BAB 2
LANDASAN TEORI

2.1.Wayang Golek

Wayang golek adalah bentuk pertunjukan boneka kayu dengan ukiran berkarakter Sunda. Pertunjukan ini biasnya dilakukan malam hari mulai pukul 22.00 hingga dini hari atau sekitar pukul 04.00, mengambil cerita dari evo Ramayana karya Valmiki atau Mahabrata karya Vyasa. Kesenian ini kerap dipergelarkan dalam rangka perayaan khitanan atau perkawinan. Pada perkembangannya, pertunjukan wayang golek ini pun kerap dipentaskan untuk event peresmian sebuah gedung atau institusi atau dalam event ulang tahun sebuah institusi.[1]

Sebagaimana alur cerita pewayangan umumnya, dalam pertunjukan wayang golek juga biasanya memiliki lakon-lakon baik galur maupun carangan yang bersumber dari cerita Ramayana dan Mahabarata dengan menggunakan bahasa Sunda dengan iringan gamelan Sunda (salendro), yang terdiri atas dua buah saron, sebuah peking, sebuah selentem, satu perangkat boning, satu perangkat boning rincik, satu perangkat kenong, sepasang gong, ditambah dengan seperangkat kendang, serta gambang dan rebab. [1]

2.1.1. Pola Pengadegan

Pola pengadegan wayang golek adalah sebagai berikut :

1. Tatalu
2. Babak unjal, paseban, dan bebegalan
3. Nagara sejen
4. Patepah
5. Perang gagal
6. Panawakan
7. Perang kembang
8. Perang raket
9. Tutug [1]

2.2. Game

Game adalah kata dalam bahasa inggris yaitu permainan. Permainan adalah sesuatu yang dapat dimainkan. Dalam pembahasan ini, akan membahas mengenai pengertian, genre, dan game edukasi.

2.2.1. Pengertian

Game adalah alat yang efektif untuk belajar karena game mampu menawarkan lingkungan hipotesis untuk siswa, di mana mereka dapat mengeksplorasi berbagai keputusan alternatif tanpa risiko kegagalan. Pemikiran dan tindakan digabungkan menjadi perilaku yang bertujuan mencapai suatu tujuan. Game mengajari kia cara menyusun strategi, mempertimbangkan altenatif, dan berpikir fleksibel. [2]

2.2.2. Komponen

Game mempunyai 5 komponen terpenting yaitu [3] :

1. Fitur

   Fitur merupakan salah satu hal yang membedakan setiap game yang ada. Fitur disini dapat merupakan fitur yang dapat membantu menggambarkan alan cerita game ke dalam bentuk-bentuk yang dapat dilihat maupun dirasakan.
2. *Gameplay*

   Gameplay dapat membantu para developer untuk mengetahui cara kerja suatu game, dimana fitur yang telah dibuat akan membentuk suatu gameplay
3. Interface

   Interface merupakan setiap tampilan yang ada di dalam game. Interface yang baik adalah dengan icon yang sangat minimum dan tidak membosankan para pemain game
4. Aturan

   Aturan merupakan kumpulan peraturan-peraturan dalam game.
5. Desain Level

   Desain level disini mencakup *style, background*, dan jalan cerita dari *game*

2.2.3. Tipe

Berikut ini adalah tipe *game* yang berhubungan dengan pembelajaran dan pendidikan [4] yaitu :

1. Arcade dan Platform Game dapat menjadi instrumen dalam perkembangan psikomotor dan spatial orientation.
2. Sports dan dynamic game dapat memudahkan koordinasi psikomotor dan meringankan stres.
3. Strategy dan role game dapat membantu menstimulasi motivasi dari dalam dan refleksi akan nilai dari game tersebut.
4. Puzzle dan question game dapat membantu untuk mengembangkan kemampuan untuk mengungkapkan alasan dan berfikir logis.
5. Game simulasi dapat membantu dalam perkembangan semua kemampuan intelektual.

