Laporan Tugas Besar 2 IF2210

Pemrograman Berorientasi Objek

**Dokumentasi Perangkat Lunak**

**Virtual Zoo Versi VZ02 dan Versi VZ03**



disusun oleh:

**Kelompok :** **The Great Zoo**

M. Irfan Maulana 13515037

Patrick Nugroho H. 13515040

Marvin Jerremy B. 13515076

Martin Lutta Putra 13515121

**Program Studi Teknik Informatika**

**Institut Teknologi Bandung**

**2017**

Kelompok : 1. M. Irfan Maulana (13515037)

2. Patrick Nugroho H. (13515040)

3. Marvin Jerremy B. (13515076)

4. Martin Lutta Putra (13515121)

Asisten Mentor : Feryandi Nurdiantoro (13513042)

Asisten Penilai :

# **1 Diagram Kelas**

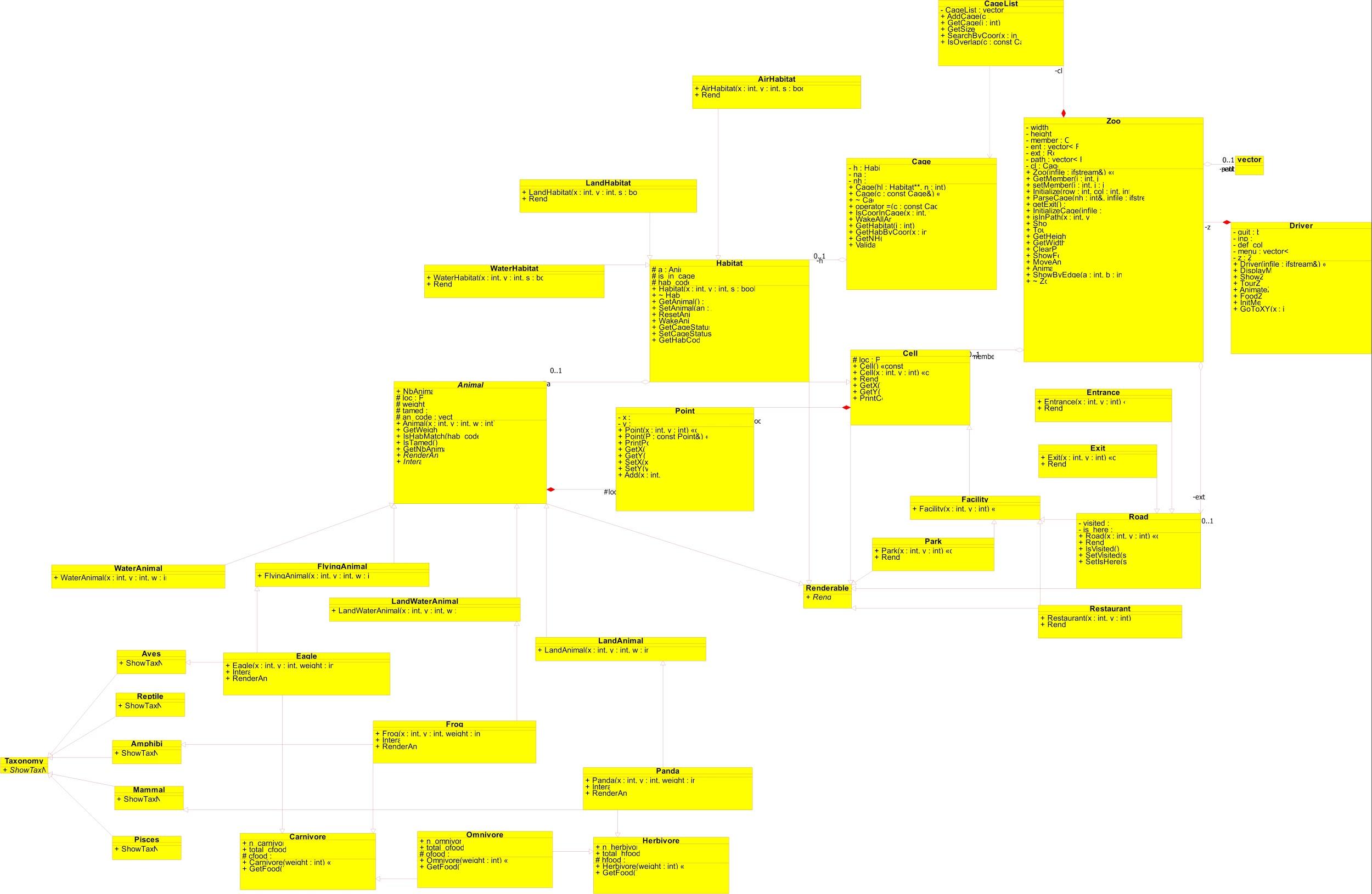
**1.1 Diagram Kelas CPP**

**1.1.1 Versi VZ02**



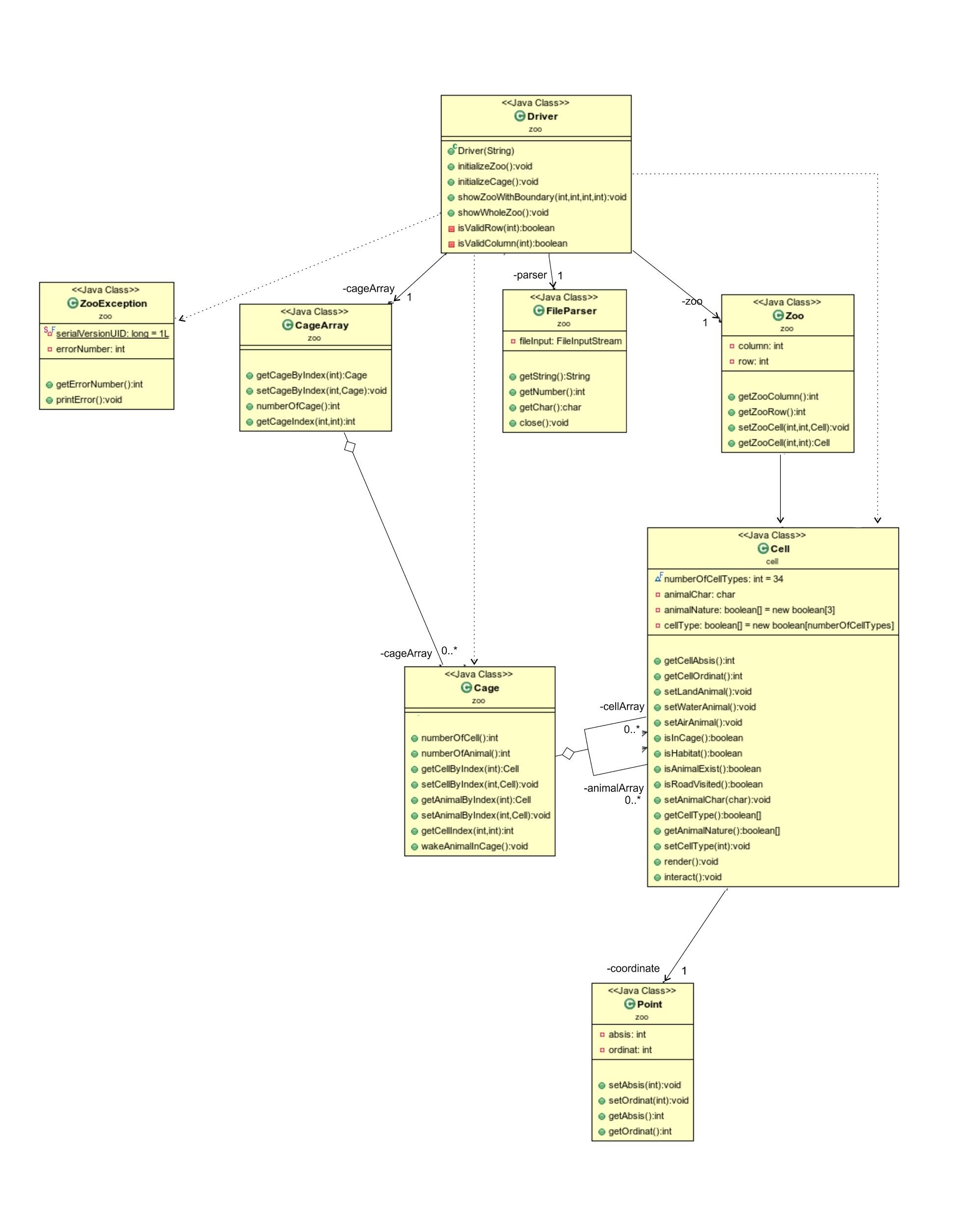
**1.1.2 Versi VZ03**

Diagram lengkap berada di lampiran berukuran kertas A3, berikut *thumbnails*nya :



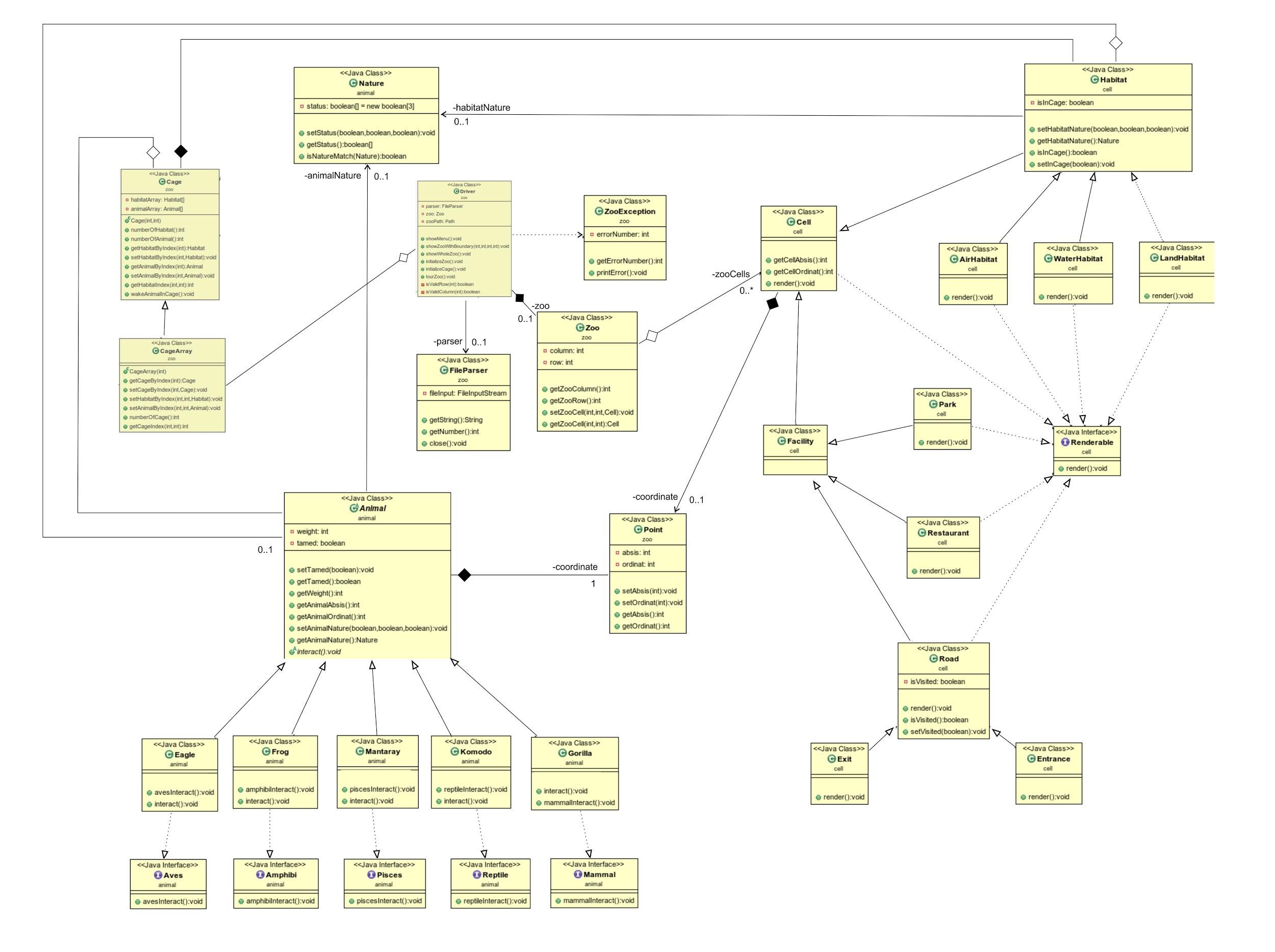
**1.2 Diagram Kelas JAVA (aplikasi baru)**

