

Rancang Bangun Collectible Card Game “Magic Masters” Berbasis Android (Front-End)

Marcello Malvin, Teknik Informatika, Universitas Ciputra, UC Town, Citraland, Surabaya 60219 *

ABSTRAK

Collectible Card Game adalah sebuah permainan yang menggunakan kartu yang bisa dikoleksi. Saat ini *Collectible Card Game Digital* semakin banyak peminatnya. Tetapi, jumlah *Collectible Card Game* pada *Android* sedikit. Oleh karena itu, perlu dibuat sebuah *Collectible Card Game* digital pada *Android*. *Collectible Card Game* dibuat dengan menggunakan *libGDX*. Bahasa yang digunakan dalam pemrograman adalah bahasa *Java*. Metodologi yang digunakan dalam pembuatan game ini adalah metode *waterfall*. Koneksi dibuat dengan menggunakan *Kryonet*. Database dibuat dengan menggunakan *MySQL*. Game ini akan digunakan untuk bermain antara pemain dengan menggunakan sebuah jaringan sebagai media penghubung kedua pemain tersebut. Dengan adanya *Collectible Card Game “Magic Masters” Berbasis Android* ini pemain akan mendapatkan opsi baru untuk bermain *Collectible Card Game* secara *digital*.

Kata kunci: Game, Android, Aplikasi, Collectible card game, smartphone.

1. Pendahuluan

1.1. Latar Belakang

Collectible Card Game adalah sebuah permainan yang menggunakan kartu-kartu kolektibel sebagai sarana permainan antar pemain. *Collectible Card Game* biasa dibuat untuk pemain yang memiliki hobi bermain permainan-permainan strategis secara kompetitif melawan pemain lain. *Collectible Card Game* juga ditujukan untuk orang-orang yang suka untuk mengoleksi kartu-kartu yang ada. Semua pemain memiliki setumpuk kartu yang dapat digunakan untuk bermain. Setumpuk kartu tersebut biasa disebut *deck*. *Deck* tersebut dibuat dengan menggunakan kartu-kartu yang dimiliki oleh pemain. Pemain dapat mengubah isi dari *deck* milik mereka dengan menggunakan kartu yang diperoleh dari sebuah *booster pack*. *Booster Pack* adalah sebuah kemasan yang berisi 5 hingga 15 kartu secara acak yang dapat digunakan pemain untuk memperkuat *deck* yang mereka miliki.

Collectible Card Game memiliki 2 jenis besar, yaitu *Collectible Card Game* konvensional dan *Collectible Card Game* digital. *Collectible Card Game* konvensional maksudnya adalah *Collectible Card Game* yang menggunakan sarana kartu fisik yang bisa dipegang secara langsung oleh pemain. Contoh dari *Collectible Card Game* konvensional adalah *Magic: the Gathering*, *Pokémon Trading Card Game*, *Cardfight!! Vanguard*, *Yu-Gi-Oh!*, dan *World of Warcraft Trading Card Game*.

Collectible Card Game digital adalah *Collectible Card Game* yang tidak menggunakan kartu fisik yang bisa dipegang secara langsung. Pemain bermain dengan

menggunakan kartu-kartu yang ada di dalam sebuah *program game*. *Collectible Card Game* digital biasa dimainkan dalam sebuah komputer ataupun *device-device* yang lain. Contoh dari *Collectible Card Game* digital adalah *Magic Online*, *Magic: the Gathering 2015: Duels of the Planeswalkers*, *Hearthstone*, *Order and Chaos Duels*, dan *Shadow Era*. *Collectible Card Game* digital yang paling banyak dimainkan, *Hearthstone*, pada awal tahun 2015 memiliki 25 juta pemain. *Collectible Card Game* digital memiliki banyak kelebihan dibandingkan dengan *Collectible Card Game* konvensional. Pertama-tama, *Collectible Card Game* konvensional membutuhkan pemain untuk berkumpul pada suatu tempat yang sama, sedangkan *Collectible Card Game* digital memungkinkan permainan dengan menggunakan internet. Kedua, *Collectible Card Game* secara digital menawarkan kemudahan kepada pemain untuk menyimpan kartu yang dimiliki pemain sehingga kartu milik pemain tidak dapat hilang atau rusak. Menurut survey yang dilakukan pada tahun 2013, diestimasikan bahwa jumlah pemain *Collectible Card Game* digital akan meningkat setiap tahunnya.

