

# LAPORAN TUGAS BESAR 2

## IF2210/Pemrograman Berorientasi Objek

### ArkavQuarium

Dipersiapkan oleh:

K02F - TimBelakangUnited

13516062 - Yusuf Rahmat Pratama

13516083 - Abram Perdanaputra

13516095 - Faza Fahleraz

13516107 - Senapati Sang Diwangkara

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	<b>Sekolah Teknik Elektro dan Informatika ITB</b>	<b>Nomor Dokumen</b>	<b>Halaman</b>
		<i>IF2210-TB-06-02</i>	20
		Revisi	0

<b>STEI- ITB</b>	<b>IF2210-TB-06-02</b>	<b>Halaman 1 dari 20 halaman</b>
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

## Daftar Isi

<b>Ringkasan</b>	<b>3</b>
<b>Penjelasan Tambahan Spesifikasi Tugas</b>	<b>3</b>
Penempatan Pellet dengan Mouse	3
Pengambilan Coin dengan Mouse	3
<b>Rancangan Kelas</b>	<b>4</b>
Perubahan dari Tugas Besar	5
<b>Rincian Kelas</b>	<b>5</b>
Aquarium	5
Aquatic (Interface)	5
Coin	5
Constants	6
Fish	6
Game Controller	7
Game View	7
Guppy	7
LinkedList<E>	7
Pellet	8
Piranha	8
Snail	9
<b>Program Utama</b>	<b>9</b>
<b>Test Script</b>	<b>9</b>
<b>Pengukuran Metriks Aplikasi</b>	<b>13</b>
<b>Pengukuran Kualitas Kode Aplikasi</b>	<b>15</b>
<b>Pembagian Kerja dalam Kelompok</b>	<b>15</b>
<b>Lampiran</b>	<b>17</b>
Form Asistensi	17
Log Activity Anggota Kelompok	18
Screenshot Program	18

# 1 Ringkasan

ArkavQuarium adalah sebuah game yang terinspirasi dari game terkenal Insaniquarium. Pada update ke 2.0 ini, ArkavQuarium diubah menjadi menggunakan bahasa Java, dengan penggunaan library nya untuk game engine, dokumentasi, unit testing, dsb. Game ini dibuat dengan menggunakan paradigma OO (*Object-oriented*), dimana setiap *sprite* yang ada di dalam game merupakan objek dari kelas yang sudah didefinisikan.

Pada laporan ini akan dibahas perubahan rancangan kelas dari versi 1.0 yang menggunakan bahasa C++, sekaligus mendefinisikan rincian kelas yang bersangkutan. Selain itu, dijelaskan juga program utama untuk menjalankan game ini. Setelahnya, dirincikan test-case yang digunakan untuk unit-testing, pengukuran metriks aplikasi, dan pengukuran kualitas kode aplikasi yang didapat dari penggunaan library yang tersedia (JUnit, JavaDoc, JDepend, JavaSwing). Pada akhir laporan akan dijelaskan pembagian kerja kelompok, sekaligus lampiran yang mendukung laporan ini.

Dari hasil Tugas besar ini dapat disimpulkan bahwa penggunaan paradigma OO sangat efektif dan relatif lebih mudah dalam membuat sebuah game. Selain itu dapat digunakan unit-testing, dan pengukuran kualitas kode untuk mempermudah dalam menghasilkan program yang bersifat *clean code*, dan cenderung *bug-free*.

