

## BAB II LANDASAN TEORI

### 2.1. Tinjauan Pustaka

Tinjauan pustaka dalam penelitian ini mengenai penilaian tampilan, menu dan fitur-fitur dari beberapa aplikasi *Game Android* dan Jurnal-jurnal ilmiah lain yang sudah ada. Dimulai dari menilai dari segi tampilan, menu dan fitur-fitur yang diberikan oleh jurnal ilmiah Andika (2012) dengan judul Perancangan dan Pembuatan Aplikasi Permainan Petualangan Gatotkaca dalam Perang Baratayuda, Jurnal ilmiah Sagita (2013) dengan judul Pengembangan Aplikasi *Game* Edukasi Karakter Gatotkaca dan 5 Pandawa untuk Anak Usia 5-8 Tahun, dan Jurnal ilmiah Juniarta (2015) dengan judul Aplikasi *Game* Petualangan I Jamong Berbasis Android.

#### 2.1.1. Perancangan dan Pembuatan Aplikasi Permainan Petualangan Gatotkaca dalam Perang Baratayuda (Andika, 2012)

Menu utama terdapat tiga tombol yaitu tombol main, tombol cara bermain, dan tombol persembahan. *Interface* menu utama disajikan pada Gambar 2.1.



Gambar 2.1. *Interface* Menu Utama

Saat pemain memilih menu main, maka pemain akan memilih menu pemilihan level yang terdiri dari tiga *level* permainan yaitu permainan *level* menggunakan fitur *touch screen* untuk menyerang musuh yang berjalan dan menggagalkan serangan bola api dari bos terakhir. Permainan *level* dua

menggunakan fitur *swipe* terhadap batu yang dipegang oleh karakter Gatotkaca yang berdiri di atas menara dan pemain harus melindungi menara dari musuh. Sedangkan permainan level tiga pemain akan diberi kontrol untuk menggerakkan Gatotkaca kesegala arah serta tombol kontrol untuk mendapatkan panah. Terdapat dua tipe musuh yaitu, musuh yang bergerak kearah Gatotkaca yaitu musuh pedang yang menyerang dari jarak dekat dan musuh tombak yang menyerang dari jarak jauh.

### 2.1.2. Pengembangan Aplikasi *Game* Edukasi Karakter Gatotkaca Dan 5 Pandawa Untuk Anak Usia 5-8 Tahun (Sagita, 2013)

Menu utama terdapat tiga tombol yaitu tombol main, tombol nilai tertinggi, dan tombol keluar. *Interface* menu utama disajikan pada Gambar 2.2.



Gambar 2.2. *Interface* Menu Utama

Saat pemain memilih menu main, pemain masuk ke dalam menu utama yang terdiri dari empat menu yaitu petualangan, permainan bebas, pengaturan, dan kembali. Menu petualangan adalah menu yang harus dilalui oleh pemain sebelum memilih menu *game* permainan bebas. Pada menu ini pemain menjadi karakter Gatotkaca dan pemain akan dibawa ke dalam sebuah cerita dimana Gatotkaca mencari lima Pandawa dalam permainan petak umpet. Menu permainan bebas adalah menu yang berisi empat mini *game* yang masing-masing dari *game* tersebut dapat dipilih secara bebas oleh pemain yaitu *game* mewarnai, mencari perbedaan, mencari barang, dan membuat kartu. Pada menu pengaturan terdapat tiga menu yang terdiri dari kamar ganti, ganti pemain, dan mengatur suara.

### 2.1.3. Aplikasi *Game* Petualangan I Jamong Berbasis *Android* (Juniartha, 2015)

Menu utama terdapat enam tombol yaitu tombol *play*, tombol *continue*, tombol *tutorial*, tombol *level*, tombol *about*, tombol *credit*, dan tombol *exit*. *Interface* menu utama disajikan pada Gambar 2.3.