Android adalah salah satu *Operating System* yang ada pada *mobile device* seperti *smartphone* dan *tablet*. *Smartphone* dan *tablet* *Android*, selain digunakan sebagai media komunikasi, juga dapat digunakan sebagai media untuk bermain *game*. Sudah banyak sekali *game* yang dibuat untuk *Android*. Pengguna *Android* sekarang sudah sangat banyak,

Berdasarkan tren dan oportunitas di atas, ada sebuah peluang bisnis yang cukup baik dalam pembuatan *Collectible Card Game* berbasis *Android*. Oleh karena itu,

* Marcello Malvin. tlp : 031-745 1699, fax : 031-745 1698,
e-mail : mmalvin@student.ciputra.ac.id

perlu dibuat sebuah *Collectible Card Game* berbasis Android yang baru.

1.2. Rumusan Masalah

Bagaimana cara merancang dan membangun *Collectible Card Game* “*Magic Masters*” berbasis Android untuk sisi *Front End*?

1.3. Ruang Lingkup

Software yang dijual dalam tugas akhir ini adalah sebuah *software Front-End* buatan sendiri yang dibuat dengan menggunakan Java. *Software* yang dibuat akan memiliki beberapa fitur sebagai berikut:

- *Login* : Fitur untuk masuk ke dalam permainan dengan menggunakan *account* yang ada.
- *Register* : Fitur untuk membuat *account* baru untuk masuk ke dalam permainan.
- *Main Menu* : Sebuah fitur untuk mengakses segala fitur yang lain.
- *Option* : Sebuah menu untuk mengubah *setting* dari game.
- *Tutorial* : Sebuah pengajaran game untuk pemula.
- *Collection* : Sebuah fitur untuk melihat semua kartu yang dimiliki.
- *Store* : Sebuah fitur untuk membeli kartu baru yang ada di dalam *Booster Pack*.
- *Deck* : Sebuah fitur untuk membuat, *edit*, melihat, menghapus, dan mengganti nama dari sebuah *deck*.
- *Ranking List* : Sebuah fitur untuk melihat *daftar* dari pemain-pemain dengan *ranking* tertinggi.
- *Play* : Seluruh fitur untuk bermain, mulai dari memilih musuh sampai *gameplay* dari permainan ini.
- *Tournament* : Fitur untuk mengikuti sebuah turnamen yang diadakan.

Penelitian ini akan dikerjakan oleh dua peneliti. Peneliti akan melaksanakan bagian-bagian berikut :

- *Gameplay Programming*
- *Front End User Interface*
- *Main Menu*
- *Collection*
- *Front End Login*
- *Front End Register*
- *Tutorial*
- *Option*
- *Ranking List*
- *Tournament*

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat *front end Collectible Card Game* “*Magic Masters*”.

1.5. Manfaat Penelitian

Manfaat yang dapat ditimbulkan dari hasil tugas akhir ini adalah membuat *game* yang dapat dimainkan di android.

Pengguna mendapatkan *game* baru di android yang berupa *Collectible Card Game* yang mirip seperti game konvensional yang masih sedikit jumlahnya.

1.6. Metode Penelitian

Metodologi yang digunakan dalam pengerjaan tugas akhir ini menggunakan teori Software Development Life Cycle (SDLC). SDLC terbagi menjadi langkah-langkah sebagai berikut:

- **Project Planning**
Dalam langkah ini, peneliti akan mencari tahu bagaimana cara membuat sebuah *Collectible Card Game*, serta alasan penting mengapa peneliti membuat proyek tersebut.
- **System Analysis**
Pada langkah ini, peneliti akan menentukan kebutuhan-kebutuhan yang diperlukan untuk membuat *Collectible Card Game*. Setelah kebutuhan untuk telah ditentukan, peneliti akan mulai membuat langkah-langkah untuk menyelesaikan proyek tersebut berdasarkan kebutuhan itu.
- **System Design**
Pada langkah ini, peneliti mendesain front-end system yang akan dibuat, yaitu User Interface dan UML yang akan digunakan.
- **Game Implementation**
Pada langkah ini, front-end system yang telah didesain akan dikerjakan.
- **Integration and Testing**
Pada langkah ini, kedua peneliti akan menggabungkan seluruh bagian dari software yang dikembangkan menjadi satu dan diuji untuk mencari kesalahan-kesalahan yang masih ada pada software.
- **Acceptance, Installation, and Deployment**
Pada langkah ini, setelah software yang dikembangkan terbebas dari error dan bug, software dimasukkan ke dalam server dan di-upload ke Google Play untuk dipublikasikan.
- **Maintenance**
Setelah proyek sudah selesai, peneliti melanjutkan pengelolaan software tersebut

2. Landasan Teori

2.1. Collectible Card Game

Collectible Card Game adalah permainan yang menggabungkan *trading card* dengan *gameplay* strategi (Marshall, 2010). *Collectible Card Game* menggunakan kartu sebagai sarana untuk menjalankan alur permainan. *Collectible Card Game* pada umumnya dimainkan dua atau lebih pemain. *Wizards of the Coast* membuat *Magic: the Gathering*, *Collectible Card Game* yang paling pertama, pada tahun 1993 (Williams, 2007).

