

Kelompok : 1. Joshua Salimin / 13414001
2. Jeremia Jason Lasiman / 13514021
3. Bervianto Leo Pratama / 13514047
4. M. Az-zahid Adhitya Silparensi / 13514095

Ass Mentor : Christ A. S.

1 Deskripsi Umum Aplikasi

Aplikasi ini berupa sebuah *game* dinamakan *The Eater* dimana pemain dapat menggerakkan satu makhluk yaitu player dan memakan makhluk lainnya dengan *powerlevel* yang lebih rendah dari dirinya. Jika pemain memakan makhluk yang lebih kuat darinya, maka makhluk pemain akan mati dan game berakhir serta menampilkan skor yang telah diraih oleh player selain itu terdapat beberapa jebakan yang tidak terlihat yang telah disiapkan dan jika player terkena jebakan tersebut maka game pun akan berakhir. Jika player berhasil memakan makhluk maka player akan mendapatkan skor sebesar 2 dan jika seluruh makhluk (kecuali jebakan) telah habis dimakan maka player telah berhasil memenangkan *game The Eater* dan skor akan ditampilkan. Pada awalnya player akan memiliki *powerlevel* sebesar 51 serta bentuk A dan jika telah berhasil memakan makhluk maka *powerlevel* akan meningkat sebanyak 1 dan jika telah memakan sebanyak 5 buah makhluk maka player dapat berevolusi sesuai alfabet yaitu dari A sampai Z.

2 Deskripsi Desain Teknis Aplikasi

Kode sudah bersih sesuai dengan ketentuan penulisan yang berlaku secara umum yaitu mengikuti *CheckStyle* namun ada beberapa yang tidak dapat kami ubah karena ada hal yang tetap dibutuhkan untuk tidak mengikuti *hint* dari *CheckStyle*. Kehebatan rancangannya ialah terletak pada desain pattern yang kami pakai yaitu MVC (Model – View – Controller) dan beberapa kelas kami jadikan Singleton.

Kelas Model : Makhluk.java, MakhlukAntiAir.java, MakhlukBurukRupa.java, MakhlukMaling.java, MakhlukMonster.java, MakhlukPlayer.java, MakhlukTerbang.java

Kelas View : Board.java

Kelas Controller : ControlCommand.java, MakhlukSpawn.java

Kelas Singleton : MakhlukPlayer.java, ControlCommand.java,

Kehebatannya disini ialah kami menerapkan dua desain pattern sekaligus yaitu MVC dan singleton karena beberapa kelas hanya dibutuhkan satu instance saja maka kami buat sebagai singleton.

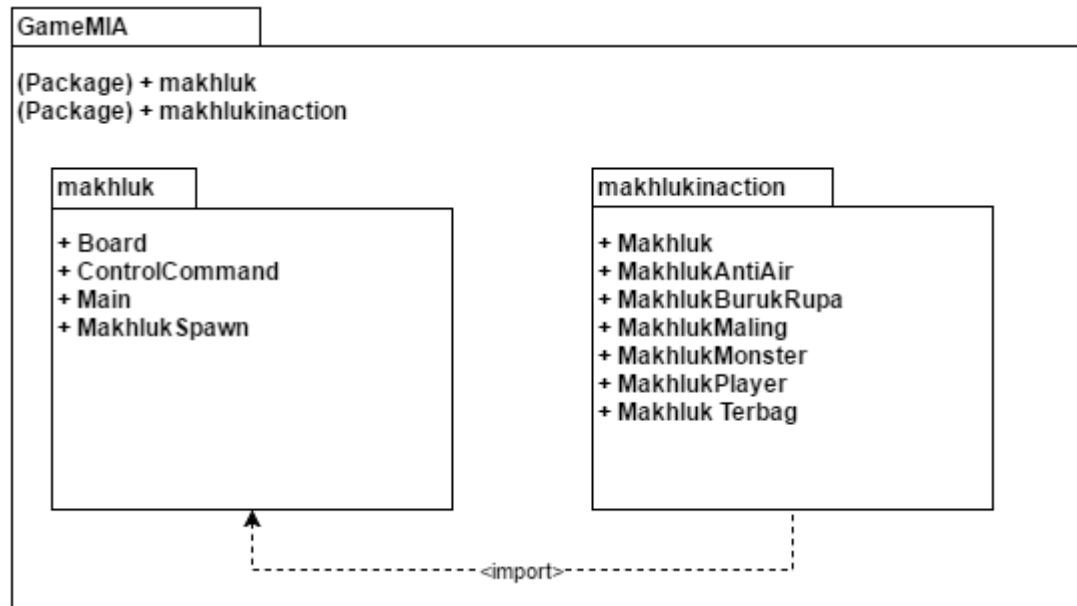
Seluruh sourcecode kami lengkap dengan javadoc nya sehingga ketika javadoc dapat mengenali dan mengenerate dengan sempurna.

