**Tugas Besar – Virtual Zoo**

**IF2210 Pemrograman Berorientasi Objek**

**Versi VZ03**

Dibuat Oleh:

**Faiz Ghifari Haznitrama** 13515010

**Abdurrahman** 13515024

**Dery Rahman Ahaddienata** 13515097

**Irfan Ariq** 13515112

****

**Program Studi Teknik Informatika**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

1. **Deskripsi Umum Aplikasi**

Aplikasi merupakan *simulator* kebun binatang sederhana. Seperti kebun binatang pada umumnya, terdapat petak untuk pengunjung atau binatang.

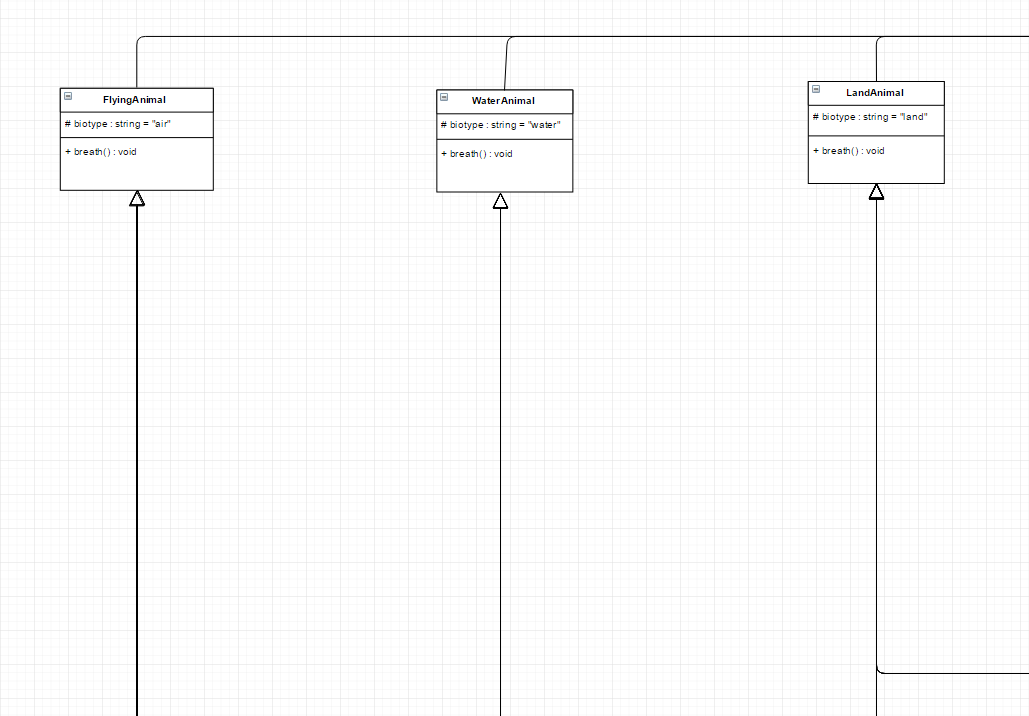
* 1. ***Animal***

Binatang yang terdapat pada kebun binatang memiliki 3 kategori umum yaitu binatang yang hidup di darat, binatang yang hidup di air, dan binatang yang hidup di udara. Pengelompokan binatang secara detail dituliskan pada bab rancangan kelas. Setiap binatang memiliki fungsi *interact* yang merupakan ekspresi yang dapat diamati dari binatang tersebut.

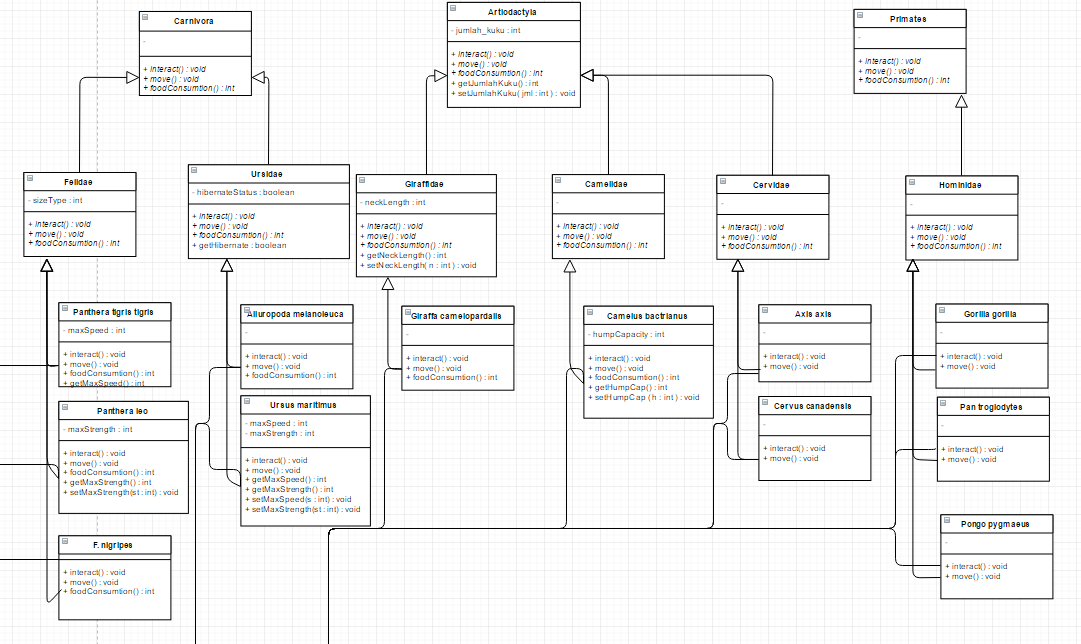
* 1. **Petak Kebun Binatang**

Kebun binatang terdiri dari matriks *cell* yang dibagi menjadi 2 yaitu *habitat* dan *facility.* Habitat merupakan tempat untuk binatang yang dibatasi dengan *cage*. Habitat secara khusus didesain untuk binatang tertentu sesuai kategori umum, sehingga terdapat *land habitat, water habitat,* dan *air habitat*. Fasilitas adalah petak di mana pengunjung bisa berjalan atau diam untuk mengamati kandang. Fasilitas yang disediakan kebun binatang adalah *Restaurant, Park,* dan *Road.* Pada sisi-sisi kebun binatang terdapat jalan khusus yang merupakan *entrance* dan *exit* untuk pengunjung.