Gambar 2.3. *Interface* Menu Utama

Pada menu *play* ini pemain dapat memainkan karakter I jamong berpetualangan mencari ayahnya ke Majapahit. Pada menu ini terdapat tombol-tombol yang digunakan untuk menggerakkan karakter I Jamong antara lain adalah tombol kanan, kiri, *pause*, *shoot*, dan *jump*. Menu *tutorial* merupakan suatu menu bantuan yang berisikan panduan dalam menggunakan aplikasi *game* petualangan I Jamong berbasis *Android* ini. Tutorial ini berisikan keterangan singkat mengenai tombol-tombol yang terdapat pada permainan. Menu *level* suatu menu yang berisi *level* permainan yang sudah dilalui oleh I Jamong untuk mencari ayahnya. Tombol *exit* digunakan untuk keluar dari aplikasi.

Berdasarkan dari tiga jurnal di atas yang menjadi referensi dan perbandingan dalam pembuatan aplikasi *game* Gatokaca berbasis *Android* ini dapat disimpulkan bahwa tampilan aplikasi yang dibuat kurang menarik.

Di bawah ini akan ditampilkan tabel perbandingan dari ketiga jurnal ilmiah yang menjadi bahan referensi untuk pembuatan aplikasi *game* Gatokaca berbasis *Android*.

Tabel 2.1. Perbandingan Aplikasi *Game* Jurnal Ilmiah

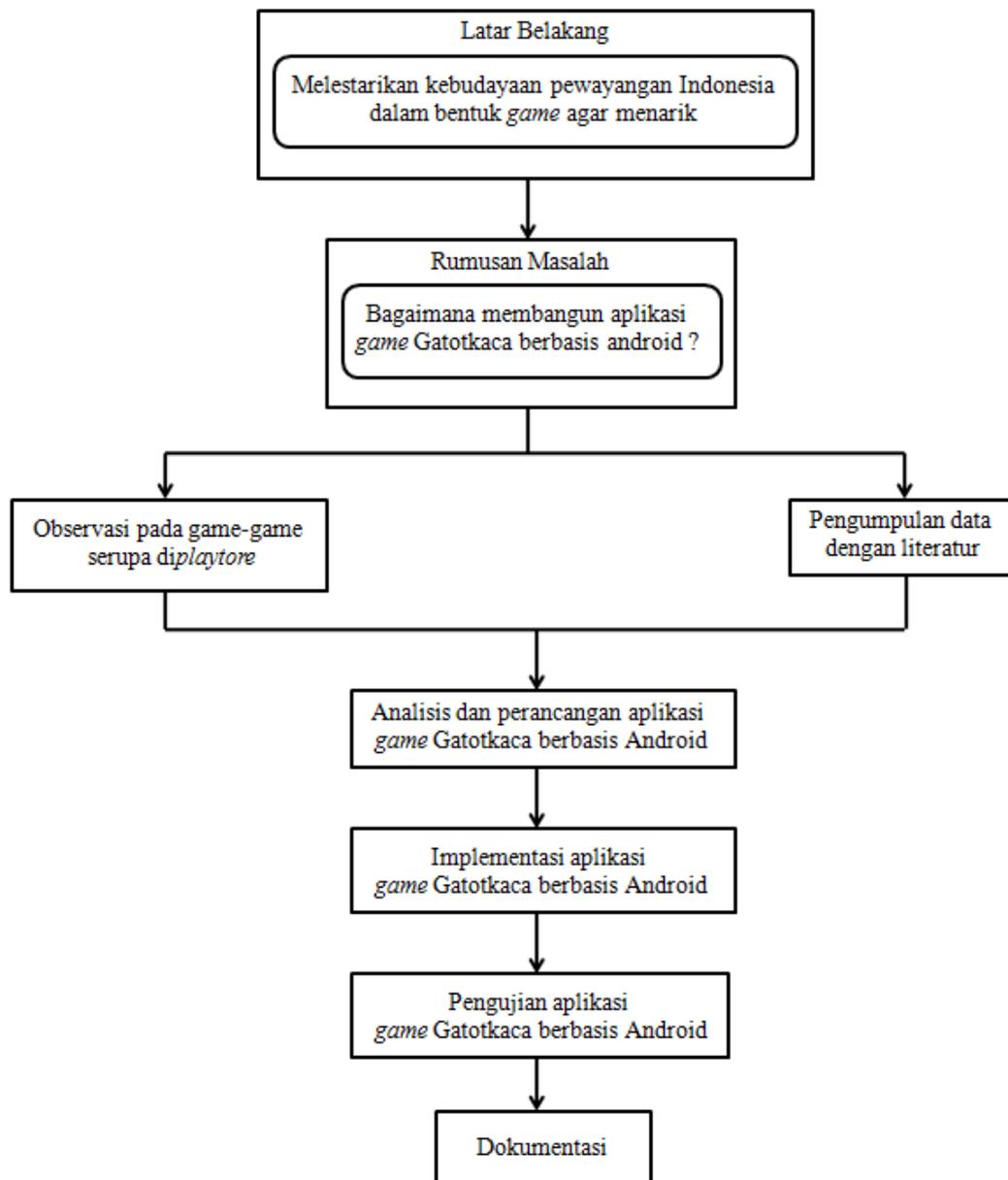
No	Judul	Aktor	Menu Utama	Software	Genre	Interface
1.	Perancangan dan Pembuatan Aplikasi Permainan Petualangan “Gatokaca Dalam Perang Baratayuda”	Gatokaca	1) Main : <i>Level 1, Level 2, dan Level 3</i> 2) Cara Bermain 3) Persembahan	<i>Adobe Flash Builder</i>	<i>Adventure</i>	Kurang Menarik ( <i>Interface</i> tampilan yang masih sederhana)
2.	Pengembangan Aplikasi <i>Game</i> Edukasi Karakter Gatokaca dan 5 Pandawa untuk Anak Usia 5-8 Tahun	Gatokaca dan 5 Pandawa	1) Main : Petualangan, permainan bebas, pengaturan, dan kembali 2) Nilai Tertinggi 3) Keluar	Multimedia	<i>Adventure dan Puzzle</i>	Kurang Menarik ( <i>Interface</i> tampilan yang masih hitam putih)
3.	Aplikasi <i>Game</i> Petualangan I Jamong Berbasis Android	I Jamong	1) <i>Play</i> 2) <i>Tutorial</i> 3) <i>About</i> 4) <i>Credit</i> 5) <i>Exit</i>	<i>Engine Unity 4.6.1</i>	<i>Adventure</i>	Menarik ( <i>Interface</i> tampilan yang <i>user friendly</i> )
4.	Aplikasi <i>Game</i> Gatokaca Berbasis <i>Android</i>	Gatokaca	1) Mulai 2) Sejarah 3) Petunjuk 4) Keluar	<i>Game Maker</i>	<i>Adventure</i>	Menarik (Hasil pengujian kuesioner nomor 5 dengan presentasi 72,2% )