# 2 Penjelasan Tambahan Spesifikasi Tugas

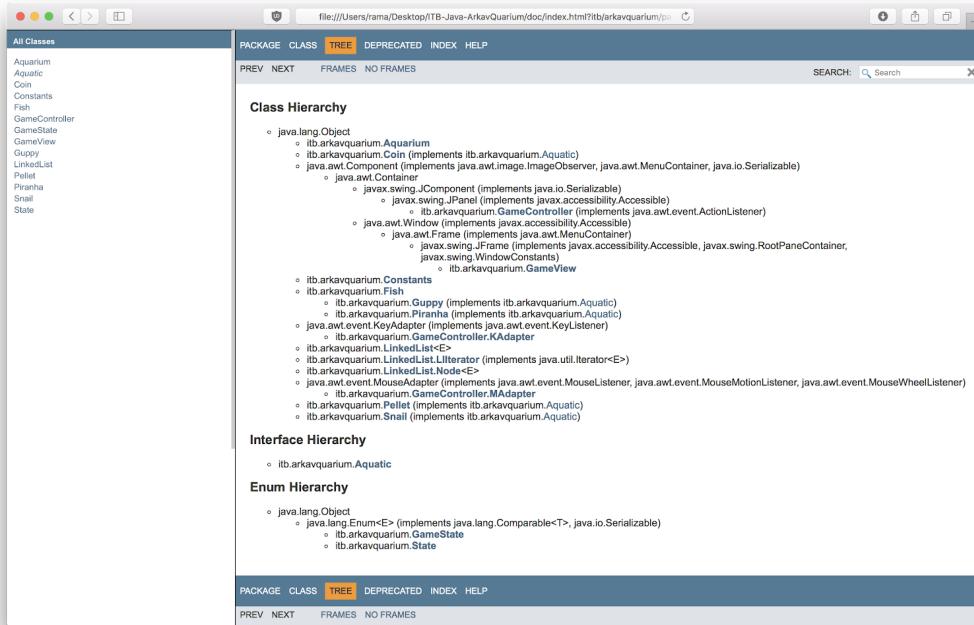
## 2.1 Penempatan Pellet dengan Mouse

Menggunakan library JavaSwing, objek Pellet dapat ditaruh di lokasi manapun dalam peta menggunakan klik mouse pada tempat yang diinginkan.

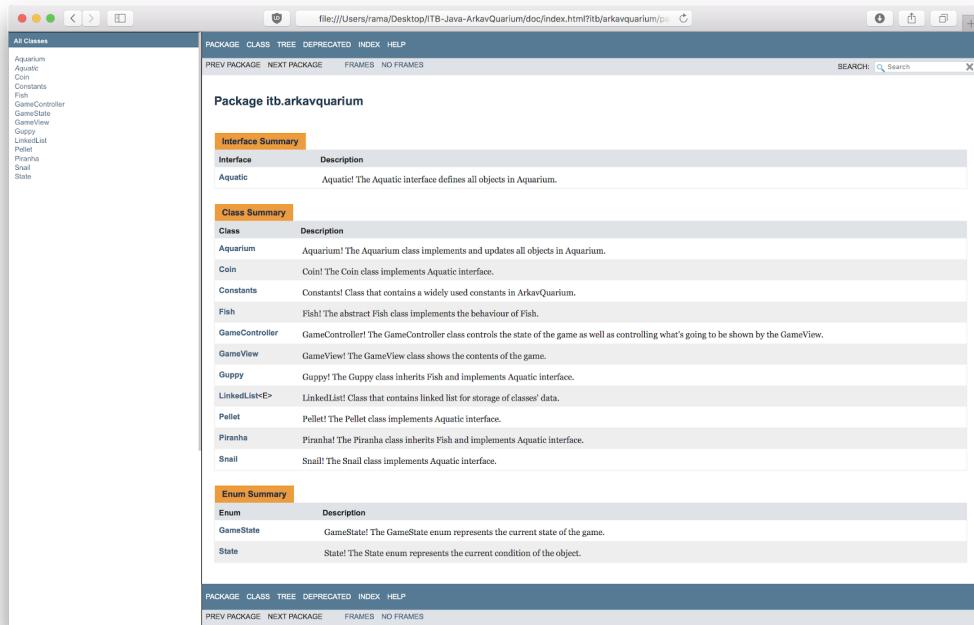
## 2.2 Pengambilan Coin dengan Mouse

Menggunakan library JavaSwing, objek Coin dapat diambil di lokasi manapun dalam peta menggunakan klik mouse pada coin yang ingin diambil.

