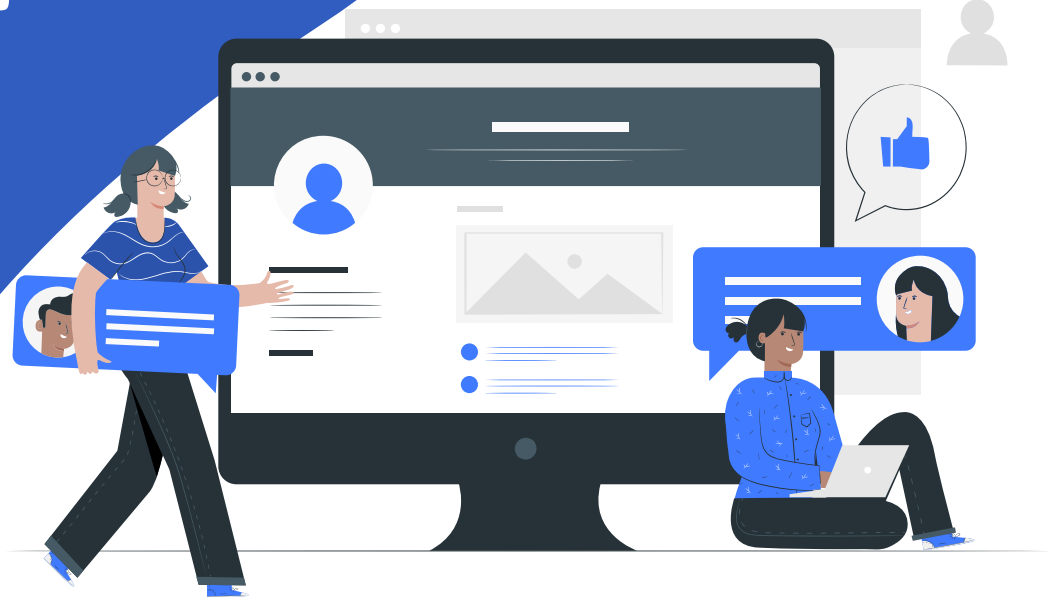


Refactoring Creational Design Pattern

ETS PPL 1



KELOMPOK B9



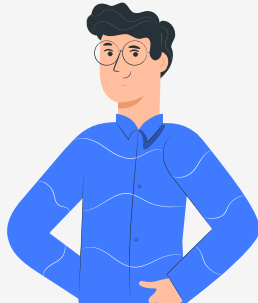
**Angga
Gemilang**

201511036



**Fitri
Fauziyah**

201511040



**Tegar Faris
Nurhakim**

2015110

01. Abstract Factory



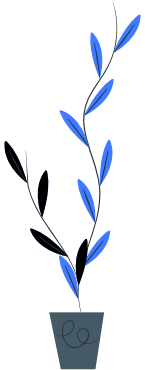
Deskripsi

Abstract Factory merupakan salah satu jenis dari Creational Design Pattern yang merangkum sekelompok class yang memiliki tema umum. Tema yang dimaksud disini yaitu method, variabel atau hal lainnya yang memiliki kesamaan. Pada game ini terdapat Class yang berfungsi untuk mengatur tingkat kesulitan dari game tersebut, yaitu Class EasyFactory, MediumFactory dan HardFactory. Yang dimana pada setiap class tersebut memiliki isi yang mirip dikarenakan meninherit dari class EnemyFactory. Setelah melihat ini saya berpikiran untuk membuat Abstract Factory yang dimana digunakan untuk memanggil product class (Easy,Medium,Hard) tanpa menspesifikasikan concrete classnya (DifficultyFactory).