**1.2.1 Versi VZ02**



**1.2.2 Versi VZ03**

Diagram lengkap berada di lampiran berukuran kertas A3, berikut *thumbnails*nya :



## **1.3 Ulasan Redesign**

**1.3.1 Versi VZ02**

Pada versi CPP, setiap hewan yang ada dalam Zoo dilambangkan dengan sebuah karakter unik. Pergerakan animal pada peta digambarkan lewat perubahan posisi karater-karakter tersebut setiap satuan waktu tertentu. Pada versi Java, pergerakan hewan tidak digambarkan lewat perubahan posisi karakter. Perubahan posisi hewan dinyatakan dengan menampilkan posisi (absis, ordinat) hewan ke layar dalam bentuk string yang terus berubah setiap satuan waktu. Sehingga, kelas Cell yang pada versi CPP memiliki atribut moving untuk menyatakan hewan sedang bergerak, kini tidak lagi memiliki atribut tersebut. Selain itu, untuk menampilkan interaksi hewan, pada versi CPP digunakan method WakeAnimal(). Di versi Java, method tersebut diganti namanya menjadi interact(), serta dilakukan perubahan pada kalimat interaksi yang diucapkan hewan.

**1.3.2 Versi VZ03**

Pada diagram kelas CPP, *real class* Animal, misalnya kelas Chimpanzee mewarisi kelas LandAnimal, Omnivore dan Mammal. Kelas Mammal, Aves, Pisces, Reptile dan Amphibi mewarisi kelas Taxonomy. Kelas Omnivore mewarisi kelas Carnivore dan Herbivore. Pengaturan makanan pada hewan ada pada kelas Herbivore dan Carnivore dengan adanya method n\_herbivore(), total\_hfood(), dan hfood().

Sedangkan pada diagram kelas Java, terdapat *package* animal, cell dan zoo. *Package* animal berisi *class*-*class* atau *interface-interface* yang berhubungan dengan hewan. *Package* cell berisi *class*-*class* atau *interface-interface* yang berhubungan dengan cell. *Package* zoo berisi *class*-*class* atau interface-interface yang berhubungan dengan zoo. Untuk *real* *class* Animal seperti yang disebutkan di atas mewarisi kelas Animal dan mengimplementasi salah satu dari *interface* Mammal, Aves, Pisces, Reptile dan Amphibi. Di dalam kelas Animal terdapat *member* dari kelas Nature dan kelas Point. Kelas Taxonomy dihapus karena perannya sudah digantikan dengan *interface* Mammal, Aves, Pisces, Reptile dan Amphibi.

Untuk menentukan jenis makanan dari tiap *real* *class* animal digunakan kelas Consumption yang dapat mengatur banyaknya makanan dan mendapatkan jumlah makanan yang dibutuhkan oleh *animal*, baik itu *meat* (untuk karnivora) maupun *vegetables* (untuk herbivora) ataupun meat dan vegetables (untuk omnivora). Untuk mengatur habitat hewan digunakan *class* Nature. Di kelas Nature ini juga dapat mencocokkan *nature* dari Habitat dengan *nature* dari Animal.

# **2 Implementasi Perubahan**

Bagian ini akan memberikan gambaran berapa % (kira-kira) kode CPP anda pakai ulang.