### 3 Rancangan Kelas



Gambar 1. Class Hierarchy dari program ArkavQuarium 2.0



Gambar 2. Daftar kelas yang ada dalam program ArkavQuarium 2.0

STEI- ITB	IF2210-TB-06-02	Halaman 4 dari 20 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

Dapat dilihat bahwa secara keseluruhan, terdapat 9 kelas utama, 1 interface, 1 kelas abstrak, dan 1 kelas bantuan (Constants yang hanya berisi atribut konstanta untuk seluruh program, serta metode random). Selain itu juga terdapat 2 enumerables dalam program ini.

### **3.1 Perubahan dari Tugas Besar**

- Penggunaan JavaSwing yang menggantikan library SDL di C++
- Penggunaan JUnit untuk testing menggantikan unit testing yang dibuat sendiri di C++
- Penggunaan JavaDoc yang menggantikan Doxygen di C++

## **4 Rincian Kelas**

### **4.1 Aquarium**

Kelas Aquarium merupakan representasi dari Aquarium yang menampung semua object dalam aquarium dengan penggunaan LinkedList, serta memberikan *signal* untuk object-object tersebut mengupdate dirinya setiap waktu sesuai dengan *current time* yang disimpan dalam Aquarium.

Methods:

- Create Coin, Guppy, Pellet, Piranha, Snail
- Delete Coin, Guppy, Pellet, Piranha, Snail
- Getter dan setCurrTime
- updateState

Attributes:

- xmin, ymin, xmax, ymax
- currTime
- contentCoin, contentGuppy, content Pellet, contentPiranha, contentSnail

### **4.2 Aquatic (Interface)**

Interface Aquatic merupakan *blueprint* dari objek yang dapat dibuat dan ditampung pada Aquarium. Semua objek Aquarium harus mengimplementasi interface Aquatic supaya penggambaran objek dari kelas yang bersangkutan dapat dilakukan.

Methods:

- move, dead
- updateProgress, updateState
- Getter - Setter

### **4.3 Coin**

Kelas Coin merupakan representasi dari koin dalam permainan. Kelas Coin mengimplement interface Aquatic, yaitu ia merupakan *inheritance* dari Aquatic.

Methods:

STEI- ITB	IF2210-TB-06-02	Halaman 5 dari 20 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

- isInside
- Getter dan Setter
- Implementasi method Aquatic

Attributes:

- value
- moveSpeed
- lastBottomTime
- aquarium
- posX
- posY
- lastCurrTime
- lastProgressTime
- currState
- progress

#### **4.4 Constants**

Kelas khusus yang berisi semua konstanta yang digunakan dalam program (selain konstanta yang merupakan atribut dari kelas tertentu).

Methods:

- random

Attributes:

- All required constants

#### **4.5 Fish**

Kelas abstrak yang merupakan rancangan dasar bagi sebuah objek ikan, dengan berisi informasi untuk membuat sebuah objek beraksi sebagai ikan secara general.

Methods:

- Getter-Setter

Attributes:

- foodThres
- eatRadius
- fullInterval
- hungerTimeout
- lastEatTime
- lastRandomTime
- lastHungerTime
- foodEaten
- hungry
- level
- dirX

STEI- ITB	IF2210-TB-06-02	Halaman 6 dari 20 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

- dirY

## 4.6 Game Controller

Kelas yang mengontrol status dari game, termasuk penggunaan game engine untuk menggambar objek yang bersangkutan.

## 4.7 Game View

Kelas yang memanggil Game Controller untuk menampilkan isi dari game nya.

## 4.8 Guppy

Kelas Guppy merupakan representasi dari ikan Guppy yang merupakan salah satu *sprite* dalam game. Kelas Guppy mengextends kelas abstrak Fish dan mengimplementasi interface Aquatic.

Methods:

- Implementasi metode Aquatic
- Implementasi metode Fish
- Getter - Setter

Attributes:

- moveSpeed
- aquarium
- nearestPellet
- lastDropCoin
- abscissa
- ordinate
- lastCurrTime
- lastProgressTime
- currState
- progress

## 4.9 LinkedList<E>

Kelas LinkedList merupakan sebuah kelas generik yang berupa pengaplikasian dari linked list, dimana kelas menyimpan node yang merupakan head, dan setiap node akan saling berhubungan dengan menyimpan address next dan prev nya. Dalam Kelas LinkedList ini, terdapat nested class berupa Kelas Node untuk pengaplikasian linked list, serta Kelas LIIterator yang mengimplementasikan interface Iterator agar LinkedList dapat dipanggil menggunakan iterator tersebut.

