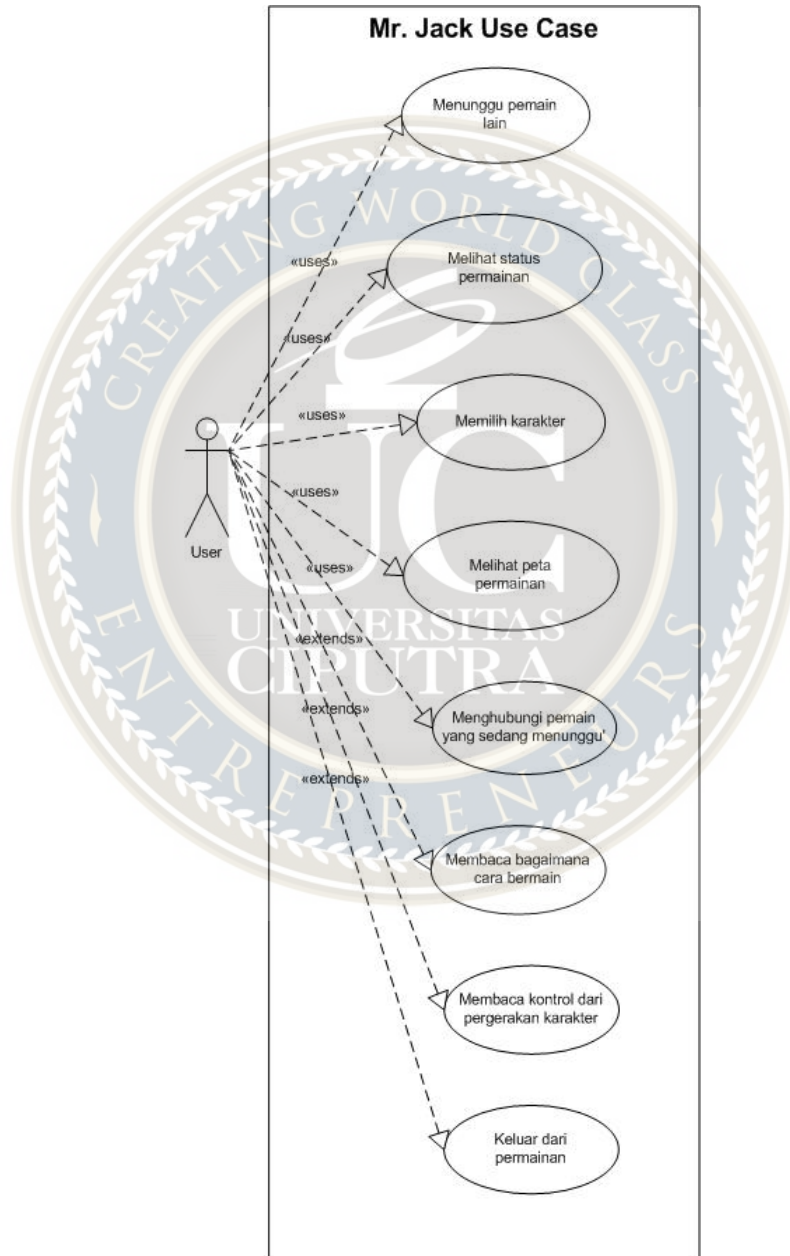


Bab III

Desain Sistem

3.1 Use Case



Gambar 3.1 Gambar Use Case dari pemain

a. Use Case menunggu pemain lain :

Syarat kondisi: User sudah melakukan Bluetooth *pairing*

1. User menjalankan aplikasi
2.
 - i. User memilih koneksi “Listening”
 - ii. User menekan tombol menu
3.
 - i. User menekan tombol start
 - ii. User menekan submenu “Listen for connections”

b. Use Case melihat status permainan :

Syarat kondisi: User sudah terhubung dengan pemain lain

1. User menekan tombol menu
2. User memilih submenu “Game State”

c. Use Case memilih karakter :

Syarat kondisi: User sudah terhubung dengan pemain lain

1. User mendapat giliran untuk memilih
2.
 - i. User menekan tombol navigasi tengah
 - ii. User menekan tombol *enter*.

d. Use Case melihat peta permainan :

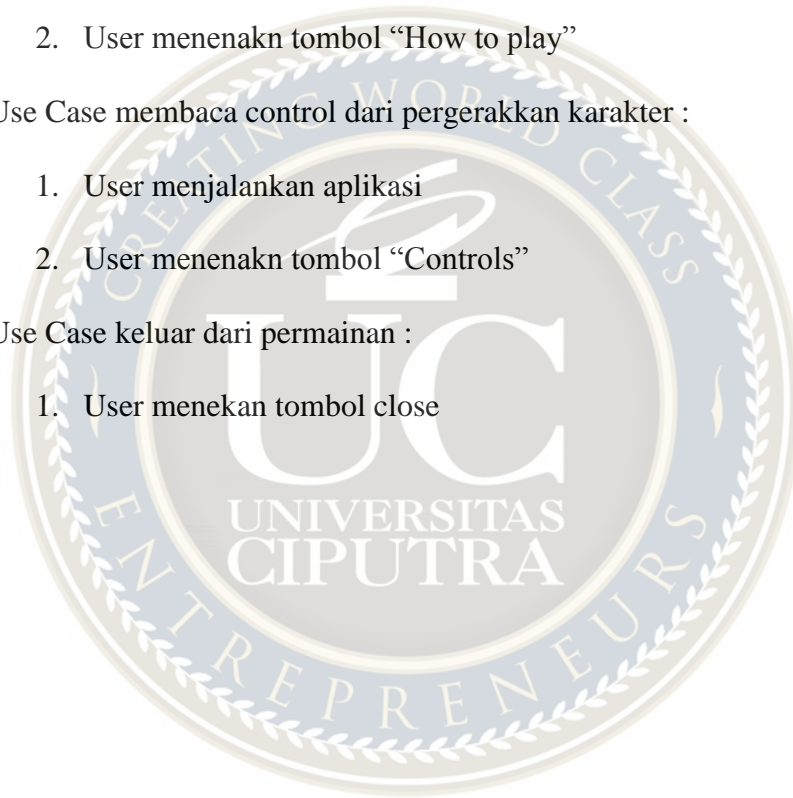
Syarat kondisi: User sudah terhubung dengan pemain lain

