Nama : Mohammad Zaenal Arif Santoso

NIM : 20051214084

Kelas : Sistem Informasi 2020B

1. Abstract Class

* Sebuah kelas yang tidak dapat di-instansiasi

Tidak bisa membuat objek dari class ini, Bisa diintansiasi lewat kelas turunannya (Hanya kelas turunannya yang bisa di-instansiasi

* Kelas ‘abstrak’

Jadi tidak dapat di-instansiasi atau dibuat objek karena masih abstrak.

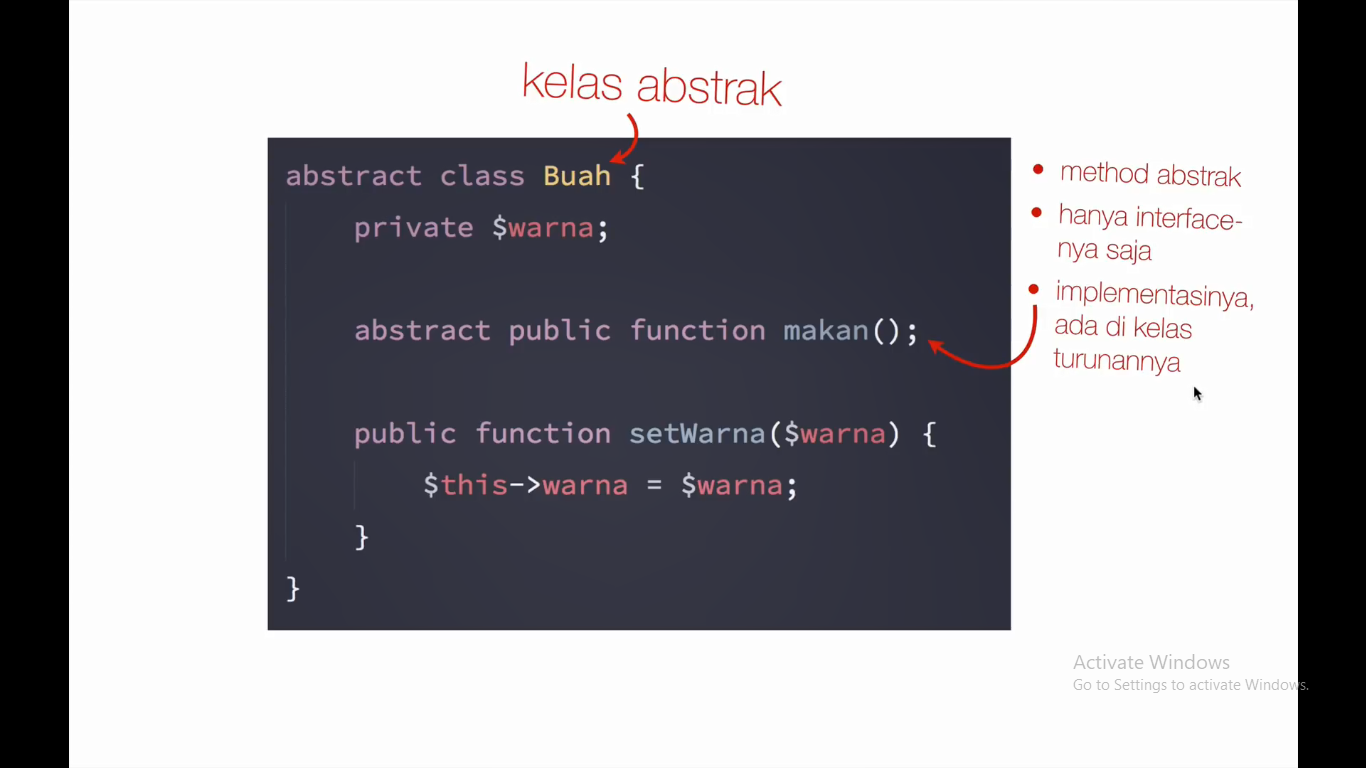
* Mendefinisikan Interface untuk kelas lain yang menjadi turunannya

Hanya nama method saja, tidak ada isinya.

* Berperan sebagai kerangka dasar untuk kelas turunannya
* Memiliki minimal 1 method abstrak

Method yang akan menjadi interface untuk kelas turunannya

* Digunakan dalam pewarisan/inheritance untuk memaksakan implementasi method abstrak yang sama untuk semua kelas turunannya.
* Semua kelas turunan, harus mengimplementasikan method abstrak yang ada dikelas abstraknya
* Kelas abstrak boleh memiliki property/method reguler (public, private, protected).
* Kelas abstrak boleh memiliki property/method static.



Kenapa harus menggunakan kelas abstrak?

* Mempresentasikan ide/konsep dasar
* “Composition over inheritance”
* Salah satu penerapan polimorphism
* Sentralisasi logic
* Mempermudah pengerjaan tim

Membuat kelas utama/base classs yang berisi method abstrac, jadi ketika membuat kelas turunan rekan setim tau bahwa ada method class yang harus diimplementasikan.

1. Interface

* Kelas abstrak yang sama sekali tidak memiliki implementasi
* Murni merupakan template untuk turunannya
* Tidak boleh memiliki properti, hanya deklarasi method saja
* Semua method harus dideklarasikan dengan visibility public
* Boleh mendeklarasikan construct
* Satu kelas boleh mendeklarasikan banyak interface
* Dengan menggunakan type-hinting bisa melakukan ‘dependency injection’

Memaksakan sebuah method untuk menerima parameternya adalah objek

* Pada akhirnya melakukan polimorphism



<?php

abstract class Buah

{

    protected $warna;

    abstract public function caraMakan();

    public function setWarna($warna)

    {

        $this->warna = $warna;

    }

    public function getWarna()

    {

        return $this->warna;

    }

}