LAPORAN TUGAS BESAR
IF2210 - PEMROGRAMAN BERORIENTASI OBJEK

Engi's Farm - Milestone 1

Disusun oleh :

Eka Novendra Wahyunadi	13517011
Reyhan Naufal Hakim	13517029
Aliffiqri Agwar	13517107
Stefanus Ardi Mulia	13517119

TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2019

DAFTAR ISI

BAB I

DESKRIPSI PERSOALAN

Chef adalah seorang pemilik peternakan yang telah pindah dari restoran miliknya. Chef kali berperan sebagai Player yang dapat menampung barang-barang Product dan air dalam jumlah terbatas. Tanah yang dimiliki pemain berisi Cell yang direpresentasikan dengan matriks 2D dimana tiap Cell berisi Land atau Facilities. Land merupakan tempat dimana para FarmAnimal dan player bisa berjalan di atasnya, sedangkan Facilities tidak dapat dilewati (namun tetap dapat dilakukan interaksi) yang berisi fasilitas-fasilitas peternakan.

Setiap Land dapat berupa Coop yang dapat dijadikan tempat untuk EggProducingAnimal, Barn untuk tempat MeatProducingAnimal dan Grassland merupakan tempat untuk MilkProducingAnimal. Seluruh hasil dari ProducingAnimal merupakan sebuah FarmProduct. Untuk setiap Facilities dapat berupa Truck yang merupakan tempat penjualan Product. Terdapat pula Mixer yang berguna untuk mengkombinasi FarmProduct sehingga menjadi SideProduct. Kemudian ada Well yang dapat mengisi ulang container air milik Player.

Player dapat melakukan pergerakan selama masih dalam area permainan. Kemudian player juga dapat melakukan interaksi dengan FarmAnimal yang berupa Talk (berbicara dengan hewan tersebut), Interact (menghasilkan susu atau telur), dan Kill (menyembelih). Selain dengan hewan, Player juga dapat berinteraksi dengan Land berupa Grow (menumbuhkan rumput selama air yang dibawa masih ada). Lalu, Player dapat melakukan Interact juga dengan Facilities sesuai dengan fasilitas yang diberlakukan interaksi.

Hewan juga dapat melakukan pergerakan secara random. Sesuai dengan naluri hewani, setiap FarmAnimal juga dapat lapar dan mati setelah 5 tick dalam keadaan lapar, sehingga harus diberi makan. Hewan akan otomatis makan rumput yang telah berkembang di area yang dia lewati. Setiap hewan juga dapat mengeluarkan suara apabila diajak berbicara oleh Player. Setiap hewan penghasil telur dan susu wajib memakan rumput untuk menghasilkan telur atau susu. Namun, hewan penghasil daging tidak perlu makan rumput untuk menghasilkan daging.