Kelas Yang Ditambahkan

DifficultyAbstractFactory

```
Public abstract class DifficultyAbstractFactory {  
    public abstract EnemyFactory etEnemy(string difficulty);  
}
```



Kelas Yang Ditambahkan

DifficultyFactory

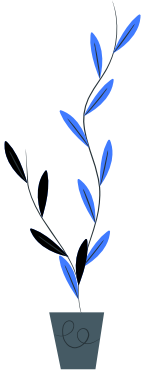
```
Public class DifficultyFactory : DifficultyAbstractFactory {  
    Public override EnemyFactory getEnemy(string difficulty) {  
        if(difficulty == "Easy"){  
            return new EasyFactory();  
        } else if(difficulty == "Easy"){  
            return new MediumFactory();  
        } else {  
            return new HardFactory();  
        }  
    }  
}
```



Kelas Yang Diubah

EasyFactory | Sebelum

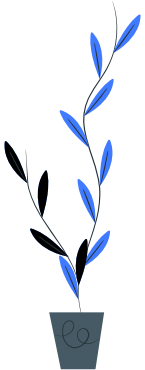
```
public class EasyFactory : EnemyFactory {  
    public override int TimeGenerate => 200;  
    public override EnemyWave GenerateEnemy(float DefaultEnemyYPosition) {  
        GameConfiguration gameConfiguration = GameConfiguration.Instance;  
        var enemyWave = new EnemyWave(WaveType.Down, DefaultEnemyYPosition);  
        var posX = UnityEngine.Random.Range(-1.6f, 1.6f);  
        var gameObject = (GameObject)Instantiate(gameConfiguration.DefaultEnemyGameObject,  
            new Vector3(posX, DefaultEnemyYPosition), Quaternion.identity);  
        enemyWave.RegisterEnemy(gameObject);  
  
        return enemyWave;  
    }  
}
```



Kelas Yang Diubah

EasyFactory | Sesudah

```
public class EasyFactory : MonoBehaviour, EnemyFactory {  
    public int TimeGenerate => 200;  
    public EnemyWave GenerateEnemy(float DefaultEnemyYPosition) {  
        GameConfiguration gameConfiguration = GameConfiguration.Instance;  
        var enemyWave = new EnemyWave(WaveType.Down, DefaultEnemyYPosition);  
        var posX = UnityEngine.Random.Range(-1.6f, 1.6f);  
        var gameObject = (GameObject)Instantiate(gameConfiguration.DefaultEnemyGameObject,  
            new Vector3(posX, DefaultEnemyYPosition), Quaternion.identity);  
        enemyWave.RegisterEnemy(gameObject);  
        return enemyWave;  
    }  
}
```



Kelas Yang Diubah

MediumFactory | Sebelum

```
public class MediumFactory : EnemyFactory {  
    public override int TimeGenerate => 100;  
    public override EnemyWave GenerateEnemy(float DefaultEnemyYPosition) {  
        GameConfiguration gameConfiguration = GameConfiguration.Instance;  
        var enemyWave = new EnemyWave(WaveType.Down, DefaultEnemyYPosition);  
        for(int i = 0; i < 5; i++)  
            AddEnemy(gameConfiguration, enemyWave, DefaultEnemyYPosition);  
        return enemyWave;  
    }  
    private void AddEnemy(GameConfiguration gameConfiguration, EnemyWave enemyWave,  
        float defaultEnemyYPosition) {  
        var posX = UnityEngine.Random.Range(-1.6f, 1.6f);  
        var gameObject = (GameObject)Instantiate(gameConfiguration.DefaultEnemyGameObject,  
            new Vector3(posX, defaultEnemyYPosition), Quaternion.identity);  
        enemyWave.RegisterEnemy(gameObject);  
    }  
}
```



Kelas Yang Diubah

MediumFactory | Sesudah

```
public class MediumFactory : MonoBehaviour, EnemyFactory {
    public int TimeGenerate => 100;
    public EnemyWave GenerateEnemy(float DefaultEnemyYPosition) {
        GameConfiguration gameConfiguration = GameConfiguration.Instance;
        var enemyWave = new EnemyWave(WaveType.Down, DefaultEnemyYPosition);
        for(int i = 0; i < 5; i++)
            AddEnemy(gameConfiguration, enemyWave, DefaultEnemyYPosition);
        return enemyWave;
    }
    private void AddEnemy(GameConfiguration gameConfiguration, EnemyWave enemyWave,
        float defaultEnemyYPosition) {
        var posX = UnityEngine.Random.Range(-1.6f, 1.6f);
        var gameObject = (GameObject)Instantiate(gameConfiguration.DefaultEnemyGameObject,
            new Vector3(posX, defaultEnemyYPosition), Quaternion.identity);
        enemyWave.RegisterEnemy(gameObject);
    }
}
```