Dalam setiap *Collectible Card Game*, setiap pemain memiliki sebuah tumpukan kartu yang digunakan untuk

bermain. Tumpukan kartu ini disebut *Deck*. Batasan dari *deck* berbeda-beda pada setiap *Collectible Card Game*. Misalnya, pada *Hearthstone deck* yang digunakan untuk bermain harus memiliki tepat 30 kartu, sedangkan *Magic : the Gathering* memiliki batasan minimal 60 kartu tanpa adanya batas maksimum. Pemain dapat mengubah isi dari *deck* mereka sesuai dengan keinginan pemain. Hal ini menyebabkan setiap pemain dapat memiliki sebuah *deck* yang unik dan berbeda dengan pemain lain.

Pemain membutuhkan cara untuk memperoleh kartu baru untuk mengubah *Deck*. Kartu baru bisa didapatkan pemain dengan cara membeli sebuah *Booster Pack*. *Booster Pack* adalah sebuah kemasan berisi beberapa kartu secara acak. Pemain kemudian bisa memasukkan kartu-kartu itu ke dalam *deck* pemain tersebut. *Booster Pack* terdiri dari 5-15 kartu. Setiap kartu memiliki tingkat kelangkaan masing-masing. Kartu yang kurang langka lebih mudah didapatkan melalui *Booster Pack*, sedangkan kartu yang lebih langka lebih mudah didapatkan melalui *Booster Pack*.

2.2. Android

Android adalah sebuah Operating System untuk telepon genggam, yang dibuat oleh Google (Todd, 2014). Nama lengkap dari *Android* adalah Android Open Source Project (Hildenbrand, 2015). Android terdapat dalam jutaan ponsel dan perangkat genggam lainnya, menjadikan Android sebagai platform utama untuk para developer-developer aplikasi.

Android sebagai sebuah system, adalah sebuah operating system berdasarkan Java Language Programming yang berjalan pada kernel Linux 2.6 System tersebut sangat ringat dan penuh dengan fitur. Aplikasi Android dikembangkan dengan menggunakan Java Language Programming dan bisa diubah ke platform lainnya dengan cukup mudah. Fitur-fitur lain dari Android termasuk accelerated 3D graphics engine (berdasarkan support dari hardware), database support dengan menggunakan SQLite dan sebuah web browser yang telah diintegrasikan.

2.3. LibGDX

LibGDX adalah sebuah game development framework yang memberikan 1 API yang bisa digunakan pada semua platform yang disupport ("libgdx," 2013). LibGDX mensupport banyak platform, yaitu Windows, Linux, Mac OS X, Android (2.2+), BlackBerry, iOS, Java Applet, dan Javascript/WebGL. Pembuatan program pada LibGDX menggunakan bahasa Java. Untuk menjalankan fungsi ini, LibGDX menggunakan banyak third-party library, yaitu OpenGL, FreeType, mpg123, xiph, Surina, Box2D, LWJGL, OpenAL, dan KissFFT.

LibGDX memungkinkan penulis program hanya satu kali untuk memasukkannya ke dalam berbagai macam platform. Hal ini memungkinkan pembuatan game di desktop tanpa harus menunggu pemasangan program di android. Hasilnya, waktu pemrograman dapat sangat dipersingkat.

Pada *libGDX*, *Stage* adalah sebuah *2D scene graph* yang mengandung sebuah hierarki dari *Actor-Actor* ("*Stage* (libgdx API)," n.d.). *Stage* berfungsi untuk mengendalikan *viewport* dan membagi-bagi *input* kepada *Actor*.

Sebuah *stage* harus menerima *input* yang ada dan mendistribusikan *input* itu kepada *Actor*. Hal ini dilakukan dengan cara menggunakan `Gdx.input.setInputProcessor`.

Actor adalah sebuah *node* dari sebuah *2d scene graph* ("*Actor* (libgdx API)," n.d.). Sebuah *Actor* memiliki posisi, ukuran secara *rectangular*, *origin*, skala, rotasi, *Z index*, dan warna. Posisi yang digunakan adalah posisi *x* dan *y* dari sudut kiri bawah *Actor*. *Actor* bisa menerima *input* sederhana dengan menggunakan *inputListener*. *Actor* juga bisa menerima *input* yang lebih kompleks seperti *ClickListener* atau *ActorGestureListener*.



