

Deskripsi Umum Tugas

Tugas ini adalah “capstone” dari semua topik yang dibahas s.d. UTS, yaitu Konsep OOP, OOD, Pemrograman secara OO dengan bahasa C++.

Mahasiswa diharap bekerja dengan sungguh-sungguh dan jujur.

Tujuan Tugas ini :

1. Mahasiswa mampu merancang dengan paradigma OO, untuk menghasilkan rancangan kelas dan pengelompokan kelas yang sesuai dengan kaidah, serta mendokumentasi rancangan dengan baik
2. Mahasiswa mampu mengimplementasi rancangannya dalam bahasa C++ dengan mengaplikasikan kaidah implementasi yang baik dan coding rules sesuai yang ditetapkan
3. Mahasiswa mampu memakai tools yang sesuai untuk merancang dan implementasi program, menganalisis statik source code, melakukan unit test, membuat dokumentasi berdasarkan source code
4. Mahasiswa mampu bekerja dalam tim dengan pembagian peran yang baik dan mengatur waktu untuk mencapai target
5. Mahasiswa mampu mengantisipasi dan mengelola perubahan spesifikasi perangkat lunak

1 Virtual Zoo

Kebun binatang adalah tempat **hewan** dipelihara dalam lingkungan buatan, dan dipertunjukkan kepada publik.¹ Salah satu fungsi utama kebun binatang adalah sebagai sarana rekreasi publik, dimana selain memiliki kandang-kandang tempat dipeliharanya hewan, kebun binatang juga menyediakan sarana rekreasi lainnya seperti taman piknik, rumah makan, dan fasilitas umum lainnya. Tugas Anda dalam tugas besar ini adalah membuat kebun binatang virtual dengan menggunakan pemrograman berorientasi objek dengan bahasa C++.

1.1 Deskripsi Virtual Zoo Universe

Sebuah kelas `Zoo` memiliki sebidang tanah yang direpresentasikan dengan sebuah matriks `Cell`, dimana setiap `Cell` merepresetasikan petak tanah berukuran 1 x 1m. Sebuah `Cell` dapat berupa `Habitat`, yaitu lingkungan dimana hewan tinggal, dan `Facility`, yaitu fasilitas umum untuk pengunjung. Lebih lanjut lagi `Habitat` dapat berupa `LandHabitat`, `WaterHabitat`, dan `AirHabitat`, untuk binatang darat, air, dan udara. Sedangkan `Facility` dapat berupa `Road`, `Park`, dan `Restaurant`. `Road` juga memiliki turunan bentuk khusus yaitu `Entrance` dan `Exit` yang merepresentasikan pintu masuk dan pintu keluar kebun binatang.

Kelas `Animal` adalah abstract base class dari semua jenis hewan. Secara umum `Animal` dikelompokkan menjadi `LandAnimal`, `WaterAnimal`, dan `FlyingAnimal`.

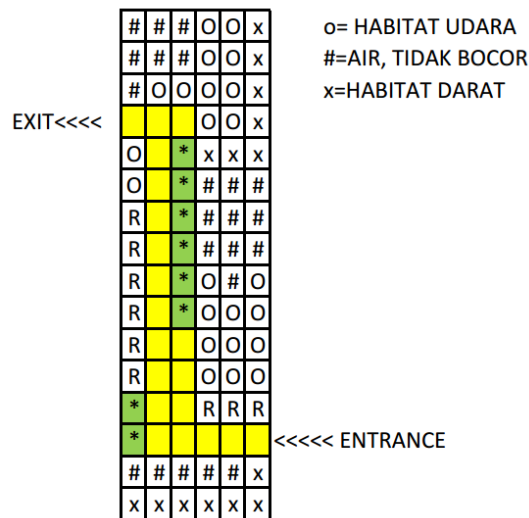
¹ https://id.wikipedia.org/wiki/Kebun_binatang

Taksonomi hewan lebih detail dapat disusun berdasarkan famili, genus, spesies hewan, misalnya *Cat*, *Fish*, dll. Anda diminta untuk mendefinisikan sendiri subset taksonomi hewan dengan mengacu ke suatu referensi yang anda tentukan untuk keperluan tugas ini, yang tak perlu mencakup semua taksonomi dalam zoology, tetapi minimal harus memunculkan 20 kelas riil hewan.