Untuk setiap versi aplikasi JAVA yang anda kembangkan, buatlah Daftar Package dan Kelas aplikasi CPP yang menjadi dasar pengembangan, dan rencana penggunaan Ulang

**2.1 Versi VZ02**

| **Package** | **Nama Kelas CPP** | **Nama Method** | **Nama Kelas JAVA** | **Nama Method JAVA** | **Reuse**  **Ya/Tidak** | **Deskripsi Perubahan** |
| --- | --- | --- | --- | --- | --- | --- |
| cell | Cell | Cell(int x,int y) | Cell | public Cell(int absis,int ordinat) | Ya | - |
| cell | Cell | virtual void Render(); | Cell | public void render() | Ya | - |
| cell | Cell | int GetX() | Cell | public int getCellAbsis() | Ya | - |
| cell | Cell | Cell(int x,int y) | Cell | public Cell(int absis,int ordinat) | Ya | - |
| cell | Cell | virtual void Render(); | Cell | public void render() | Ya | - |
| cell | Cell | int GetX() | Cell | public int getCellAbsis() | Ya | - |
| zoo | Cage | Cage(Habitat \*\* hl,int nh) | Cage | public Cage(int numberOfHabitat,int numberOfAnimal) | Tidak | Pada C++ parameternya berupa array of habitat dan jumlah habitat, sedangkan pada Java parameternya adalah jumlah habitat dan jumlah hewan |
| zoo | Cage | int GetNH() const | Cage | public int numberOfHabitat() | Ya | Perubahan nama method |
| zoo | Cage | int GetHabByCoor(int x,int y) const | Cage | public Habitat getHabitatByIndex(int index) | Ya | - |
| zoo | Driver | Driver(ifstream& infile) | Driver | public Driver(String filePath) throws FileNotFoundException,IOException, ZooException | Ya | Pada C++ Zoo dibuat sebagai parameter, pada Java Zoo dibuat sebagai member dari kelas Driver |
| zoo | Driver | void DisplayMenu() | Driver | public void showMenu() | Ya | - |
| zoo | Driver | void ShowZoo() | Driver | public void showZooWithBoundary() throws ZooException | Ya | - |
| zoo | Driver | void TourZoo() | Driver | public void tourZoo() | Ya | - |
| zoo | Point | int GetX() const | Point | public int getAbsis() | Ya | Perubahan nama method |
| zoo | Point | int GetY() const | Point | public int getOrdinat() | Ya | Perubahan nama method |
| zoo | Point | void SetX() const | Point | public void setAbsis(int absis) | Ya | Perubahan nama method |
| zoo | Point | void SetY() const | Point | public void setOrdinat(int absis) | Ya | Perubahan nama method |
| zoo | Point | Point(int x, int y) | Point | public Point(int absis,int ordinat) | Ya | - |
| zoo | Zoo | Zoo(ifstream& infile) | Zoo | public Zoo(int column, int row) | Tidak | Pada C++ parameter berupa file eksternal, pada Java parameter berupa banyak baris dan kolom yang ingin dibuat |
| zoo | Zoo | int GetHeight() const | Zoo | public int getZooColumn() | Ya | Perubahan nama method |
| zoo | Zoo | int GetWidth() const | Zoo | public int getZooRow() | Ya | Perubahan nama method |

Kesimpulan Penggunaan Ulang :

Dengan adanya failitas *package* pada Java, pembagian dan pengurusan kelas dapat lebih fleksibel dan mudah.