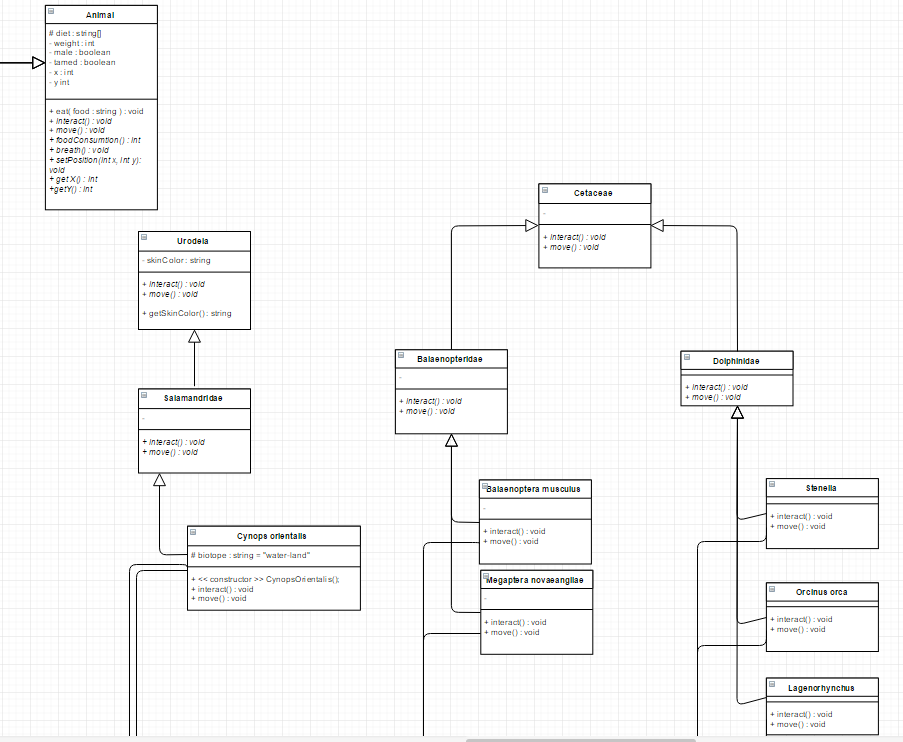
1. **Rancangan Kelas**
   1. **Rancangan-1**
      1. ***Animal***

****

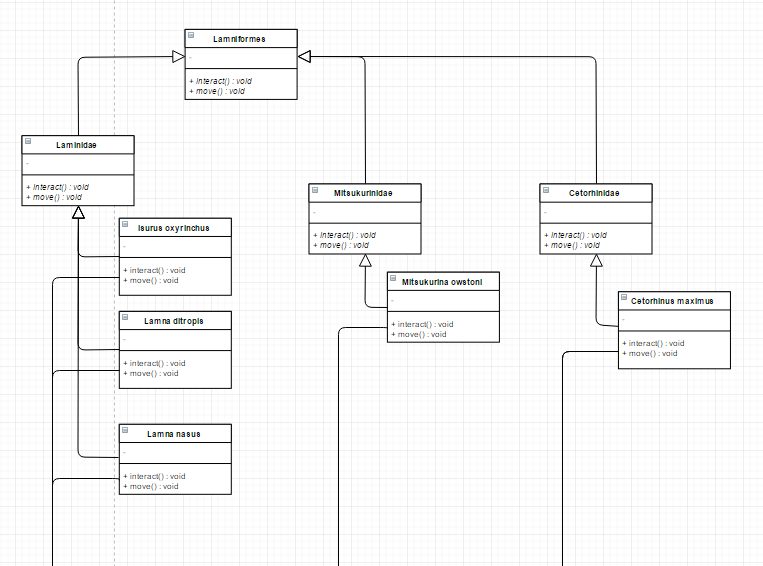
**Gambar 2.1.1 Struktur Inheritance Kelas Animal (Potongan 1: Turunan 1)**

****

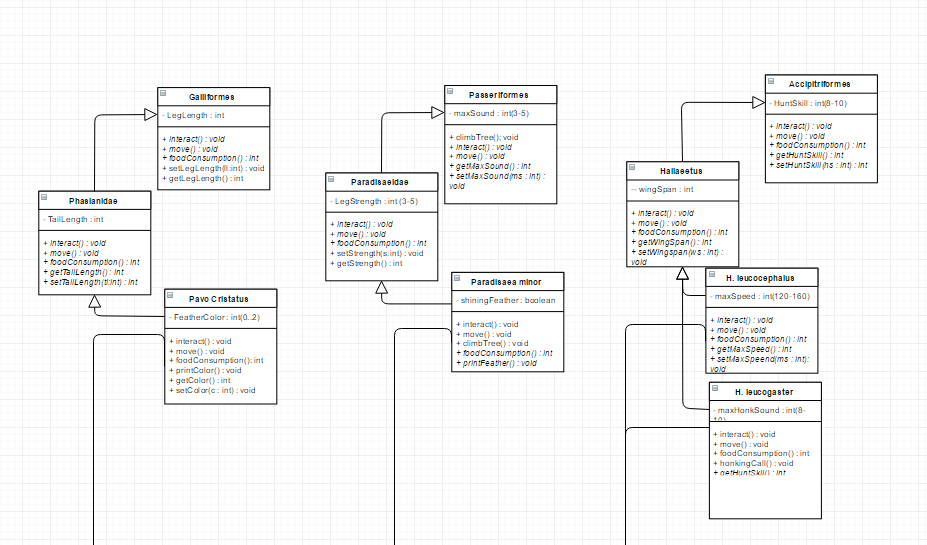
**Gambar 2.1.2 Struktur Inheritance Kelas Animal (Potongan 2: Turunan LandAnimal)**