Methods:

- add
- isEmpty
- iterator
- remove

STEI- ITB	IF2210-TB-06-02	Halaman 7 dari 20 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

- Getter - Setter

Attributes:

- head
- length

#### **4.10 Pellet**

Kelas Pellet merupakan representasi dari pellet yang merupakan makanan bagi Guppy. Kelas ini mengimplementasikan interface Aquatic.

Methods:

- Implementasi methods Aquatic
- isInside
- Getter-Setter

Attributes:

- moveSpeed
- aquarium
- abscissa
- ordinate
- lastCurrTime
- lastProgressTime
- currState
- progress

#### **4.11 Piranha**

Kelas Piranha merupakan kelas yang merepresentasikan piranha. Kelas ini mengextends dari kelas abstrak Fish, dan mengimplementasikan interface Aquatic.

Methods:

- Implementasi metode Fish
- Implementasi metode Aquatic
- Getter-Setter

Attributes:

- moveSpeed
- nearestGuppy
- aquarium
- abscissa
- ordinate
- lastCurrTime
- lastProgressTime
- currState
- progress

STEI- ITB	IF2210-TB-06-02	Halaman 8 dari 20 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

## 4.12 Snail

Kelas Snail merupakan kelas yang merepresentasikan snail, dimana snail merupakan *sprite* khusus pengambilan Coin. Kelas ini mengimplementasikan interface Aquatic.

Methods:

- resetCoin
- Implementasi metode Aquatic
- Getter-Setter

Attributes:

- snailRadius
- moveSpeed
- nearestCoin
- holdCoinValue
- dirX
- aquarium
- abscissa
- ordinate
- lastCurrTime
- lastProgressTime
- currState
- progress

## 5 Program Utama

Program utama terdapat dalam kelas GameController, dimana saat dijalankan, akan membuat sebuah Runnable yang akan menjalankan GameView untuk menampilkan main menu nya. GameView kemudian akan menginisialisasi GameController. GameController akan menginisialisasi engine nya dan juga Aquarium. Dari situ program akan *looping* dengan menjalankan game sampai keadaan menang atau kalah tercapai. Runnable digunakan karena JSwing pada dasarnya menggunakan multi-threading, tetapi dalam program ini kami hanya menggunakan satu thread saja.

## 6 Test Script

No.	Kelas	Nama File Driver	Fitur/Method yang diuji	Kasus Pengujian	Hasil yang Diharapkan	Hasil yang Keluar
1	Aquarium	AquariumTest.java	updateState	Pemanggilan updateState dengan delta waktu 50 satuan	Waktu aquarium bertambah 50	Waktu aquarium bertambah 50. Return 0 (success)
2	Aquarium	AquariumTest.java	createObjects	Pemanggilan createPiranha,	Objek-objek terinisialisasi	Objek-objek berhasil terinisialisasi dan

				createGuppy, createSnail, createPellet, dan createCoin dari Aquarium yang masih kosong	dan masuk ke LinkedList	masuk ke LinkedList. Return 0 (success)
3	Aquarium	AquariumTest.java	deleteObjects	Pemanggilan createObjects, kemudian dilakukan deletePiranha, deleteGuppy, deleteSnail, deletePellet, dan deleteCoin terhadap objek yang telah dibuat	Objek-objek yang bersangkutan terhapus dari LinkedList	Objek-objek berhasil dihapus dari LinkedList. Return 0 (success)
4	LinkedList	LinkedListTest.java	isEmpty	Menginisialisasi LinkedList kemudian memanggil metode isEmpty	True, karena LinkedList masih kosong	True. Return 0 (success)
5	LinkedList	LinkedListTest.java	isEmpty	Menginisialisasi LinkedList kemudian menambahkan elemen baru ke dalam LinkedList, setelah itu memanggil metode isEmpty	False, karena LinkedList sudah terisi	False. Return 0 (success)
6	LinkedList	LinkedListTest.java	add	Menginisialisasi LinkedList kemudian menambahkan elemen (node) baru dengan metode add. Kemudian mencari elemen tersebut dalam LinkedList	Elemen baru ditemukan	Elemen baru berhasil ditemukan. Return 0 (success)
7	LinkedList	LinkedListTest.java	remove	Menginisialisasi LinkedList, menambahkan elemen baru dengan add, kemudian memanggil	Elemen tidak ditemukan karena sudah dihapus	Elemen tidak ditemukan. Return 0 (success)