Sourcecode kami pun dapat dibuat metriknya dengan jDepend dengan baik.

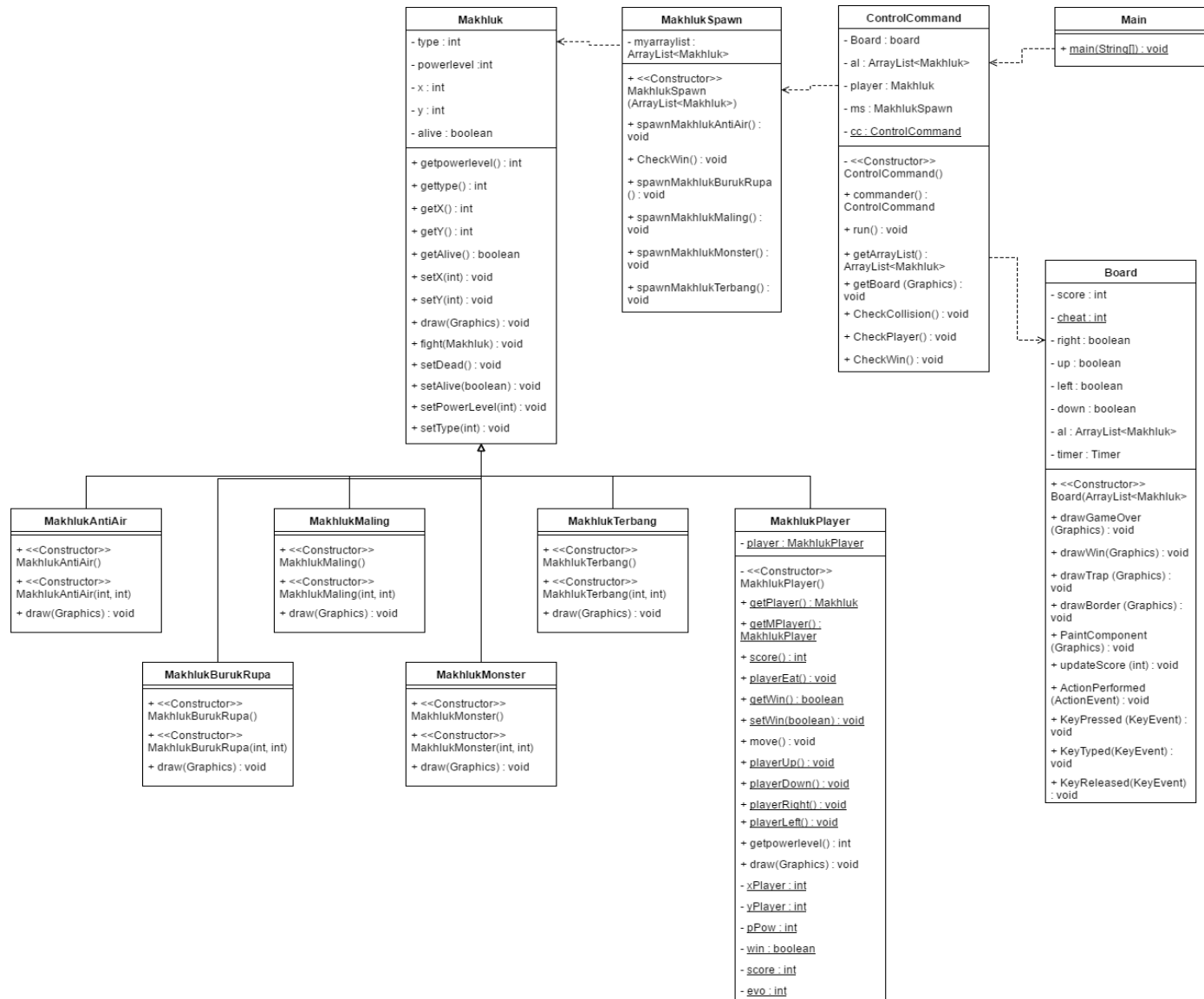
Terdapat junit yang membuktikan bahwa setiap unitnya telah di test dengan baik.