BAB II

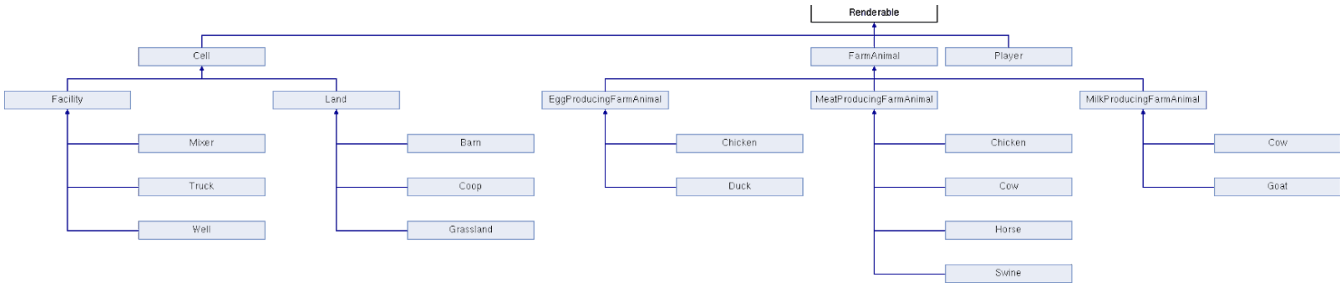
HIERARKI KELAS

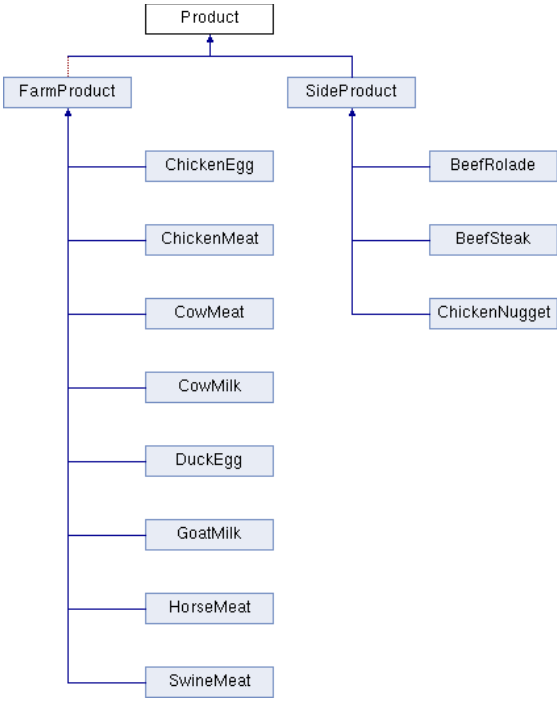
Solusi yang dibentuk untuk persoalan ini dibagi dengan hierarki sebagai berikut:

1. Scene
2. Time
3. LinkedList <T>
4. Counter <T>
5. Renderable
    - 5.1. Cell
        - 5.1.1. Facility
            - 5.1.1.1. Mixer
            - 5.1.1.2. Truck
            - 5.1.1.3. Well
        - 5.1.2. Land
            - 5.1.2.1. Barn
            - 5.1.2.2. Coop
            - 5.1.2.3. Grassland
    - 5.2. FarmAnimal
        - 5.2.1. EggProducingFarmAnimal
            - 5.2.1.1. Chicken
            - 5.2.1.2. Duck
        - 5.2.2. MeatProducingFarmAnimal
            - 5.2.2.1. Chicken
            - 5.2.2.2. Cow
            - 5.2.2.3. Horse
            - 5.2.2.4. Swine
        - 5.2.3. MilkProducingFarmAnimal
            - 5.2.3.1. Cow
            - 5.2.3.2. Goat
    - 5.3. Player
6. Product
    - 6.1. FarmProduct
        - 6.1.1. ChickenEgg
        - 6.1.2. ChickenMeat
        - 6.1.3. CowMeat
        - 6.1.4. CowMilk
        - 6.1.5. DuckEgg

6.1.6. GoatMilk
6.1.7. HorseMeat
6.1.8. SwineMeat
6.2. SideProduct
6.2.1. BeefRolade
6.2.2. BeefSteak
6.2.3. ChickenNugget

Sehingga apabila digambarkan dalam tree akan membentuk seperti gambar berikut





BAB III

DOKUMENTASI KELAS

1. Scene

Kelas Scene adalah blueprint dari objek yang menampung koleksi objek pada game serta melakukan update terhadap peta pada Farm.

```
class Scene {
   private:
      const int DEFAULT_FARM_MAP_HEIGHT;      // Tinggi peta farm default
      const int DEFAULT_FARM_MAP_WIDTH;       // Lebar peta farm default

      string sessionName;                     // nama pemain

      LinkedList<Cell*> field;                // LinkedList yang menyimpan koleksi Cell
      LinkedList<FarmAnimal*> animals;        // LinkedList yang menyimpan koleksi Animal

      Player player;                          // Instansiasi player

      char** farmMap;                         // Kumpulan hasil render karakter dari objek-objek pada farm
      int farmMapHeight;                      // Tinggi peta farm
      int farmMapWidth;                       // Lebar peta farm

      // Memanggil method render setiap objek dalam Fields dan menyimpannya
      // dalam Canvas, kemudian melakukan hal yang sama dengan Animals dan
      // terakhir player.
      void UpdateFarmMap();

   public:
      // CTOR
      Scene();
      // CTOR user defined
      Scene(string _sessionName, int _farmMapHeight, int _farmMapWidth);
      // CCTOR
      Scene(Scene &oldScene);
      // DTOR
      ~Scene();

      // menampilkan jendela permainan di konsol
      void DrawScene();
};
```