				metode remove. Setelah itu mencari elemen tersebut dalam LinkedList		
8	LinkedList	LinkedListTest.java	iterator	Menginisialisasi LinkedList dan menambahkan elemen dalam LinkedList sesuai dengan isi dari sebuah array yang ditentukan. Kemudian metode iterator dipanggil dan digunakan untuk mengecek isi dengan array	Metode iterator berhasil mengeluarkan sebuah Iterator, dan pengecekan menggunakan Iterator menghasilkan elemen yang sama	Metode terpanggil, Iterator terbentuk, pengecekan dengan Iterator menghasilkan isi sama dengan array. Return 0 (success)
9	Coin	CoinTest.java	updateState	Pemanggilan updateState dengan currTime dijalankan. Mengecek apakah Coin benar turun	Posisi y Coin bertambah setelah dipanggil updateStae	Posisi y Coin bertambah. Return 0 (success)
10	Coin	CoinTest.java	updateState	Pemanggilan updateState dengan currTime dijalankan. Mengecek apakah Coin bergerak dalam arah x	Posisi x Coin tidak berubah sesuai spek	Posisi x Coin tidak berubah. Return 0 (success)
11	Coin	CoinTest.java	updateState	Mengecek apabila Coin sudah sampai bawah (yMax) apakah akan terus ke bawah	Posisi y Coin tidak bertambah apabila telah sampai pada yMax	Posisi y Coin tidak bertambah. Return 0 (success)
12	Coin	CoinTest.java	updateState	Mengecek apabila Coin di yMax terhapus apabila melewati rentang waktunya (COIN_DELETION_INTERVAL )	Coin terhapus	Coin berhasil terhapus. Return 0 (success)

13	Pellet	PelletTest.java	updateState	Pemanggilan updateState dengan currTime dijalankan. Mengecek apakah Pellet benar turun	Posisi y Pellet bertambah setelah dipanggil updateStae	Posisi y Pellet bertambah. Return 0 (success)
14	Pellet	PelletTest.java	updateState	Pemanggilan updateState dengan currTime dijalankan. Mengecek apakah Pellet bergerak dalam arah x	Posisi x Pellet tidak berubah sesuai spek	Posisi x Pellet tidak berubah. Return 0 (success)
15	Pellet	PelletTest.java	updateState	Mengecek apabila Pellet di yMax terhapus	Pellet terhapus	Pellet berhasil terhapus. Return 0 (success)
16	Guppy	GuppyTest.java	updateState	Mengecek apakah jika Guppy lapar akan mengejar Pellet terdekat	Guppy bergerak ke arah Pellet terdekat	Guppy berhasil bergerak ke arah Pellet terdekat. Return 0 (success)
17	Guppy	GuppyTest.java	updateState	Mengecek apakah jika Guppy lapar dan ada Pellet di radius nya akan dimakan, dan menghasilkan Coin	Guppy makan dan menghasilkan Coin	Guppy berhasil memakan Pellet dan menghasilkan Coin. Return 0 (success)
18	Guppy	GuppyTest.java	updateState	Mengecek apakah dalam interval tertentu Guppy menghasilkan Coin	Guppy menghasilkan Coin	Guppy berhasil menghasilkan Coin. Return 0 (success)
19	Guppy	GuppyTest.java	updateState	Mengecek apakah jika Guppy tidak makan dalam waktu interval maka akan mati	Guppy mati	Guppy berhasil mati. Return 0 (success).
20	Piranha	PiranhaTest.java	updateState	Mengecek apakah jika Piranha lapar akan mengejar Guppy terdekat	Piranha bergerak ke arah Guppy terdekat	Piranha berhasil bergerak ke arah Guppy terdekat. Return 0 (success)
21	Piranha	PiranhaTest.java	updateState	Mengecek apakah jika	Piranha makan dan	Piranha berhasil memakan Guppy dan