Penelitian pada Tugas Akhir dengan judul aplikasi *game* Gatokaca berbasis Android menggunakan tokoh Gatokaca sebagai aktor pada *game* yang

akan dibuat. Menu utama *game* ini terdiri dari 4 menu yaitu menu main, sejarah, petunjuk, dan keluar. *Software* yang digunakan dalam pembuatan aplikasi *game* Gatotkaca berbasis Android ini adalah *Game Maker*.

## 2.2. Kerangka Pemikiran

Diagram kerangka pemikiran yang dijalankan dalam penelitian ini dapat dilihat pada Gambar 2.4.



Gambar 2.4. Diagram Kerangka Pemikiran

Keterangan dari Gambar 2.4. :

1) Latar Belakang

Melestarikan kebudayaan pewayangan Indonesia dalam bentuk *game* agar menarik.

2) Rumusan Masalah

Bagaimana membangun aplikasi *game* Gatotkaca berbasis Android?

3) Observasi

Penulis mengamati beberapa aplikasi *game* yang sudah ada di *playstore*, baik dari karya ilmiah, buku, atau internet yang dapat dijadikan referensi untuk aplikasi *game* Gatotkaca berbasis Android.

4) Pengumpulan Data dengan Literatur

Penulis mengumpulkan data dengan mempelajari literatur, yang berupa buku-buku, diktat ataupun bentuk lain yang berhubungan dengan objek yang dipelajari guna mendukung penelitian.

5) Analisis dan Perancangan Sistem

Penulis menganalisa sistem yang sudah ada yaitu dari jurnal-jurnal ilmiah dan *game-game* yang ada di *playstore* serta membangun sistem yang akan dibangun seperti apa, bagaimana desainnya, dan apa saja isinya, sehingga sistem ini dapat dibuat sesuai dengan tujuan penelitian.