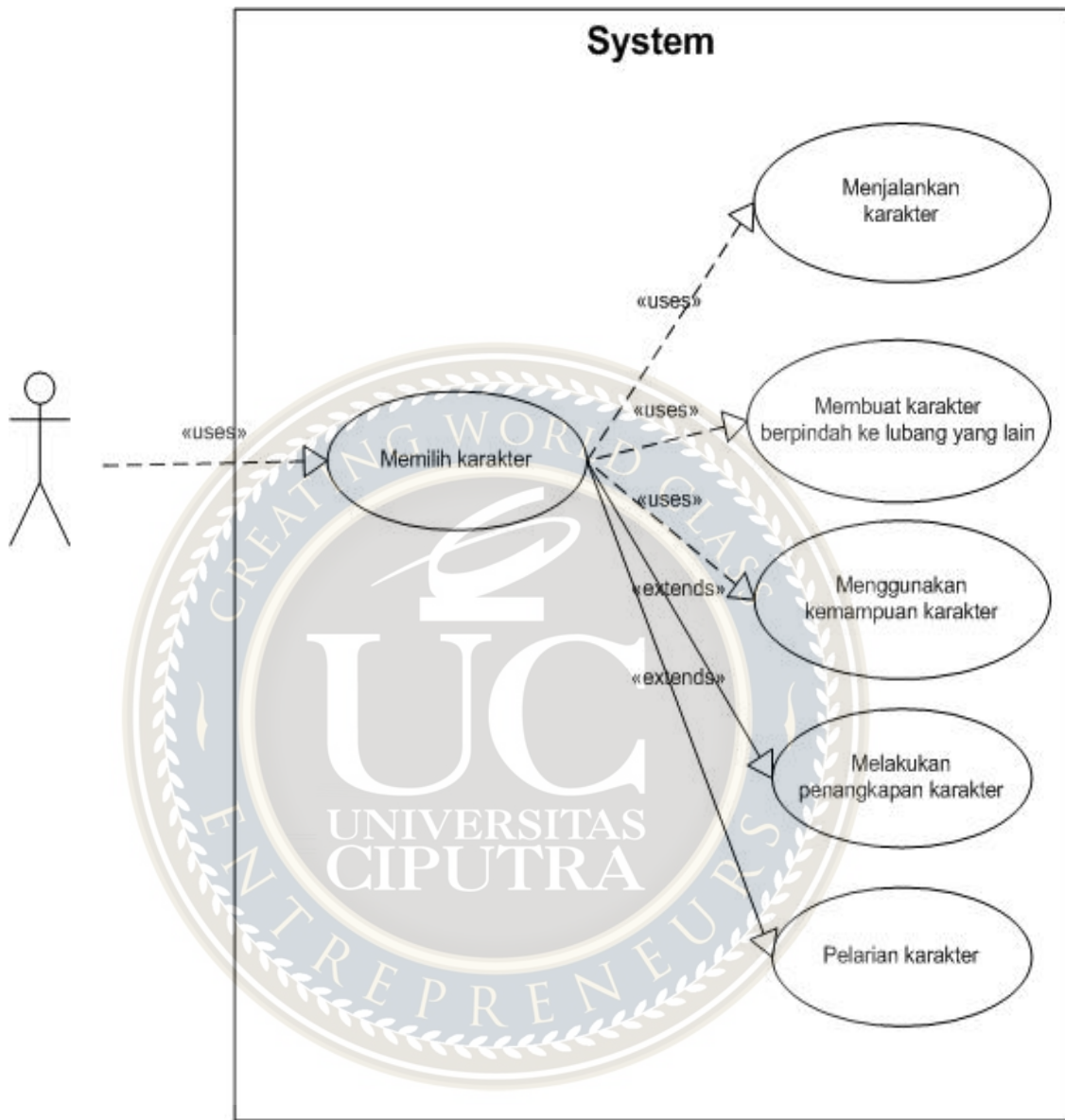
1. User menekan tombol menu
2. User memilih submenu “Game Screen”

e. Use Case menghubungi pemain lain yang sedang menunggu :

Syarat kondisi: User sudah melakukan Bluetooth *pairing*

1. User menjalankan aplikasi
 2. User memilih koneksi
 3. User menekan tombol “New Game”
- f. Use Case membaca bagaimana cara bermain :
1. User menjalankan aplikasi
 2. User menekan tombol “How to play”
- g. Use Case membaca control dari pergerakan karakter :
1. User menjalankan aplikasi
 2. User menekan tombol “Controls”
- h. Use Case keluar dari permainan :
1. User menekan tombol close





Gambar 3.2 Gambar Use case menjalankan karakter

i. Use Case menjalankan karakter

Syarat kondisi: User sudah terhubung dengan pemain lain

1. User memilih salah satu karakter
2. User menekan salah satu tombol untuk menggerakkan karakter

j. Use Case membuat karakter berpindah ke lubang yang lain

Syarat kondisi: User sudah terhubung dengan pemain lain

1. User memilih salah satu karakter
2. User menekan tombol navigasi tengah
3. User menggerakkan arah *trackpad*
4.
 - i. User menyelesaikan pilihan dengan menekan tombol *enter*
 - ii. User menyelesaikan pilihan dengan menekan tombol navigasi tengah

k. Use Case menggunakan kemampuan karakter

Syarat kondisi: User sudah terhubung dengan pemain lain

1. User memilih salah satu karakter
2. User menekan tombol menu
3. User memilih submenu "Use Ability"

l. Use Case melakukan penangkapan karakter (Untuk user yang berperan sebagai polisi)

Syarat kondisi: User sudah terhubung dengan pemain lain

1. User memilih salah satu karakter
2. User menjalankan karakter pada petak yang berisi karakter lain
3. User memilih "Yes" pada pilihan yang ditampilkan

m. Use Case melakukan pelarian karakter (Untuk user yang berperan sebagai Jack)

Syarat kondisi: User sudah terhubung dengan pemain lain

1. User memilih salah satu karakter
2. User menjalankan karakter menuju jalan keluar yang tidak ditutup
3. User memenangkan permainan

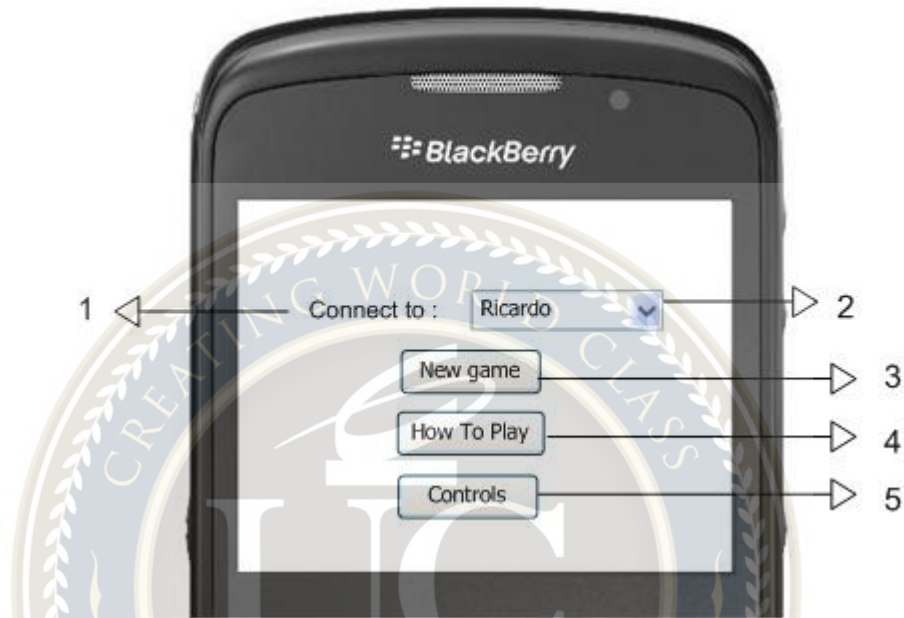
3.2 Daftar Fungsi

Tabel 3.1 *Feature List*

No	Nama Fitur	Jack	Polisi
1	Penjelasan cara bermain	X	X
2	Penjelasan kontrol dari karakter	X	X
3	Memilih karakter	X	X
4	Tampilan keadaan karakter yang dipilih	X	X
5	Tampilan peta permainan	X	X
6	Menggerakkan karakter	X	X
7	Memindah karakter dari satu lubang ke lubang yang lain	X	X
8	Penangkapan karakter		X
9	Pelarian karakter	X	
10	Pemindahan letak lubang	X	X
11	Pemindahan letak penutup jalan keluar	X	X
12	Pemindahan letak lampu	X	X
13	Mendekatkan seorang karakter satu petak dari karakter yang sedang dijalankan	X	X
14	Menembus tembok	X	X
15	Penggerakkan arah lentera	X	X
16	Pengambilan alibi	X	X
17	Menukar letak karakter	X	X
18	Menunggu pemain lain	X	X
19	Memanggil pemain lain	X	X
20	Melihat kondisi terang / gelapnya Jack		X
21	Melihat identitas Jack	X	
22	Melihat daftar alibi	X	X
23	Keluar dari permainan	X	X

3.3 Rancangan Tampilan Aplikasi (UI Mockup)

3.3.1 Start Screen



Gambar 3.3 Mockup Start Screen

StartGameScreen merupakan halaman pertama sewaktu aplikasi dijalankan. Start Screen berguna untuk memilih lawan pada awal permainan, melihat tombol kendali dan cara bermain. Desain StartScreen dapat dilihat pada gambar 3.3.