**2.2 Versi VZ03**

| **Package** | **Nama Kelas CPP** | **Nama Method** | **Nama Kelas JAVA** | **Nama Method JAVA** | **Reuse**  **Ya/Tidak** | **Deskripsi Perubahan** |
| --- | --- | --- | --- | --- | --- | --- |
| animal | Animal | Animal(int x, int y, int w) | Animal | public Animal(int weight,int absis,int ordinat) | Ya | Perubahan nama parameter |
| animal | Animal | int GetWeight() const | Animal | public int getWeight() | Ya | - |
| animal | Animal | bool IsTamed() const | Animal | public boolean getTamed() | Ya | - |
| animal | Animal | virtual void Interact()=0 | Animal | public abstract void interact() | Ya | - |
| animal | Bat | Bat(int x, int y, int weight) | Bat | public Bat(int weight, int absis, int ordinat) | Ya | - |
| animal | Bat | void Interact() | Bat | public void interact() | Ya | - |
| animal | Cendrawasih | Cendrawasih(int x, int y, int weight) | Cendrawasih | public Cendrawasih(int weight, int absis, int ordinat) | Ya | - |
| animal | Cendrawasih | void Interact() | Cendrawasih | public void interact() | Ya | - |
| animal | Cheetah | Cheetah(int x, int y, int weight) | Cheetah | public Cheetah(int weight,int absis,int ordinat) | Ya | - |
| animal | Cheetah | void Interact() | Cheetah | public void interact() | Ya | - |
| animal | Chimpanzee | Chimpanzee(int x, int y, int weight) | Chimpanzee | public Chimpanzee(int weight,int absis,int ordinat) | Ya | - |
| animal | Chimpanzee | void Interact() | Chimpanzee | public void interact() | Ya | - |
| animal | Dolphin | Dolphin(int x, int y, int weight) | Dolphin | public Dolphin(int weight,int absis,int ordinat) | Ya | - |
| animal | Dolphin | void Interact() | Dolphin | public void interact() | Ya | - |
| animal | Eagle | Eagle(int x, int y, int weight) | Eagle | public Eagle(int weight, int absis, int ordinat) | Ya | - |
| animal | Eagle | void Interact() | Eagle | public void interact() | Ya | - |
| animal | Frog | Frog(int x, int y, int weight) | Frog | public Frog(int weight, int absis, int ordinat) | Ya | - |
| animal | Gorilla | Gorilla(int x, int y, int weight) | Gorilla | public Gorilla(int weight,int absis,int ordinat) | Ya | - |
| animal | Gorilla | void Interact() | Gorilla | public void interact() | Ya | - |
| animal | Hyena | Hyena(int x, int y, int weight) | Hyena | Hyena(int x, int y, int weight) | Ya | - |
| animal | Hyena | void Interact() | Hyena | public void interact() | Ya | - |
| animal | Kangaroo | Kangaroo(int x, int y, int weight) | Kangaroo | public Kangaroo(int weight,int absis,int ordinat) | Ya | - |
| animal | Kangaroo | void Interact() | Kangaroo | public void interact() | Ya | - |
| animal | Coala | Coala(int x, int y, int weight) | Koala | public Koala(int weight,int absis,int ordinat) | Ya | - |
| animal | Coala | void Interact() | Koala | public void interact() | Ya | - |
| animal | Kolibri | Kolibri(int x, int y, int weight) | Kolibri | public Kolibri(int weight, int absis, int ordinat) | Ya | - |
| animal | Kolibri | void Interact() | Kolibri | public void interact() | Ya | - |
| animal | Komodo | Komodo(int x, int y, int weight); | Komodo | public Komodo(int weight, int absis, int ordinat) | Ya | - |
| animal | Komodo | void Interact() | Komodo | public void interact() | Ya | - |
| animal | Lion | Lion(int x, int y, int weight) | Lion | public Lion(int weight,int absis,int ordinat) | Ya | - |
| animal | Lion | void Interact() | Lion | public void interact() | Ya | - |
| animal | Mantaray | Mantaray(int x, int y, int weight) | Mantaray | public Mantaray(int weight,int absis,int ordinat) | Ya | - |
| animal | Mantaray | void Interact() | Mantaray | public void interact() | Ya | - |
| animal | Ostrich | Ostrich(int x, int y, int weight) | Ostrich | public Ostrich(int weight,int absis,int ordinat) | Ya | - |
| animal | Ostrich | void Interact() | Ostrich | public void interact() | Ya | - |
| animal | Orangutan | Orangutan(int x, int y, int weight) | Orangutan | public Orangutan(int weight,int absis,int ordinat) | Ya | - |
| animal | Orangutan | void Interact() | Orangutan | public void interact() | Ya | - |
| animal | Peacock | Peacock(int x, int y, int weight) | Peacock | public Peacock(int weight,int absis,int ordinat) | Ya | - |
| animal | Peacock | void Interact() | Peacock | public void interact() | Ya | - |
| animal | Shark | Shark(int x, int y, int weight) | Shark | public Shark(int weight,int absis,int ordinat) | Ya | - |
| animal | Shark | void Interact() | Shark | public void interact() | Ya | - |
| animal | Whale | Whale(int x, int y, int weight) | Whale | public Whale(int weight,int absis,int ordinat) | Ya | - |
| animal | Whale | void Interact() | Whale | public void interact() | Ya | - |
| cell | AirHabitat | AirHabitat(int x,int y,bool s) | AirHabitat | public AirHabitat(int absis, int ordinat) | Ya | Variabel boolean dihilangkan |
| cell | AirHabitat | void Render() | AirHabitat | public void render() | Ya | - |
| cell | Cell | Cell(int x,int y) | Cell | public Cell(int absis,int ordinat) | Ya | - |
| cell | Cell | virtual void Render(); | Cell | public void render() | Ya | - |
| cell | Cell | int GetX() | Cell | public int getCellAbsis() | Ya | - |
| cell | Cell | Cell(int x,int y) | Cell | public Cell(int absis,int ordinat) | Ya | - |
| cell | Cell | virtual void Render(); | Cell | public void render() | Ya | - |
| cell | Cell | int GetX() | Cell | public int getCellAbsis() | Ya | - |
| cell | Entrance | Entrance(int x,int y) | Entrance | public Entrance(int absis, int ordinat) | Ya | - |
| cell | Entrance | void Render() | Entrance | public void render() | Ya | - |
| cell | Exit | Exit(int x,int y) | Exit | public Exit(int absis, int ordinat) | Ya | - |
| cell | Exit | void Render() | Exit | public void render() | Ya | - |
| cell | Facility | Facility(int x,int y) | Facility | public Facility(int absis, int ordinat) | Ya | - |
| cell | Habitat | Habitat(int x,int y,bool s) | Habitat | public Habitat(int absis, int ordinat) | Ya | Variabel boolean dihilangkan |
| cell | LandHabitat | LandHabitat(int x,int y,bool s) | LandHabitat | public LandHabitat(int absis, int ordinat) | Ya | Variabel boolean dihilangkan |
| cell | Park | Park(int x,int y) | Park | public Park(int absis, int ordinat) | Ya | - |
| cell | Park | void Render() | Park | public void render() | Ya | - |
| cell | Renderable | virtual void Render()=0 | Renderable | public void render() | Ya | Pada C++ disebut sebagai Abstract Base Class (ABC), pada Java disebut interface. Keduanya sama-sama tidak diimplementasi pada method tersebut. |
| cell | Restaurant | Restaurant(int x,int y) | Restaurant | public Restaurant(int absis, int ordinat) | Ya | - |
| cell | Restaurant | void Render() | Restaurant | public void render() | Ya | - |
| cell | Road | Road(int x,int y) | Road | public Road(int absis, int ordinat) | Ya | - |
| cell | WaterHabitat | WaterHabitat(int x,int y,bool s) | WaterHabitat | public WaterHabitat(int absis, int ordinat) | Ya | Variabel boolean dihilangkan |
| cell | WaterHabitat | void Render() | WaterHabitat | public void render() | Ya | - |
| zoo | Cage | Cage(Habitat \*\* hl,int nh) | Cage | public Cage(int numberOfHabitat,int numberOfAnimal) | Tidak | Pada C++ parameternya berupa array of habitat dan jumlah habitat, sedangkan pada Java parameternya adalah jumlah habitat dan jumlah hewan |
| zoo | Cage | int GetNH() const | Cage | public int numberOfHabitat() | Ya | Perubahan nama method |
| zoo | Cage | int GetHabByCoor(int x,int y) const | Cage | public Habitat getHabitatByIndex(int index) | Ya | - |
| zoo | Driver | Driver(ifstream& infile) | Driver | public Driver(String filePath) throws FileNotFoundException,IOException, ZooException | Ya | Pada C++ Zoo dibuat sebagai parameter, pada Java Zoo dibuat sebagai member dari kelas Driver |
| zoo | Driver | void DisplayMenu() | Driver | public void showMenu() | Ya | - |
| zoo | Driver | void ShowZoo() | Driver | public void showZooWithBoundary() throws ZooException | Ya | - |
| zoo | Driver | void TourZoo() | Driver | public void tourZoo() | Ya | - |
| zoo | Point | int GetX() const | Point | public int getAbsis() | Ya | Perubahan nama method |
| zoo | Point | int GetY() const | Point | public int getOrdinat() | Ya | Perubahan nama method |
| zoo | Point | void SetX() const | Point | public void setAbsis(int absis) | Ya | Perubahan nama method |
| zoo | Point | void SetY() const | Point | public void setOrdinat(int absis) | Ya | Perubahan nama method |
| zoo | Point | Point(int x, int y) | Point | public Point(int absis,int ordinat) | Ya | - |
| zoo | Zoo | Zoo(ifstream& infile) | Zoo | public Zoo(int column, int row) | Tidak | Pada C++ parameter berupa file eksternal, pada Java parameter berupa banyak baris dan kolom yang ingin dibuat |
| zoo | Zoo | int GetHeight() const | Zoo | public int getZooColumn() | Ya | Perubahan nama method |
| zoo | Zoo | int GetWidth() const | Zoo | public int getZooRow() | Ya | Perubahan nama method |