6) Implementasi Sistem

a) Perancangan Karakter pada *game*

Perancangan karakter yang akan digunakan pada *game* menggunakan *software corel draw* dan referensi buku pewayangan.

b) Perancangan *Background*

Membuat desain *background* untuk tampilan *game*.

c) Perancangan Aplikasi

Membuat aplikasi *game* Gatotkaca berbasis Android sesuai dengan tujuan penelitian.

7) Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui jika ternyata masih ada kesalahan atau kekurangan pada aplikasi yang dibuat.

## 8) Dokumentasi

Pada tahap akhir, dimana aplikasi telah siap digunakan setelah melewati tahap pengujian dan membuat dokumentasi dari keseluruhan Tugas Akhir.

## 2.3. Teori - teori Pendukung

### 2.3.1. *Game*

*Game* atau permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan *refreshing* (Muchtari, 2005 dikutip oleh Amrullah, 2012).

Klasifikasi *game* dimaksudkan untuk memudahkan pengelompokan jenis *game*. Beberapa klasifikasi *game* adalah sebagai berikut (Martono, 2015) :

- 1) *Game as Game*, *Game* yang dimaksud adalah *game* untuk kesenangan atau *fun*.
- 2) *Game as Media*, tujuan utama dari *game as media* adalah untuk menyampaikan pesan tertentu dari pembuat *game* tersebut.
- 3) *Game as Beyond Game (Gamification)*, merupakan penerapan konsep atau cara berfikir *game design* ke dalam lingkungan *non-game*.

Selain klasifikasi *game*, terdapat jenis *platform* yang digunakan dalam pengembangan atau pengaplikasian *game*. Beberapa jenis *platform game* adalah sebagai berikut :

- 1) *Arcade Games*, yaitu yang sering disebut ding-dong di Indonesia, biasanya berada di daerah/tempat khusus dan memiliki box atau mesin yang memang khusus di *design* untuk jenis video *games* tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa “masuk” dan “menikmati”, seperti pistol, kursi khusus, sensor gerakan, sensor injakan dan stir mobil.
- 2) *PC Games*, yaitu video *game* yang dimainkan menggunakan *Personal Computers*.
- 3) *Console Games*, yaitu video *game* yang dimainkan menggunakan *console* tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii.

- 4) *Handheld Games*, yaitu dimainkan di *console* khusus video game yang dapat dibawa kemana-mana, seperti Nintendo DS dan Sony PSP.
- 5) *Mobile Games*, yaitu yang dapat dimainkan atau khusus untuk *mobile phone* atau PDA.

### 2.3.2. *Storyboard*

*Storyboard* merupakan istilah untuk kertas yang berisi ide cerita dan berupa susunan gambar serta detail dari setiap adegan. Dengan *storyboard*, proses pembuatan karakter dan animasi dapat lebih cepat, mudah, dan efisien (Ristina, 2013).

### 2.3.3. *Smartphone*

*Smartphone* merupakan *cellphone* yang menggabungkan fungsi-fungsi *Personal Digital Assistant (PDA)* seperti kalender, *personal schedule*, *address book*, dan memiliki kemampuan untuk mengakses *internet*, membuka *email*, membuat dokumen, bermain *game*, serta membuka aplikasi lainnya. Istilah *smartphone* merupakan istilah yang digunakan untuk mendeskripsikan *mobile device* yang menggabungkan fungsi *cellphone*, *audio player*, *digital camera*, *camcorder*, *Global Positioning System (GPS) receiver*, dan *Personal Computer (PC)* (Wijayanto, 2015).

