

Kelompok : 1. Muhammad Raihan Asyraf / 13517027  
2. Irfan Sofyana Putra / 13517078  
3. Kintan Sekar Adinda / 13517102  
4. Paulus Siahaan / 13517111

Asisten Mentor : Kevin Jonathan K

Asisten Penilai : .....

## 1. Deskripsi Umum Aplikasi

Aplikasi ini merupakan permainan yang memanfaatkan program berorientasi objek dalam bahasa C++. *Player* memiliki wadah air yang dapat menampung air dalam jumlah terbatas dan tas yang dapat menampung sejumlah terbatas *Product*. Pada Engi's Farm, terdapat *Cell* yang direpresentasikan dengan sebuah matriks 2D dengan ukuran (X,Y). Setiap *Cell* dapat berupa *Land*, daerah untuk beternak hewan; atau *Facility*, yaitu fasilitas peternakan. *Land* digunakan untuk tempat hidup hewan. *Land* dapat dikategorikan sebagai *Coop*, *Grassland*, atau *Barn*. *Coop* digunakan untuk beternak hewan penghasil telur, *Grassland* digunakan untuk beternak hewan penghasil susu, *Barn* digunakan untuk beternak hewan penghasil daging.

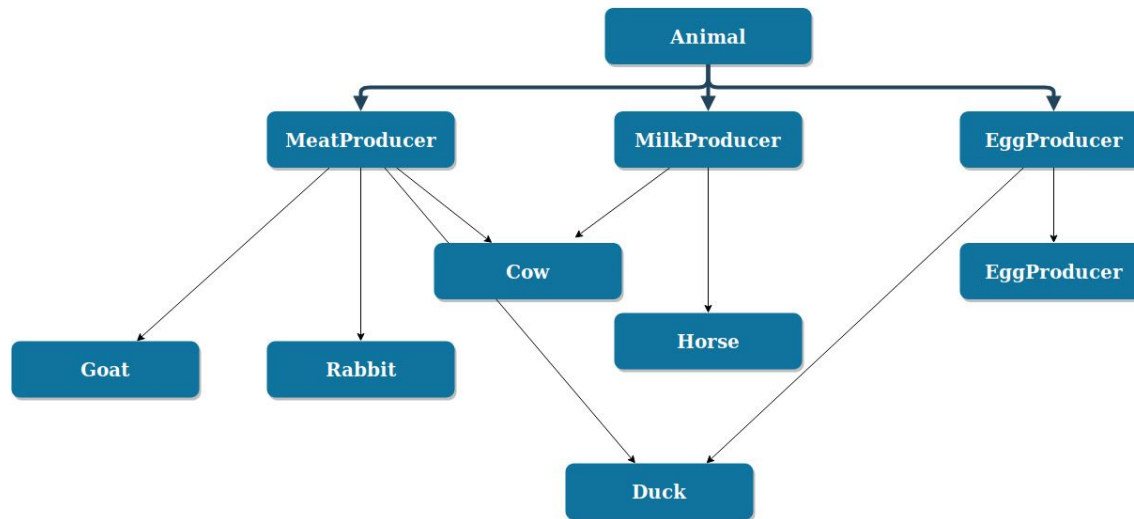
*Facility* dapat berupa *Well*, *Mixer*, atau *Truck*. *Well* digunakan untuk mengisi wadah air yang dimiliki *Player*; *Mixer* digunakan untuk membuat produk sampingan dari produk hewan; dan *Truck* digunakan untuk menjual barang hasil ternak. *Farm Animal* dapat dikategorikan sebagai *Egg Producing Farm Animal*, *Meat Producing Farm Animal*, dan *Milk Producing Farm Animal*. *Product* yang dijual dapat dikategorikan sebagai *Farm Product* dan *Side Product*.

*Player* dapat bergerak bebas pada semua *Cell* yang bertipe *Land* selama tidak ada hewan pada *Cell* tersebut. Setiap *Player* memiliki beberapa aksi yang dapat dilakukan selain bergerak, yaitu *Talk*, *Interact*, *Kill*, dan *Grow*. *Talk* digunakan untuk berbicara dengan hewan; *Interact* digunakan untuk berinteraksi dengan *FarmAnimal* atau *Facility* di samping *Player*; *Kill* untuk menyembelih hewan; *Grow* digunakan untuk menyiram *Land* dengan wadah air yang dimiliki dan menumbuhkan rumput di petak *Land* tempat *Player* berdiri.