Source code kami telah digenerate pula PMD tanpa adanya error yang signifikan.

3 Diagram Paket



4 Diagram Kelas



4.1 Ulasan Redesign

Terdapat beberapa perbedaan karena kami menggunakan javax.swing sebagai GUI.

1. Tidak adanya MakhlukPlayer karena pada CPP tidak adanya gamifikasi
2. ControlCommand isinya berubah menyesuaikan javax.swing
3. View.cpp berubah menjadi Board.cpp (import java swing) dimana memiliki fungsi yang sama sebagai view untuk dirender tampilannya namun berisi atribut dan method yang berbeda.
4. Beberapa method dalam Makhluk dihapus karena tidak dibutuhkan lagi dan ada beberapa method yang ditambahkan karena membutuhkan informasi yang sebelumnya tidak di butuhkan.
5. Terdapat kelas baru MakhlukSpawn yang sebagai spawner daripada makhluk.

5 Penggunaan Ulang Kelas dari Tugas-tugas yang lalu

5.1 Penggunaan Ulang kelas CPP

No	Nama Kelas CPP	Nama Method	Deskripsi Perubahan
1.	Makhluk	getpowerlevel()	Tidak ada perubahan
		gettype()	Tidak ada perubahan
		getX()	Tidak ada perubahan
		getY()	Tidak ada perubahan
2.	MakhlukAntiAir	Ctor, ctor dengan parameter	Tidak ada perubahan
3.	MakhlukBurukRupa	Ctor, ctor dengan parameter	Tidak ada perubahan
4.	MakhlukMaling	Ctor, ctor dengan parameter	Tidak ada perubahan
5.	MakhlukMonster	Ctor, ctor dengan parameter	Tidak ada perubahan
6.	MakhlukTerbang	Ctor, ctor dengan parameter	Tidak ada perubahan