### 2.3.4. *Android*

*Android* adalah kumpulan perangkat lunak yang ditujukan bagi perangkat bergerak mencakup sistem operasi, *middleware*, dan aplikasi kunci. *Android* dikembangkan oleh *Google* bersama *Open Handset Alliance (OHA)* yaitu aliansi perangkat seluler terbuka yang terdiri dari 47 perusahaan *hardware*, *software*, dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat *mobile* (Amrullah, 2012). Sampai saat ini *Android* telah memiliki beberapa versi sejak pertama kali dirilis, yaitu :

- 1) *Android* versi 1.0
- 2) *Android* versi 1.1
- 3) *Android* versi 1.5 (*Cupcake*)
- 4) *Android* versi 1.6 (*Donut*)
- 5) *Android* versi 2.0 / 2.0.1 / 2.1 (*Eclair*)

- 6) *Android* versi 2.2 (*Frozen Yogurt / Froyo*)
- 7) *Android* versi 2.3 (*Gingerbread*)
- 8) *Android* versi 3.0 / 3.1 / 3.2 (*Honeycomb*)
- 9) *Android* versi 4.0 (*Ice Cream Sandwich*)
- 10) *Android* versi 4.1 / 4.3 (*Jelly Bean*)
- 11) *Android* versi 4.4 (*Kitkat*)
- 12) *Android* versi 5.0 (*Lollipop*)
- 13) *Android* versi 6.0 (*Marshmallow*)
- 14) *Android* versi 7.0 (*Nougat*)

### **2.3.5. Android SDK**

*Android SDK* adalah *tools API (Application Programming Interface)* yang diperlukan untuk mengembangkan aplikasi pada *platform Android* yang menggunakan bahasa pemrograman *Java*. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi yang di-*release* oleh Google. Saat ini disediakan *Android SDK (Software Development Kit)* sebagai alat bantu dan *API (Application Programming Interface)* untuk mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. Sebagai aplikasi *platform Android* aplikasi netral, *Android* memberi kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone/ Smartphone* (Nasrudin, 2012).

### **2.3.6. JDK (Java Development Kit)**

*Java Development Kit (JDK)* merupakan sebuah produk yang dikembangkan oleh *Oracle* yang ditujukan untuk para *Developer java*. *JDK* dilengkapi dengan banyak komponen untuk melakukan pemrograman. *JDK* juga berisi paket *Java Runtime Environment (JRE)* yang lengkap, biasanya disebut *private runtime* dari *JRE* reguler dan dilengkapi dengan konten tambahan, yaitu terdiri atas *Java Virtual Machine* dan semua *class library* yang ada di *environment* produk dan juga *library* tambahan yang berguna untuk *developer* (Fadlullah, 2012).

### 2.3.7. Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju (Pradana, 2015).

### 2.3.8. Game Maker

*Game Maker* merupakan *Integrated Development Enviroment* (IDE) untuk sistem operasi Windows dan Mac OS yang dikembangkan oleh Mark Overmars dalam bahasa pemograman Delphi (2006). Saat ini *Game Maker* dikembangkan dan diterbitkan oleh Yoyo Games, sebuah perusahaan perangkat lunak dimana Overmars bekerja. *Game Maker* memungkinkan pengguna untuk mengembangkan permainan komputer dengan mudah, meskipun belum memiliki banyak pengalaman dalam pemograman komputer. *Game Maker Studio IDE* memiliki fitur-fitur untuk pembuatan permainan seperti *font*, suara, gambar, musik serta *resource* lain yang dibutuhkan untuk membuat sebuah aplikasi permainan. Komponen utama yang digunakan untuk membangun sebuah permainan menggunakan *Game Maker* antara lain adalah *sprite*, *sounds*, *objects*, *events*, *room*, dan *background*. Selain itu ada pula *scripts* atau *coding* dengan menggunakan GML (*Game Maker Language*) untuk mengembangkan permainan menjadi lebih kompleks. Komponen-komponen *Game Maker* mencakup berbagai genre, termasuk aksi, petualangan, permainan *puzzle* dan yang lainnya. Untuk membuat *game* yang bagus dan handal harus mengetahui semua fungsi yang ada pada *Game Maker* (Komputer, 2013).

### 2.3.9. Wayang

Istilah “wayang” menurut Kamus Besar Bahasa Indonesia (Edisi Ketiga, Jakarta: Pusat Departemen Pendidikan Nasional dan Balai Pustaka, 2005), diartikan sebagai boneka tiruan orang yang terbuat dari pahatan kulit atau kayu dan sebagainya yang dapat dimanfaatkan untuk memerankan tokoh dalam pertunjukan drama tradisional (Bali, Jawa, Sunda, dan sebagainya), biasanya dimainkan oleh seseorang yang disebut dalang (Yasasusastra, 2011).

