

NAMA : LIEM RIKI WIBOWO

NIM : A11.2020.12486

UAS PBO

1. Jelaskan definisi dan berikan contoh implementasi pada hal berikut ini dengan menggunakan Bahasa pemrograman Kotlin:

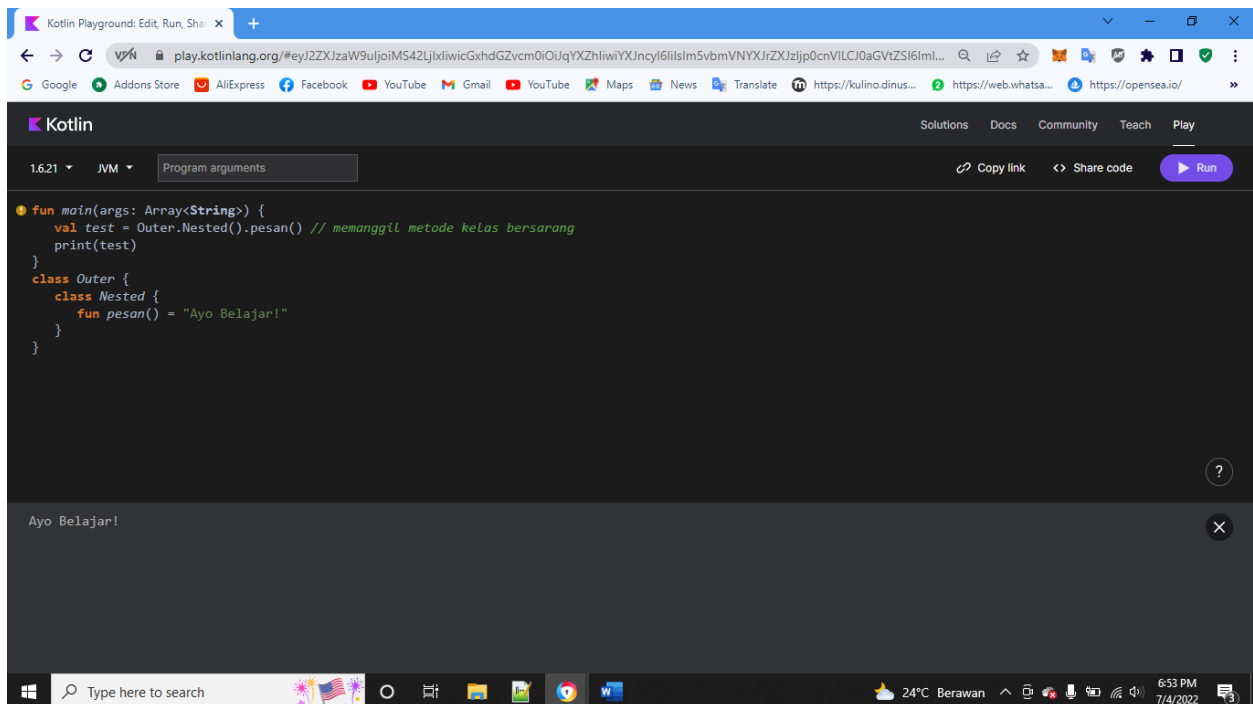
a. Nested class

Nested Class adalah class yang sudah di buat di dalam class lain. Di kotlin Nested Class akan bersifat statis secara default, sehingga dapat diakses tanpa membuat objek apapun dari class tersebut.

CODE NESTED CLASS:

```
fun main(args: Array<String>) {  
    val test = Outer.Nested().pesan() // memanggil metode kelas bersarang  
    print(test)  
}  
  
class Outer {  
    class Nested {  
        fun pesan() = "Ayo Belajar!"  
    }  
}
```

OUTPUT DARI NESTED CLASS:



b. Singleton

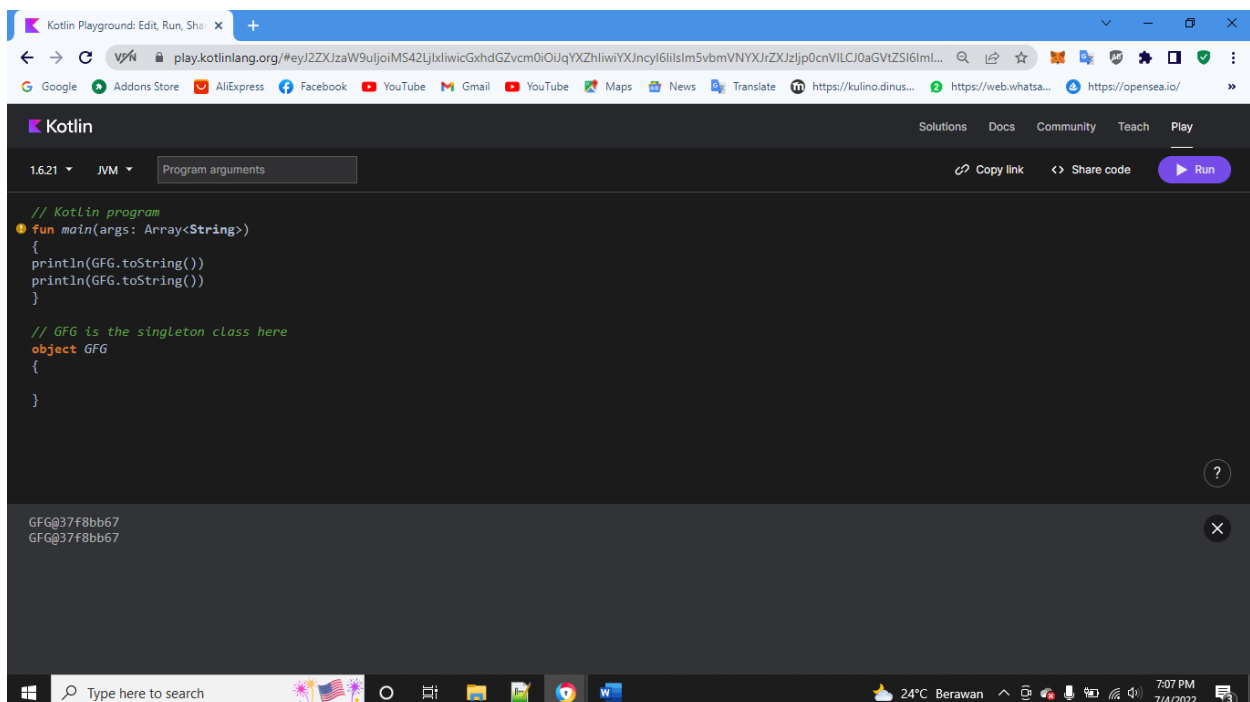
Singleton Class Kotlin disebut sebagai Singleton Object di Kotlin. Class Singleton adalah class yang hanya mempunyai satu instance dari kelas yang dapat dibuat dan digunakan di mana saja.

CODE SINGLETON :

```
// Kotlin program
fun main(args: Array<String>)
{
    println(GFG.toString())
    println(GFG.toString())
}

// GFG is the singleton class here
object GFG
{
}
```

OUTPUT SINGLETON:

The image is a screenshot of a web browser displaying the Kotlin Playground interface. The browser's address bar shows the URL 'play.kotlinlang.org/#eyJ2ZXJzaW9uIjoiaS42LjlxliwicGxhdGZvcn0iOiJqYXZhiwiYXJncyI6Ilslm5vbmVNYXJrZXJzJjp0cnVlLCJ0aGVtZSI6ImL...'. The Kotlin Playground editor shows the same code as the previous block: a Kotlin program with a main function that prints 'GFG' twice, and a singleton object 'GFG' with an empty body. The 'Run' button is highlighted. Below the code editor, the output is displayed: 'GFG@37f8bb67' on two separate lines. The bottom of the image shows a Windows taskbar with the search bar, task view button, and several application icons. The system tray on the right shows the temperature as 24°C, the location as Berawan, and the time as 7:07 PM on 7/4/2022.