Hewan tinggal di dalam *Cage* yang berada di atas satu atau lebih *cell Habitat* sejenis yang saling berhubungan melalui sebuah “penyekat” yang bisa dibuka-tutup saat kebun binatang. Perhatikan bahwa sebuah *Cage* akan mencakupi satu/banyak *cell Habitat* sedangkan satu *cell Habitat* hanya dimiliki oleh satu *Cage*. Asumsikan bahwa semua hewan di dalam Virtual Zoo ini sudah dijinakkan, sehingga hewan yang berbeda jenis tidak akan saling serang. Karena itu satu *Cage* bisa berisi lebih dari satu jenis hewan. Tetapi peletakan hewan di dalam *Cage* harus memperhatikan jenis habitat yang cocok untuk tempat tinggalnya.

1.2 Ilustrasi deskripsi struktural Virtual Zoo

Ilustrasi sebuah Mini Zoo berukuran 6x16:



1.3 Deskripsi Fungsional dan Perilaku

Proses inisiasi sebuah Zoo dilakukan dengan cara:

- Menginisasi bidang tanah dengan cara membuat objek-objek *Habitat* dan *Facility* dan meletakkannya ke dalam matriks *Cell* sesuai dengan desain layout kebun binatang yang dirancang.
- Membuat objek-objek *Cage* dan meletakkannya di atas *Habitat* yang sesuai yang sudah dibuat di langkah ke-i.
- Membuat objek-objek binatang dan meletakkannya di dalam *Cage* yang sesuai.

Sebuah kelas abstrak *Renderable* merepresentasikan perilaku objek yang dapat digambar di atas layar. Kelas ini mendefinisikan `render()` yang

mengembalikan satu karakter yang merepresentasikan bentuk objek yang bersangkutan di atas console teks (pada gambar di atas : {*, #, x, O, R}). Kelas `Cell` dan `Cage` adalah contoh sebuah `Renderable`. Karakter tersebut bisa dikonfigurasi untuk tampilan lain selain gambar di atas, misalnya, `WaterHabitat` digambarkan sebagai `w`, `Cage` di atas `WaterHabitat` digambarkan sebagai `W`, `Road` sebagai `+`, `Park` sebagai `#`, dan sebagainya. Dengan bantuan method `render` ini, Anda dapat menampilkan Virtual Zoo di atas console teks.

Kelas `Animal` memiliki virtual method `interact()` yang mengirimkan sebuah yang menggambarkan yang dapat didengar, dirasakan, atau dilihat oleh seorang pengunjung saat berdekatan dengan jenis hewan tersebut. Semua turunan riil dari kelas `Animal` harus mengimplementasikan method ini, misalnya `Lion` “auum”, `Crocodile` “splash”, `Fish` “wet”, `Bird` “cuit”, dan seterusnya.

2 Deskripsi Tugas

Buatlah program berorientasi objek yang merepresentasikan deskripsi di atas. Kelas, properti, dan method yang dideskripsikan di atas adalah spesifikasi minimal yang harus dimiliki oleh program yang di buat. Silahkan tambahkan lagi definisi kelas, properti, dan method lain yang dianggap perlu.

Buatlah kelas `Driver` yang akan menginisiasikan dunia Virtual Zoo yang Anda rancang. Proses inisiasi dan konfigurasi objek-objek dapat dilakukan di dalam kelas `Driver`.

Bonus: inisiasi dan konfigurasi objek-objek dilakukan dengan cara membaca konfigurasi dari sebuah file. Anda dapat menggunakan mesin pembaca pita karakter untuk melakukan ini.

`Driver`, anda realisasi sebagai sebuah menu berbasis teks yang memberikan menu pilihan kepada pengguna sebagai berikut:

1. “Display Virtual Zoo”

Jika pengguna memilih menu ini, program akan menampilkan Zoo di atas layar dengan memanfaatkan method `render`. Untuk membatasi luas bidang yang akan ditampilkan, pengguna diminta memasukkan koordinat atas-kiri, dan bawah-kanan.

2. “Tour Virtual Zoo”

Jika pengguna memilih menu ini, program akan secara acak memilih salah satu `Entrance`, kemudian membuat sebuah jalur tour yang dilalui pengunjung. Untuk setiap `Cell` yang dilalui oleh pengunjung, program menampilkan serangkaian yang akan dialami pengunjung pada `Cell` itu berdasarkan interaksi dengan hewan-hewan yang ada di setiap `Cage` yang bersinggungan dengan `Cell` tersebut. Algoritma pemilihan jalur tour dapat dilakukan dengan cara memilih `Cell` (bertipe `Road`) yang bersinggungan dengan `Cell` yang belum pernah

dikunjungi sebelumnya. Jika ada lebih dari satu `Cell` bertipe `Road` yang dapat dipilih, maka pilihlah secara acak. Penelusuran akan berhenti saat sudah tidak ada lagi `Road` yang dapat dipilih, atau telah mencapai `Cell` bertipe `Exit`.