****

**Gambar 2.1.3 Struktur Inheritance Kelas Animal (Potongan 3: Turunan WaterAnimal dan Animal)**

****

**Gambar 2.1.4 Struktur Inheritance Kelas Animal (Potongan 4: Turunan WaterAnimal)**

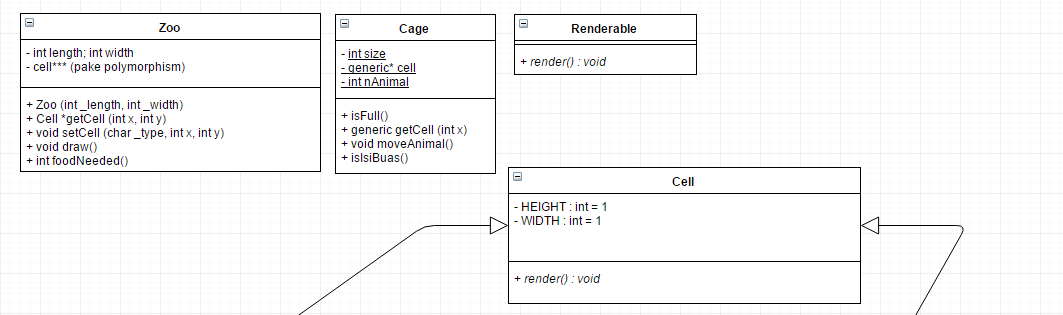
****

**Gambar 2.1.5 Struktur Inheritance Kelas Animal (Potongan 5: Turunan FlyingAnimal)**

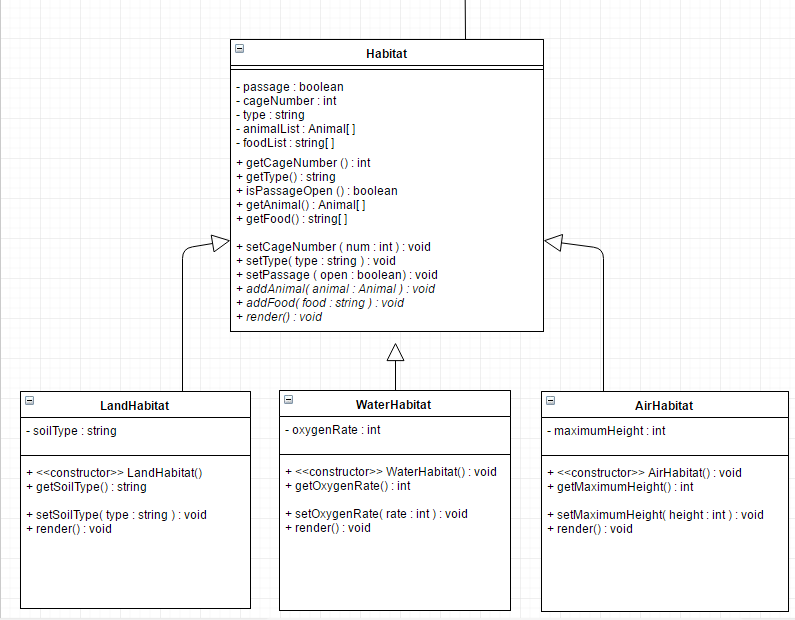
Desain awal digunakan struktur *inheritance* biasa dengan referensi pengelompokan *family* dan *genus* dilanjutkan dengan kelas riil spesies. Di rancangan 1 ini kami sudah sampai melakukan pendefinisian memberdan method untuk setiap kelas, baik itu kelas riil maupun kelas abstrak (dapat dilihat di 5 gambar diatas). Setiap kelas mempunyai member yang menjadi ciri khas dari kelas tersebut serta beberapa method yang berhubungan dengan member tersebut.

Tetapi yang menjadi masalah pada rancangan 1 Animal ini adalah diagram kelas yang dibuat sangatlah kompleks. Mulai dari ukuran yang diagram yang sangat besar, serta jumlah method dan member yang sangat banyak dari tiap kelas. Hal inilah yang sangat berbeda dengan rancangan 2. Pada akhirnya ketika sudah mulai mendesain header masing-masing setiap kelas, kami merasa bahwa desain yang sudah dibuat kurang bagus dan akan sangat memakan waktu untuk mengimplementasikan seluruh hal yang telah didesain pada diagram kelas ini. Maka kami memutuskan untuk merubah desain diagram kelas menjadi rancangan 2 yang selanjutnya kami gunakan dan implementasikan.

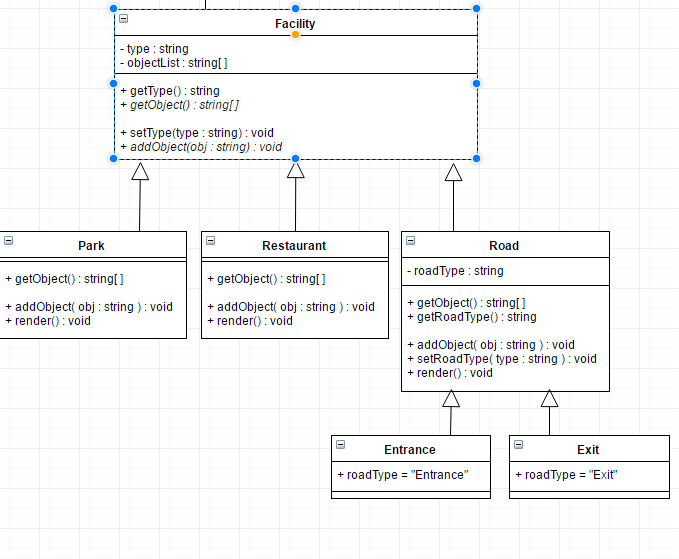
* + 1. **Petak Kebun Binatang**

****

**Gambar 2.1.6 Detail Kelas Zoo, Renderable, Cage, dan Cell**

****

**Gambar 2.1.7 Detail Struktur Kelas Habitat**

****

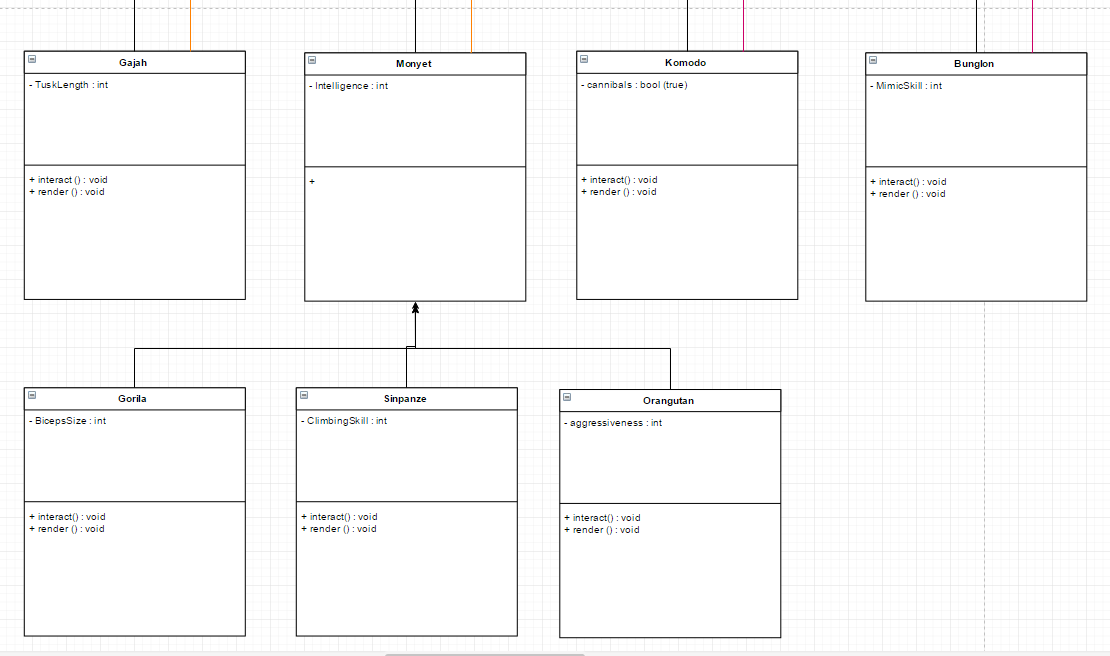
**Gambar 2.1.8 Detail Struktur Kelas Facility**

Kelas Zoo adalah kelas independen yang mempunyai attribute seperti definisinya yaitu panjang, lebar, dan matriks of pointer to Cell. Terdapat 1 kelas independen lainnya dengan nama Renderable yang mempunyai 2 turunan kelas yaitu Cage dan Cell. Renderable adalah *abstract base class* yang mempunyai pure method virtual render(). Tiap kelas Cage menyimpan atribut kelas lainnya yaitu Cell sehingga pengaksesan Cell di kelas tersebut pada kebun binatang virtual ini melalui cage. Cell sendiri merupakan kelas yang menurunkan beberapa kelas menurut peran dan fungsinya dalam kebun binatang. Peran cell yang pertama adalah petak untuk lingkungan tempat tinggal binatang yang dibagi secara umum ke 3 kelas yaitu Land Habitat, Water Habitat, dan Air Habitat. Sedangkan peran cell yang kedua sebagai petak fasilitas untuk pengunjung berupa Restaurant, Park, dan Road. Pada objek Road pengunjung dapat menelusuri kebun binatang, dimulai dari objek Road bertipe Entrance hingga objek Road bertipe Exit.