Beberapa jenis-jenis wayang antara lain sebagai berikut (Yasasusastra, 2011) :

- 1) wayang Gedbog, jenis wayang ini berupa boneka-boneka wayang yang terbuat dari kulit, tipis, dan juga ditantah.
- 2) Wayang Golek, jenis wayang yang wujudnya berupa boneka terbuat dari kayu dalam bentuk tiga dimensi.
- 3) Wayang Klitibik, jenis wayang ini terbuat dari kayu pipih dan ada bagian yang terbuat dari kulit.
- 4) Wayang Beber, jenis wayang ini tidak memperlihatkan tokoh cerita satu persatu, melainkan pertunjukannya berupa lembaran kain yang dilukisi dengan gambar-gambar berupa jalannya cerita atau adegan-adegan.
- 5) Wayang Wong (Orang), jenis wayang yang mempergelarkan cerita yang diperankan oleh orang dengan syarat para pemainnya dapat menari.
- 6) Wayang Suluh, pertunjukan yang diadakan sebagai kelanjutan dari apa yang disebut “Wayang Wahana”.
- 7) Wayang Menak, jenis wayang berbentuk boneka kayu yang diyakini muncul pertama kali di daerah Kudus.

#### **2.3.10. Gatotkaca**

Gatotkaca (bahasa Sanskerta : Ghattotkacha) adalah seorang tokoh dalam wiracarita Mahabharata yang dikenal sebagai putra Bimasena atau Wrekodara dari keluarga Pandawa. Ibunya bernama Hidimbi (Harimbi) berasal dari bangsa raksasa, sehingga ia pun dikisahkan memiliki kekuatan luar biasa. Gatotkaca menjadi tokoh pewayangan yang sangat populer. Misalnya dalam pewayangan Jawa ia dikenal dengan ejaan Gatutkaca (bahasa Jawa : Gathutkaca). Kesaktiannya dikisahkan luar biasa, diantara lain mampu terbang diangkasa tanpa menggunakan sayap, serta terkenal dengan julukan “*otot kawat tulang besi*” (Yasasusastra, 2011).

#### **2.3.11. UML (*Unified Modeling Language*)**

UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’ (Nugroho, 2010). Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks demikian rupa sehingga lebih mudah dipelajari dan dipahami.

1) *Use Case Diagram*

*Use case diagram* yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case* (Haviluddin, 2011). Simbol-simbol yang digunakan pada *use case diagram* disajikan pada Tabel 2.2.

Tabel 2.2. Simbol-simbol *Use Case Diagram*

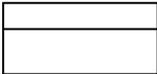
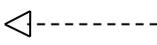
No	Gambar	Nama	Keterangan
1.		<i>Aktor</i>	Idealization orang eksternal, proses, atau hal yang berinteraksi dengan sistem, subsistem atau kelas.
2.		<i>Use case</i>	Sebuah <i>use case</i> menggambarkan interaksi dengan <i>actor</i> sebagai urutan pesan antarsistem dan aktor satu atau lebih.
3.		<i>System Boundary</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
4.		<i>Generalization</i>	Hubungan antara <i>use case</i> umum dan <i>use case</i> yang lebih spesifik yang mewarisi dan menambahkan fitur.
5.		<i>Comunication Association</i>	Jalur komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi didalam
6.	--<<extend>>-->	<i>Extend</i>	Penyisipan perilaku tambahan kedalam basis <i>use case</i> yang tidak tahu tentang hal itu.
7.	--<<include>>-->	<i>Include</i>	Penyisipan perilaku tambahan kedalam basis <i>use case</i> yang secara eksplisit menggambarkan penyisipan.

Sumber : Kusumo (2004:3)

## 2) Class Diagram

*Class diagram* menggambarkan struktur statis dari kelas dalam sistem dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat (Haviluddin, 2011). Simbol-simbol yang digunakan pada *class diagram* disajikan pada Tabel 2.3.

