Spesifikasi Tugas Besar IA
IF2210 - Pemrograman Berorientasi Objek
Engi's Farm

I. Deskripsi Umum Tugas

Tugas ini adalah penerapan dari topik yang telah dibahas terkait konsep Pemrograman Berorientasi Objek.

II. Tujuan

1. Mahasiswa mampu merancang kelas dengan paradigma berorientasi objek, untuk menghasilkan rancangan kelas dan pengelompokan kelas yang sesuai dengan aturan serta dokumentasi rancangan dengan baik
2. Mahasiswa mampu mengantisipasi dan mengelola perubahan spesifikasi perangkat lunak

III. Deskripsi Persoalan

Sudah hampir setahun semenjak Chef membuka restoran Engi's Kitchen. Chef telah lelah mengurus bisnis restorannya yang sudah berkembang pesat. Banyaknya peran yang harus ia ambil setelah ekspansi kurang lebih empat bulan lalu membuatnya lelah. Hari itu adalah hari yang cerah, Chef, sedang bebersih rumahnya. Saat membersihkan rumahnya, ia menemukan sebuah amplop berwarna putih bersih tanpa tulisan apapun di atasnya. Chef membuka amplop putih tersebut dan menemukan sepucuk surat yang ditulis oleh kakeknya yang telah meninggal sepuluh tahun lalu. Pada surat tersebut, tertulis bahwa kakeknya telah mewariskan padanya sebuah peternakan untuknya. Melihat kesempatan ini, Chef pun menutup restoran yang telah ia buka nyaris setahun. Chef tidak membuang waktu dan segera mencari peternakan warisan kakeknya untuk membuka Engi's Farm. Petualangan Chef di Engi's Farm pun dimulai.

Pemain (**`Player`**) memiliki wadah air yang dapat menampung air dalam jumlah terbatas dan tas yang dapat menampung sejumlah terbatas `Product`.

Pada Engi's Farm, terdapat kumpulan petak tanah (**`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.

Hewan ternak (**`Farm Animal`**) dapat dikategorikan sebagai penghasil telur (**`Egg Producing Farm Animal`**, penghasil daging (**`Meat Producing Farm Animal`**), dan penghasil susu (**`Milk Producing Farm Animal`**). Hasil yang dapat dijual oleh peternakan (**`Product`**) dapat dikategorikan sebagai hasil ternak (**`Farm Product`**) dan hasil sampingan peternakan (**`Side Product`**).

IV. Deskripsi Mekanisme

IV.1. Mekanisme Umum

Pada saat memulai permainan, inisialisasi seluruh Cell berdasarkan *layout* yang telah dibuat sebelumnya (baik melalui file eksternal ataupun *hardcode*). Peta harus memuat semua kelas yang dapat ditampilkan (misal : FarmAnimal, Facility, Land, Player). Hewan yang lapar harus ditampilkan berbeda dengan hewan yang tidak lapar. Begitu pula dengan Land yang memiliki rumput dan tidak

Jika perlu, buatlah sebuah kelas abstrak **`Renderable`** dengan virtual method `render()` yang mengembalikan karakter yang merepresentasikan objek tersebut bila ditampilkan untuk membantu dalam menampilkan objek.

Berikut merupakan contoh Ilustrasi Engi's Farm (warna pada ilustrasi hanya untuk mempermudah visualisasi)

C	o	**c**	-	x	**G**	x	x	-	-	**T**	
o	*	*	-	x	x	x	x	-	-	**M**	
o	*	o	-	x	x	@	@	-	-	-	**Engi's Farm**
-	-	-	-	x	x	x	@	-	-	**W**	Inventory
-	-	-	-	x	x	**g**	x	-	-	-	- ChickenEgg
-	-	-	-	x	x	x	x	-	-	-	- ChickenEgg
-	-	-	-	-	-	-	-	-	-	P	- BeefRolade
-	-	#	-	-	-	-	**H**	-	-	-	
-	-	#	#	-	**H**	-	-	-	-	-	Money : 0
-	-	-	#	#	-	-	-	-	-	-	Water : 0
Command: **INTERACT**											

Keterangan:
C : Ayam
G : Kambing
H : Kuda
T : Truck
M : Mixer
W : Well

P : Player

- : Grassland
x : Barn
o : Coop

*, @, # : Land dengan rumput

IV.2. Mekanisme `Player`

`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.