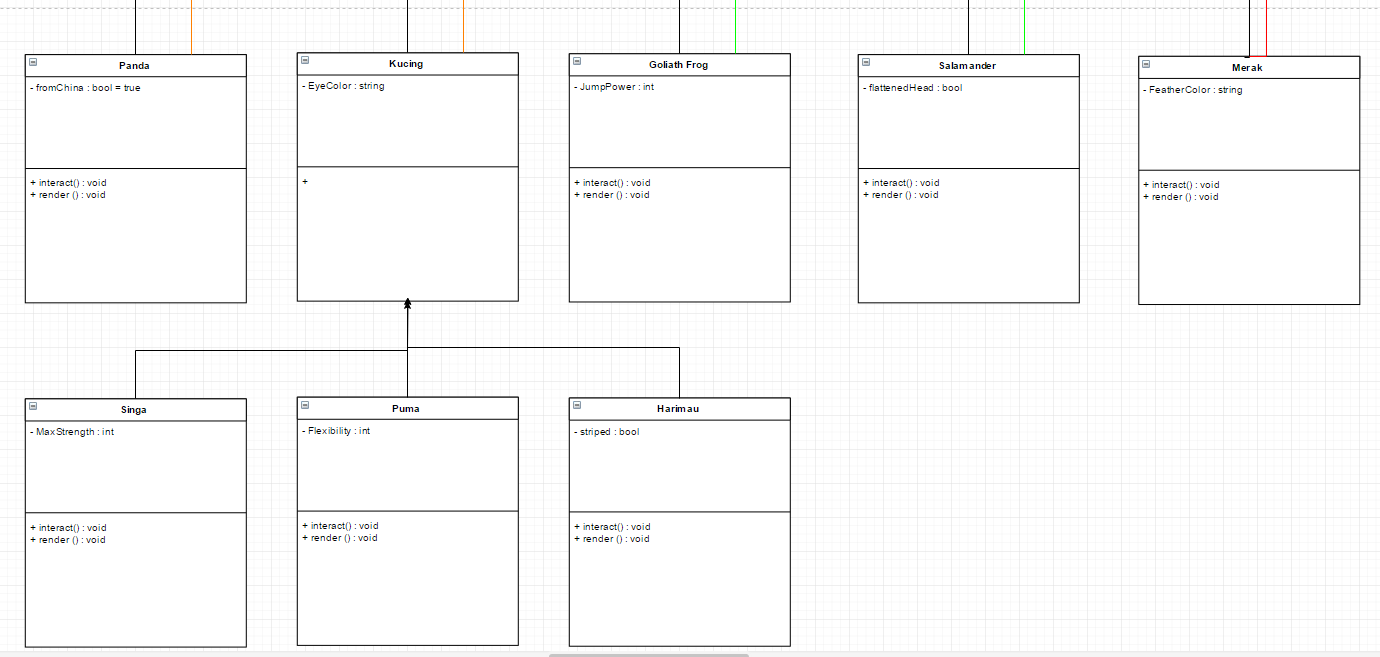
Rancangan petak kebun binatang pada versi VZ03 rancangan 1 ini tidak banyak berubah dari rancangan petak kebun binatang pada versi VZ01. Hal ini dikarenakan perubahan dari VZ01 ke VZ03 tidak terlalu banyak pada aspek petak kebun binatang, tetapi lebih banyak di Animal. Sehingga di petak kebun binatang sendiri tidak terjadi perubahan yang signifikan. Perubahan hanya terjadi di bagian Zoo dan Cage yaitu beberapa penambahan dan pengurangan member dan method untuk penyesuaian dengan *change request* yang ada.

Perbedaan dengan rancangan 2 yang kami gunakan dan implementasikan terletak pada member dan method dari setiap kelas. Secara umum, diagram yang digunakan antara rancangan 1 dan 2 masih sama. Tetapi setelah kami mencoba mendesain header dan sedikit mengimplementasikan beberapa bagian, kami mendapati bahwa dengan member dan method seperti ini Virtual Zoo tidak dapat dibuat dengan baik. Keterkaitan antar kelas pada rancangan 1 ini kurang baik dan menyebabkan pembuatan Virtual Zoo menjadi tidak optimal. Terlihat mulai dari Zoo yang memiliki member matriks of pointer to Cell sebagai wujud dari desain Zoo secara keseluruhan, tetapi Cell disini malah hanya memiliki method render serta ukuran panjang x lebar sebesar 1x1. Sangat jelas bahwa *polymorphism* yang kami rencanakan tidak akan berjalan dengan baik disini. Hal ini belum ditambah keterhubungan dengan kelas Animal. Maka dari itu, karena desain kelas Animal pun memiliki beberapa masalah, maka rancangan 1 ini tidak kami lanjutkan dan beralih ke rancangan 2.

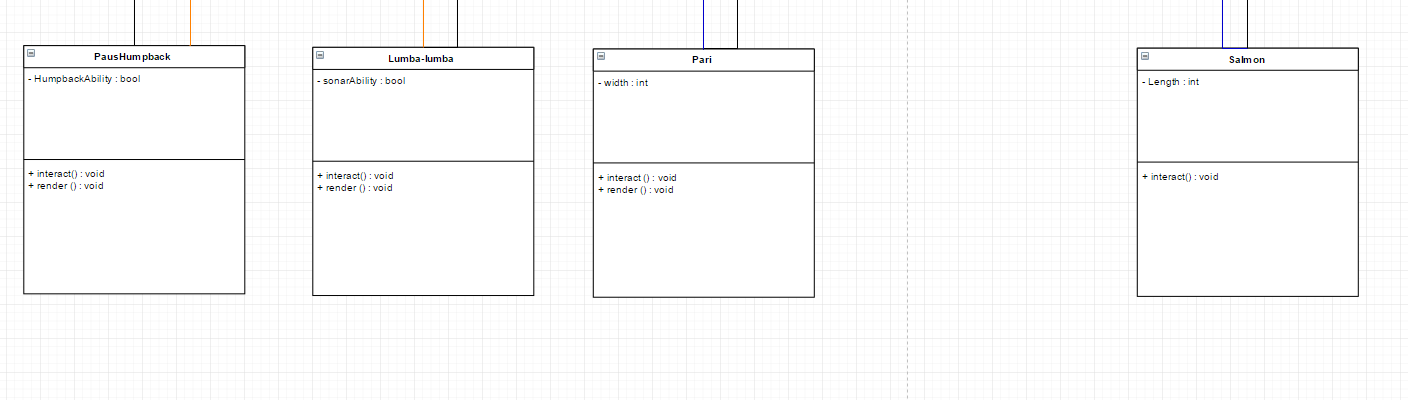
* 1. **Rancangan-2**
     1. ***Animal***