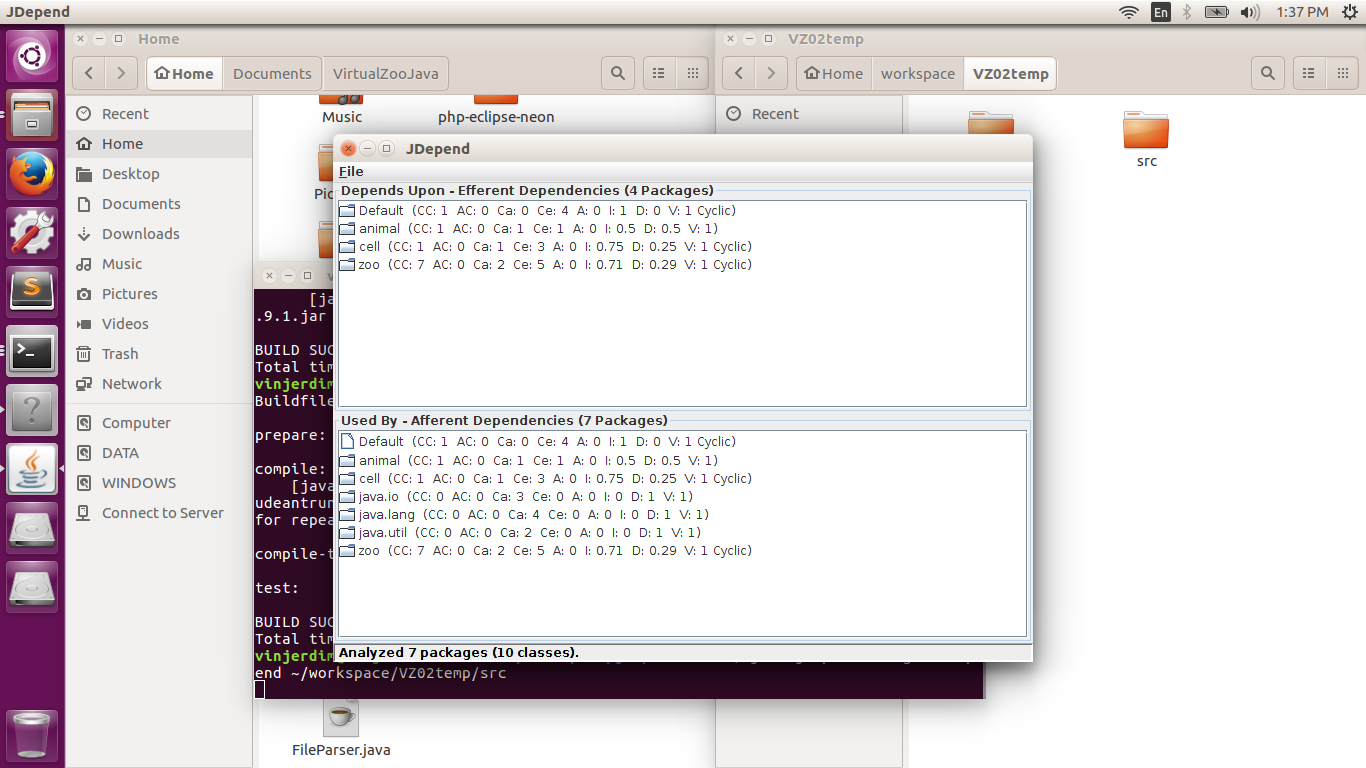
Kesimpulan Penggunaan Ulang :

Dengan penggunaan package pada java, maka pembuatan kelas-kelas lebih teratur.