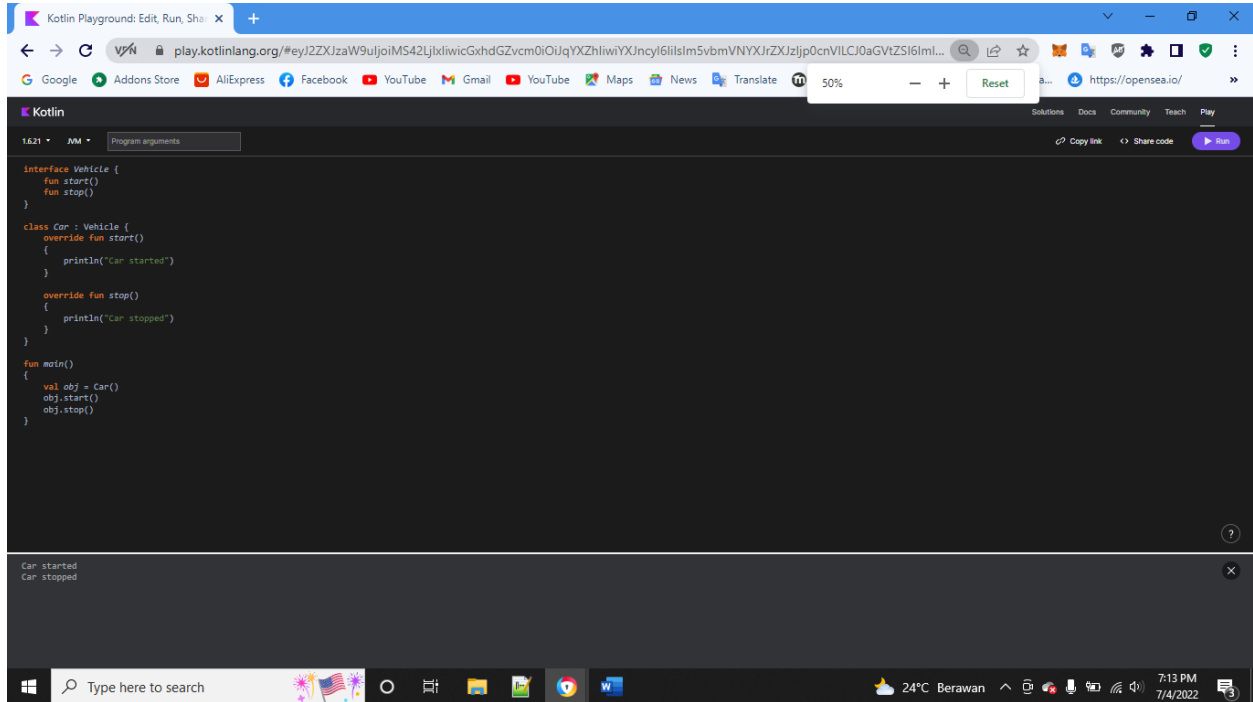
c. Interface

Class interface adalah implementasi metode serta deklarasi metode abstrak/custom dan dapat diimplementasikan oleh kelas untuk menggunakan fungsionalitas yang ditentukan.

CODE INTERFACE :

```
interface Vehicle {  
    fun start()  
    fun stop()  
}  
  
class Car : Vehicle {  
    override fun start()  
    {  
        println("Car started")  
    }  
    override fun stop()  
    {  
        println("Car stopped")  
    }  
}  
  
fun main()  
{  
    val obj = Car()  
    obj.start()  
    obj.stop()  
}
```

OUTPUT INTERFACE :



The screenshot shows a web browser window with the URL `play.kotlinlang.org/#eyJ2ZXJzaW9uIjoiaS42LjlxliwicGxhdGZvcn0iOiJqYXZlbiwYXJncyI6IiIsIm5vbmVNYXJrZXJzIjp0cnVILCJ0aGVtZSI6ImI...`. The page title is "Kotlin Playground: Edit, Run, Share". The main content area displays Kotlin code for a simple car simulation. Below the code editor, the output of the program is shown in a terminal-like window.

```
interface Vehicle {
    fun start()
    fun stop()
}

class Car : Vehicle {
    override fun start() {
        println("Car started")
    }

    override fun stop() {
        println("Car stopped")
    }
}

fun main() {
    val obj = Car()
    obj.start()
    obj.stop()
}
```

Car started
Car stopped

2. Untuk membuat suatu aplikasi kita dapat menggunakan Bahasa pemrograman Java ataupun Kotlin. Pada Kotlin sendiri kita dapat menggunakan library yang ada di Java. Apa persamaan dan perbedaan antara Bahasa pemrograman Java dan Kotlin?

perbedaan Java dan Kotlin 100% dapat dioperasikan . Anda dapat memanggil kode Kotlin dari Java, dan Anda dapat memanggil kode Java dari Kotlin. Jadi dimungkinkan untuk memiliki kelas Kotlin dan Java secara berdampingan dalam proyek yang sama, dan semuanya akan tetap dikompilasi.

3. Buatlah sebuah aplikasi desktop sederhana yang dapat menampilkan daftar mahasiswa yang dapat ditambahkan, diedit, ataupun dihapus. Dengan ketentuan sebagai berikut:

- a. Data mahasiswa meliputi: nama, nim, IPK, mata kuliah aktif, foto.
 - b. Terdiri dari minimal 2 halaman aplikasi.
 - c. Terdapat implementasi listener.
 - d. Terdapat implementasi atau koneksi ke database baik local maupun online.
- * Note: upload aplikasi tersebut sebagai tugas terakhir ke github masing-masing.
Kemudian presentasikan saat ujian praktek.