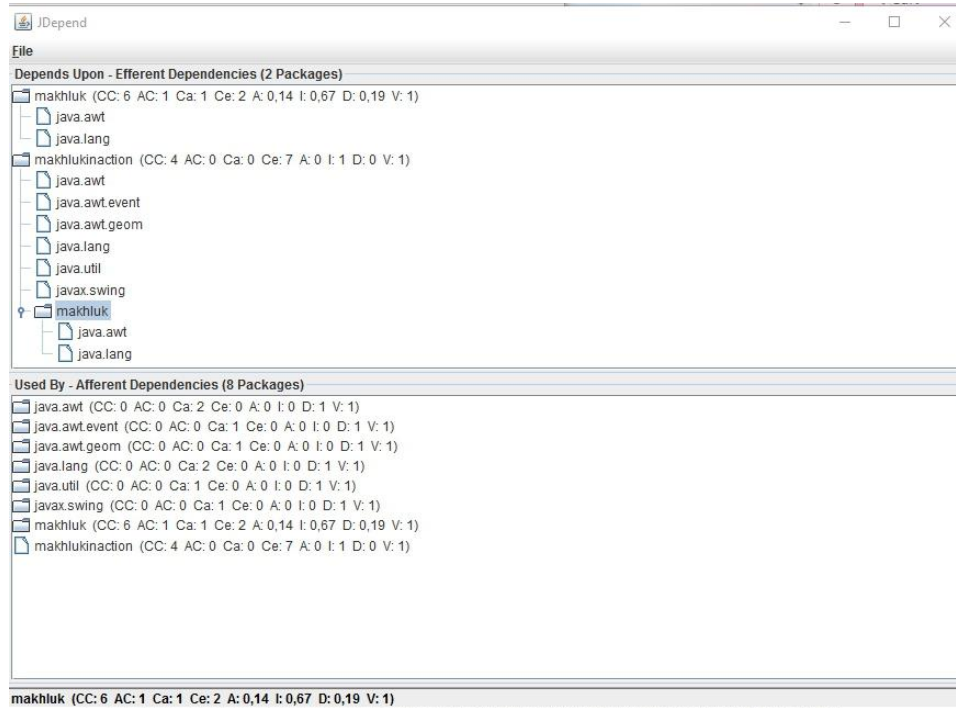
5.2 Penggunaan Ulang kelas Java

No	Package	Nama Kelas JAVA	Nama Method JAVA	Deskripsi Perubahan
1	Makhluk	Makhluk.java	getpowerlevel()	Tidak ada perubahan
			gettype()	Tidak ada perubahan
			getX()	Tidak ada perubahan
			getY()	Tidak ada perubahan
2.	Makhluk	MakhlukAntiAir	Ctor, ctor dengan parameter	Tidak ada perubahan
3.	Makhluk	MakhlukBurukRupa	Ctor, ctor dengan parameter	Tidak ada perubahan
4.	Makhluk	MakhlukMaling	Ctor, ctor dengan parameter	Tidak ada perubahan
5.	Makhluk	MakhlukMonster	Ctor, ctor dengan parameter	Tidak ada perubahan
6.	Makhluk	MakhlukTerbang	Ctor, ctor dengan parameter	Tidak ada perubahan

6 Pengukuran Metriks Aplikasi

Berikut ini adalah ringkasan dari pengukuran metriks dengan JDepend :

IF2210 – Tugas Besar 3 Aplikasi desktop dalam bahasa JAVA



Ulasan : Source Code kami telah dapat digenerate oleh Jdepend dan dapat dikeluarkan metriksnya

7 Pengukuran Kualitas Kode Aplikasi

Ulasan : Secara keseluruhan kualitas kode aplikasi kami sangat baik, terlihat dari hasil generate *CheckStyle*, dikarenakan kami menggunakan *plug-in CheckStyle* secara langsung. Hanya terdapat 2 kelas yang *CheckStyle* - nya error yaitu pada kelas *ControlCommand* (4 line) dan *Main* (1 line) dikarenakan pada kelas *ControlCommand* kami tidak dapat membuat final dan ada beberapa atribut static yang dianggap *out of order* oleh *CheckStyle* dan pada kelas *Main* karena menurut *CheckStyle* seharusnya kelas *Main* yang hanya berisi driver tidak seharusnya dibuat (tidak ada atribut atau method). Selebihnya tidak ada error pada *CheckStyle*. Pada hasil *generate PMD* pun tidak ada error yang keluar sama sekali hal tersebut menandakan bahwa kode kami sudah sangat bersih.

8 Dokumentasi Rinci Kelas

Dokumentasi Rinci Kelas adalah hasil generation JAVADoc .

Tidak perlu diprint. Setor dalam file terpisah (softcopy) ke Oddyseus