# **3 Pengukuran Metriks Aplikasi**

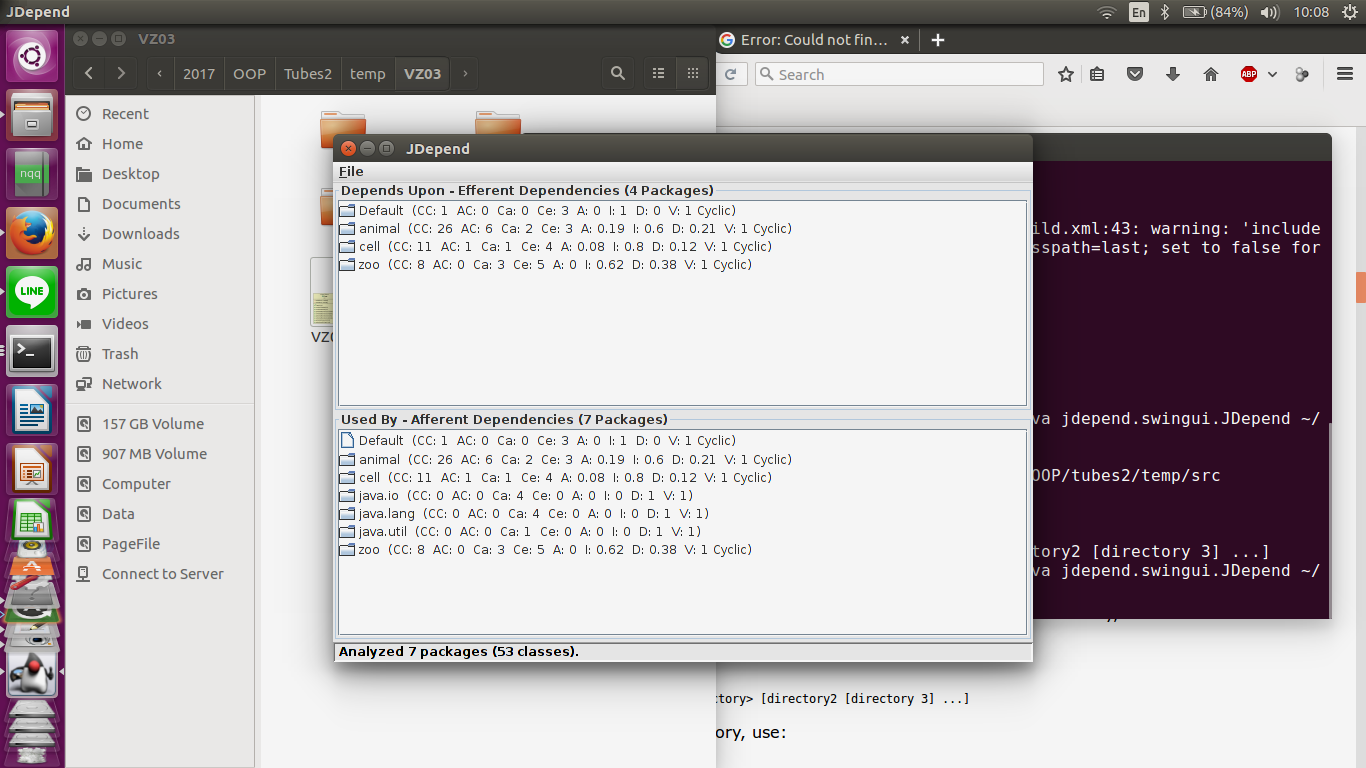
**3.1 Versi VZ02**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Class Count | Abstract Class Count | Ca | Ce | A | I | D | V |
| Default | 1 | 0 | 0 | 4 | 0 | 1 | 0 | 1 |
| animal | 1 | 0 | 1 | 1 | 0 | 0.5 | 0.5 | 1 |
| cell | 1 | 0 | 1 | 3 | 0 | 0.75 | 0.25 | 1 |
| zoo | 7 | 0 | 2 | 5 | 0 | 0.71 | 0.29 | 1 |