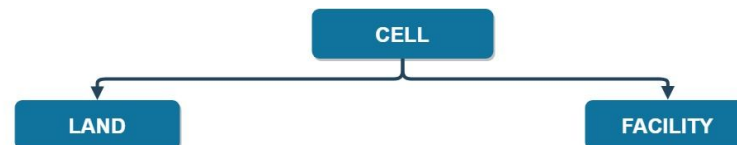
## 2. Diagram Kelas

### 2.1 Diagram Kelas

#### 2.1.1. Animal Class



#### 2.1.2. Cell Class



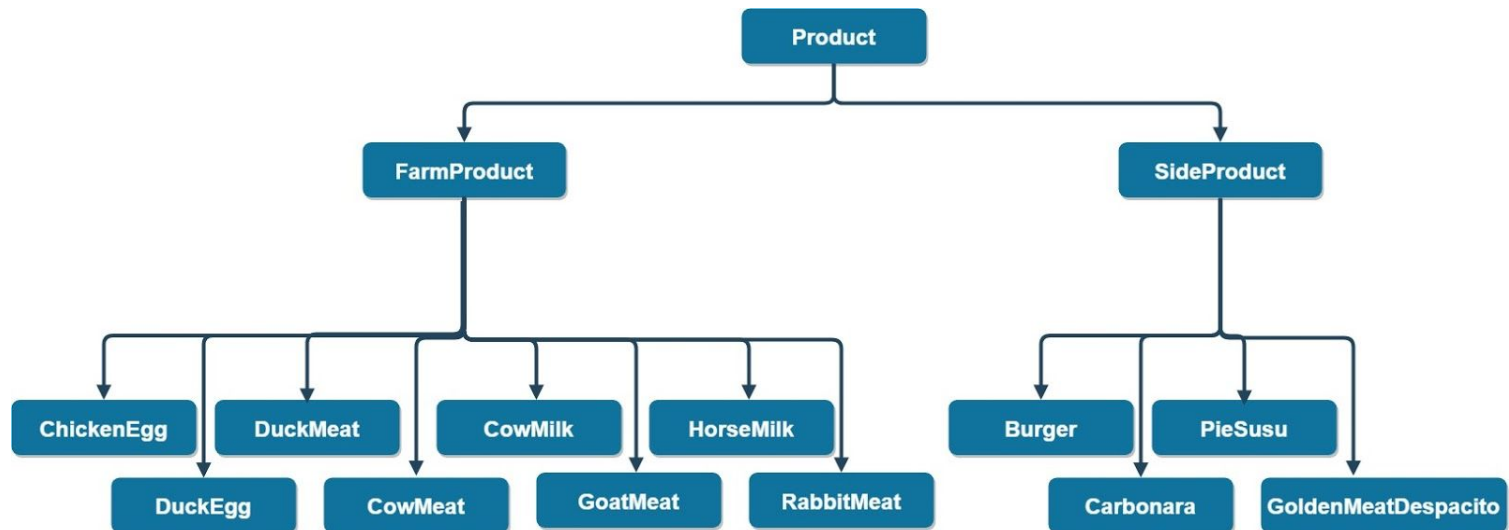
2.1.3. LinkedList Class



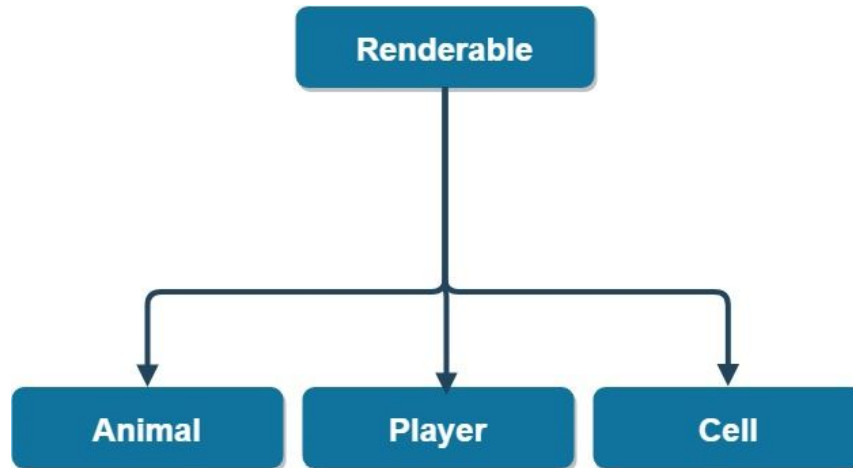
2.1.4. Player Class



2.1.5. Product Class



#### 2.1.6. Renderable Class



## 2.2 Ulasan Rancangan Final

### 2.2.1 Alasan Dipilihnya Desain Rancangan

Alasan kami memilih desain rancangan ini adalah kami merasa ini adalah desain yang mudah untuk diimplementasikan.