****

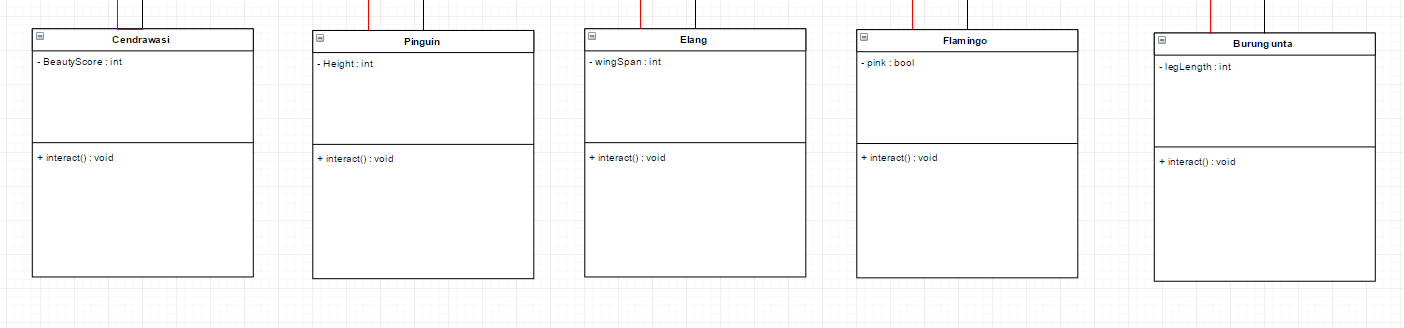
**Gambar 2.2.1 Struktur Inheritance Kelas Animal dan Turunannya (Potongan 1)**

****

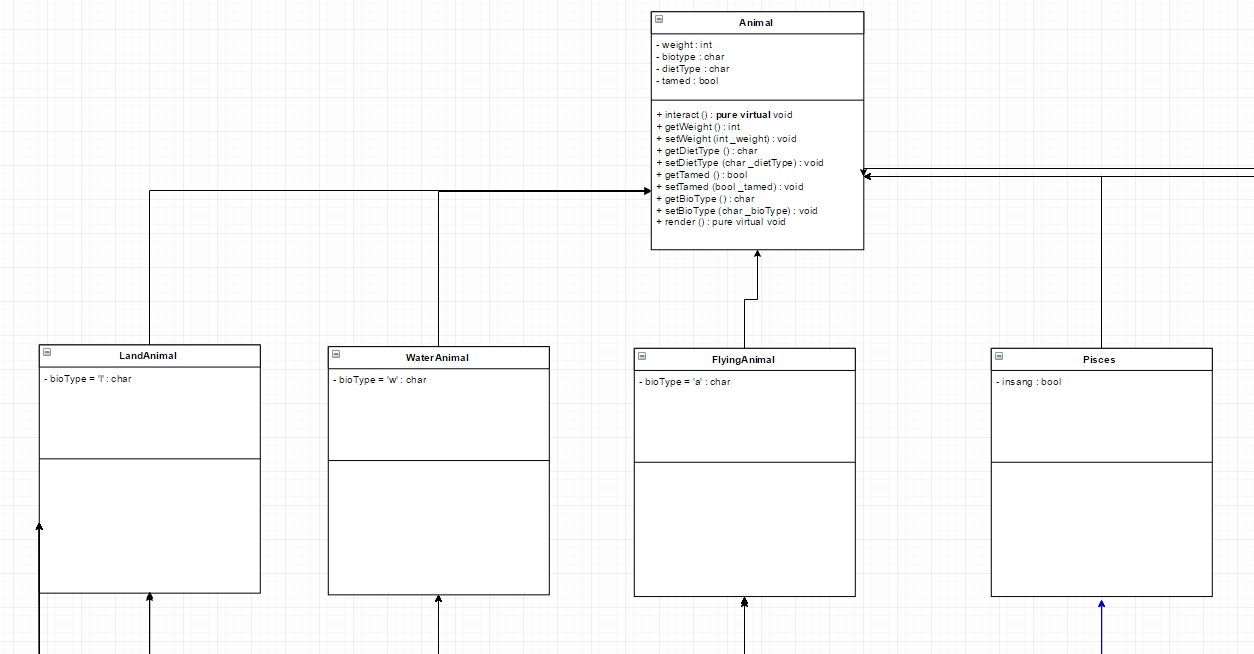
**Gambar 2.2.2 Struktur Inheritance Kelas Animal dan Turunannya (Potongan 2)**

****

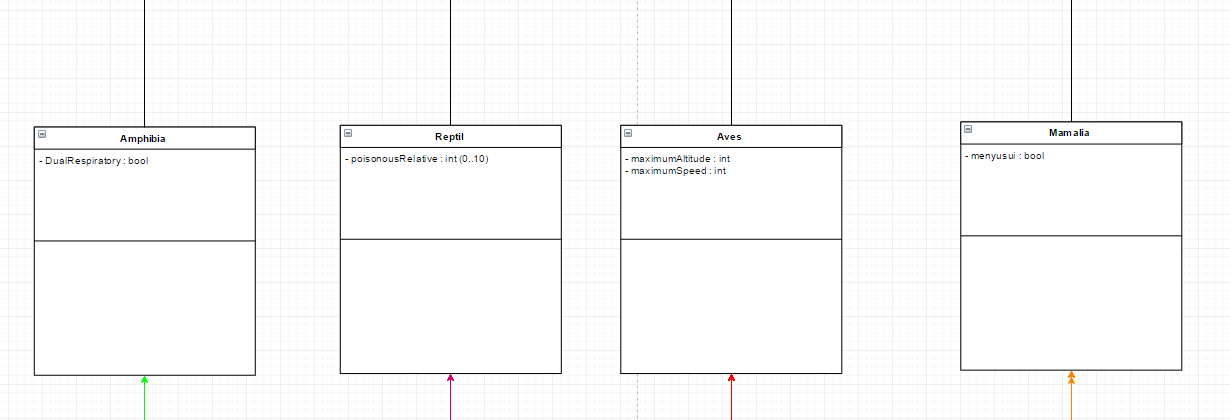
**Gambar 2.2.3 Struktur Inheritance Kelas Animal dan Turunannya (Potongan 3)**

****

**Gambar 2.2.4 Struktur Inheritance Kelas Animal dan Turunannya (Potongan 4)**

****

**Gambar 2.2.5 Struktur Inheritance Kelas Animal dan Turunannya (Potongan 5)**

****

**Gambar 2.2.6 Struktur Inheritance Kelas Animal dan Turunannya (Potongan 6)**

Pada rancangan 2 ini kami melakukan evaluasi berdasarkan hal-hal yang menjadi kekurangan pada rancagan 1. Kami melakukan penyederhanaan yang cukup signifikan pada rancangan 1 untuk kemudian tetap diberikan member dan method yang menjadi ciri khas masing-masing kelas serta melakukan definisi dan implementasi dari member dan method umum (dimiliki oleh seluruh Animal) di *abstract base class* Animal. Suatu hal yang cukup berbeda dari rancangan sebelumnya.

