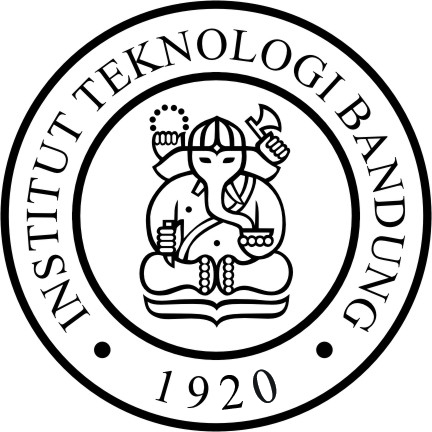
Laporan Tugas Besar IF2210 Pemograman Berorientasi Objek

Semester II tahun 2016/2017

**Virtual Zoo VZ03**



Kelompok ‘Sebut Saja Mawar’

Anggota:

Alivia Dewi Parahita 13515018

Helena Suzane Graciella 13515032

Emilia Andari Razak 13515056

Luthfi Fadillah 13515072

INSTITUT TEKNOLOGI BANDUNG

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

JALAN GANESHA NO.10 BANDUNG

# Deskripsi Umum Aplikasi

Sebuah aplikasi kebun binatang. Kebun binatang sendiri merupakan tempat memelihara hewan di lingkungan buatan dan menjadi wisata publik. Tidak hanya lingkuan buatan untuk hewan saja, untuk kenyamanan pengujung, juga disediakan fasilitas seperti taman piknik, rumah makan, serta fasilitas umum lainnya. Dalam tugas besar ini dibuatlah kelas Zoo yang memiliki sebidang tanah yang direpresentasikan matriks Cell, dimana Cell merepresentasikan tanah 1x1m. Sebuah Cell dapat berupa Habitat(LandHabitat, WaterHabitat, AirHabitat) atau Facility(Road, Park, Restaurant). Pada Road terdapat suatu pintu khusus yang dinamakan Enterance dan Exit.

Selain Zoo dan Cell, terdapat kelas absrak Animal yang merupakan base class dari semua jenis hewan. Secara umum Animal dikelompokkan menjdi LandAnimal, WaterAnimal, dan FlyingAnimal. Pendetailan pada hewan selanjutnya didefinisikan sebagai bubset taksonomi hewal dengan mengacu ke suatu referensi biologi.

Untuk menggabungkan Cell dan Animal, dibuat suatu kelas Cage yang berada diatas satu atau lebih cell Habitat yang saling berhubungan. Didalam Cage ini pulalah tempat Animal tinggal. Sebuah Zoo akan terdiri atas beberapa Cell

| **No Rilis** | **Deskripsi Umum Rilis** | **Deskripsi Umum Perubahan** |
| --- | --- | --- |
| VZ03 | Tugas ini adalah permintaan /penambahan terhadap aplikasi Virtual Zoo yang sedang anda kerjakan. Versi ini sudah mengandung Inheritance dan tidak mengandung Generik | Adanya inheritance dan/atau generik untuk hubungan antar kelas yang telah dibuat. |

# Rancangan Kelas + Ulasan kalo ada evolusi perubahan dari rancangan sebelumnya

# *Snapshot* Struktur Direktori atau Sub direktori

# Daftar Lampiran Tidak di Cetak (Diagram kelas hasil reverse engineering, doxygen, unit test dgn google test, Static Code Test dgn CppCheck)

* 1. **Skenario Test**

**Form Penilaian Functional Test**

Versi: VZ03 dengan generik/inheritance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skenario** | **Keterangan** | **Fakta** | **Nilai Mhs** | **Nilai Ass** |
| **General** | | | | |
| Inisiasi, View kemudian quit | OK |  | A |  |
| **Display Virtual Zoo** | | | | |
| Display Map (Partial) | OK |  | A |  |
| Display Map (Full) | OK |  | A |  |
| Move Animal | OK |  | A |  |
| Display Jumlah Makanan | OK |  | A |  |
| **Tour Virtual Zoo** | | | | |
| Tampilkan Narasi Tour | OK |  | A |  |
| **Bonus** | | | | |
| Save ke File | No |  | **E** |  |
| Retrieve dari File | OK |  | **A** |  |