### 2.2.2 Bagian yang Mengalami Perubahan

1. Bagian kelas yang mengalami perubahan adalah Kelas Cell. Sebelumnya kelas ini memiliki *child class* coop, Grassland, Barn, Well, Mixer, dan Truck. Pada desain sekarang kami mengubah kelas cell hanya memiliki turunan yaitu kelas land dan facility.

### 2.2.3 Hal yang Berubah dari Rancangan Awal dan Alasan Melakukan Perubahan

1. Game.h

Pada rancangan sebelumnya belum ada Game.h. Game.h adalah sebuah file yang berisi kelas tentang Game dari Engi's Farm. Game.h memiliki atribut `map_facility`, `map_land`, dan `map_animal` yang merepresentasikan map dari permainan. Kemudian ada atribut `player` yang merepresentasikan dari player yang sedang bermain game.

## 2. Animal.h

Pada desain sebelumnya ada method `moveUp()`, `moveDown()`, `moveLeft()`, `moveRight()`. Pada desain sekarang kami menghilangkan method ini karena setelah implementasi, animal tidak membutuhkan method untuk bergerak dan masalah pergerakan animal diatur pada kelas `matrixOfAnimal`.

Kemudian kami menambahkan atribut `endurance`, `endurance_default` yang digunakan untuk mengatur apakah animal masih hidup atau tidak. Selain itu kami menambahkan getter dan setter untuk atribut `endurance` dan `endurance_default`. Kemudian kami perlu menambahkan pure virtual method dari `endurance`.

## 3. Chicken.h

Pada desain sebelumnya ada atribut `timeHungryChicken`, pada desain sekarang kami menghilangkan ini karena kami merasa ini tidak perlu. Sebagai gantinya kami menggunakan atribut `endurance` yang dapat diakses dari *parent class* nya yaitu `Animal.h`. Selain itu, di kelas ini ada implementasi dari getter dan setter `endurance`.

## 4. Cow.h

Pada desain sebelumnya ada atribut `timeHungryCow`, pada desain sekarang kami menghilangkan ini karena kami merasa ini tidak perlu. Sebagai gantinya kami menggunakan atribut `endurance` yang dapat diakses dari *parent class* nya yaitu `Animal.h`. Selain itu, di kelas ini ada implementasi dari getter dan setter `endurance`.

## 5. Duck.h

Pada desain sebelumnya ada atribut `timeHungryDuck`, pada desain sekarang kami menghilangkan ini karena kami merasa ini tidak perlu. Sebagai gantinya kami menggunakan atribut `endurance` yang dapat diakses dari *parent class* nya yaitu `Animal.h`. Selain itu, di kelas ini ada implementasi dari getter dan setter `endurance`.

## 6. Goat.h

Pada desain sebelumnya ada atribut `timeHungryGoat`, pada desain sekarang kami menghilangkan ini karena kami merasa ini tidak perlu. Sebagai gantinya kami menggunakan atribut `endurance` yang dapat diakses dari *parent class* nya yaitu `Animal.h`. Selain itu, di kelas ini ada implementasi dari getter dan setter `endurance`.

7. `Horse.h`

Pada desain sebelumnya ada atribut `timeHungryHorse`, pada desain sekarang kami menghilangkan ini karena kami merasa ini tidak perlu. Sebagai gantinya kami menggunakan atribut `endurance` yang dapat diakses dari *parent class* nya yaitu `Animal.h`. Selain itu, di kelas ini ada implementasi dari getter dan setter `endurance`.

8. `Rabbit.h`

Pada desain sebelumnya ada atribut `timeHungryRabbit`, pada desain sekarang kami menghilangkan ini karena kami merasa ini tidak perlu. Sebagai gantinya kami menggunakan atribut `endurance` yang dapat diakses dari *parent class* nya yaitu `Animal.h`. Selain itu, di kelas ini ada implementasi dari getter dan setter `endurance`.