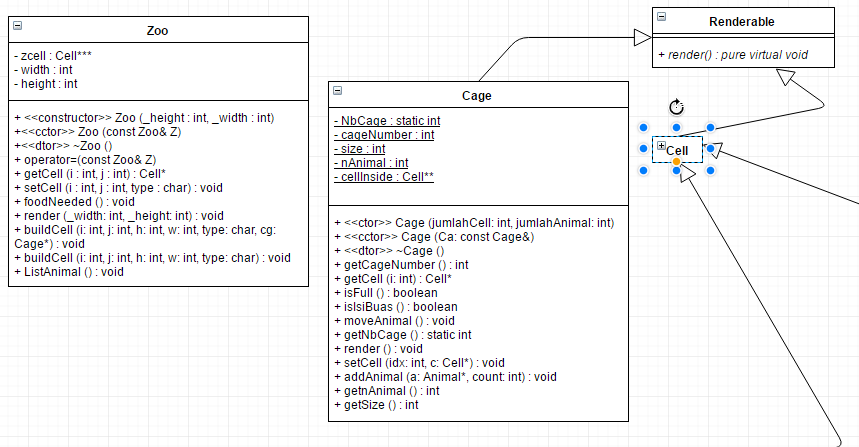
Kebutuhan akan definisi dan implementasi di kelas Animal dikarenakan kami bermaksud menggunakan fitur *polymorphism* yang ada pada C++. Hal ini erat kaitannya dengan kelas Cell, yang pada rancangan 2 ini diubah menjadi bentuk pointer to Animal. Sehingga ketika Virtual Zoo dibuat, Cell yang merupakan pointer to Animal dapat menunjuk ke seluruh kelas turunan dari Animal dengan adanya efek *polymorphism*. Fitur ini yang menjadi andalan kami dalam rancangan 2 ini.

Penyederhanaan desain diagram kelas dari rancangan 1 ke rancangan 2 kami lakukan berdasarkan fungsionalitas dari setiap member dan method yang telah dibuat di rancangan 1. Kami melakukan seleksi mana saja member dan method yang sebenarnya benar-benar dibutuhkan keberadaannya pada Virtual Zoo ini. Kami mengurangi member dan method yang tidak penting dan menghapus beberapa kelas abstrak yang berada di antara kelas riil dengan kelas Animal. Selain juga agar mudah untuk diimplementasikan, diagram kelas yang lebih sederhana pada rancangan 2 ini memungkinkan orang lain untuk lebih mudah memahami cara kerja aplikasi Virtual Zoo secara structural.

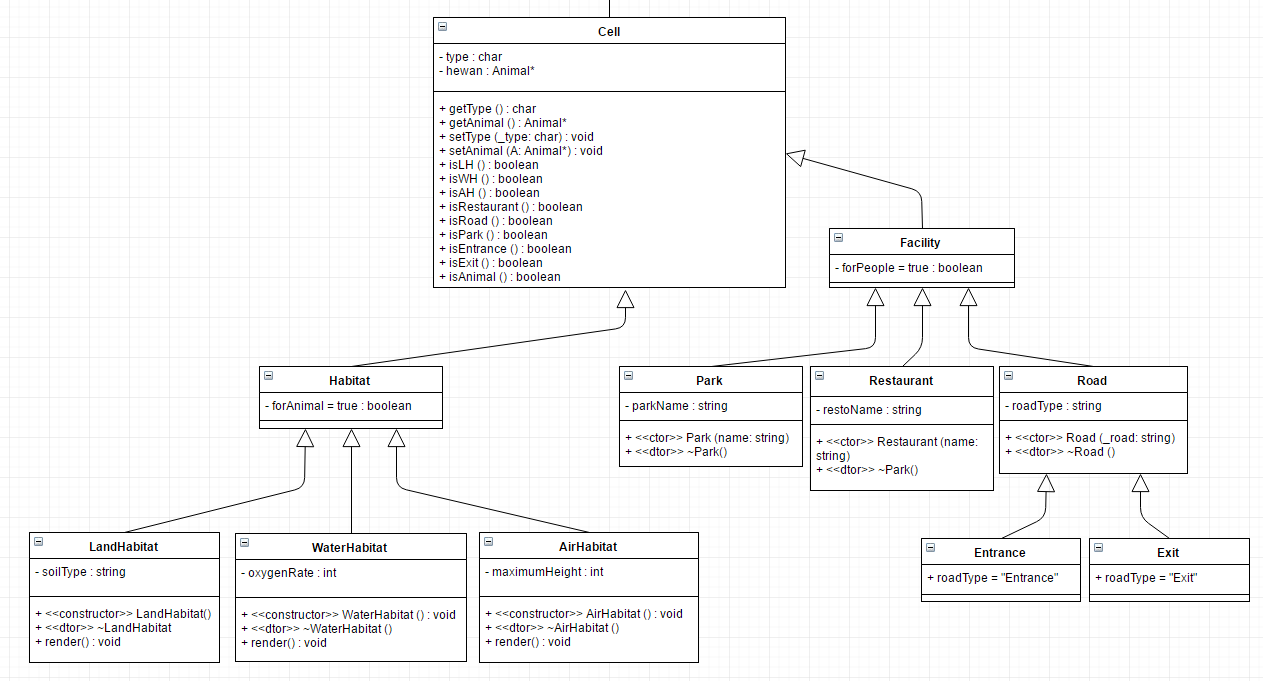
Berikut merupakan daftar kelas yang termasuk pada Animal:

|  |  |  |
| --- | --- | --- |
| **Nama Kelas** | **Gambaran Member** | **Gambaran Method** |
| Animal | Seluruh atribut yang menjadi informasi seekor Animal | Seluruh method terkait dengan atribut Animal dan keperluan penurunan setiap method ke seluruh turunannya |
| LandAnimal | Menunjukkan ciri khas hewan pada ordo tersebut | Getter member tersebut |
| WaterAnimal | Menunjukkan ciri khas hewan pada ordo tersebut | Getter member tersebut |
| FlyingAnimal | Menunjukkan ciri khas hewan pada ordo tersebut | Getter member tersebut |
| Aves | Menunjukkan ciri khas hewan pada ordo tersebut | Getter member tersebut |
| **Nama Kelas** | **Gambaran Member** | **Gambaran Method** |
| Mamalia | Menunjukkan ciri khas hewan pada ordo tersebut | Getter member tersebut |
| Pisces | Menunjukkan ciri khas hewan pada ordo tersebut | Getter member tersebut |
| Reptil | Menunjukkan ciri khas hewan pada ordo tersebut | Getter member tersebut |
| Amphibia | Menunjukkan ciri khas hewan pada ordo tersebut | Getter member tersebut |
| Kucing | Menunjukkan ciri khas hewan pada kelas tersebut | Getter member tersebut |
| Monyet | Menunjukkan ciri khas hewan pada kelas tersebut | Getter member tersebut |
| Gajah | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Simpanze | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Orangutan | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Gorila | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Komodo | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Bunglon | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Panda | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Puma | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Harimau | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Singa | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| GoliathFrog | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| **Nama Kelas** | **Gambaran Member** | **Gambaran Method** |
| Salamander | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Merak | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| PausHumpback | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| LumbaLumba | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Salmon | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Cendrawasi | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Pinguin | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Elang | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| Flamingo | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |
| BurungUnta | Menunjukkan ciri khas hewan tersebut | Getter member tersebut dan sifatnya ketika ditampilkan ke layar serta interaksi saat dikunjungi |