Keterangan :

1. Label keterangan
2. Sebelum memulai permainan, kedua pemain harus mendengarkan koneksi Bluetooth, melakukan pairing antara kedua *device*, dan membuka aplikasi kembali untuk dapat melakukan koneksi. Hal ini dilakukan karena permainan dilakukan dengan menggunakan koneksi Bluetooth.
3. tombol untuk memulai permainan

4. tombol untuk melihat peraturan permainan
5. tombol untuk melihat kendali karakter

3.3.2 Game States



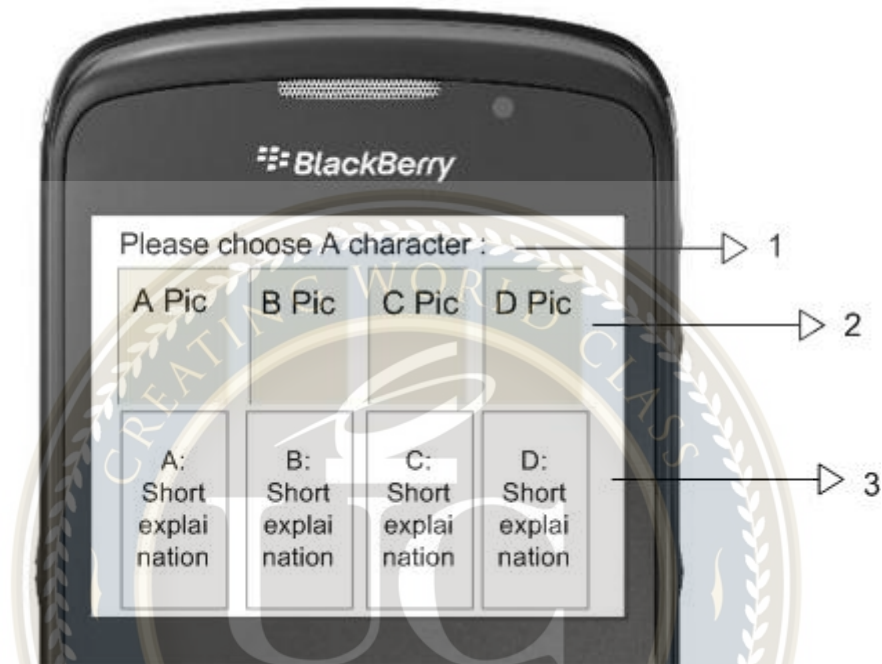
Gambar 3.4 Mockup Game States

Game State Screen merupakan layar untuk membantu pemain mengingat kondisi permainan yang terdiri dari sisa langkah karakter yang telah dipilih, penggunaan kemampuan karakter, gambar dari Jack, dan semua alibi yang didapat oleh pemain selama permainan berlangsung.

Keterangan :

1. Keterangan untuk membantu mengingat sisa langkah yang dapat dilakukan oleh karakter, sebuah karakter mempunyai tiga sampai empat jumlah langkah, namun ada juga karakter yang tidak dapat melangkah jika menggunakan kemampuan khususnya.
2. Keterangan untuk membantu mengingat penggunaan kemampuan karakter, kemampuan karakter bisa digunakan sebelum, sesudah atau sebagai pengganti langkah dari karakter.
3. Pada layer pemain yang memerankan Jack, gambar merupakan alibi dari karakter Jack, pada layer polisi gambar akan berupa kondisi terang dan gelapnya Jack. Daerah terang adalah daerah di sekeliling lampu dan di sekeliling karakter, daerah gelap adalah daerah sisanya. Jack tidak dapat keluar dari peta selama keadaan terang. Kondisi terang dan gelap ini akan berubah pada akhir dari setiap ronde.
4. Gambar dari semua alibi yang didapatkan dengan kemampuan karakter Sherlock

3.3.3 Character Screen



Gambar 3.5 Mockup Character Screen

CharacterScreen merupakan layar untuk pemilihan karakter yang dapat dijalankan pada peta permainan. CharacterScreen akan berubah menjadi gambar 3.5 setelah pemain lawan melakukan hubungan *Bluetooth* dengan alat yang kita gunakan.

Keterangan :

1. Label keterangan agar pemain memilih salah satu karakter
2. Gambar karakter yang dapat dimainkan dalam sebuah ronde. Pada ronde ganjil keempat karakter adalah hasil pengacakan dari delapan karakter yang

ada. Pada ronde genap, karakter yang ditampilkan merupakan sisa dari karakter yang ditampilkan dari ronde ganjil.

3. Penjelasan singkat dari kemampuan karakter yang ditampilkan

3.3.4 Controls Screen



Gambar 3.6 Mockup Controls Screen

Controls Screen merupakan layar yang berisi tombol yang digunakan untuk menggerakkan karakter

Keterangan :

1. Judul dari layar yang bertujuan untuk member keterangan bahwa user sedang berada di layar Controls

2. Penjelasan dari penggerakkan karakter dalam permainan Mr. Jack

3.3.5 How To Play Screen



Gambar 3.7 Mockup How To Play Screen

How To Play Screen merupakan layar yang berisi petunjuk bagaimana peraturan dalam permainan Mr. Jack

Keterangan :

1. Judul dari layar yang bertujuan untuk member keterangan bahwa user sedang berada di layar How To Play
2. Penjelasan dari bagaimana cara bermain dalam permainan Mr. Jack