Gambar 1. Layout Game

3. Desain Sistem

3.1. Aturan Permainan

Permainan dilakukan oleh 2 orang pemain. Setiap pemain memiliki satu *Hero*, empat *Minion*, dan satu *Deck* yang berisi kartu *Spell*. Pemain juga memiliki sebuah *resource* yang disebut *Mana*. Pemain akan memenangkan pertandingan jika *Hero* milik lawan dikalahkan. Pada layar pemain, Bagian milik pemain akan ditampilkan di bawah layar, dan bagian milik lawan akan ditampilkan di atas layar. Layout yang digunakan bisa dilihat pada Gambar 1.

Pada awal permainan, pemain akan mendapatkan empat kartu secara acak dari *deck*. Kartu-kartu ini disebut kartu tangan pemain. Permainan dilakukan secara bergiliran. Pada setiap giliran pemain, pemain akan mendapatkan satu kartu lagi dari *deck*. Kemudian pemain dapat melakukan 2 hal, yaitu menyerang dengan menggunakan *Hero* atau *Minion* dan menggunakan kartu *Spell* dari kartu tangan.

Kartu *Hero* dan *Minion* memiliki dua *property*. *Property* yang pertama adalah serangan *Hero* dan *Minion*. *Property* yang kedua adalah besarnya nyawa *Hero* dan *Minion*. *Hero* atau *Minion* tersebut mati jika nyawa tersebut mencapai 0.. Setiap *Hero* dan *Minion* hanya dapat menyerang sekali setiap giliran.

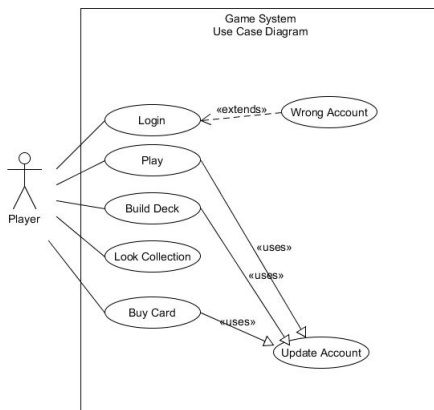
Untuk mempengaruhi keadaan permainan pemain menggunakan kartu *Spell* dari kartu tangan pemain. Setiap

Spell memiliki efek yang berbeda-beda. Contoh dari efek yang ada misalnya memberi damage pada sebuah *Minion* atau memberi tambahan serangan pada sebuah *Hero*. Pemain menggunakan kartu ini dengan membayar *Mana* sesuai yang dibutuhkan masing-masing kartu *Spell*.

Mana berfungsi sebagai *resource* untuk membayar sebuah *Spell*. Pada awal permainan, jumlah maksimum *Mana* yang dimiliki pemain adalah 0. Setiap awal giliran, jumlah maksimum tersebut akan bertambah satu. Setelah itu, pemain akan mendapatkan *Mana* sampai mencapai batas maksimum itu. Pemain dapat menggunakan *Spell* sebanyak-banyaknya selama jumlah *Mana* yang dimiliki masih mencukupi untuk menggunakan *Spell*.

3.2. Use Case Diagram

Gambar 2 menjelaskan *Use Case Diagram* dari permainan yang akan dilakukan oleh pemain. Pemain dapat melakukan 3 hal dalam *game*, yaitu *login*, membeli kartu, melihat kartu yang dimiliki, membuat *deck*, dan bermain melawan pemain lain. Jika perlu, maka *game* akan *update account* yang dimiliki pemain agar sesuai dengan yang dilakukan pemain.