9. `Cell.h`

Pada desain sebelumnya belum ada implementasi pure virtual dari `render`. Pada desain sekarang kami mengimplementasikannya

10. `Facility.h`

Pada desain sebelumnya belum ada implementasi pure virtual dari `render`. Pada desain sekarang kami mengimplementasikannya

11. `ListOfFacility.h`

Pada desain sebelumnya, kami membuat `ListOfFacility.h` sekarang kami menggantikannya dengan `MatrixOfFacility` karena dengan bentuk matrix maka proses pencarian fasilitas menjadi lebih cepat.

12. `MatrixOfFacility.h`

Pada desain yang sekarang, kami membuat `MatrixOfFacility.h` yang berisi matrix of kelas `facility`. Alasan melakukan perubahan agar lebih efektif dalam pengaksesan fasilitas.

13. `ListOfLand.h`

Pada desain sebelumnya, kami membuat ListOfLand.h sekarang kami menggantikannya dengan MatrixOfLand karena dengan bentuk matrix maka proses pencarian fasilitas menjadi lebih cepat.

14. MatrixOfLand.h

Pada desain yang sekarang, kami membuat MatrixOfLand.h yang berisi matrix of kelas facility. Alasan melakukan perubahan agar lebih efektif dalam pengaksesan fasilitas.

15. MatrixOfAnimal.h

Pada desain yang sekaang kami membuat MatrixOfAnimal yang berisi matrix pointer to child class dari Animal. Hal ini digunakan untuk memproses Animal yang ada pada game.

16. Player.h

Pada desain sekarang terdapat penambahan atribut FillWaterContainer(), SellAllProduct() yang digunakan untuk membantu implementasi method Interact. Kemudian terdapat penambahan method Print\_Inventory() dan Print\_Attribute() untuk mencetak status atribut dari player dan isi Inventory yang dimiliki oleh seorang player.

17. Product.h

Pada desain sekarang kami menambahkan atribut name pada product untuk masalah penampilan inventory. Oleh karena itu kami juga menambahkan getter dan setter untuk atribut name ini.

18. Renderable.h

Pada desain sebelumnya kami mendeklarasikan getter dan setter dari atribut posX dan posY menjadi virtual tetapi pada desain yang sekarang kami mendeklarasikan getter dan setter tersebut menjadi method biasa karena implementasi posX dan posY dari child class Renderable tetap sama.

#### 2.2.4 Keunggulan dan Kekurangan Rancangan

Keunggulan Rancangan dari kelompok kami adalah mudah digunakan karena menggunakan sedikit tentang pointer dan pembagian kelasnya jelas. Sementara kekurangannya adalah rancangan kami yang sekarang sangat berbeda jauh dengan desain awal rancangan.

#### 4. Skenario Test

Skenario	Hasil	Lolos (Y/N)
Memasukan command "MOVE UP"	Player bergerak ke atas	Y
Memasukan command "MOVE DOWN"	Player bergerak ke bawah	Y
Memasukan command "MOVE LEFT"	Player bergerak ke kiri	Y
Memasukan command "MOVE RIGHT"	Player bergerak ke kanan	Y
Memasukan command "TALK"	Hewan yang ada di sekitar Player bersuara	Y
Memasukan command "INTERACT"	Player akan berinteraksi dengan facility atau animal yang ada di sekitar player.	Y
Memasukan command "KILL"	Player akan membunuh hewan yang ada disekitarnya. Lalu jika hewan tersebut bisa menghasilkan product maka product akan masuk ke Inventory Player	Y



Memasukan command "GROW"	Player akan menyiram "Land" di posisi dia berada. Kemudian pada "Land" itu akan muncul rumput.	Y
Memasukan command "MIX"	Player akan mencampur beberapa product yang ada di inventory dan menghasilkan side product	Y
Memasukan command "HELP"	Program akan menampilkan menu HELP dan penjelasan semua yang ada di Game	Y
Waktu pada game merupakan kelipatan 5	Animal yang ada pada game akan bergerak secara random	Y
Jika Player sudah lapar dan 5 satuan waktu pada game berlalu	Animal akan mati	Y
Semua hewan yang ada game sudah mati	Program akan berhenti	Y

## 5. Pembagian Tugas dan Log Activity

### 5.1 Detail Pembagian Tugas Koding

Package	Class	Designer	Implementor	Tester
	Product, FarmProduct, SideProduct, DuckEgg, DuckMeat, ChickenEgg, CowMeat, CowMilk, HorseMilk, GoatMeat, RabbitMeat, Burger, PieSusu, Carbonara, GoldenMeatDespacito	M. Raihan Asyraf D. / 13517027	M. Raihan Asyraf D. / 13517027	M. Raihan Asyraf D. / 13517027