`Player` dapat berinteraksi dengan `Well` untuk **mengisi air**, dan berinteraksi dengan `Truck` untuk mengosongkan tas dan **menjual seluruh** `Product` pada inventory dan mendapatkan uang. Setiap `Product` memiliki harga yang berbeda. Setelah Player berinteraksi dengan `Truck`, maka `Truck` tidak dapat digunakan untuk jangka waktu tertentu. Permainan akan selesai apabila jumlah hewan pada Engi's Farm sudah habis.

`Player` dapat menggunakan command **Mix** pada `Mixer` untuk membuat produk sampingan (`SideProduct`). `Player` harus mengkombinasikan bahan-bahan dari tas (minimal dua bahan [boleh sama]; cara implementasi dan interaksi dibebaskan) untuk mendapatkan sebuah `SideProduct`, (misal: memasukkan `ChickenEgg` dan `CowMeat` untuk mendapatkan `BeefRolade`).

Perintah Kill hanya dapat digunakan pada hewan **`MeatProducingFarmAnimal`** untuk mendapatkan `Product` daging. Namun hewan tersebut akan mati. Sebaliknya, perintah Interact hanya dapat digunakan pada hewan **`EggProducingFarmAnimal`** dan **`MilkProducingFarmAnimal`**

IV.3. Mekanisme `FarmAnimal`

Setiap petak `Land` pada Engi's Farm hanya dapat diisi oleh satu ekor hewan pada satu waktu. Setiap petak `Land` dapat ditumbuhi rumput sebagai makanan hewan ketika disiram air dengan wadah air yang dimiliki pemain.

Pada Engi's Farm Hewan ternak **dapat bergerak secara acak** selama berada di area Land yang sesuai. Hewan juga dapat mengeluarkan suara ketika diajak berbicara (berupa teks yang dicetak pada layar). Misal ayam berbunyi "petok" atau sapi berbunyi "moo". Seekor hewan **dapat dikategorikan sebagai lebih dari satu jenis hewan ternak.**

Seperti halnya hewan ternak biasa, hewan ternak di Engi's Farm juga bisa merasakan rasa lapar. Hewan ternak **akan merasa lapar setelah waktu tertentu**. Jika setelah hewan merasa lapar, hewan tidak mendapat makan setelah 5 tick maka hewan tersebut akan **mati**. Hewan ternak harus makan rumput yang tumbuh pada `Land` agar tidak lapar. Setiap jenis hewan memiliki waktu yang berbeda sebelum merasakan lapar. Hewan akan **secara otomatis memakan rumput** pada tempat ia berdiri jika merasa lapar.

Seekor hewan ternak dapat menghasilkan sebuah **`FarmProduct`** setelah memakan rumput. Ketika pemain berinteraksi dengan hewan ternak atau menyembelih hewan tersebut, maka pemain akan mendapatkan `FarmProduct` milik hewan tersebut (misal: `Chicken` [`EggProducingFarmAnimal`] dapat menghasilkan `ChickenEgg`). Jika pemain

belum mengambil hasil ternak yang sebelumnya dihasilkan, maka hewan tidak akan menghasilkan produk tambahan jika hewan memakan rumput lagi.

Pada **`MeatProducingFarmAnimal`**, hewan tidak perlu memakan rumput untuk menghasilkan daging ketika disembelih.

Untuk tugas ini buat setidaknya **6** (enam) kelas riil `FarmAnimal`, **6** (enam) kelas riil `FarmProduct`, dan **3** (tiga) kelas riil `SideProduct` (total 15 kelas)

V. Spesifikasi

Buatlah desain kelas untuk diimplementasikan yang merepresentasikan deskripsi Engi's Farm di atas. Kelas, properti, dan method yang dideskripsikan di atas adalah spesifikasi minimal yang harus dimiliki oleh kelas yang dibuat. Diperbolehkan untuk menambahkan definisi kelas, properti, dan method lain jika diperlukan.