3.3.6 Game Screen



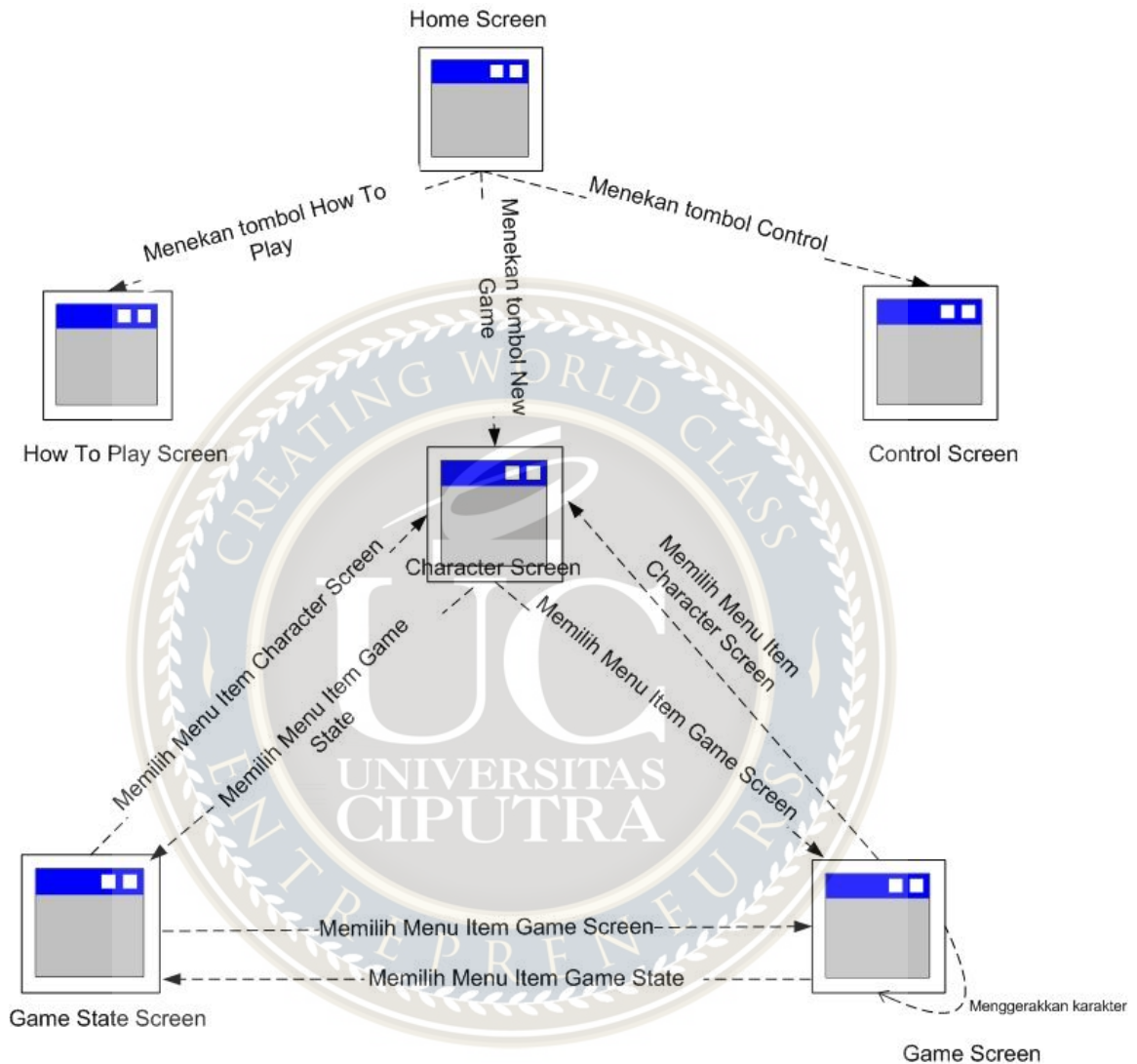
Gambar 3.8 Mockup Game Screen

Game Screen merupakan layar permainan utama, dalam layar ini, karakter yang telah dipilih dapat dijalankan dan digunakan kemampuan khususnya. Dalam layar ini juga segala logika permainan dijalankan.

Keterangan :

Peta dari permainan Mr. Jack yang ditampilkan semirip mungkin dengan versi *boardgame*.

3.4 Navigasi Tampilan Aplikasi (UI Navigation)



Gambar 3.9 UI Navigation

3.5 Arsitektur Aplikasi

Aplikasi permainan Mr. Jack menggunakan arsitektur *Peer to Peer(One Tier)* dengan menggunakan teknologi *Bluetooth*. Dalam melakukan koneksi, salah satu alat akan mendengarkan, dan lainnya mencari koneksi.



Gambar 3.10 Bluetooth Pairing

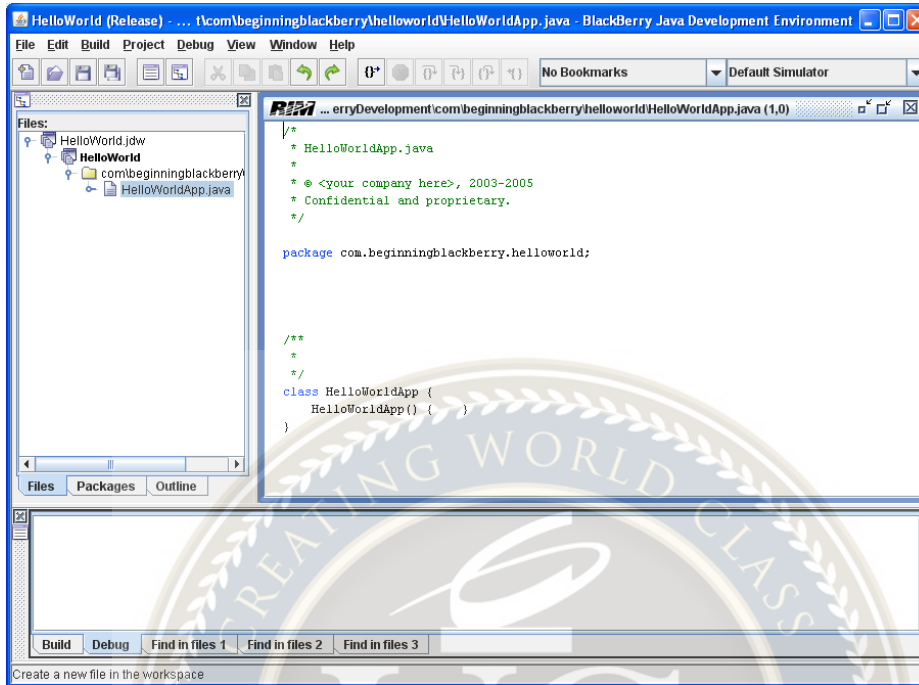
3.5.1 Pemilihan Framework dan Tools

Tabel 3.2 Perbandingan Framework

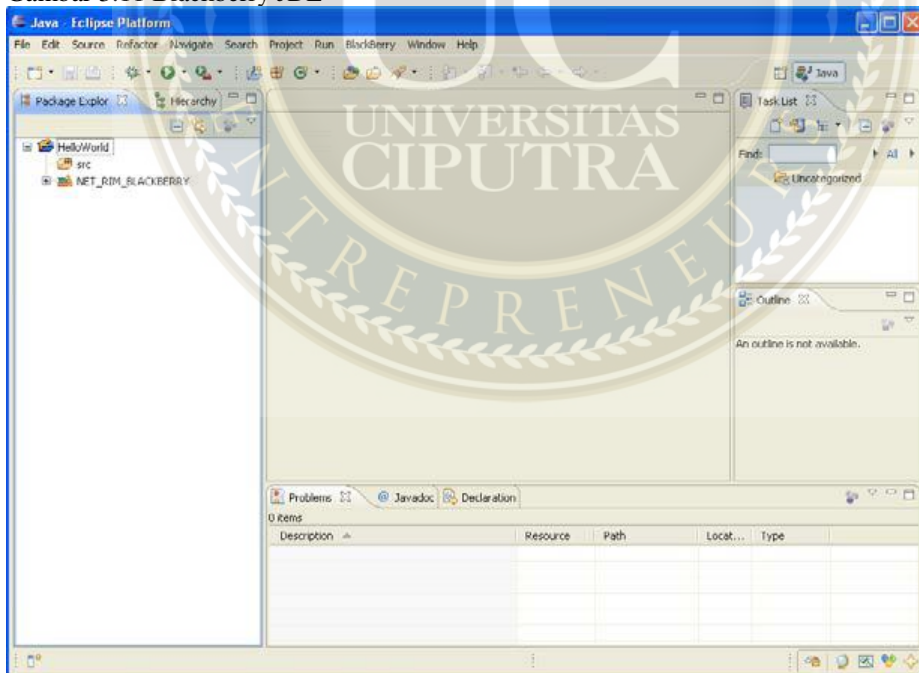
Framework	Perbandingan
Java ME Mobile Information Device Profile (MIDP)	Tidak dapat menggunakan semua bagian dari Blackberry API.
Blackberry API	Mendukung MIDP dan mempunyai API lain yang tidak termasuk MIDP.

Tabel 3.3 Perbandingan Tools

Tools	Perbandingan
Blackberry JDE	<p>menyediakan semua alat – alat yang diperlukan untuk mengembangkan, menguji aplikasi baik dalam simulator, maupun dalam alat aslinya. Mempunyai desain tampilan yang selangkah tertinggal dibandingkan <i>BlackBerry JDE Plug-in untuk Eclipse</i>. Namun tidak perlu melakukan banyak pengaturan sebelum mengembangkan aplikasi</p>
BlackBerry JDE Plug-in untuk Eclipse	<p>Belum lama dikembangkan, sehingga terdapat beberapa hal yang tidak berjalan 100 persen sewaktu – waktu, hanya mendukung JDE dengan versi di atas 4.2.1.</p> <p>Namun, dengan banyak fitur yang tidak terdapat dalam <i>Blackberry JDE</i>, dan dengan tampilan yang lebih menarik.</p>

