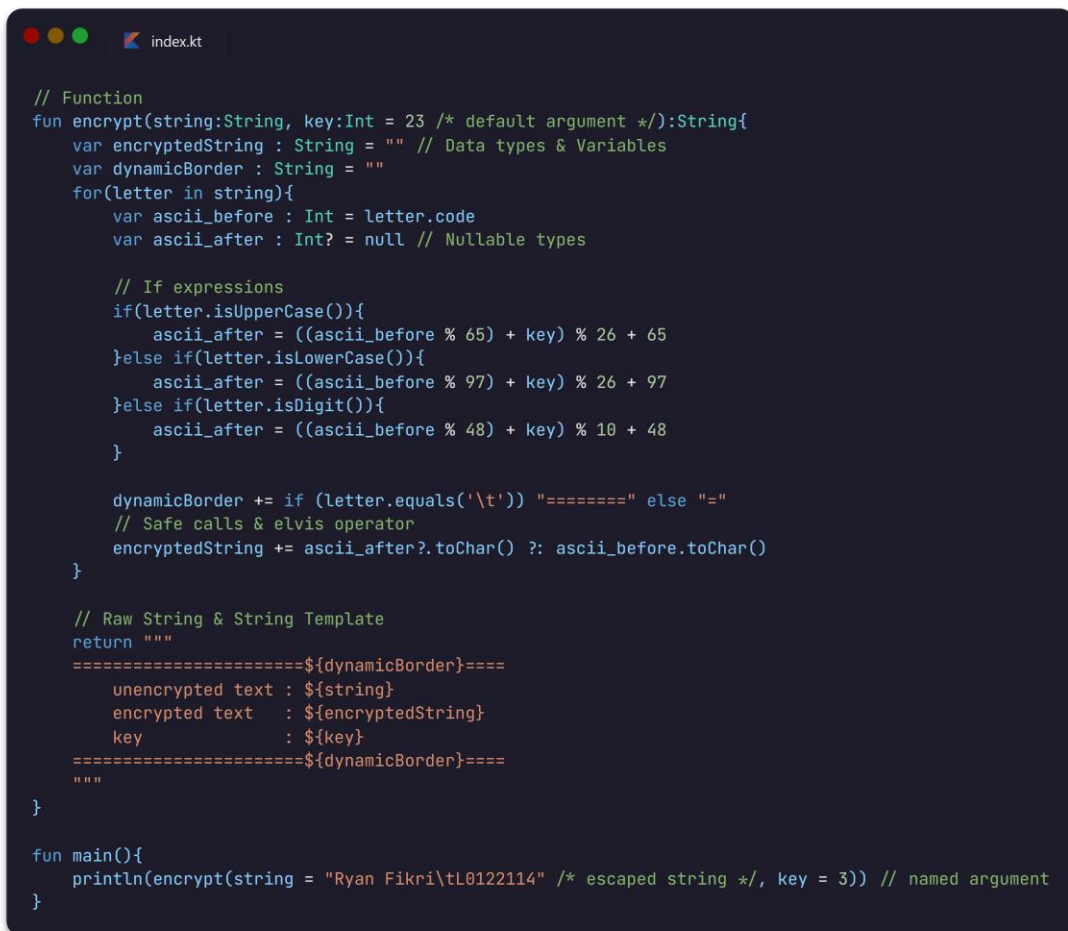
PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

2024

1. Screenshot Source Code



```
// Function
fun encrypt(string:String, key:Int = 23 /* default argument */):String{
    var encryptedString : String = "" // Data types & Variables
    var dynamicBorder : String = ""
    for(letter in string){
        var ascii_before : Int = letter.code
        var ascii_after : Int? = null // Nullable types

        // If expressions
        if(letter.isUpperCase()){
            ascii_after = ((ascii_before % 65) + key) % 26 + 65
        }else if(letter.isLowerCase()){
            ascii_after = ((ascii_before % 97) + key) % 26 + 97
        }else if(letter.isDigit()){
            ascii_after = ((ascii_before % 48) + key) % 10 + 48
        }

        dynamicBorder += if (letter.equals('\t')) "======" else "="
        // Safe calls & elvis operator
        encryptedString += ascii_after?.toChar() ?: ascii_before.toChar()
    }

    // Raw String & String Template
    return """
    =====${dynamicBorder}====
    unencrypted text : ${string}
    encrypted text   : ${encryptedString}
    key              : ${key}
    =====${dynamicBorder}====
    """
}

fun main(){
    println(encrypt(string = "Ryan Fikri\tL0122114" /* escaped string */, key = 3)) // named argument
}
```

Source code ini merupakan source code yang digunakan untuk praktikum minggu 1 saya. Source code ini mencakup materi berikut

A. Data Types & Variables



```
var encryptedString : String = "" // Data types & Variables
var dynamicBorder : String = ""
```

Variable encryptedString disini memiliki tipe data String dan bersifat mutable. Variable ini digunakan untuk menyimpan data string hasil enkripsi.