Kelas Yang Diubah

HardFactory | Sebelum

```
public class HardFactory : EnemyFactory {  
    public override int TimeGenerate => 50;  
    public override EnemyWave GenerateEnemy(float DefaultEnemyYPosition) {  
        GameConfiguration gameConfiguration = GameConfiguration.Instance;  
        var enemyWave = new EnemyWave(WaveType.Down, DefaultEnemyYPosition);  
        for (int i = 0; i < 5; i++)  
            AddEnemy(gameConfiguration, enemyWave, DefaultEnemyYPosition);  
        return enemyWave;  
    }  
    private void AddEnemy(GameConfiguration gameConfiguration, EnemyWave enemyWave,  
        float defaultEnemyYPosition) {  
        var posX = UnityEngine.Random.Range(-1.6f, 1.6f);  
        var gameObject = (GameObject)Instantiate(gameConfiguration.DefaultEnemyGameObject,  
            new Vector3(posX, defaultEnemyYPosition), Quaternion.identity);  
        enemyWave.RegisterEnemy(gameObject);  
    }  
}
```



Kelas Yang Diubah

HardFactory | Sesudah

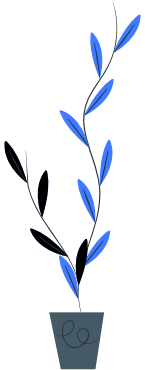
```
public class HardFactory : MonoBehaviour, EnemyFactory {
    public int TimeGenerate => 50;
    public EnemyWave GenerateEnemy(float DefaultEnemyYPosition) {
        GameConfiguration gameConfiguration = GameConfiguration.Instance;
        var enemyWave = new EnemyWave(WaveType.Down, DefaultEnemyYPosition);
        for (int i = 0; i < 5; i++)
            AddEnemy(gameConfiguration, enemyWave, DefaultEnemyYPosition);
        return enemyWave;
    }
    private void AddEnemy(GameConfiguration gameConfiguration, EnemyWave enemyWave,
        float defaultEnemyYPosition) {
        var posX = UnityEngine.Random.Range(-1.6f, 1.6f);
        var gameObject = (GameObject)Instantiate(gameConfiguration.DefaultEnemyGameObject,
            new Vector3(posX, defaultEnemyYPosition), Quaternion.identity);
        enemyWave.RegisterEnemy(gameObject);
    }
}
```



Kelas Yang Diubah

EnemyFactory | Sebelum

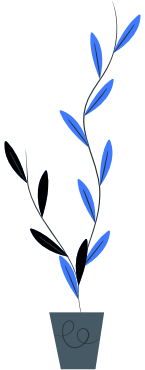
```
public abstract class EnemyFactory : MonoBehaviour {  
    public abstract int TimeGenerate { get; }  
    public abstract EnemyWave GenerateEnemy(  
        float DefaultEnemyYPosition);  
}
```



Kelas Yang Diubah

EnemyFactory | Sesudah

```
public abstract class EnemyFactory {  
    public int TimeGenerate { get; }  
    public EnemyWave GenerateEnemy(float DefaultEnemyYPosition);  
}
```



02. _enemy



Sebelum Refactoring

```
private EnemyFactory _enemyFactory;  
private static GameConfiguration _instance;  
private List<EnemyWave> _enemyWaves;  
void Start()
```


Class Enemy tidak dipanggil pada game apapun sehingga isi dari class tersebut tidak dapat dijalankan. Karenanya, ditambahkan pemanggilan class Enemy pada class GameConfiguration agar class Enemy dapat dijalankan



Setelah Refactoring

```
private EnemyFactory _enemyFactory;  
private Enemy _enemy;  
private static GameConfiguration _instance;  
private List<EnemyWave> _enemyWaves;  
void Start()
```

Terima Kasih!

Ada Pertanyaan?