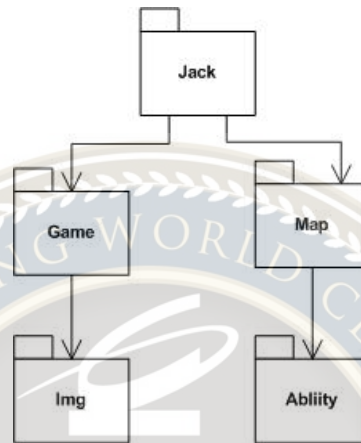


Gambar 3.11 Blackberry JDE



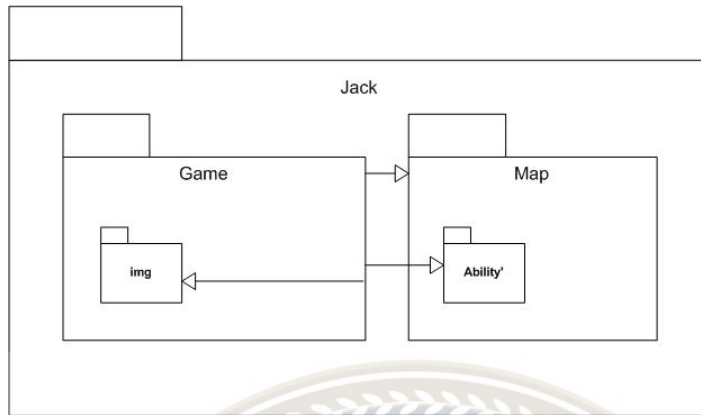
Gambar 3.12 Eclipse JDE

3.6 Diagram Package



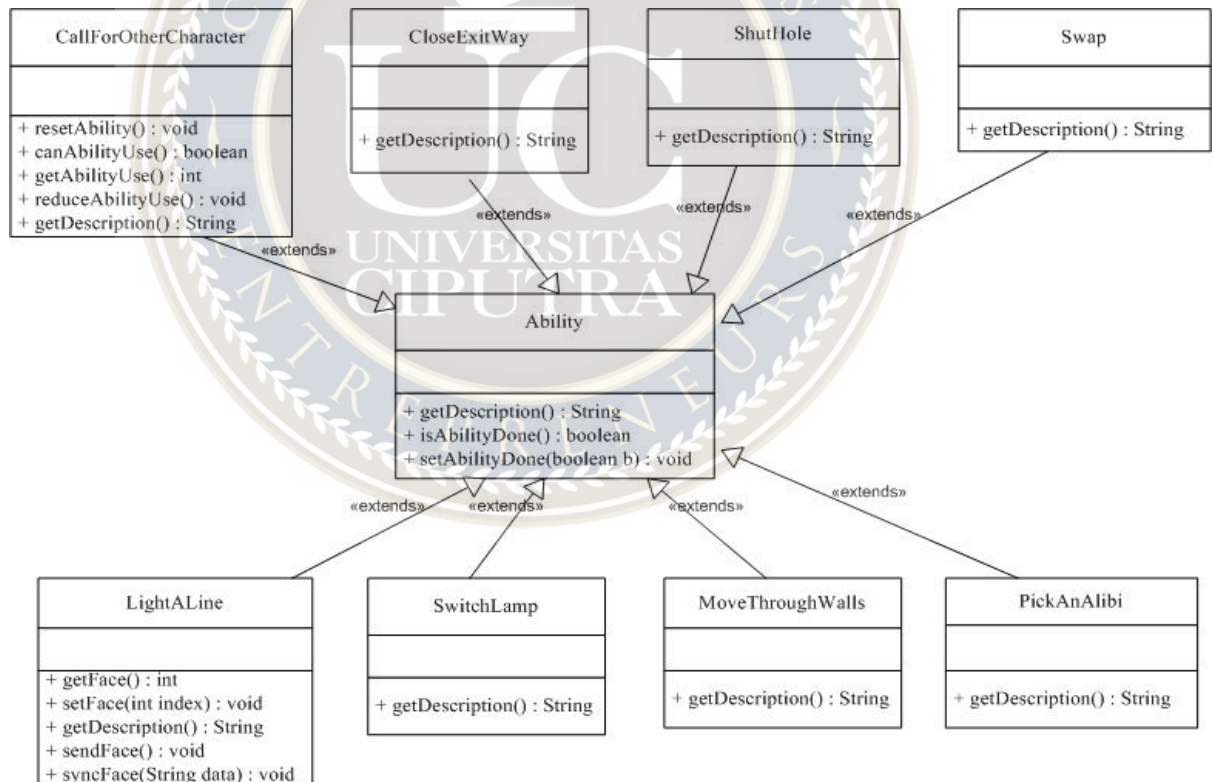
Gambar 3.13 Struktur *Package*

- a) *Package* Game: mengatur semua layar, hubungan *Bluetooth*, algoritma serta *main class* dari permainan Mr. Jack.
- b) *Package* img: mengatur semua gambar dalam permainan
- c) *Package* Map: mengatur semua objek dan atributnya dalam permainan
- d) *Package* Ability : mengatur atribut yang diperlukan oleh penggunaan kemampuan khusus dari setiap karakter permainan