* + 1. **Petak Kebun Binatang**

****

**Gambar 2.1.6 Detail Kelas Zoo, Renderable, dan Cage**

****

**Gambar 2.1.7 Detail Struktur Kelas Cell dan turunannya**

Kelas Zoo adalah kelas independen yang mempunyai attribute seperti definisinya yaitu panjang, lebar, dan matriks of pointer to Cell. Terdapat 1 kelas independen lainnya dengan nama Renderable yang mempunyai 2 turunan kelas yaitu Cage dan Cell. Renderable adalah *abstract base class* yang mempunyai pure method virtual render(). Tiap kelas Cage menyimpan atribut kelas lainnya yaitu Cell sehingga pengaksesan Cell di kelas tersebut pada kebun binatang virtual ini melalui Cage. Cell sendiri merupakan kelas yang menurunkan beberapa kelas menurut peran dan fungsinya dalam kebun binatang. Peran cell yang pertama adalah petak untuk lingkungan tempat tinggal binatang yang dibagi secara umum ke 3 kelas yaitu Land Habitat, Water Habitat, dan Air Habitat. Sedangkan peran cell yang kedua sebagai petak fasilitas untuk pengunjung berupa Restaurant, Park, dan Road. Pada objek Road pengunjung dapat menelusuri kebun binatang, dimulai dari objek Road bertipe Entrance hingga objek Road bertipe Exit.

Rancangan petak kebun binatang pada versi VZ03 rancangan 2 ini masih sama, tidak banyak berubah baik dari versi VZ01 ataupun VZ03 rancangan 1. Tapi yang menjadi perhatian adalah perubahan pada member dan method di masing-masing kelas. Disini terjadi perubahan fungsionalitas kelas secara signifikan dikarenakan penggunaan fitur *polymorphism* pada Cell.

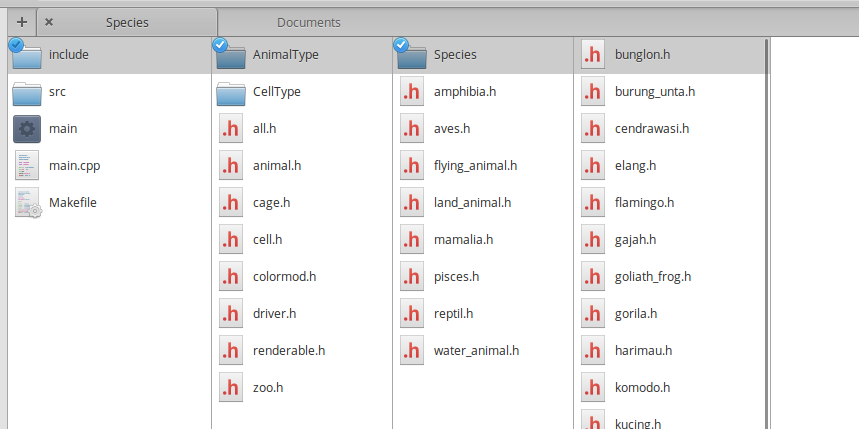
Cell ditambah member berupa pointer to Animal. Zoo memiliki matriks of pointer to Cell. Dalam rancangan 2 ini, untuk mengimplementasikan Zoo dengan adanya member matriks of pointer to Cell menjadi cukup mudah. Karena pointer to Cell dapat melakukan *polymorphism* terhadap seluruh turunan dari Cell. Ditambah lagi dengan member pointer to Animal dari Cell itu sendiri, yang artinya pointer to Cell pada Zoo bisa berperan sebagai Animal juga. Sehingga disini matriks of Cell pada Zoo menjadi sangat fleksibel dengan adanya fitur *polymorphism* untuk berperan sebagai kelas Cell atau Animal dan turunannya. Kekurangan dari penggunaan fitur ini adalah tingkat dependensi antar kelas, terutama Zoo dengan kelas lainnya menjadi cukup besar, meskipun pada kelas lain utamanya Cell dan Animal tingkat dependensinya sudah cukup baik. Tingkat dependensi antara kelas turunan dan kelas induk pun menjadi cukup besar dikarenakan untuk mewujudkan efek *polymorphism* kebanyakan method dan member didefinisikan dan diimplementasikan langsung di kelas induk. Sedangkan di kelas turunan hanya berisi member dan method yang menjadi ciri khas kelas tersebut.

Untuk kelas Cage sendiri tidak memiliki banyak perubahan dari VZ01 dan VZ03 rancangan 1. Cage tetap sesuai dengan fungsionalitasnya sebagai Cage yaitu sebagai pembatas antar Cell dengan menyimpan larik seluruh Cell yang berada di dalam Cage tersebut.

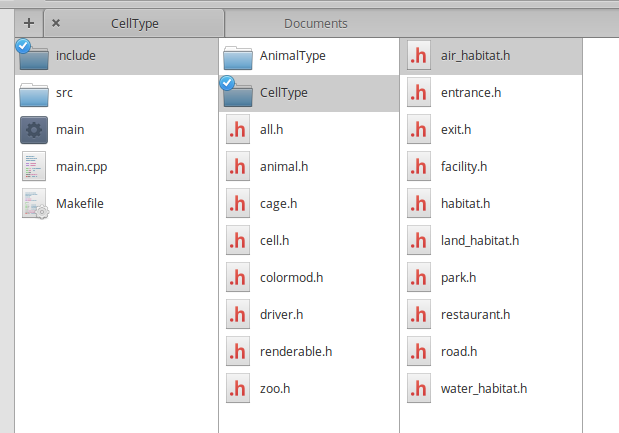
Berikut merupakan daftar kelas yang termasuk pada Petak Kebun Binatang:

|  |  |  |
| --- | --- | --- |
| **Nama Kelas** | **Gambaran Member** | **Gambaran Method** |
| Zoo | Berupa matriks of pointer to Cell | Berhubungan dengan penempatan Cell pada Zoo |
| Cage | Berupa Array of pointer to Cell | Berhubungan dengan pembatasan Animal di Cell tertentu |
| Cell | Berupa pointer to Animal | Berhubungan dengan sifat sebuah Cell dan juga penempatan Animal di Cell tersebut |
| Renderable | N/A | Berisi pure virtual render() |
| Habitat | Berupa keterangan apakah bisa dilewati buat animal | Kelas abstrak turunan dari Cell dan induk dari seluruh habitat |
| **Nama Kelas** | **Gambaran Member** | **Gambaran Method** |
| LandHabitat | Berupa tipe tanah (soil\_type) | Berhubungan dengan sifat LandHabitat dengan Animal atau ketika ditampilkan ke layar |
| WaterHabitat | Berupa kadar oksigen (oxygen\_rate) | Berhubungan dengan sifat WaterHabitat dengan Animal atau ketika ditampilkan ke layar |
| AirHabitat | Berupa kecepatan dan ketinggian maksimum hewan (max\_altitude dan max\_speed) | Berhubungan dengan sifat AirHabitat dengan Animal atau ketika ditampilkan ke layar |
| Facility | Berupa keterangan apakah bisa dilewati orang | Kelas abstrak turunan dari Cell dan induk dari seluruh fasilitas |
| Park | Nama dari taman tersebut (park\_name) | Berhubungan dengan sifat Park ketika ditampilkan ke layar |
| Restaurant | Nama dari restoran tersebut (resto\_name) | Berhubungan dengan sifat Restaurant ketika ditampilkan ke layar |
| Road | Jenis dari jalan tersebut, dapat berupa pintu masuk, pintu keluar, atau jalan biasa | Berhubungan dengan sifat road ketika ditampilkan ke layar |