B. Functions (selain main function)

```
// Function
fun encrypt(string:String, key:Int = 23 /* default argument */):String{
    var encryptedString : String = "" // Data types & Variables
    var dynamicBorder : String = ""
    for(letter in string){
        var ascii_before : Int = letter.code
        var ascii_after : Int? = null // Nullable types

        // If expressions
        if(letter.isUpperCase()){
            ascii_after = ((ascii_before % 65) + key) % 26 + 65
        }else if(letter.isLowerCase()){
            ascii_after = ((ascii_before % 97) + key) % 26 + 97
        }else if(letter.isDigit()){
            ascii_after = ((ascii_before % 48) + key) % 10 + 48
        }

        dynamicBorder += if (letter.equals('\t')) "======" else "="
        // Safe calls & elvis operator
        encryptedString += ascii_after?.toChar() ?: ascii_before.toChar()
    }

    // Raw String & String Template
    return """
    =====${dynamicBorder}====
    unencrypted text : ${string}
    encrypted text   : ${encryptedString}
    key              : ${key}
    =====${dynamicBorder}====
    """
}
```

Function `encrypt(string:String, key:Int)` mengembalikan data bertipe `String`. Fungsi ini digunakan untuk mengenkripsi string menggunakan Caesar Cipher. Fungsi ini memiliki 2 argumen: `string` bertipe data `String` yang menampung kata yang ingin dienkripsi dan `key` bertipe data `Integer` yang menampung data `key` yang digunakan untuk mengenkripsi string.

C. Nullable Types

```
var ascii_before : Int = letter.code
var ascii_after : Int? = null // Nullable types
```

Variable `ascii_after` merupakan sebuah variable dengan tipe data `Integer` yang dapat bernilai `null`, variable ini menampung `ascii` value untuk huruf yang telah melalui proses enkripsi

D. Safe Calls

```
// Safe calls & elvis operator
encryptedString += ascii_after?.toChar() ?: ascii_before.toChar()
```

Safe calls digunakan untuk mengakses variable dengan nullable types. Disini digunakan safe calls untuk mengakses data di variable `ascii_after` dan menggunakan elvis operator untuk memberikan nilai default, yakni `ascii_before` atau value `char` yang tidak terenkripsi

E. If Expression

```
// If expressions
if(letter.isUpperCase()){
    ascii_after = ((ascii_before % 65) + key) % 26 + 65
}else if(letter.isLowerCase()){
    ascii_after = ((ascii_before % 97) + key) % 26 + 97
}else if(letter.isDigit()){
    ascii_after = ((ascii_before % 48) + key) % 10 + 48
}
```

If expression digunakan untuk pemisahan 3 skenario, yakni jika hurufnya merupakan uppercase, lowercase, atau digit numerik.

F. Named Argument

```
fun main(){
    println(encrypt(string = "Ryan Fikri\tL0122114" /* escaped string */, key = 3)) // named argument
}
```

Named argument merupakan cara memanggil fungsi dengan menyebutkan nama parameter secara eksplisit. Di dalam fungsi main terdapat penggunaan fungsi encrypt dengan menyebut nama parameternya, yakni string dan key.

G. Default Argument

```
// Function
fun encrypt(string:String, key:Int = 23 /* default argument */):String{
```

Default argument menetapkan nilai default untuk parameter fungsi. Disini parameter key memiliki default argument 23 yang didasarkan pada halaman https://en.wikipedia.org/wiki/Caesar_cipher.

H. Escaped String

```
fun main(){
    println(encrypt(string = "Ryan Fikri\tL0122114" /* escaped string */, key = 3)) // named argument
}
```

Escaped String memungkinkan kita untuk mengurangi ambiguitas nilai yang berada di dalam sebuah String. Disini terdapat penggunaan escaped string ‘\t’ pada argument pemanggilan fungsi encrypt di fungsi main.

I. Raw String

```
// Raw String & String Template
return """
===== ${dynamicBorder} =====
    unencrypted text : ${string}
    encrypted text   : ${encryptedString}
    key              : ${key}
===== ${dynamicBorder} =====
"""
```

Raw String yang memungkinkan untuk menuliskan multiline dan arbitrary text. Disini raw string digunakan untuk membuat format return dari fungsi encrypt.

J. String Template

```
// Raw String & String Template
return """
===== ${dynamicBorder} =====
    unencrypted text : ${string}
    encrypted text   : ${encryptedString}
    key              : ${key}
===== ${dynamicBorder} =====
"""
```

String template adalah fitur yang memungkinkan penggabungan nilai variabel ke dalam string. Disini string template digunakan di raw string yang akan direturn oleh fungsi encrypt. String template digunakan untuk menggabungkan nilai raw string, encrypted string, dan key kedalam raw string.

K. Elvis Operators

```
dynamicBorder += if (letter.equals('t')) {""} else {""}
// Safe calls & elvis operator
encryptedString += ascii_after?.toChar() ?: ascii_before.toChar()
}
```

Elvis operator memungkinkan kita untuk memberikan nilai default jika variabel nullable bernilai null. Disini jika ascii_after bernilai null, maka akan digunakan nilai default yakni unencrypted string pada variable ascii_before.

2. Screenshot Terminal

Berikut adalah hasil pada terminal apabila source code dijalankan.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● PS D:\Developer\Playgrounds\MobileAppsDevelopment\PPAB-01_L0122114_MuhammadRyanFikriFakhrezi\sourcecode>
  cd "d:\Developer\Playgrounds\MobileAppsDevelopment\PPAB-01_L0122114_MuhammadRyanFikriFakhrezi\sourcecod
  e\" ; if ($?) { kotlinc main.kt -include-runtime -d main.jar } ; if ($?) { java -jar main.jar }

=====
unencrypted text : Ryan Fikri    L0122114
encrypted text   : Ubdq Ilnul   03455447
key              : 3
=====
```