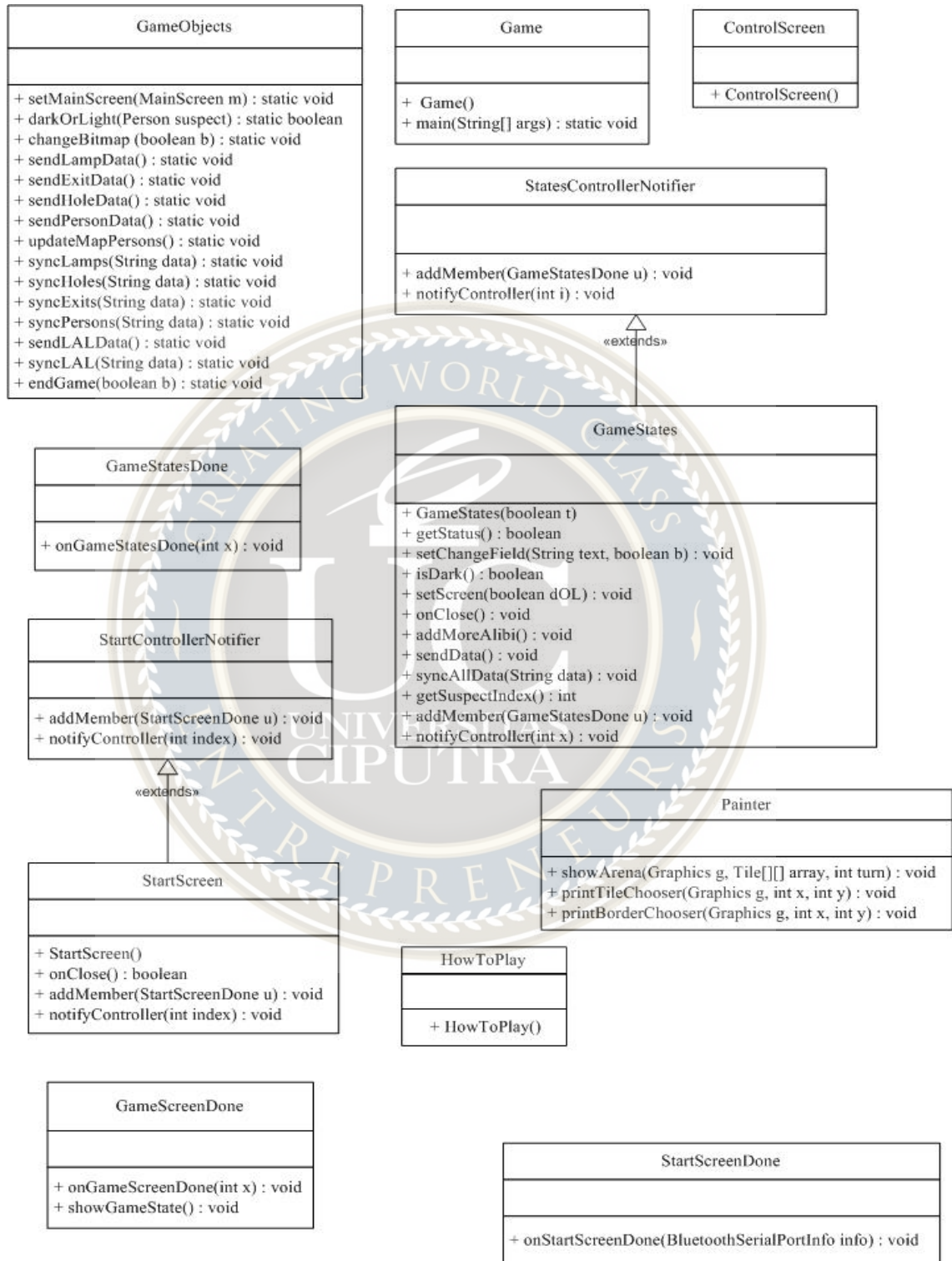


Gambar 3.14 Package Dependencies

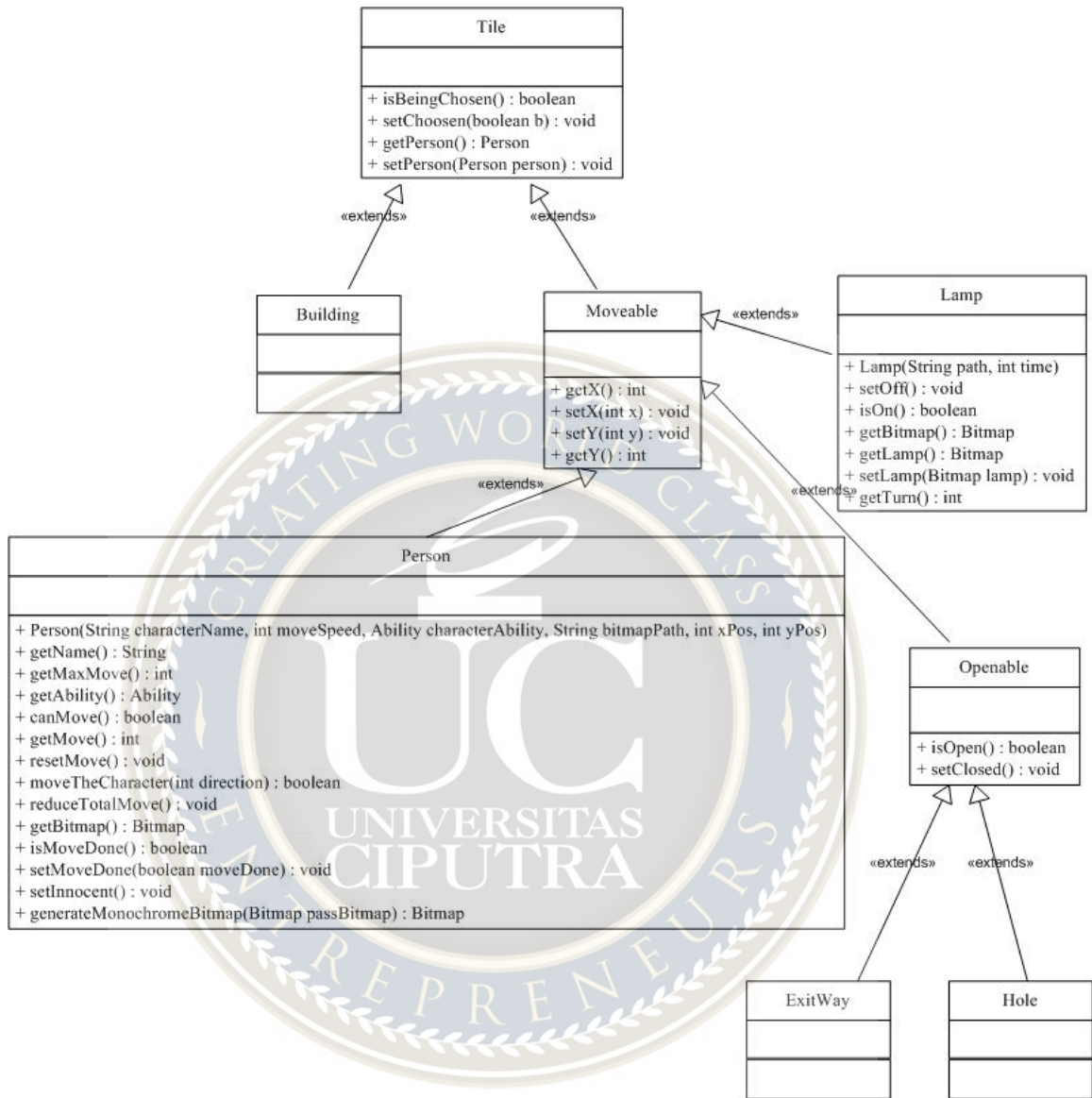
3.7 Diagram & Deskripsi Class



Gambar 3.15 Class Diagram (1)



Gambar 3.16 Class Diagram (2)



Gambar 3.17 Class Diagram (3)

1. Class Ability

Fungsi: *class* ini adalah *interface* dari semua kemampuan karakter, sehingga setiap karakter dapat mempunyai karakter yang berbeda - beda

Berelasi dengan: semua *class* kemampuan karakter *class* Person, dan *class* GameScreen

2. *Class* ShutHole

Fungsi: kemampuan karakter untuk memindah letak tutup lubang pada peta

Berelasi dengan: *class* Person dan gamescreen

3. *Class* CloseExitWay

Fungsi: kemampuan karakter untuk membuka dan menutup jalan keluar

Berelasi dengan: *class* Person dan gamescreen

4. *Class* MoveThroughWalls

Fungsi: kemampuan karakter untuk menembus dinding

Berelasi dengan: *class* Person dan gamescreen

5. *Class* LightALine

Fungsi: kemampuan karakter untuk menerangi jalan untuk membantu polisi menangkap *Jack*

Berelasi dengan: *class* Person dan gamescreen

6. *Class* SwitchLamp

Fungsi: kemampuan karakter untuk memindah lampu pada peta

Berelasi dengan: *class* Person dan gamescreen

7. *Class* CallForOtherCharacter

Fungsi: kemampuan karakter untuk memanggil karakter lain mendekat sebanyak satu petak ke karakter yang menggunakan kemampuan ini

Berelasi dengan: *class* Person dan gamescreen

8. *Class PickAnAlibi*

Fungsi: kemampuan karakter untuk mengambil alibi dari karakter untuk membantu polisi menangkap *Jack*

Berelasi dengan: *class* Person dan *gamescreen*

9. *Class Swap*

Fungsi: kemampuan karakter untuk menukar posisi dengan karakter lain

Berelasi dengan: *class* Person dan *gamescreen*

10. *Class ControlScreen*

Fungsi: berisi petunjuk bagaimana cara mengendalikan karakter di permainan

Berelasi dengan: *class* startScreen

11. *Class HowToPlay*

Fungsi: berisi peraturan permainan

Berelasi dengan: *class* startScreen

12. *Class Game*

Fungsi: merupakan *controller* dari semua *screen*.

Berelasi dengan: *class* StartScreen, *class* BluetoothController, *class* GameScreen, *class* GameStates, *class* CharacterScreen, *interface* BluetoothListener, *interface* CharacterControllerNotifier, *interface* CharacterScreenDone, *interface* GameControllerNotifier, *interface* GameScreenDone, *interface* GameStatesDone, *interface* StartControllerNotifier, *interface* StartScreenDone, *interface* StatesControllerNotifier.

13. *Class BluetoothController*

Fungsi: pengiriman dan penerimaan data lewat *Bluetooth*

Berelasi dengan: *class gameScreen*, *class Game*, *class CharacterScreen* dan *class GameState*

14. *Class GameObjects*

Fungsi: pengaturan map, karakter, kemampuan karakter, dan gelap terangnya peta

Berelasi dengan: *class gameScreen*

15. *Class GameScreen*

Fungsi: penanganan interaksi user pada peta permainan serta logika permainan.

Berelasi dengan: *class GameObjets*, *class ability*, *class Person*, *class Hole*, *class Building*, *class Tile*

16. *Class Painter*

Fungsi: menggambar peta dan semua objek yang ada di dalam peta permainan.

Berelasi dengan: *class gameScreen*, *class BluetoothController*

17. *Class GameStates*

Fungsi: menampilkan alibi dari tiap karakter, keadaan *Jack* , sisa langkah adri karakter dan penggunaan kemampuan karakter.