	Animal, MeatProducer, MilkProducer, EggProducer, Chicken, Duck, Goat, Rabbit, Cow, Horse	Kintan Sekar A. / 13517102	Kintan Sekar A. / 13517102	Kintan Sekar A. / 13517102
	Renderable	Irfan Sofyana Putra / 13517078	Irfan Sofyana Putra / 13517078	Kintan Sekar A. / 13517102
	LinkedList	Irfan Sofyana Putra / 13517078	Irfan Sofyana Putra / 13517078	
	Cell, Land, Facility	Paulus Siahaan / 13517111	Paulus Siahaan / 13517111	Paulus Siahaan / 13517111
	MatrixOfLand, MatrixOfFacility, MatrixOfAnimal	Paulus Siahaan / 13517111, Irfan Sofyana Putra / 13517078	Paulus Siahaan / 13517111, Irfan Sofyana Putra / 13517078	Paulus Siahaan / 13517111
	Player	M. Raihan Asyraf D. / 13517027, Irfan Sofyana Putra / 13517078	M. Raihan Asyraf D. / 13517027, Irfan Sofyana Putra / 13517078	Paulus Siahaan / 13517111
	Game, main(driver kelas game)	M. Raihan Asyraf D. / 13517027, Irfan Sofyana Putra / 13517078	M. Raihan Asyraf D. / 13517027, Irfan Sofyana Putra / 13517078	Paulus Siahaan / 13517111

## 5.2 Detail Pembagian Tugas Dokumentasi

Elemen Dokumentasi	Penanggung Jawab
Deskripsi Umum Aplikasi	Kintan Sekar Adinda/13517102

Diagram Kelas	Kintan Sekar Adinda/13517102
Ulasan Rancangan Final	Irfan Sofyana Putra/13517078
Skenario Test	Irfan Sofyana Putra/13517078, Kintan Sekar Adinda/13517102, M.Raihan Asyraf D./13517027
Pembagian Tugas dan Log Activity	M. Raihan Asyraf D./13517027, Kintan Sekar Adinda/13517102

### 5.3 Log Activity

No	Mulai	Selesai	Kegiatan	Hasil	Keterangan
1.	29 Maret 18.00	31 Maret 21.00	Implementasi kelas masing-masing	File.cpp dari masing-masing kelas	Irfan Sofyana Putra/13517078, Kintan Sekar Adinda/13517102, M.Raihan Asyraf D./13517027, Paulus Siahaan 13517111
2.	1 April 17.00	3 April 22.00	Menggabungkan file-file dan membuat main.cpp	main.cpp	Irfan Sofyana Putra/13517078, Kintan Sekar Adinda/13517102, M.Raihan Asyraf D./13517027, Paulus Siahaan 13517111
3.	4 April 19.00	5 April 20.00	Debugging program dan membuat dokumentasi	Program yang siap dijalankan dan dokumentasi	Irfan Sofyana Putra/13517078, Kintan Sekar Adinda/13517102, M.Raihan

					Asyraf D./13517027, Paulus Siahaan 13517111
--	--	--	--	--	--

## 5.4 Lampiran

### ASISTENSI TUBES OOP

#### 1. DESKRIPSI WAWANCARA

Nama Kelompok	Gazzz
Anggota yang hadir saat wawancara	1. M. RAHAN ASYRAF 2. PAULUS SIHAHAN 3. IRFAN SOFYANA 4. KINTAN SEKAR
Tempat, tanggal	Lab Pro, 26 Maret 2019
Waktu	13.00 WIB
Nama Asisten	Kevin Jonathan K

#### 2. HASIL WAWANCARA

No.	Pertanyaan	Jawaban
1	Bagaimana cara diinjak pindah kotak? & waktu pindahnya fix?	bebas, disarain 1 kotak, waktu pindah bisa fix
2	Hewan apakah langsung ada, atau bisa nambah?	bebas, sarain make file eksternal.
3	<del>Apakah</del> Apakah hewan bisa menghasilkan produk > 1 kali & bisa makan saat sudah menghasilkan produk?	untuk susu & telur bisa, bisa makan & jika blm diinteract.
4	Jika ada 2 hewan/lebih saat memasukkan command, bagaimana perilaku player? yg diot	bebas, bisa langsung interact semua.
5	Ada berapa batas max tas & apakah jika jml barang, tas langsung kosong?	batas max dibebasin, jml barang tas langsung kosong.

