**Nama : Afida Kholifatus Zahro**

**NIM : 19051397040**

**Kelas : D4 Manajemen Informatika 2019 B**

**Tugas PBO**

1. Bentuk standart UML class diagram

* **Abstrak class**

**Abstract Class** merupakan Class yang khusus dibuat untuk keperluan inheritance. Tujuan dari pembuatan abstract class ini ialah untuk membuat definisi umum bagi class-class yang akan menjadi turunan darinya.

Abstract Class tidak bisa diinstansiasi. Abstract method tidak mempunyai implementasi, sehingga penulisannya dilakukan hanya dengan mengikutkan semikolon, bukan blok method {} seperti biasanya. Abstract method ini bisa digunakan oleh Class turunannya dengan melakukan override.

Abstract class hanya bisa digunakan sebagai super class, tapi juga bisa diturunkan dari class abstract lainnya. Untuk mendeklarasikan sebuah abstract class digunakan keyword abstract, [abstract] class [class\_name] Sebuah abstract class pada dasarnya tidak hauh beda dengan class lainnya, yakni juga berisi method yang menggambarkan carakteristik dari kelas abstract tersebut, bedanya yakni sebuah abstract class bisa berisi method tanpa diimplementasikan artinya sebuah method tanpa body, methid seperti ini disebut method abstract.

* **Interface**

Sedangkan interface secara filosofis lebih berfungsi sebagai antarmka yang membentuk komunikasi dengan code lain. Misalnya membentuk hubungan antara sebuah object dengan object yang lain atau hubungan antara object sebagai penyedia dengan code pengguna. Karena itulah interface ini banyak digunakan dalam dunia design pattern. Sebab kebanyakan dari design pattern ini menekankan hubungan antar object.

**Interface** adalah jenis khusus dari blok yang hanya berisi method signature (atau constant). Interface mendefinisikan sebuah(signature) dari sebuah kumpulan method tanpa tubuh. Interface mendefinisikan sebuah cara standar dan umum dalam menetapkan sifat-sifat dari class-class. Mereka menyediakan class-class, tanpa memperhatikan lokasinya dalam hirarki class, untuk mengimplementasikan sifat-sifat yang umum.

Dengan catatan bahwa interface-interface juga menunjukkan polimorfisme, dikarenakan program dapat memanggil method interface dan versi yang tepat dari method yang akan dieksekusi tergantung dari tipe object yang melewati pemanggil method interface.