Berelasi dengan: *class gameScreen*, *class BluetoothController*

18. *Class StartScreen*

Fungsi: memulai permainan, menunggu hubungan *Bluetooth* , atau mencari pemain lawan

Berelasi dengan: *class HowToPlay*, *class ControlScreen*, *class Game*

19. *Class CharacterScreen*

Fungsi: pemilihan karakter

Berelasi dengan: *class* GameScreen, *class* GameStates, *class* BluetoothController

20. *Interface* GameScreenDone

Fungsi: memberi tahu *controller* agar mengubah *screen* yang sedang aktif

Berelasi dengan: *class* Game, Class GameScreen.

20 *Interface* StartScreenDone

Fungsi: memberi tahu *controller* agar mengubah *screen* yang sedang aktif

Berelasi dengan: *class* Game, Class GameScreen.

21. *Interface* CharacterScreenDone

Fungsi: memberi tahu *controller* agar mengubah *screen* yang sedang aktif

Berelasi dengan: *class* Game, Class GameScreen.

21. *Interface* StartControllerNotifier

Fungsi: memberi tahu *controller* agar mengubah *screen* yang sedang aktif

Berelasi dengan: *class* Game, Class StartScreen.

22. *Interface* GameControllerNotifier

Fungsi: memberi tahu *controller* agar mengubah *screen* yang sedang aktif

Berelasi dengan: *class* Game, Class GameScreen.

23. *Interface* CharacterControllerNotifier

Fungsi: memberi tahu *controller* agar mengubah *screen* yang sedang aktif

Berelasi dengan: *class* Game, Class CharacterScreen.

24. *Class* Tile

Fungsi: penggerakan karakter di peta

Berelasi dengan: *class* GameObjects, *class* GameScreen

25. *Class* Building

Fungsi: penggerakan karakter di peta

Berelasi dengan: *class* GameObjects, *class* GameScreen

26. *Class* Moveable

Fungsi: penggerakan objek, dan letak objek di peta

Berelasi dengan: *class* GameObjects, *class* GameScreen, *class* Lamp, *class* Openable

27. *Class* Openable

Fungsi: status terbuka atau tertutupnya suatu objek di peta

Berelasi dengan: *class* GameObjects, *class* GameScreen, *class* Hole, *class* ExitWay, *class* Moveable

28. *Class* Hole

Fungsi: lubang di peta

Berelasi dengan: *class* GameObjects, *class* GameScreen, *Class* Openable

29. *Class* Lamp

Fungsi: lampu di peta

Berelasi dengan: *class* GameObjects, *class* GameScreen, *class* Moveable

30. *Class* Person

Fungsi: segala sesuatu yang dibutuhkan oleh karakter di permainan dicakup disini

Berelasi dengan: *class* GameObjects, *class* GameScreen, *Class* Moveable

31. *Class* ExitWay

Fungsi: jalan keluar di peta

Berelasi dengan: *class* GameObjects, *class* GameScren, *Class* Openable

32. Interface BluetoothListener

Fungsi: interface dari class BluetoothController

Berelasi dengan: *class* Game, *class* BluetoothController

33. Interface GameStatesDone

Fungsi: memberi tahu *controller* agar merubah *screen* yang sedang aktif

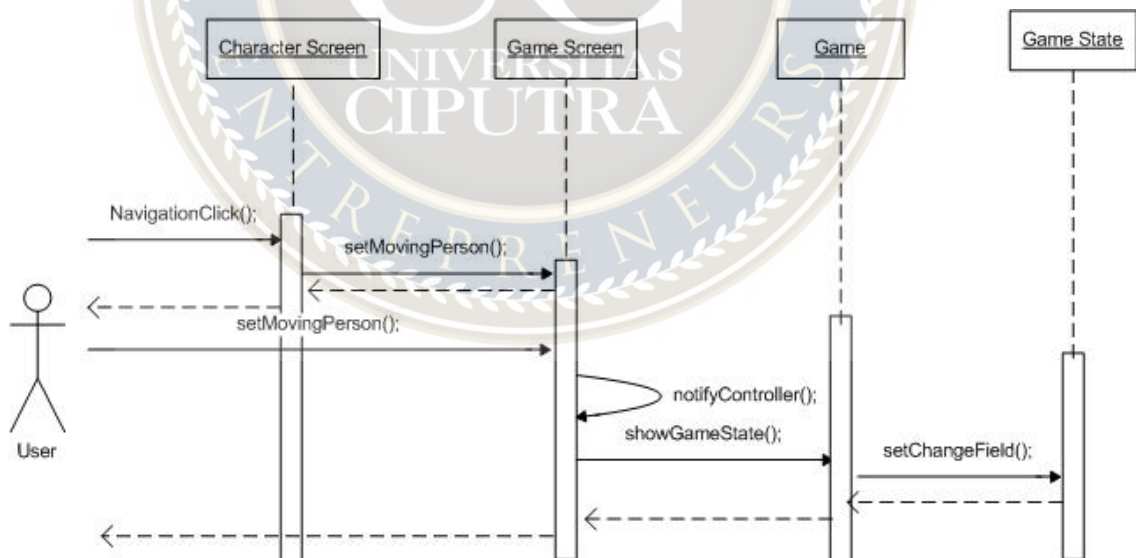
Berelasi dengan: *class* GameStates, *class* Game

34. Interface StatesControllerNotifier

Fungsi: memberi tahu *controller* agar mengubah *screen* yang sedang aktif

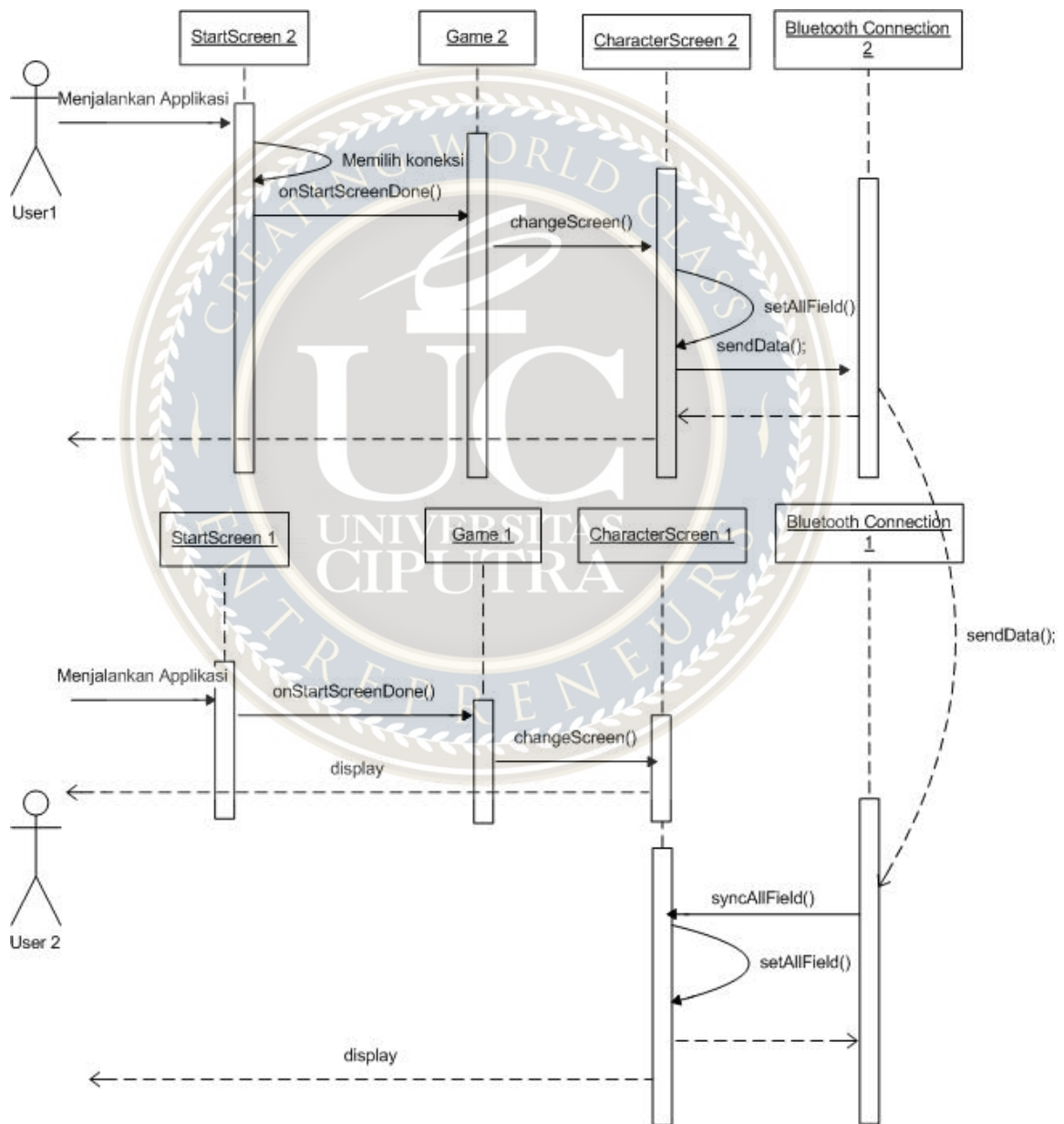
Berelasi dengan: *class* GameStates, *class* Game