Disamping itu, *game* juga dapat menekankan pemain untuk melakukan kegiatan :

1. Reading (membaca) : Game juga dapat digunakan untuk mempromosikan membaca buku sesuai dengan game yang sedang dimainkan, misalnya lord of the rings.
2. Logical Thinking : Game membantu dalam berfikir bagaimana memcahkan masalah dengan mengusulkan strategi, mengorganisir elemen dalam mengantisipasi tujuan.
3. Observation : Pemain menggunakan kemampuan observasi dalam bermain dengan mengamati perbedaan visual dan ruang serta jumlah elemen dalam layar.
4. Spatially, Geography : Mengembangkan kemampuan pemain dalam membaca peta dan mengenali bentuk ruang

2.3. Mobile

Penggunaan perangkat *mobile* seperti *smarthphone* dan tablet kian meningkat. Menurut Kementrian Komunikasi dan Informatika, angka masyarakat pengguna gadget di Indonesia menyentuh 240 juta unit mengingat jumlah penduduk Indonesia yang saat ini berada di kisaran angka 230 juta jiwa. [5]

Hal lainnya yaitu mengakses internet mengalami pertumbuhan yang cepat, mulai dari 40 % pada 2012 menjadi 68% pada 2016 . [6]

Walaupun kecepatan akses masih tebilang biasa saja dan tarif internet di Indonesia masih relatif mahal. Pengguna *smartphone* di Indonesia sangat suka mengunduh aplikasi. Rata-rata pengguna mengunduh aplikasi yaitu 0,65 aplikasi per hari. Kebanyakan paling banyak diunduh pengguna *smartphone* adalah game dengan persentase mencapai 43,71% dan yang kedua adalah tool dengan angka 8,96 %. [7]

2.4. Smartphone

Smartphone adalah telepon genggam yang mempunyai kemampuan tingkat tinggi, kadang-kadang dengan fungsi yang menyerupai komputer. *Smartphone* juga merupakan telepon yang bekerja menggunakan seluruh perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. *Smartphone* menyajikan fitur canggih seperti surel (suart elektronik), internet, dan kemampuan membaca buku elektronik.

Macam-macam sistem operasi *smartphone* adalah sebagai berikut :

1. Android
2. Blackberry
3. iOS
4. Windows Phone
5. Bada
6. FireFox
7. MeeGo
8. Palm
9. Symbian
10. Tizen
11. Ubuntu [8]

2.5. Android

Android merupakan sistem operasi yang mendominasi peredaran *smartphone* di tanah air dengan pembagian pasar sebesar 59,91 %. [9]

Android adalah sistem operasi untuk perangkat mobile berbasis linuk yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. [10]

2.5.1. Sejarah

Awalnya, Google Inc. Membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qulcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat mobile. Dilain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi erangkat lunak dan *open platform* perangkat seluler.

Sekitar September 2007 Google mengenalkan *Nexus One*, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Pada masa saat ini sebagian besar *vendor-vendor smartphone* sudah memproduksi *smartphone* berbasis android, *vendor-vendor* itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, Acer, Philips, T-Mobile, Nexian, Imo, Asus dan masih banyak lagi *vendor smartphone* di dunia yang memproduksi android. Karena android itu adalah sistem operasi yang *open source* sehingga bebas didistibusikan dan dipakai oleh *vendor* manapun. [10]

2.5.2. Versi

Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut:

1. Android versi 1.0 (2008)
2. Android versi 1.5 *Cupcake* (2009)
3. Android versi 1.6 *Donut* (2009)
4. Android versi 2.0 *Eclair* (2009)
5. Android versi 2.2 *Froyo* (2010)
6. Android versi 2.3 *Gingerbread* (2010)
7. Android versi 3.0 *Honeycomb* (2011)
8. Android versi 4.0 *Ice Cream Sandwich* (2011)
9. Android versi 4.1 *Jelly Bean* (2012)
10. Android versi 4.4 *KitKat* (2013)
11. Android versi 5.0 *Lollipop* (2014)
12. Android versi 6.0 *Marshmallow* (2015

13. Android versi 7.0 *Nougat* (2016) [11]

2.6. OOP (Object Oriented Programming)

OOP adalah suatu cara untuk mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Setiap objek mempunyai sifat yang melekat pada identitasnya. [12] Pendekatan berorientasi objek merupakan paradigma baru dalam rekayasa perangkat lunak yang memandang sistem sebagai kumpulan objek-objek diskrit yang saling berinteraksi. Yang dimaksud beriorientasi objek adalah bahwa mengorganisasikan perangkat lunak sebagai kumpulan objek-objek diksrit yang berkerja sama antara informasi atau struktur data dan perilaku yang mengaturnya. [13]

2.6.1. Objek (object)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek juga bisa merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.[14]

Semua objek mempunyai identitas yang berbeda dengan lainnya. Misalnya saja dua apel mempunyai warna, permukaan dan tekstur yang sama tapi tetap saja dipandang sebagai individual apel. [12]