Anda **hanya perlu membuat header file dari setiap kelas** (file .h atau .hpp) untuk tugas ini. Adapun spesifikasi lain pada tugas ini adalah:

1. Setiap kelas, method, serta atribut harus **dilengkapi dengan komentar** yang menjelaskan kegunaan kelas, method, serta atribut (dapat dipahami orang lain).
2. Setiap kelas dituliskan pada **file terpisah**.
3. Lampirkan diagram kelas untuk keseluruhan kelas. Gunakan tools seperti **Doxygen** untuk mengenerate diagram kelas bila perlu. Pastikan diagram kelas tersebut **dapat dibaca dengan mudah.**
4. Desain kelas akan digunakan pada tugas besar selanjutnya. Buatlah desain kelas yang paling tepat menurut anda. **Perubahan desain kelas pada tugas besar selanjutnya harus disertai dengan alasan perubahan.**
5. Buatlah kelas dengan penulisan nama yang teratur baik dari sisi penamaan maupun aturan penulisan (huruf besar), nama dari sebuah kelas haruslah berupa kata benda. Begitu pula dengan nama method seharusnya berupa kata kerja. Jika perlu ikuti aturan penulisan yang tersedia pada beberapa website.
6. Desain harus memiliki elemen pemrograman berorientasi objek, di antaranya:
    a. Inheritance
    b. Method overriding
    c. Aggregation
    d. Polymorphism
    e. Generic (ditandai dengan penggunaan kelas pada dua tipe berbeda)
    f. Abstract Class
    g. Function/method overloading
    h. Interface (abstract class tanpa implementasi)
    i. Multiple inheritance

7. Buatlah kode dengan baik dengan memanfaatkan elemen pemrograman berorientasi objek, yaitu kode yang:
    a. Tidak memiliki kode duplikat
    b. Memiliki struktur kelas yang mudah dipahami
    c. Implementasi yang tidak terlalu kompleks
    d. Mengikuti prinsip SOLID
8. Selain entitas yang sudah disebutkan sebelumnya, program yang Anda buat diwajibkan mengimplementasikan kelas linked list yang dapat menyimpan tipe generic. Method minimal yang harus dibuat diantaranya
    a. find(T element) → int
    Mengembalikan indeks dimana elemen ditemukan, -1 jika tidak ada
    b. isEmpty() → boolean
    Mengembalikan True jika linked list kosong
    c. add(T element) → void
    Menambahkan elemen sebagai elemen paling akhir
    d. remove(T element) → void
    Membuang elemen dari linked list
    e. get(int indeks) → T
    Mengembalikan elemen pada indeks

VI. Protokol Pengerjaan Tugas

Bagian ini berisi deskripsi tentang pembagian kelompok, waktu pengerjaan, dan *deliverables* yang harus dikumpulkan.

VI.1. Deskripsi Umum

1. Tugas ini dikerjakan secara berkelompok. Satu kelompok beranggotakan **4 orang** dari kelas yang sama. Nama dan anggota kelompok didaftarkan pada *spreadsheet* yang dapat diakses di Olympia.
2. Diperbolehkan membentuk kelompok dengan anggota 5 orang dengan jumlah maksimum 3 kelompok tiap kelas.

VI.2. Deliverables

Pada saat pengumpulan tugas, mahasiswa diharuskan untuk mengumpulkan *deliverables* sebagai berikut.

1. **Softcopy** dokumen, dimasukkan pada folder **doc** yang berisi:
    a. File gambar diagram kelas **yang dapat dibaca**
    b. Dokumentasi kelas dalam bentuk pdf berisi deskripsi kelas dan method

2. **Kode Sumber**, dimasukkan pada folder **src** yang berisi:
    a. Seluruh file header yang telah dibuat (sebaiknya dimasukkan ke subfolder untuk memperjelas pembagian kelas)

Seluruh deliverable akan **di-zip** dengan format nama

EF1A_<NIM1>_<NIM2>_<NIM3>_<NIM4>.zip

Deliverables untuk Tugas Besar 1A dikumpulkan selambat-lambatnya pada 20 Maret 2019 pukul 15.00 WIB melalui *uploader* yang disediakan di Olympia

VII. Catatan Perubahan Deskripsi Tugas

Versi	Tanggal Rilis	Keterangan
v1.0	27 Februari 2019	-
v1.0.1	4 Maret 2019	Perubahan return function pada *linked list*