3.8 Diagram Sequence



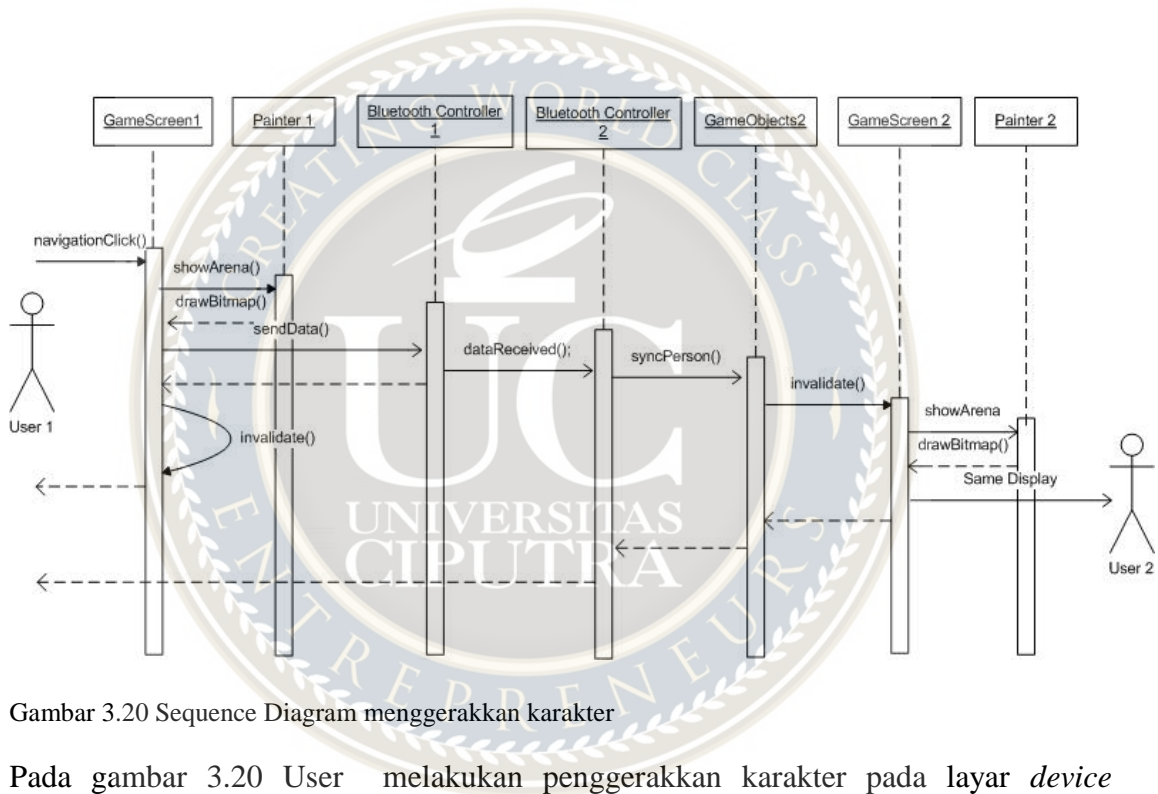
Gambar 3.18 Sequence Diagram pemilihan karakter

Pada gambar 3.18 user melakukan *request* menggunakan *event*. Karakter yang dipilih kemudian disiapkan untuk digerakkan pada layar permainan, user kemudian melakukan *request* untuk mengganti layar menuju layar kondisi permainan



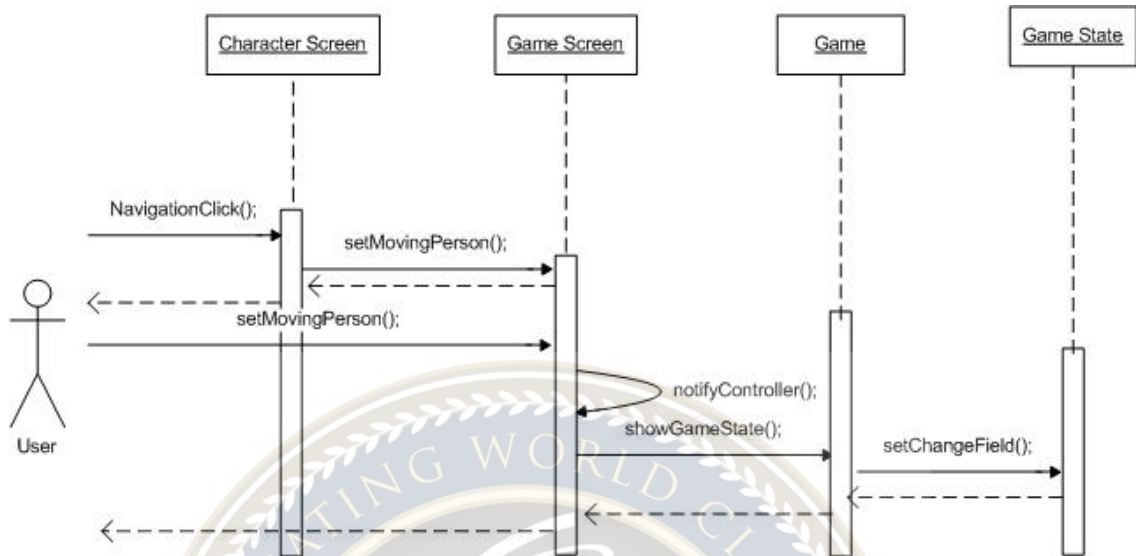
Gambar 3.19 Sequence Diagram hubungan Bluetooth

Pada Gambar 3.19 user melakukan koneksi dari *device* yang digunakan menuju *device* pemain lain dengan menggunakan koneksi Bluetooth. Setelah kedua pemain terhubung, pemain yang memanggil kemudian mengirimkan data permainan ke pemain yang menunggu. Data ini kemudian akan ditampilkan oleh *device* pemain lain sehingga kedua *device* mempunyai data dan tampilan yang sama



Gambar 3.20 Sequence Diagram menggerakkan karakter

Pada gambar 3.20 User melakukan penggerakkan karakter pada layar *device* miliknya, data dari letak karakter kemudian dirubah bentuknya menjadi paket dengan *header* agar *device* pemain lain dapat membedakan data yang diterima dan dikirimkan melalui Bluetooth ke *device* pemain lain. Setelah data diterima dan diolah menjadi informasi yang berguna, letak seluruh karakter dari pemain yang sedang menggerakkan karakter akan sama dengan pemain yang menunggu giliran.



Gambar 3.21 Sequence Diagram memindahkan lampu

Pada gambar 3.21 User melakukan kemampuan dari salah satu karakter yaitu penggerakkan lampu pada layar *device* miliknya, data dari letak lampu kemudian dirubah bentuknya menjadi paket dengan *header* agar *device* pemain lain dapat membedakan data yang diterima dan dikirimkan melalui Bluetooth ke *device* pemain lain. Setelah data diterima dan diolah menjadi informasi yang berguna, letak seluruh lampu dari pemain yang sedang menggerakkan karakter akan sama dengan pemain yang menunggu giliran.