2. Time

Kelas Time adalah blueprint dari objek yang menentukan waktu pada permainan.

```
class Time {
private:
   static int gameTime;              // Menampung game time dalam satuan tick
public:
   static void advanceGameTime();    // Menambah 1 tick pada game time
   static int getGameTime();   // Getter gameTime
};
```

3. LinkedList <T>

Class LinkedList adalah blueprint dari objek yang dapat menampung data dengan representasi LinkedList.

```
template <class T>
class LinkedList{
        private :
                struct Node{
                        T info;                 // Menampung informasi yang disimpan pada LinkedList
                        Node *next, *prev;              // Pointer
                };

                Node *list;                     // Untaian node yang membentuk linked list

        public :
                LinkedList();                           // Constructor

                bool isEmpty();                         // Mengembalikan True jika LinkedList kosong
                int find(T element);
        // Mengembalikan indeks di mana elemen ditemukan, -1 jika tidak ada
                void add(T element);                    // Menambahkan elemen sebagai elemen paling akhir
                void remove(T element);         // Membuang elemen dari linked list
                T get(int index);               // Mengembalikan elemen pada indeks

};
```

4. Counter <T>

Counter adalah kelas yang melakukan perhitungan terhadap objek-objek yang ada.

```
template <typename T>
class Counter
{
   private:
      // atribut yang mencatat obyek yang telah diinstansiasi
      static int objects_created;
      static int objects_alive;

   public:
      // CTOR
      Counter();
      // DTOR
      virtual ~Counter();
};

template <typename T> int Counter<T>::objects_created( 0 );
template <typename T> int Counter<T>::objects_alive( 0 );
```

5. Renderable

Class Renderable adalah blueprint dari objek yang di-inherit oleh objek-objek yang memiliki posisi.

```
class Renderable {
            public :
      Renderable();                    // Default ctor
      Renderable(int x, int y);
      // User defined ctor : menentukan koordinat x dan y dalam ctor

      virtual char render() const = 0;        // pure virtual dari method render
            void setX(int x);                 // setter X
            void setY(int y);                 // setter Y
            int getX() const;                 // getter X
            int getY() const;                 // getter Y

            private :
                  int x;                              // posisi secara horizontal
                  int y;                              // posisi secara vertikal
};
```

5.1. Cell

Class Cell untuk pengelompokan kelas dan akan diinherit oleh Land dan Facilities

```
class Cell : public Renderable {
   private:
      bool walkable;
      // Boolean walkable = true untuk objek yang dapat ditempati oleh Player atau Animal
   public:
      Cell();                 // Initialize walkable menjadi True

      // getter untuk walkable
            bool isWalkable() const;
            void setWalkable();       // setter untuk walkable
};
```

5.1.1. Facility

Class Facility adalah blueprint dari objek turunan "Cell" yang tidak dapat ditempati oleh Animal atau Player

```
class Facility : public Cell {
public:
   Facility();        // Ctor: assign walkable menjadi False
};
```

5.1.1.1. Mixer

Class Mixer adalah blueprint dari objek turunan "Facility" yang dapat menghasilkan SideProduct dari gabungan FarmProduct

```
class Mixer : public Facility {
        private :
                Product **listSideProduct;
                //struktur array
                // | Side Product | Farm Product 1 | Farm Product 2 |

        public :
                //CTOR
                Mixer();

                //DTOR
                ~Mixer();

                //Render
                char render() const override;

                //METHOD
                void mix();
};
```

5.1.1.2. Truck

Class Truck adalah blueprint dari objek yang digunakan untuk menjual Product

```
class Truck : public Facility {
        public :
                //CTOR
                Truck();

                //DTOR
                ~Truck();

                //Render
                char render() const override;                   // Mengembalikan karakter T

                //METHOD
                void convertToGold(Product product);    //proses penjualan, menghasilkan gold dengan menjual product
yang dipilih
};
```

5.1.1.3. Well

Class Truck adalah blueprint dari objek yang digunakan untuk menambah air yang dibawa player

```
class Well : public Facility {
        public :
                char render() const override;           // Mengembalikan karakter W
};
```