2.6.2. Kelas

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (relationship) dan arti. [14]

2.6.3. Metode (method)

Metode (*method*) disebut juga service atau operator adalah prosedur atau fungsi seperti yang terdapat dalam bahasa pascal, tetapi cara kerjanya agak

berlainan. Metode adalah subprogram yang tergabung dalam objek bersama-sama dengan atribut. [12]

Sebuah kelas boleh memiliki lebih dari satu metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Metode dapat berasal dari event, aktivitas atau aksi keadaan, fungsi, atau kelakuan dunia nyata. Contoh metode atau operasi misalnya Read, Write, Move, Copy, dan sebagainya. [14]

2.6.4. Atribut

Atribut adalah variable global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut berarti properti, kualitas dan karakteristik yang dapat ditentukan pada orang atau barang dan data dimana objek atau kelas memiliki sebuah nilai. [14]

2.6.5. Abstraksi

Abstraksi bisa juga prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan. [14]

Tujuan abstraksi adalah untuk melakukan isolasi aspek-aspek penting demi suatu maksud dan menindak aspek-aspek yang tidak penting. Abstraksi harus selalu mempunyai maksud, karena maksud yang menentukan apa yang penting dan apa yang tidak penting. Model yang bagus adalah model yang dapat menagkap aspek-aspek paling penting dari masalah dan menghilangkan aspek-aspek lain yang tidak penting.[13]

2.6.6. Enkapsulasi (encapsulation)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya. [14] Enkapsulasi merupakan daar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas dalam bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. [12]

2.6.7. Polimorfisme (polymorphism)

Polimorfisme yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyi bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda. Polimorfisme berarti suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program. [14]

2.6.8. Pewarisan (Inheritance)

Pewarisan adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek induk diturunkan kepada anak objek, demikian seterusnya. [12] Pewarisan merupakan mekanisme yang memungkinkan satu objek mewarisi sebagaian atau seluruh definisi dan objek lain sebagai bagian dan dirinya. [14]

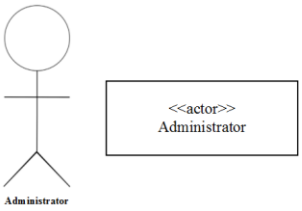
2.7. UML *(*Unified Modeling Language)

UML adalah bahasa grafis untuk mendokumentasi, menspesifikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstaksi, tidak bergantung proses pengenbangan, tidak bergantung bahasa teknologi. Pemaduan beberapa di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan lewat XML (XMI). Standar UML dikelola oleh OMG (*Object Management Group*). UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek [14] UML menyediakan berbagai diagram diantaranya sebagai berikut:

2.7.1. Diagram Use Case

Diagram use case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang berhak menggunakan fungsi-fungsi itu. Ada dua hal yang utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case. [14]

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambaran orang, tapi aktor belum tentu merupakan orang.



Gambar 2.1 Aktor

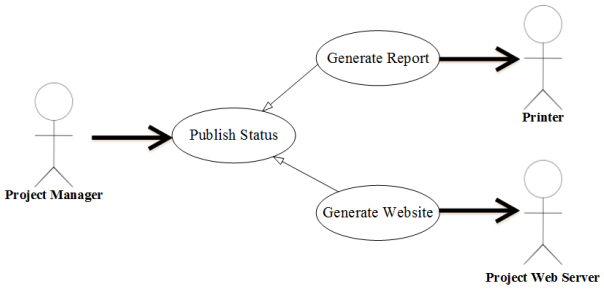
2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.



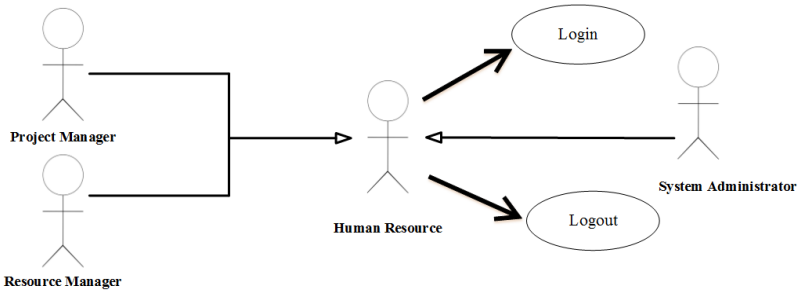
Gambar 2.2 User Registration Use Case

Adapun relasi yang terdapat pada *use diagram* yaitu :

1. *Generalization* disebut sebagai pewarisan, sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.