Tabel 2.3. Simbol-simbol *Class Diagram*

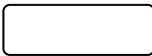
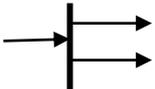
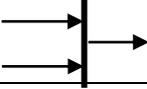
No	Gambar	Nama	Keterangan
1.		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagai perilaku dan struktur data dari objek yang ada diatas objek induk ( <i>ancestor</i> ).
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.
3.		<i>Class</i>	Himpunan dari objek-objek yang terbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yan tidak mandiri.
7		<i>Association</i>	Untuk menghubungkan objek satu dengan objek yang lainnya.

Sumber : Kusumo (2004:3)

### 3) *Activity Diagram*

*Activity diagram* menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas (Haviluddin, 2011). Simbol-simbol yang digunakan pada *activity diagram* disajikan pada Tabel 2.4.

Tabel 2.4. Simbol-simbol *Activity Diagram*

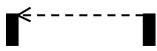
No	Gambar	Nama	Keterangan
1.		<i>Activity State</i>	Aktivitas yang mewakili pelaksanaan dalam pernyataan dalam prosedur atau pelaksanaan kegiatan dalam alur kerja.
2.		<i>Branch/Merge</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
3.		<i>Initial State</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
4.		<i>Final State</i>	Status akhir yang dilakukan sistem.
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
6.		<i>Join Node</i>	Beberapa aliran masukan tertentu menjadi satu aliran.

Sumber : Kusumo (2004:4)

### 4) *Sequence Diagram*

*Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram* (Haviluddin, 2011). Simbol-simbol yang digunakan pada *sequence diagram* ditampilkan pada Tabel 2.5.

Tabel 2.5 Simbol-simbol *Sequence Diagram*

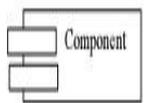
No	Gambar	Nama	Keterangan
1.		<i>Lifeline</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Message</i>	Message ditampilkan sebagai anak panah dari lifeline dari satu objek ke objek yang lain.
3.		<i>Return</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Sumber : Kusumo (2004:4)

### 5) *Component Diagram*

*Component diagram* menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan (Haviluddin, 2011). Simbol-simbol yang digunakan pada *component diagram* disajikan pada Tabel 2.6.

Tabel 2.6 Simbol-simbol *Component Diagram*

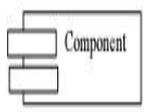
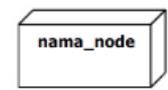
No	Gambar	Nama	Keterangan
1.		<i>Component</i>	Sebuah komponen melambangkan sebuah entitas <i>software</i> dalam sebuah sistem.
2.		<i>Dependency</i>	Sebuah <i>Dependency</i> digunakan untuk menotasikan relasi antara dua komponen.

Sumber : Kusumo (2004:4)

### 6) *Deployment Diagram*

*Deployment diagram* memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment diagram* dapat dianggap sebagai ujung 26pectrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan system (Haviluddin, 2011). Simbol-simbol yang digunakan pada *deployment diagram* disajikan pada Tabel 2.7.

Tabel 2.7 Simbol-simbol *Deployment Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Component</i>	Sebuah komponen melambangkan sebuah entitas <i>software</i> dalam sebuah sistem.
2.		<i>Node</i>	Node menggambarkan bagian-bagian <i>hardware</i> dalam sebuah sistem.
3.		<i>Association</i>	Sebuah <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara komponen-komponen <i>hardware</i> .

Sumber : Kusumo (2004:4)

#### 2.4. Metode Pengujian Sistem

Metode yang digunakan dalam menguji aplikasi *game* Gatotkaca berbasis Android ini adalah :

##### 1) Metode Pengujian *Blackbox*

Pengujian *blackbox* adalah pengujian yang memungkinkan *software engineer* mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan persyaratan fungsional untuk suatu program (Pressman, 2005). Metode pengujian *blackbox* dilakukan untuk menguji fungsional aplikasi sehingga dapat diketahui ada atau tidaknya *bug* pada aplikasi.

##### 2) Metode Pengujian Kuesioner

Metode pengujian kuesioner dilakukan dengan menggunakan skala likert. Skala likert merupakan metode pengukuran yang digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau kelompok orang tentang fenomena sosial (Sugiono, 2012). Pengujian kuesioner dilakukan untuk menguji kepuasan pengguna aplikasi.