5.1.2. Land

Class Land adalah blueprint dari objek turunan "Cell" yang dapat ditempati oleh Animal atau Player

```
class Land : public Cell {
            protected :
                        bool grassed;
            // Menyimpan boolean untuk mengetahui Land tersebut ditumbuhi rumput atau tidak
                        // menentukan secara acak apakah Land akan berrumput (dipanggil oleh CTOR)
                        void randomGrass();

            public :
                        //CTOR : Memaanggil randomGrassed()
                        Land();

                        //SELECTOR
                        void setGrass(bool input);      // setter boolean grassed
                        bool getGrass() const;          // getter boolean grassed
};
```

5.1.2.1. Barn

Class Barn adalah blueprint dari objek turunan "Land" khusus MeatProducingFarmAnimal

```
class Barn : public Land {
    public:
        char render() const override;       // Mengembalikan karakter x
};
```

5.1.2.2. Coop

Class Coop adalah blueprint dari objek turunan "Land" khusus EggProducingFarmAnimal

```
class Coop : public Land {
    public:
        char render() const override;       // Mengembalikan karakter o
};
```

5.1.2.3. Grassland

Class Grassland adalah blueprint dari objek turunan "Land" khusus MilkProducingFarmAnimal

```
class Grassland : public Land {
    public:
        char render() const override;       // Mengembalikan karakter -
};
```

5.2. FarmAnimal

Class FarmAnimal adalah abstract base class yang memiliki method animal dasar dan menjadi dasar pembentukan class animal lainnya.

```
class FarmAnimal : public Renderable {
    protected:
        bool interactable;
        int hungryLevel;

    public:
        // CTOR User defined
        FarmAnimal(bool _interactable);

        // getter
        bool isInteractable() const;
        int getHungryLevel() const;

        // setter
        int setHungryLevel() const;

        // method untuk mengurangi nilai hungryLevel animal
        void eat();

        // method pure virtual untuk pergerakan animal
        // menerima List of Renderable* untuk mengecek cell yang akan ditempati
        virtual void autoMove(LinkedList<Renderable*> farmMap) = 0;

        // virtual method untuk mendapatkan suara hewan
        virtual string sound() const = 0;

        // virtual method untuk mengembalikan FarmProduct yang dihasilkan suatu hewan
        virtual Product* getProduct() const = 0;
};
```

5.2.1. EggProducingFarmAnimal

Class EggProducingFarmAnimal digunakan untuk mengelompokkan animal yang menghasilkan telur

```
class EggProducingFarmAnimal : public FarmAnimal {
    public:
        // method untuk pergerakan animal, hanya dapat bergerak di Coop
        // menerima List of Renderable* untuk mengecek cell yang akan ditempati
        void autoMove(LinkedList<Renderable*> farmMap) override;
};
```

5.2.1.1. Chicken

Class Chicken adalah blueprint dari objek FarmProduct yang menggambarkan ayam

```
class Chicken : public EggProducingFarmAnimal, public MeatProducingFarmAnimal {
    public:
        // mengembalikan karakter C
        char render() const override;

        // mengembalikan suara ayam misal "Petok petok!"
        string sound() const override;

        // mengembalikan FarmProduct yang behubungan dengan ayam
        Product* getProduct() const override;
};
```

5.2.1.2. Duck

Class Duck adalah blueprint dari objek FarmProduct yang menggambarkan bebek

```
class Duck : public EggProducingFarmAnimal {
    public:
        // mengembalikan karakter D
        char render() const override;

        // mengembalikan suara bebek misal "Kwek kwek!"
        string sound() const override;

        // mengembalikan FarmProduct yang behubungan dengan bebek
        Product* getProduct() const override;
};
```

5.2.2. MeatProducingFarmAnimal

Class MeatProducingFarmAnimal digunakan untuk mengelompokkan animal yang menghasilkan daging

```
class MeatProducingFarmAnimal : public FarmAnimal {
    public:
        // method untuk pergerakan animal, hanya dapat bergerak di Barn
        // menerima List of Renderable* untuk mengecek cell yang akan ditempati
        void autoMove(LinkedList<Renderable*> farmMap) override;
};
```