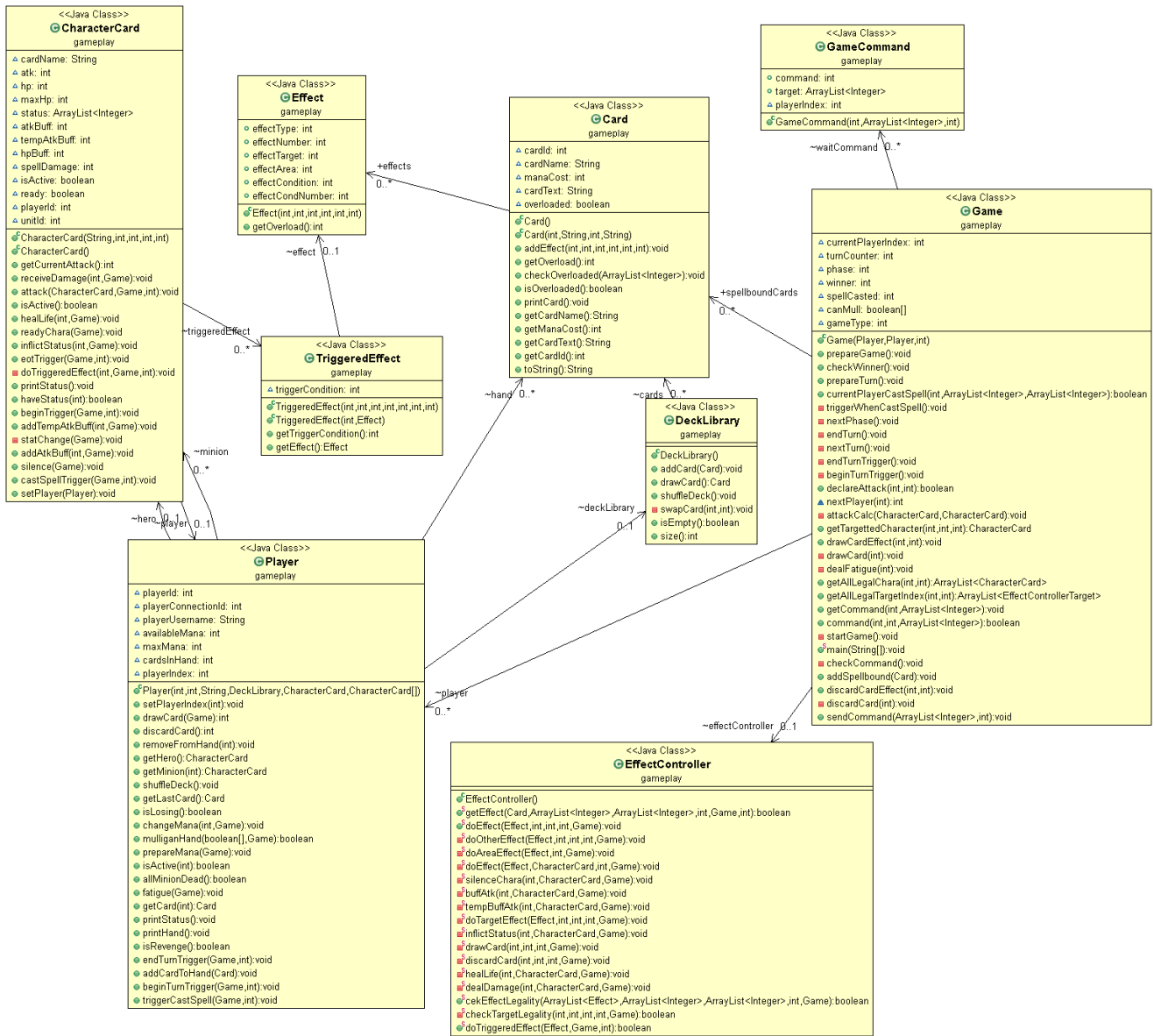
Gambar. 2. Use Case Diagram

3.3. Class Diagram

Gambar 3 adalah diagram dari *class-class* yang akan digunakan pada *Game Server*. *Class-class* ini berfungsi untuk menjalankan sebuah *game* yang akan dimainkan oleh pemain. *Class* utama yang akan digunakan untuk membuat permainan adalah *class Game*. *Class Game* berfungsi sebagai media dasar untuk kedua pemain bermain. *Game* akan mengandung 2 *class Player*. *Class Player* berfungsi untuk melambungkan masing-masing pemain. Masing-masing *Player* akan memiliki satu *DeckLibrary* dan 5 *CharacterCard*. *Player* juga memiliki kartu tangan *Class DeckLibrary* digunakan untuk melambungkan *deck* masing-masing pemain dengan menggunakan *ArrayList<Card>*. Masing-masing *deck* memiliki 25 kartu yang diacak secara *random* dan diletakkan ke kanan. Kartu-kartu yang dimiliki pemain akan dilambungkan dengan 2 *class*, yaitu *CharacterCard* untuk kartu karakter dan *Card* untuk melambungkan kartu yang lain. *Card* memiliki beberapa *Effect* untuk membuat kartu bisa berjalan, sedangkan *CharacterCard* bisa memiliki *TriggeredEffect*.

Class Effect dan *TriggeredEffect* digunakan untuk melambungkan efek dari masing-masing kartu dan kartu karakter.