Gambar 2.3 Generalisasi Use Case



Gambar 2.4 Generalisasi Aktor

2. *Association*, menghubungkan link antar element
3. *Depedency*, sebuah element bergantung dalam beberapa cara ke element lainnya.
4. *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya

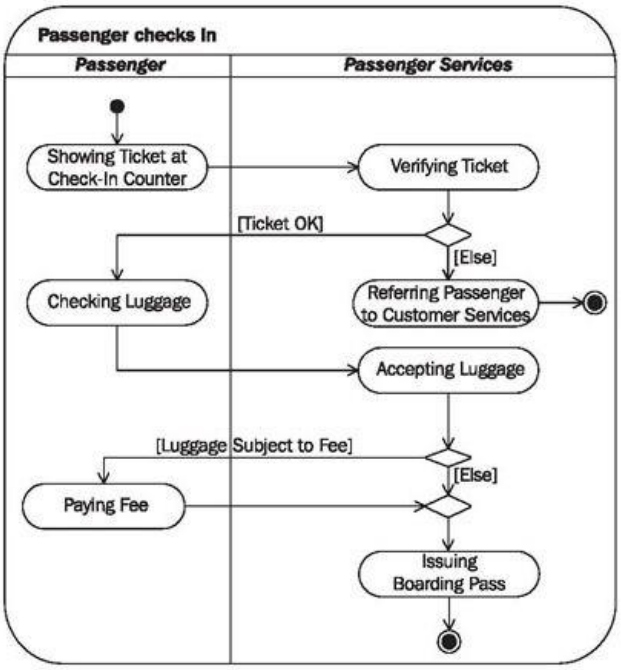
Tipe relasi yang mungkin terjadi pada *use case* diagram :

1. <<include>>, yaitu relasi *use case* terhadap *use case* lainnya, dimana *use case* yang di tambahkan memerlukan *use case* ini untuk menjalankannya.
2. <<extends>>, yaitu relasi *use case* tambahan ke sebuah *use case*, dimana *use case* yang di tambahkan dapat berdiri sendiri.

2.7.2. Diagram Aktivitas

Diagram aktivitas adalah suatu aktivitas dari sebuah sistem atu proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, tapi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas banyak digunakan untuk mendefinisikan hal-hal berikut [14] :

1. Rancangan menu yang ditampilkan pada perangkat lunak
2. Setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.



Gambar 2.5 Activity Diagram

2.7.3. Diagram Kelas

Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. [14]

Dalam UML, *class* digambarkan dengan segi empat yang dibagi beberapa bagian seperti pada Gambar 2.5

NamaKelas
- DaftarAttribut
+ DaftarOperasi()

Gambar 2.6 Class Diagram

Atribut dan operasi dapat memiliki salah satu sifat yaitu :

1. *Private*, sifat ini tidak dapat dipanggil dari luar *class* kecuali dengan *class* yang bersangkutan.
2. *Protected*, sifat ini hanya bisa dipanggil oleh *class* yang bersangkutan dan class yang mewarisinya.
3. *Public*, sifat ini dapat dipanggil oleh semua *class*.

2.7.4. Diagram Sekuen

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. [14]

Diagram sekuen menunjukkan objek sebagai garis vertikal dan tiap kejadian sebagai panah horisontal dari objek pengirim ke objek penerima. Waktu berlalu dari atas ke bawah dengan lama waktu tidak relevan. Diagram ini hanya menunjukkan barisan kejadian, bukan pewaktuan nyata. Kecuali untuk sistem waktu nyata yang mengharuskan konstrain barisan kejadian. [14]

2.8. BPMN (Business Process Model and Notation)

Pada tahun 2001 BPMI (Business Process Management Initiative) mulai mengembangkan BPML (Business Process Modeling Language). Pada bulan Mei 2004, BPMN versi 1.0 di rilis ke publik. Sejak itu lebih dari 50 perusahaan telah mengembangkan dan menerapkan BPMN sebagai standar mereka. Pada bulan Februari 2006 BPMN menjadi milik OMG (Open Management Group). Pada bulan Februari 2008. Pihak OMG merilis versi 1.1 dan selanjutnya adalah versi 1.2. BPMN 2.0 dirilis pada awal pertengahan tahun 2009. [15]

2.9. Checklists Testing

Checklists testing berupa kumpulan kasus uji yang mencakup area permainan tertentu atau permainan yang akan diuji. Pengujian ini memungkinkan masing-masing fungsi diperiksa. *Checklists testing* memuat daftar periksa dari suatu game yang akan diuji dari awal sampai akhir. Hasil dari pengujian *checklists* yaitu *Pass* dan *Fail*. *Pass* ketika fungsi sesuai atau berhasil dan *Fail* ketika fungsi gagal. [16] Contoh dari *checklists testing* ditunjukkan pada tabel 2.1 berikut.