5.2.2.1. Chicken (sama dengan 5.2.1.1)

5.2.2.2. Cow

Class Cow adalah blueprint dari objek FarmProduct yang merepresentasikan sapi

```
class Cow : public MeatProducingFarmAnimal, public MilkProducingFarmAnimal {
    public:
        // mengembalikan karakter W
        char render() const override;

        // mengembalikan suara sapi misal "Moo!"
        string sound() const override;

        // mengembalikan FarmProduct yang behubungan dengan sapi
        Product* getProduct() const override;
};
```

5.2.2.3. Horse

Class Horse adalah blueprint dari objek FarmProduct yang merepresentasikan kuda

```
class Horse : public MeatProducingFarmAnimal {
    public:
        // mengembalikan karakter H
        char render() const override;

        // mengembalikan suara kuda misal "Ngihii!"
        string sound() const override;

        // mengembalikan FarmProduct yang behubungan dengan kuda
        Product* getProduct() const override;
};
```

5.2.2.4. Swine

Class Swine adalah blueprint dari objek FarmProduct yang merepresentasikan babi

```
class Swine : public MeatProducingFarmAnimal {
    public:
        // mengembalikan karakter S
        char render() const override;

        // mengembalikan suara babi misal "Ngookk"
        string sound() const override;

        // mengembalikan FarmProduct yang behubungan dengan babi
        Product* getProduct() const override;
};
```

5.2.3. MilkProducingFarmAnimal

Class MilkProducingFarmAnimal digunakan untuk mengelompokkan animal yang menghasilkan susu

```
class MilkProducingFarmAnimal : public FarmAnimal {
    public:
        // method untuk pergerakan animal, hanya dapat bergerak di Grassland
        // menerima List of Renderable* untuk mengecek cell yang akan ditempati
        void autoMove(LinkedList<Renderable*> farmMap) override;
};
```

5.2.3.1. Cow (sama dengan 5.2.2.2)

5.2.3.2. Goat

Class Goat adalah blueprint dari objek FarmProduct yang mewakilkan kambing

```
class Goat : public MilkProducingFarmAnimal {
    public:
        // mengembalikan karakter G
        char render() const override;

        // mengembalikan suara kambing misal "Mbheee!"
        string sound() const override;

        // mengembalikan FarmProduct yang behubungan dengan kambing
        Product* getProduct() const override;
};
```

5.3. Player

Class player merupakan kelas yang menggambarkan pemain utama yang menyimpan data-data seperti gold, currentWater, dan player facing serta memiliki method untuk player berinteraksi dengan farm-nya

```
#define NULL nil

class Player : public Renderable {
    // Data member:
private:
    const int maxWater; //kapasitas maksimum air yang dapat dibawa

    int gold; //uang yang dimiliki player
    int currentWater;

    int facing; //1 : facing up, 2 : facing down, 3 : facing right, 4 : facing left

    LinkedList<Product*> productInventory; // list of pointer to Product

public:
                //CTOR
                Player();
```

```
        //DTOR
        ~Player();

        //Render
        char render() const override;

        // mengembalikan true jika productInventory kosong / memiliki 0 elemen
        bool isInventoryEmpty();
        // menambahkan Product ke inventory
        void addToInventory(Product* element);
        // menghapus Product dari inventory
        void removeFromInventory(Product* element);
        // mengembalikan objek Renderable yang berada di depan player
        Renderable* getInFront(LinkedList<Renderable*>);

        // getter
        int getGold() const;
        int getFacing() const;
        int getCurrentWater() const;
        Product* getProductFromInventory(int index) const;
        //setter
        void setGold() const;
        void setCurrentWater(int water);

    // method untuk mengubah posisi player ke arah tertentu.
        // player mengubah orientasi tanpa bergerak jika collision dengan
        // obyek yang tidak dapat dilewati (isWalkable()==false).
    void move(int direction);

        // menerima objek Renderable, jika merupakan FarmAnimal mencetak hasil dari fungsi sound()
    void talk(Renderable*);

        // menerima objek Renderable dan melakukan interaksi berdasarkan jenis kelas turunannya
    void interact(Renderable*);

        // menerima objek Renderable, jika merupakan FarmAnimal membunuhnya dengan cara memanggil
        // dekostruktornya.
    void kill(Renderable*);                                 // Marked by Kim: Kill untuk masing-masing hewan menghasilkan product
berbeda, apa mending template aja?

        // menyiram (mengurangi air) petak yang ditempati player sehingga tumpuh rumput (grass)
    void grow();
};
```