				Piranha lapar dan ada Guppy di radius nya akan dimakan, dan menghasilkan Coin	menghasilkan Coin	menghasilkan Coin. Return 0 (success)
22	Piranha	PiranhaTest.java	updateState	Mengecek apakah jika Piranha tidak makan dalam waktu interval maka akan mati	Piranha mati	Piranha berhasil mati. Return 0 (success).
23	Snail	SnailTest.java	updateState	Mengecek apakah Snail akan mengejar Coin terdekat	Snail bergerak ke arah Coin terdekat	Snail berhasil bergerak ke arah Coin terdekat. Return 0 (success).
24	Snail	SnailTest.java	updateState	Mengecek apakah Snail bergerak hanya pada posisi x	Posisi y Snail tidak berubah	Posisi y Snail tidak berubah. Return 0 (success)
25	Snail	SnailTest.java	updateState	Mengecek apakah jika ada Coin yang berada di radius Snail akan dimakan oleh Snail	Coin berhasil dihapus (dimakan oleh Snail)	Coin berhasil dihapus. Return 0 (success)

## 7 Pengukuran Metriks Aplikasi

```
- Package: itb.arkavquarium
```

Stats:

Total Classes: 28  
Concrete Classes: 26  
Abstract Classes: 2

Ca: 0  
Ce: 5

A: 0.07  
I: 1  
D: 0.07

**Abstract Classes:**

itb.arkavquarium.Aquatic  
itb.arkavquarium.Fish

**Concrete Classes:**

itb.arkavquarium.Aquarium  
itb.arkavquarium.AquariumTest  
itb.arkavquarium.Coin  
itb.arkavquarium.CoinTest  
itb.arkavquarium.Constants  
itb.arkavquarium.FishTest  
itb.arkavquarium.GameController\$1  
itb.arkavquarium.GameController\$KAdapter  
itb.arkavquarium.GameController\$MAdapter  
itb.arkavquarium.GameState  
itb.arkavquarium.GameView  
itb.arkavquarium.Guppy  
itb.arkavquarium.Guppy\$1  
itb.arkavquarium.GuppyTest  
itb.arkavquarium.LinkedList  
itb.arkavquarium.LinkedList\$L1Iterator  
itb.arkavquarium.LinkedList\$Node  
itb.arkavquarium.LinkedListTest  
itb.arkavquarium.Pellet  
itb.arkavquarium.PelletTest  
itb.arkavquarium.Piranha  
itb.arkavquarium.Piranha\$1  
itb.arkavquarium.PiranhaTest  
itb.arkavquarium.Snail  
itb.arkavquarium.SnailTest  
itb.arkavquarium.State