EffectController berfungsi untuk mengendalikan efek-efek tersebut. *EffectController* tidak memiliki *property* tertentu. *Class-class* memanggil *EffectController* untuk menerjemahkan data dari *Effect* menjadi suatu *action* pada *game*

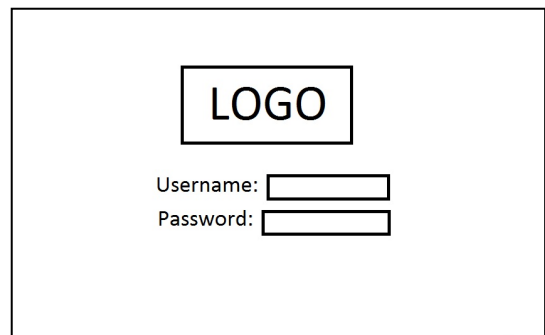


Gambar 3. Class Diagram

3.4. User Interface

Berikut akan diperlihatkan gambar-gambar UI *Mid Fidelity* dari game yang akan dibuat. Game akan dibuat secara *landscape*. Ini dipilih agar pemain bisa melihat lebih nyaman ketika sedang bermain.

Gambar 4 adalah gambar dari UI halaman *Login* dalam bentuk *Mid Fidelity*. Halaman ini hanya dimunculkan ketika pertama kali bermain atau pemain memutuskan untuk logout. Jika pemain belum memiliki *account*, maka pemain bisa menggunakan tombol *register*. Pemain akan diarahkan ke *website* untuk membuat *account*.

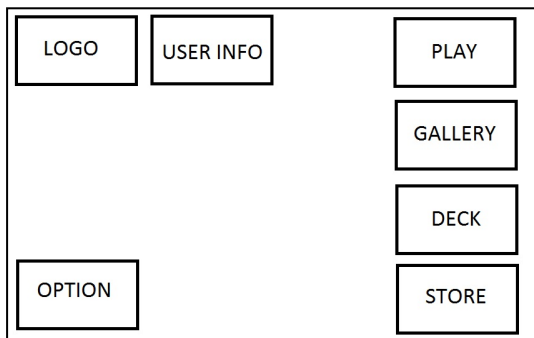


Gambar 4. UI Login Screen

Pada Gambar 5 diperlihatkan desain dari *Main Menu* yang akan dibuat. *Main Menu* yang dibuat berisi tombol-tombol yang menuju halaman-halaman yang lain. Selain

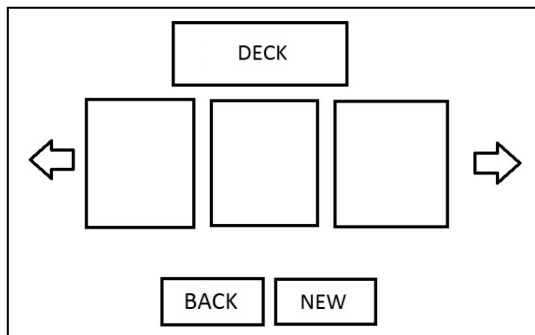
* Marcello Malvin. tlp : 031-745 1699, fax : 031-745 1698, e-mail : mmalvin@student.ciputra.ac.id

itu, *Main Menu* juga menampilkan informasi penting tentang pemain.



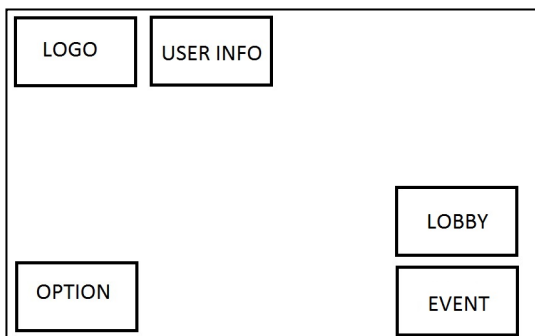
Gambar. 5. *UI Main Menu Screen*

Pada Gambar 6 diperlihatkan desain dari halaman *Deck*. Dalam halaman ini pemain bisa memilih *deck* yang ia gunakan dalam permainan. *Deck* yang ditampilkan pada halaman ini ada 3. Pemain bisa menggeser *deck* pada tampilan untuk melihat *deck* yang lain. Untuk melambungkan *deck* ditampilkan kartu penting dari *deck* yang dipilih.

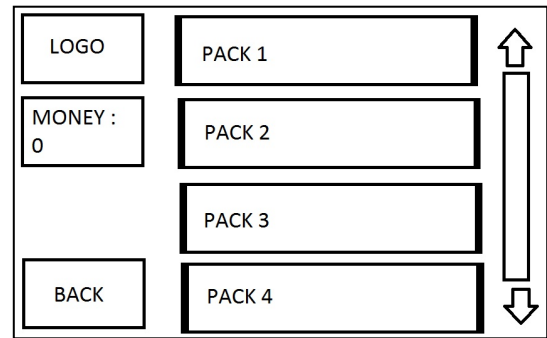


Gambar. 6. *UI Deck Screen*

Jika pemain menekan tombol *Play* pada *Main Menu*, maka pemain akan masuk ke dalam halaman yang pertama. Halaman pertama ini terlihat pada Gambar 7. Pemain akan bergerak ke halaman kedua apabila pemain menekan tombol *Lobby*. Pada kedua halaman, pemain menggunakan bagian kanan halaman untuk navigasi. Bagian atas digunakan untuk informasi, dan tombol untuk kembali ke halaman sebelumnya berada di bagian bawah kiri halaman.