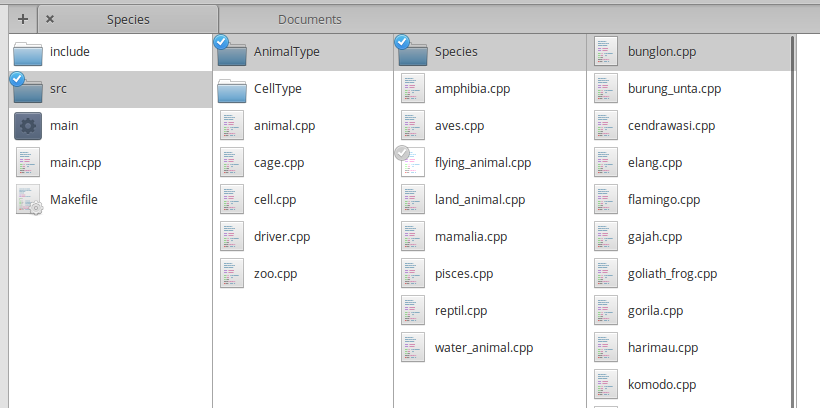
1. **Struktur Direktori**

****

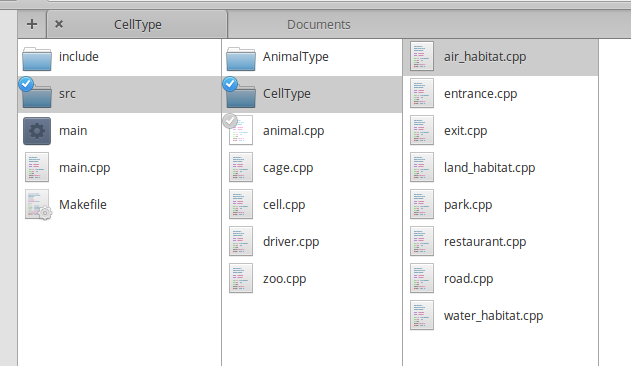
**Gambar 3.1 Direktori Root, include, AnimalType, dan Species**

****

**Gambar 3.2 Direktori Root, include, dan CellType**

****

**Gambar 3.3 Direktori Root, src, AnimalType, dan Species**

****

**Gambar 3.4 Direktori Root, src, dan CellType**

1. **Daftar Lampiran**
   1. **Skenario Tes**

Terlampir

* 1. **Penghitungan Metriks Perangkat Lunak**

Terlampir

* 1. ***Log Activity***

Terlampir

* 1. **Diagram Kelas**

Terlampir

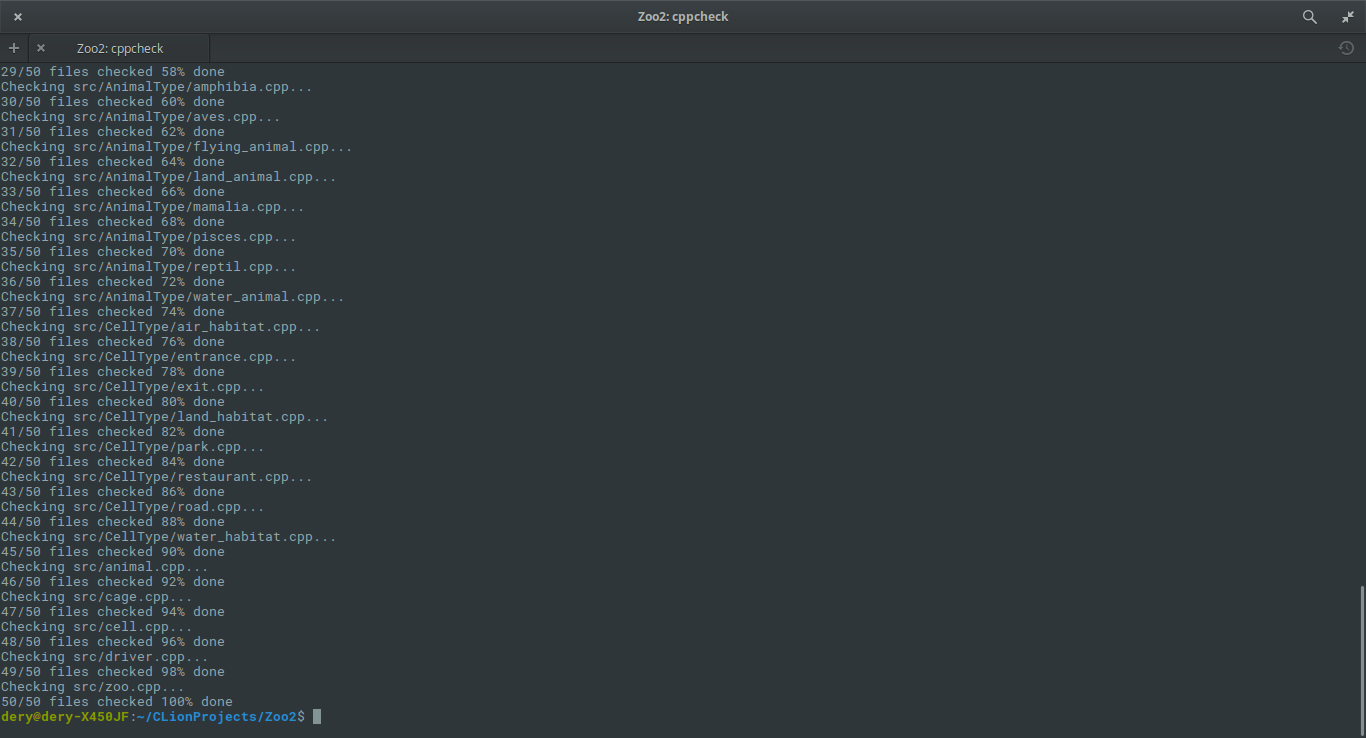
* 1. **Hasil Dokumentasi dengan *Doxygen***

Terlampir

* 1. **Hasil Unit Tes dengan *GoogleTest***

Terlampir

* 1. **Hasil *Static Code* dengan *CppCheck***



**Gambar 4.7.1 Snapshot Hasil Static Code dengan *CppCheck***