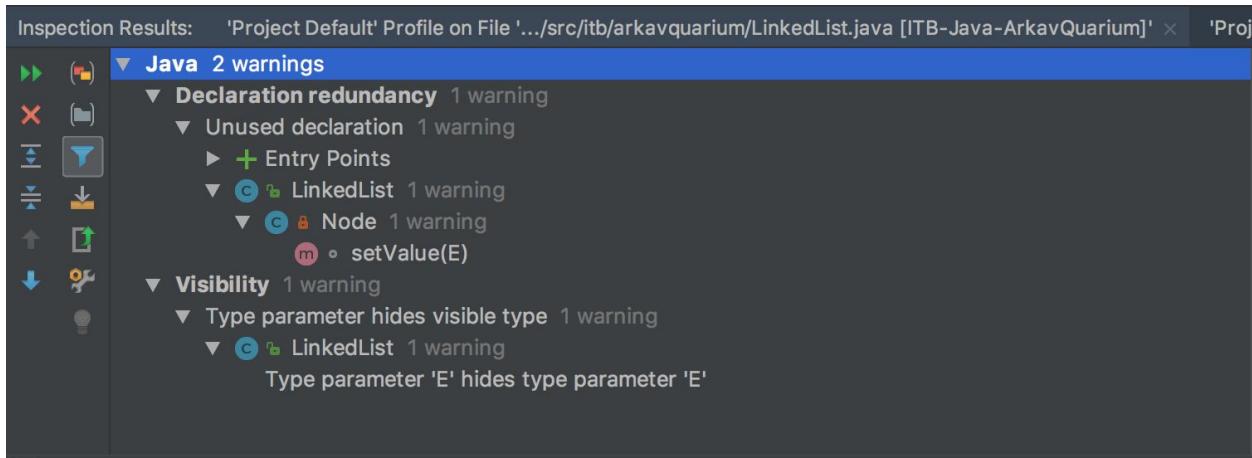
**Depends Upon:**

java.awt.event  
java.lang  
java.util  
javax.swing  
org.junit

**Used By:**

Not used by any packages.

## 8 Pengukuran Kualitas Kode Aplikasi



The screenshot shows the 'Inspection Results' window in IntelliJ IDEA. The title bar says 'Inspection Results: 'Project Default' Profile on File .../src/itb/arkavquarium/LinkedList.java [ITB-Java-ArkavQuarium]' × 'Proj'. The main pane is titled 'Java 2 warnings'. It lists two types of warnings:

- Declaration redundancy**: 1 warning
  - Unused declaration: 1 warning
    - Entry Points
    - LinkedList: 1 warning
      - Node: 1 warning
        - setValue(E)
- Visibility**: 1 warning
  - Type parameter hides visible type: 1 warning
    - LinkedList: 1 warning
      - Type parameter 'E' hides type parameter 'E'

Gambar 3. Hasil checkstyle

Dari checkstyle didapat bahwa tidak ada lagi Error atau typo yang ada, tetapi terdapat 2 warnings dimana satu berisi bahwa setValue tidak pernah dipakai, tetapi dibuat untuk digunakan sebagai getter-setter. Sedangkan satu lagi karena tipe E generik digunakan untuk membangun kelas yang bersifat private (Node) di dalam kelas lain (LinkedList)

## 9 Pembagian Kerja dalam Kelompok

Nama	Pembagian Kerja	Role
Yusuf Rahmat Pratama	Membuat LinkedList, JUnit untuk AquariumTest dan LinkedListTest. Membuat laporan, membuat JavaDoc	Developer / Tester
Abram Perdanaputra	Mengkonversi program dari C++ ke Java. Melakukan CheckStyle dan membenarkannya. Melakukan JDepend, membuat JavaDoc	Developer
Faza Fahleraz	Mengkonversi game engine dari SDL ke JavaSwing. Membuat GameController dan GameView	Developer

Senapati Sang Diwangkara	Membuat JUnit untuk CoinTest, FishTest, GuppyTest, PelletTest, PiranhaTest, dan SnailTest	Tester
--------------------------	----------------------------------------------------------------------------------------------------	--------

<b>STEI- ITB</b>	<b>IF2210-TB-06-02</b>	<b>Halaman 16 dari 20 halaman</b>
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

## 10 Lampiran

### 10.1 Form Asistensi

Asistensi II		Catatan Asistensi:
Tanggal : 19/04/2010		
Tempat : Laboratorium Pemrograman		
Kehadiran Anggota Kelompok:		
No		
NIM		
Tanda tangan		
1 13516062 	- Pembahasan total boleh tapi dibentuk di laporan	
2 13516095 	- Driver boleh dia main class	
3 13516083 Abraam	- Style bebas tapi dibentuk	
4	- Interface dibuat	
5	- Extend + implement dibuat	
6	- Unit testing gaada minimal	
	Tanda Tangan Asisten: 	

## **10.2 Log Activity Anggota Kelompok**

Logact dapat dilihat di file doc/logact.txt

## **10.3 Screenshot Program**