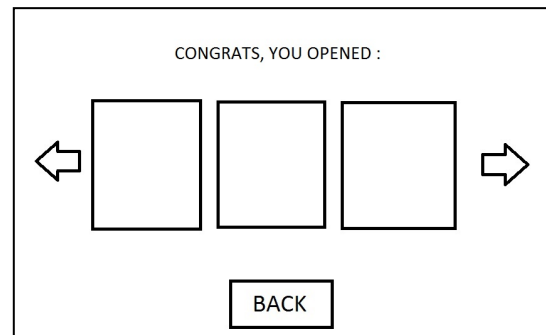


Gambar. 7. *UI Play Screen*



Gambar. 8. *UI Shop Screen*

Gambar 8 memperlihatkan desain UI untuk *Shop*. Pada UI *Shop*, diperlihatkan barang-barang yang dijual dan dapat dibeli dengan menggunakan uang yang dimiliki pemain. Pemain bisa mendapatkan informasi tentang barang yang mereka pilih dengan menekan barang yang diinginkan. Jika pemain mau membeli, maka akan ditampilkan konfirmasi. Jika pemain menekan ya, maka barang akan dibeli.



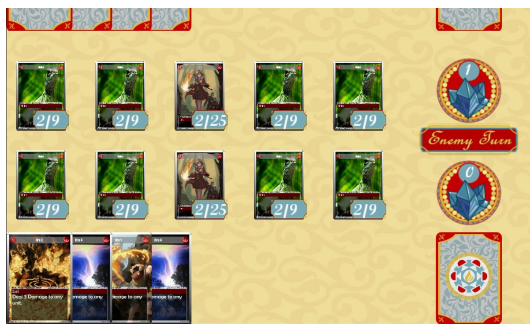
Gambar. 9. *UI Shop Screen 2*

Gambar 9 menunjukkan desain UI dari halaman yang diperlihatkan setelah membeli. Setelah membeli, maka kartu-kartu yang didapatkan ditampilkan di sini. Pemain bisa melihat kartu secara lebih detail dengan menekan kartu. Tombol *back* digunakan untuk kembali ke halaman *shop*.

4. Hasil dan Pembahasan

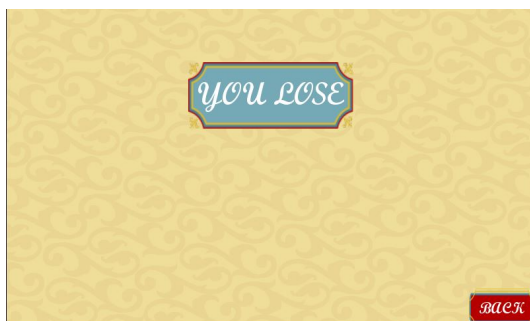
Aplikasi yang telah dibuat diuji coba oleh tim penguji, untuk mengetahui fungsionalitas, dan ada tidaknya bug yang muncul pada software ini. Berikut adalah hasil dan screenshot dari uji coba tersebut.

4.1. Main Game



Gambar. 10. Main Game Screen

Main Game dilakukan dengan bermain 2 orang. Pemain bermain dengan menggunakan screen seperti pada Gambar 10. Dalam uji coba ini, pemain bisa menggunakan kartu yang pemain miliki dengan double tap pada pada hand. Pemain juga bisa menyerang dengan menge tap karakter. Untuk berganti giliran pemain bisa menggunakan button di kanan. Permainan berlangsung hingga salah satu hero pemain mencapai hp 0. Jika demikian, maka screen akan berganti ke resultScreen seperti Gambar 11.