6. Product

Kelas Product adalah *blueprint* dari seluruh produk yang dapat disimpan dan dijual.

```
class Product {
    private:
        int harga;              // Menyimpan harga dari Product

    public:
        int getHarga() const;   // Getter harga
        void setHarga();        // Setter harga
};
```

6.1. FarmProduct

Kelas FarmProduct adalah Product yang dihasilkan oleh FarmAnimal

```
class FarmProduct : Product {
    public:
        //Ctor
        FarmProduct();
};
```

6.1.1. ChickenEgg

Kelas ChickenEgg adalah FarmProduct dari Chicken.

```
class ChickenEgg: public FarmProduct, public Counter<ChickenEgg> {
    public:
    //Ctor yang mengatur harga awal
        ChickenEgg();
};
```

6.1.2. ChickenMeat

Kelas ChickenMeat adalah FarmProduct dari Chicken.

```
class ChickenMeat : public FarmProduct, public Counter<ChickenMeat> {
    public:
        //Ctor yang mengatur harga awal
        ChickenMeat();
};
```

6.1.3. CowMeat

Kelas CowMeat adalah FarmProduct dari Cow.

```
class CowMeat : public FarmProduct, public Counter<CowMeat> {
    public:
    //Ctor yang mengatur harga awal
        CowMeat();
};
```

6.1.4. CowMilk

Kelas CowMilk adalah FarmProduct dari Cow.

```
class CowMilk : public FarmProduct, public Counter<CowMilk> {
    public:
    //Ctor yang mengatur harga awal
        CowMilk();
};
```

6.1.5. DuckEgg

Kelas DuckEgg adalah FarmProduct dari Duck.

```
class DuckEgg : public FarmProduct, public Counter<DuckEgg> {
     public:
     //Ctor yang mengatur harga awal
        DuckEgg();
};
```

6.1.6. GoatMilk

Kelas GoatMilk adalah FarmProduct dari Goat.

```
class GoatMilk : public FarmProduct, public Counter<GoatMilk> {
    public:
    //Ctor yang mengatur harga awal
        GoatMilk();
};
```

6.1.7. HorseMeat

Kelas HorseMeat adalah FarmProduct dari Horse.

```
class HorseMeat : public FarmProduct, public Counter<HorseMeat> {
    public:
    //Ctor yang mengatur harga awal
        HorseMeat();
};
```

6.1.8. SwineMeat

Kelas SwineMeat adalah FarmProduct dari Swine.

```
class SwineMeat : public FarmProduct, public Counter<SwineMeat> {
    public:
        //Ctor yang mengatur harga awal
        SwineMeat();
};
```

6.2. SideProduct

Kelas SideProduct adalah *blueprint* dari Product yang dihasilkan dari gabungan FarmProduct.

```
class SideProduct : public Product {};
```

6.2.1. BeefRolade

Kelas BeefRolade adalah *blueprint* dari objek SideProduct yang bernama BeefRolade dan dihasilkan dari penggabungan FarmProduct CowMeat dan ChickenEgg.

```
class BeefRolade : public SideProduct, public Counter<BeefRolade> {
public:
    // Ctor BeefRolade, menentukan harga BeefRolade
    BeefRolade();
};
```

6.2.2. BeefSteak

Kelas BeefSteak adalah *blueprint* dari objek SideProduct yang bernama BeefSteak dan dihasilkan dari hasil penggabungan FarmProduct CowMeat dan CowMilk.

```
class BeefSteak : public SideProduct, public Counter<BeefSteak> {
    public:
        // Ctor BeefSteak, menentukan harga BeefSteak
        BeefSteak();
};
```

6.2.3. ChickenNugget

Class ChickenNugget adalah blueprint dari objek SideProduct yang bernama ChickenMeat dimana ChickenNugget merupakan gabungan ChickenMeat dan ChickenEgg.

```
class ChickenNugget : public SideProduct, public Counter<ChickenNugget> {
    public:
        // Ctor ChickenNugget, menentukan harga ChickenNugget
        ChickenNugget();
};
```