**Form Penilaian Unit Test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelas / Keluarga Kelas** | **Method** | **Kasus** | **OK/NO** | **Nilai** |
| Beluga | Beluga() | - |  |  |
|  | Interact() | - |  |  |
|  | Render() | Mengecek apakah Render sudah sesuai Jenis Hewan | OK |  |
|  | Cetacea() | - |  |  |
|  | Animals() | - |  |  |
|  | ~Animals() | - |  |  |
|  | GetBerat() | Mengecek apakah berat sudah sesuai keadaan ketika dihidupkan | OK |  |
|  | SetBerat() | - |  |  |
|  | GetKoordinat() | Mengecek apakah koordinat sudah sesuai keadaan ketika dihidupkan | OK |  |
|  | SetKoordinat() | - |  |  |
|  | IsLandAnimal() | Mengecek apakah hewan sesuai dengan habitatnya | OK |  |
|  | IsWaterAnimal() | Mengecek apakah hewan sesuai dengan habitatnya | OK |  |
|  | IsAirAnimal() | Mengecek apakah hewan sesuai dengan habitatnya | OK |  |
|  | IsJinak() | Mengecek apakah hewan Jinak | OK |  |
|  | IsLivable() | - |  |  |
|  | GetMakanan() | Mendapatkan jumlah makanan hewan setiap hari | OK |  |
| WaterHabitat | WaterHabitat(Indices I) |  |  |  |
|  | Render() | Mengecek apakah Render sudah sesuai Jenis Habitat | OK |  |
|  | Habitat(Indices I, int type, char code) |  |  |  |
|  | IsLand() | Mengecek apakah Cell Merupakan daratan? | OK |  |
|  | IsWater() | Mengecek apakah Cell Merupakan perairan? | OK |  |
|  | IsAir() | Mengecek apakah Cell Merupakan udara? | OK |  |
|  | Cell (Indices I, int type, char code) |  |  |  |
|  | ~Cell() |  |  |  |
|  | GetKoordinat() | Mengecek apakah koordinat sudah sesuai keadaan ketika dihidupkan | OK |  |
|  | IsHabitat() | Mengecek jenis Cell | OK |  |
|  | IsFacility() | Mengecek jenis Cell | OK |  |
|  | GetCode() | Mengecek kode Cell | OK |  |
| Zoo | Zoo() |  |  |  |
|  | ~Zoo() |  |  |  |
|  | Move() |  |  |  |
|  | Print() |  |  |  |
|  | HitungMakanan() |  |  |  |
|  | IsInteractable(Indices I, Cage C) | Mengecek apakah di posisi terdapat kandang dan dapat berinteraksi | OK |  |
|  | GetKandang() |  |  |  |

* 1. **Perhitungan Metriks Perangkat Lunak**

Versi: VZ03 dengan generik/inheritance

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Metriks** | **Besarnya** | **Keterangan** |
| 1 | Number of packages |  | Package adalah sub direktori |
| 2 | Number of Classes |  | Jumlah kelas rill |
| 3 | Number of Abstact class |  | Jumlah ABC |
| 4 | Afferent Couplings (Ca) |  | The number of classes in other packages that depend upon classes within the package is an indicator of the package's responsibility. Afferent = incoming |
| 5 | Efferent Coupling (Ce) |  | The number of classes in other packages that the classes in the package depend upon is an indicator of the package's dependence on externalities. Efferent = outgoing |
| 6 | Asbtractness (A) |  | Rasio jumlah kelas abstrak dibanding kelas riil |
| 7 | Instability (I) |  | The ratio of efferent coupling (Ce) to total coupling (Ce + Ca) such that I = Ce / (Ce + Ca). This metric is an indicator of the package's resilience to change. The range for this metric is 0 to 1, with I=0 indicating a completely stable package and I=1 indicating a completely unstable package. |
| 8 | Package Dependency Cycle |  | Adakah dependency Cycle? Artinya paket yang saling tergantung ? seharusnya tidak boleh ada karena tidak bisa dibuilt |
| 9 | Kelas Generik |  | Sebutkan nama kelas generik |

* 1. **Log Activity**

**Pembagian Peran**

(Deskripsi Umum)

Berikut dilampirkan tabel sebagai bentuk deskripsi rinci tanggung jawab pengerjaan tiap unit kelas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelas** | **File** | **Designer** | **Implementor/Koder** | **Tester** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Selain itu juga dilampirkan tabel pembagian tugas untuk dekomposisi dokumen sebagai berikut.

|  |  |  |
| --- | --- | --- |
| **Elemen Dokumentasi** | **Writer** | **Reviewer** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**RIncian Kegiatan**

Berikut perincian kegiatan tiap individu anggota

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Dari Tgl.. Pk…** | **S.d Tgl.. Pk..** | **Kegiatan** | **Hasil** | **Keterangan** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Total waktu yang dibutuhkan untuk mengerjakan tugas adalah…

(Rangkuman Capaian)