Tabel 2. 1 Checklists Testing

Test	Hasil
Audio	Fail
Collision	Fail
Main Weapon	Fail
Save	Pass

2.10. Non Functional Requirements

Persyaratan non fungsional menentukan atribut dari sistem yang dihasilkan. Non fungsional dibagi menjadi 3 yaitu *Product Requirements*, *Process Requirements,* dan *External Requirements*. [17]

Adapun penjelasan dari ketiga hal tersebut adalah sebagai berikut.

1. *Product Requirements* adalah karakteristk yang harus dimiliki sistem atau subsistem. Contohnya sebagai berikut.
   a. Sistem x harus memiliki ketersediaan 999/1000 atau 99%. Ini adalah persyaratan keandalan yang berarti bahwa dari setiap 1000 permintaan untuk layanan ini, 999 harus dipuaskan.
   b. Sistem Y akan memproses minimal 8 transaksi per detik. Ini adalah persyaratan kinerja.
   c. Sistem harus mengenkripsi semua komunikasi eksternal dengan menggunakan algoritma RSA.
   d. Sistem menyediakan bantuan berupa pesan kesalahan atau pesan informasi untuk menjelaskan semua fungsi dan cara mencapai fungsi tersebut.
2. *Process Requirements* adalah kendala yang ditempatkan pada proses pengembangan sistem. Process requirements meliputi metode pengembangan, alat pengembangan yang digunakan, dan laporan manajemen. Contohnya sebagai berikut.
   a. Sistem dikembangkan secara exsplisit dengan standar ISO 9000.
   b. Sistem ini dikembangkan dengan menggunakan perangkat XYZ.
   c. Sistem yang diidentifikasi harus diproduksi setiap dua minggu sekali.

3. External Requirement adalah suatu standar yang ditempatkan sesuai produk dan prosesnya, dan berasal dari lingkungan dimana sistem dikembangkan. Contohnya sebagai berikut.
   a. Sistem data medis: Petugas perlindungan data organisasi harus memastikan bahwa semua data dipelihara sesuai dengan undang-undang perlindungan data sebelum sistem dioperasikan.
   b. Sistem dapat digunakan untuk semua pengguna yaitu setiap orang dapat berbagi
   c. Sistem web yang dikembangkan untuk pemerintah federal NRW harus mematuhi undang-undang persamaan perlakuan terhadap orang-orang penyandang disabilitas.

2.11. Construct 2

Construct 2 merupakan sebuah game engine 2D untuk HTML 5 yang dikembangkan oleh Scirra Ltd, sebuah perusahaan *start up* yang bermarkas di London. Game engine ini dikembangkan dengan konsep *bahavior dan event attachment*. [18]

Construct pertama kali dirilis dengan nama Construct Classic pada tahun 2007. Pada tahun 2011, Scirra merilis versi terbaru Construct yaitu Construct 2 (atau biasa juga dikenal dengan nama C2. [18]

2.12. Game Development Life Cycle

Menurut Blitz Game Studio, metodologi pengembangan game terdiri dari enam tahap, yaitu pitch, pre-production, main production, alpha, beta, dan master. Keenam tahap ini harus berurutan dalam praktiknya. Berikut adalah GDLC yang ditunjukkan seperti pada gambar 1.2



Gambar 2.7 Game Development Life Cycle

Berikut adalah penjelasan dari Gambar 2.6 :

1. *Pitch*

   Tahap pitch adalah tahap persiapan / konsep dasar dari game yang akan dibangun.

2. *Pre-Production*

   Tahap pre-production adalah tahap perancangan desain game.

3. *Main Production*

   Tahap main production adalah tahap pembuatan aset dan implementasi.

4. *Alpha*

   Tahap alpha adalah tahap pengujian dimana pengecekan berupa aset yang masih belum selesai tetapi game yang dimainkan sudah bisa sampai akhir.

5. *Beta*

   Tahap beta adalah tahap pengujian dimana semua aset sudah ada dan sudah selesai, namun masih ada kesalah (*bugs*)

6. *Master*

   Tahap master adalah tahap pendistribusian game. [19]