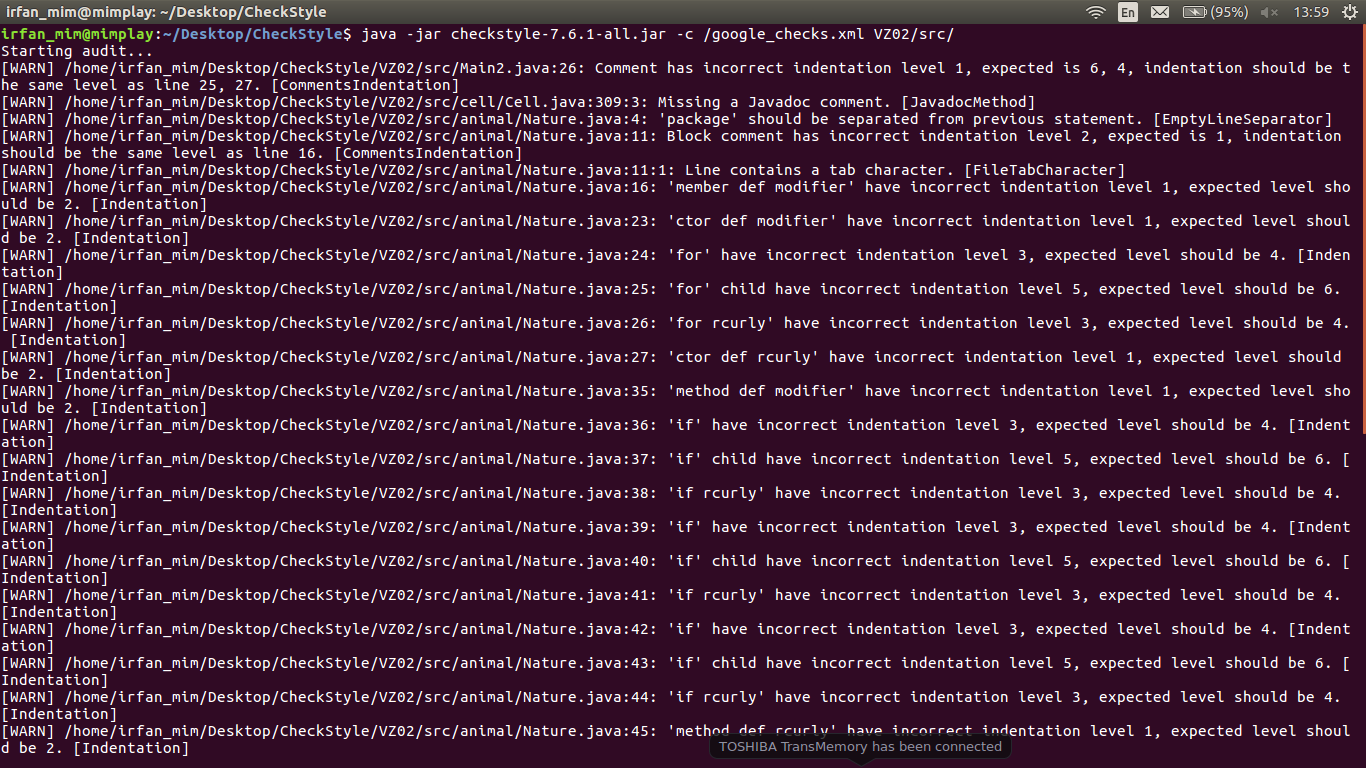
**3.2 Versi VZ03**

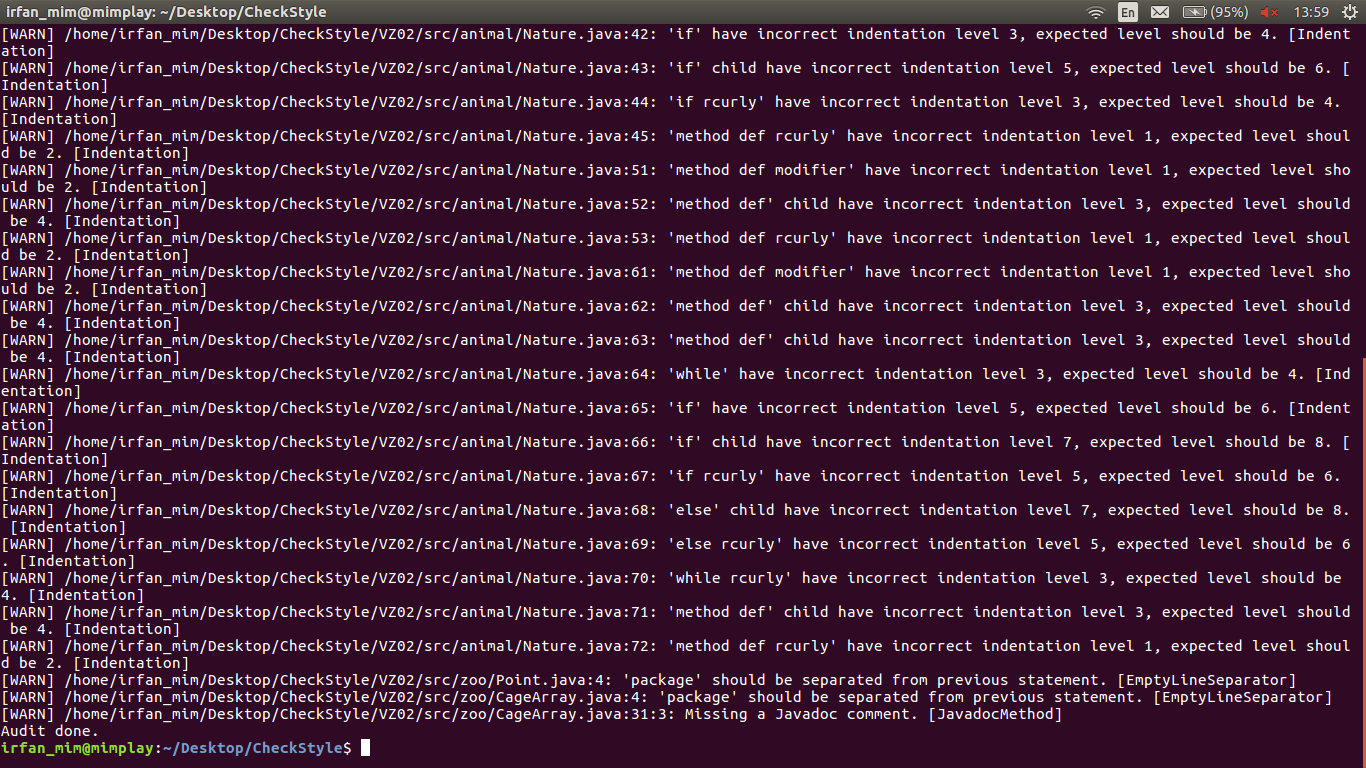
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Class Count | Abstract Class Count | Ca | Ce | A | I | D | V |
| Default | 1 | 0 | 0 | 3 | 0 | 1 | 0 | 1 |
| animal | 11 | 6 | 2 | 3 | 0.19 | 0.6 | 0.21 | 1 |
| cell | 26 | 1 | 1 | 4 | 0.08 | 0.8 | 0.12 | 1 |
| zoo | 8 | 0 | 3 | 5 | 0 | 0.62 | 0.38 | 1 |



# **4 Pengukuran Kualitas Kode Aplikasi**

**3.1 Versi VZ02**

****

****

**3.2 Versi VZ03**

# 

# **5 Dokumentasi Rinci Kelas**

Di-*upload* dalam bentuk softcopy di olympia.

**LAMPIRAN**