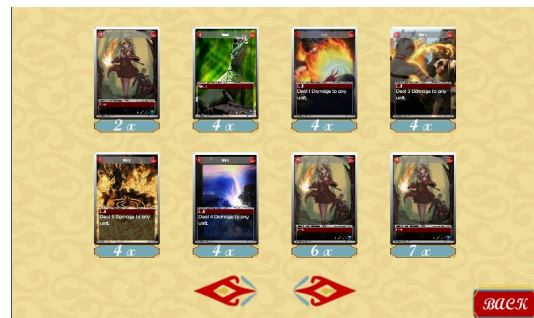
Gambar. 11. Result Screen

Proses permainan dari *game* ini dikendalikan dengan menggunakan sebuah class *Game*. Method yang pertama adalah *prepareGame* yang memulai game. Setelah itu, *game* akan masuk ke dalam *method prepareTurn* yang berguna untuk memulai giliran dari salah satu pemain. Setelah itu, *game* menerima perintah-perintah dari pemain. Perintah-perintah tersebut diterjemahkan ke dalam beberapa *method*. *Method currentPlayerCastSpell* digunakan untuk menggunakan kartu yang ada di tangan pemain. *Method declareAttack* digunakan untuk menyerang menggunakan kartu karakter. *Method endTurn* digunakan untuk mengakhiri giliran. *Program* akan mengulangi giliran setiap pemain sampai ada salah satu pemain yang kalah. Pengecekan ini digunakan dengan menggunakan *checkWinner*. Jika sudah, maka *program* akan mengakhiri *game*.

4.2. Collection

CollectionScreen digunakan untuk melihat Collection yang dimiliki pemain. CollectionScreen ini dibuat seperti pada Gambar 12. Dalam screen ini bisa dilihat bahwa

screen memperlihatkan kartu-kartu yang dimiliki dalam kelipatan 8. Jumlah kartu diperlihatkan di bawah masing-masing kartu. Tombol kiri dan kanan bisa digunakan untuk browse kartu yang lain.



Gambar. 12. Collection Screen

4.3. Tournament & Ranking List

Tournament dijalankan pada TournamentScreen. Ranking List juga diperlihatkan pada Tournament Screen. TournamentScreen ini dibuat seperti pada Gambar 13. Di sini diperlihatkan top 8 pemain yang memiliki poin paling banyak dari semua pemain. Poin pemain sendiri dicantumkan di bawah, di sebelah tombol play. Jika menekan button play, maka akan diarahkan ke pada screen yang sama apa bila bermain biasa, akan tetapi di server akan diletakkan di tempat yang berbeda dan ditemukan dengan orang yang memiliki jumlah poin yang mirip.



Gambar. 13. Tournament Screen

5. Kesimpulan dan Saran

5.1. Kesimpulan

Dengan selesainya tugas akhir “Rancang Bangun Collectible Card Game “Magic Masters” berbasis Android (Front-end)”, dapat disimpulkan bahwa:

- Semua fitur-fitur pada aplikasi ini (dari sisi *front-end*) dapat berkomunikasi dengan *server* dan berjalan dengan baik.
- Aplikasi dapat berkomunikasi dengan server secara secara lokal (*intranet*).

5.2. Saran

- Dalam game yang dibuat gunakan *Drawable*. *Drawable* adalah fitur Android yang memungkinkan penggunaan

gambar dengan ukuran yang sesuai dengan ukuran layar pemain. Ini dibutuhkan agar tampilan sesuai dengan ukuran screen yang dimiliki pemain.

- Fitur tutorial dibuat secara interaktif agar memudahkan pengguna dalam memahami cara bermain game tersebut
- Meningkatkan user experience dalam penggunaan game tersebut untuk meningkatkan tingkat kemudahan dalam menggunakan dan tingkat kepuasan pengguna.

DAFTAR PUSTAKA

Actor (libgdx API). (n.d.). Diperoleh 29 November 2015 dari [https://libgdx.badlogicgames.com/nightlies/docs/api/com/badlogic/gdx/scenes/scene2d/Actor.html](https://libgdx.badlogicgames.com/nightlies/docs/api/com.badlogiclogic/gdx/scenes/scene2d/Actor.html)

Hildenbrand, J. (2015, 16 Mei). What is Android? | Android Central. Diperoleh 29 November 2015 dari <http://www.androidcentral.com/what-android>

libgdx. (2013). Diperoleh 20 Maret 2015 dari <http://libgdx.badlogicgames.com/features.html>

Marshall, B. D., Van Dreunen, J., & Wang, M. (2010). Trading Card Game Industry From the T to the C to the G.

Stage (libgdx API). (n.d.). Diperoleh 29 November 2015 dari [https://libgdx.badlogicgames.com/nightlies/docs/api/com/badlogic/gdx/scenes/scene2d/Stage.html](https://libgdx.badlogicgames.com/nightlies/docs/api/com.badlogiclogic/gdx/scenes/scene2d/Stage.html)

Todd, A. (2014, 23 Oktober). What Is Android? What Is An Android Phone? Beginners Guide To Android. Diperoleh 23 Maret 2015 dari https://recombu.com/mobile/article/what-is-android-and-what-is-an-android-phone_M12615.html

Williams, J. P. (2006). Gaming as culture: essays on reality, identity and experience in fantasy games. Jefferson, N.C.: McFarland & Co..