Untuk mentest driver tersebut secara otomatis (penilaian oleh autograder system Olympia), akan diberikan contoh yang dapat diakses di Olympia.

3 Protokol Pengerjaan Tugas

Bagian ini memuat Pembagian Kelompok, Estimasi Waktu Pengerjaan, dan Deliverable

3.1 Deskripsi Umum

1. Tugas ini dikerjakan secara berkelompok. Satu kelompok beranggotakan 4 orang. Nama dan anggota kelompok didaftarkan pada spreadsheet yang dapat diakses di Olympia.
2. Semua deliverable dikumpulkan sesuai dengan yang dijelaskan pada bab 3.2. Deliverable yang terlambat, atau tak hadir saat demo, akan diberi nilai NOL.
3. Tugas ini dikerjakan selama 3 minggu dengan tahapan sebagai berikut:
 - a. Minggu pertama: desain. Deliverable: draft desain program
 - b. Minggu kedua: coding. Deliverable: unfinished program
 - c. Minggu ketiga: finishing, testing. Deliverable: finished program, dokumentasi final (berisi desain final, hasil testing, evaluasi kualitas program, dan deskripsi pembagian tugas anggota kelompok)Beberapa jam kuliah akan dialokasikan untuk mengerjakan TB ini.
4. Di akhir minggu ketiga, setiap kelompok akan mendemonstrasikan programnya.

3.2 Deliverables

Penilaian akan dilakukan berdasarkan deliverables sebagai berikut

1. **Hardcopy**, dokumentasi TB1, 1 eksemplar dicetak, per kelompok, yang minimal berisi butir-butir sbb, diserahkan sesuai jadwal di lotek TU Program Studi teknik Informatika ITB. Setoran yang terlambat tidak akan dinilai (nilai hardcopy adalah NOL):
 - a. Deskripsi umum aplikasi (user point of view)
 - b. Rancangan Kelas (dalam bentuk diagram kelas rancangan, dan ekstraksi diagram kelas menggunakan tools). Anda boleh menyerahkan **beberapa** versi rancangan dan hanya **sebuah** rancangan yang terbaik yang anda implementasi.
 - c. Ulasan tentang rancangan (atau alternatif beberapa rancangan) yang anda rancang dan kemudian akhirnya satu yang dipilih untuk diimplementasi, dan keunggulan serta kekurangannya. Ulasan ini boleh mengacu ke hal-hal detail yang hanya diserahkan dalam bentuk softcopy
 - d. Skenario test (bukan unit test), sebagai persiapan demo
 - e. Log-activity kegiatan pengerjaan
2. **Softcopy** disubmisi lewat sistem submisi Olympia :
 - a. **Rancangan** kelas hasil ekstraksi dari source code, dalam bentuk diagram kelas UML, dengan menggunakan tools **<TBD>**

- b. Keseluruhan source code disertai dengan test cases yang minimal dapat mencoba driver dan builder (makefile) yang disubmisi ke sistem Olympia, untuk digrading secara otomatis
 - c. Hasil generate dokumentasi dengan Doxygen
 - d. Hasil pemakaian unit testing C++ menggunakan tools <TBD>
 - e. Hasil pemeriksaan white code dengan tools yang ditetapkan (<TBD>)
3. **Artefak untuk demo (eksekusi, check)** : Semua source code dan executable code siap untuk demo, harus ada di laptop masing-masing.
Setiap laptop harus siap untuk dipakai demo: berisi tools yang dipakai, yaitu IDE, dan semua tools yang dipakai (static analyser untuk mengases dan mengukur metriks source code, unit testing, dokumentasi, reverse engineering menjadi UML)

3.3 Jadwal Pengerjaan Tugas

Pada bagian ini hanya dituliskan milestone pengerjaan. Di luar milestone ini, mahasiswa bekerja secara mandiri.

Milestone pengerjaan

Week	Tanggal/Perioda	Deskripsi Aktivitas	Deliverable
W05	17-Feb-2017	Rilis Tugas	Kelompok ditentukan
W07	23-Feb-2017	Instalasi tools yang dibutuhkan, mentoring dengan asisten mentor	