**LAPORAN PRAKTIKUM**

**AGGREGATION, INHERITANCE, DAN COMPOSITION PADA ULAR TANGGA**



Oleh :

Dhani Yanuar Erdiansyah 4210161021

D4 TEKNOLOGI GAME

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

SURABAYA

2018

**Dasas Teori**

* **Aggregation**

Aggregation adalah suatu hubungan khusus antara suatu object dengan object lainnya, namun setiap object itu dapat berdiri sendiri dan memiliki attribute dan variable nya sendiri. Hubungan khusus di sini yang dimaksud jika salah satu object di hapus maka object yang memiliki hubungan dengan object yang telah di hapus tidak akan terpanguruh.

* **Inheritance**

Inheritance adalah suatu salah satu mengatur object pada Object Oriented Programing (OOP), yang dimana object memiliki hubungan parent dan child, yang dimana object parent akan menurunkan method dan variable nya kepada child class nya.

* **Composition**

Composition adalah suatu hubungan yang lebih spesifik yang dimana class atau object akan bergantung dengan method dari class atau object lainnya. Jika object atau class owner nya dihapus maka class atau object lainnya tidak akan berjalan atau tidak akan memiliki method.

* **Tugas**

Tugas kali ini, mahasiswa mendapat tugas untuk menganalisa ketiga teory di atas akan dipakai dibagian mana pada game ular tangga. Untuk lebih jelas kita buat contoh class yang kemungkinan yang ada untuk di Analisa. Seperti di bawah ini.

**Pembahasan**

* Aggregation

Pada game ular tangga terdapat player yang memiliki behavior untuk jalan. Seperti di bawah ini :

class player1

{

private:

int a;

public:

void player1 ()

{

a=0;

}

Void move(){

**Dice** myDice = new Dice();

myDice.rolldice();

}

int player1 ()

{

return a;

}

int player1 (int ra)

{

a+=ra;

}

int player1 (int ma)

{

a-=ma;

}

operator int()

{

return a/1;

}

};

class Dice

{

private:

int dice1;

public:

Dice();

int rolldice(void);

operator int()

{

return dice1/1;

}

};

Di atas terdapat 2 contoh class yang akan dipakai dalam game ular tangga, object dice dibuat di luar object Player sehingga jika Object player dihancurkan maka maka tidak akan berpengaruh dengan class dice. Dan class dice akan tetap ada.

* Inheritance

Digame ini juga terdapat tiles, dan juga beberapa tiles yang memiliki fungsi yang berbeda beda jika kita membuat nya dalam oop, maka kira – kira akan terlihat sebagai berikut :

Class Tiles(){

Attribute and Method here

};

Class Good\_tiles() : Public Tiles{

Method dari Tiles

Dan tambahan fungsi untuk membuat pemain pindah untuk berpindah maju.

}

* Composition

Di dalam game ular tangga tentu ada design ui, yang di antara nya ada button untuk roll dadu.

Class button(){

Void OnClick(){

}

};

Class ButtonRoll : public button{

Void OnClick (){

**Dice** myDice = new Dice();

myDice.rolldice();

}

};

class Dice

{

private:

int dice1;

public:

Dice();

int rolldice(void);

operator int()

{

return dice1/1;

}

};

Pada contoh di atas class ButtonRoll hanya memiliki lifecycle atau method untuk roll dadu, yang dimana roll dadu dimilki oleh dice, jadi jika class dice di destroy maka class ButtonRoll tidak memilki lifecycle atau method, jadi class ButtonRoll ini tidak bias berdiri sendiri.

**Referensi**

* https://adnansetiawan.com/2017/01/15/perbedaan-asosiasi-agregasi-dan-komposisi-pada-object-oriented-programming/
* http://cplusplus.happycodings.com/games/code7.html