BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya / Kajian Pustaka

Penelitian sebelumnya banyak juga yang membuat *game* edukasi sebagai alternatif membantu seseorang dalam proses belajar dengan cara belajar sambil bermain *game*. *Game* edukasi memiliki tujuan penting yaitu khusus dirancang untuk mengajarkan *user* pada pembelajaran tertentu dan pemahaman serta membimbing *user* dalam meningkatkan skill dan memotivasi *user* untuk memainkannya.

Pada Penelitian yang dilakukan oleh D. Damayanti, M. F. Akbar, dan H. Sulistiani, pada tahun 2020 peneliti menggunakan construct 2 sebagai *tool* pembuatan *game*. *Game* dibuat untuk membantu pengguna dalam mengenal hewan langka, serta memberikan edukasi kepada *user* dengan cara bermain *game* di *smartphone* dan mendapatkan edukasi[2]. Akan tetapi pada penelitian masih memiliki kekurangan yaitu untuk hewan yang diperkenalkan masih belum banyak atau masih kurang.

Pada penelitian yang dilakukan oleh P. Putra, H. Mubarok, A. Rachman, pada tahun 2020 yang bertujuan untuk media hiburan yang mengenalkan budaya yang ada dijawa barat serta melatih kemampuan dan imajinasi, memperbanyak wawasan, pengetahuan, logika, seni, dan daya ingat. Menggunakan metode *Luther-Sutopo* sebagai pengembang multimedia. Dalam pembuatan *game* menggunakan *tools* construct 2[3]. Akan tetapi pada penelitian ini proses pengumpulan data masih kurang rinci dalam menjelaskan proses mendapatkan data.

Pada penelitian yang dilakukan oleh R. F. Wijaya, R. B. Utomo, D. Y. Niska, dan K. Khairul, pada tahun 2019 yang bertujuan sebagai media bantu petani-petani muda yang masih minim pengetahuan tentang bagaimana melakukan kegiatan petani seperti menanam dan memelihara tanaman / tumbuhan yang baik agar menghasilkan hasil panen yang baik. Menggunakan UML untuk memvisualisasikan *game* yang akan dibuat[4].

Akan tetapi masih memiliki kekurangan pada penelitian yaitu belum ada tahap pengujian dari aplikasi yang telah di buat.

Pada penelitian yang dilakukan oleh A. Chusyairi, J. S. L. Wibowo, dan A. K. Winata, pada tahun 2020. Dalam penelitian ini peneliti mempunyai tujuan yang sangat baik untuk *game* yang dibuat yaitu untuk memperkenalkan budaya dari banyuwangi kepada remaja(penerus). *Game* ini dibuat menggunakan metode GDLC yang setiap langkah-langkahnya ada inisiasi, pre-produksi, produksi, pengujian, *beta* dan rilis. *Game* ini sangat bermanfaat untuk remaja (penerus)dari banyuwangi karena bisa menambahkan wawasan kepada penerus dan bisa antusias terhadap budaya yang ada di banyuwangi[5]. Pada proses penelitian pada metode GDLC peneliti hanya membahasa secara ringkas tidak menjelaskan secara detail bagaimana tahap pengujian dan lain sebagainya.

Pada penelitian yang dilakukan V. Nirwana, Ira; Karnadi, pada tahun 2021 yang merancang sebuah *game* edukasi dengan menggunakan metode MDLC yang memiliki tahapan seperti konsep, desain, mengumpulkan data, perancangan dengan *tools* construct 2 dan pengujian serta menggunakan UML dalam memvisualisasikan *game* dengan membuat diagram-diagram. *Game* ini dibuat memiliki maksud untuk menambah wawasan dan mengukur logika *user* dalam membaca gambar[6]. Namun penelitian masih kurang menjelaskan proses penelitian secara detail mendapatkan hasil uji coba.

Dari penelitian-penelitan sebelumnya terdapat beberapa penelitian yang menggunakan metode GDLC dan MDLC. Pada penelitian ini nantinya peneliti akan memilih metode GDLC karena menurut peneliti metode ini bisa menghasilkan penelitian yang memiliki hasil yang bagus. Karena perbedaan dari GDLC dan MDLC itu terletak pada prosesnya yaitu GDLC memiliki proses beta untuk meminta orang lain menilai kelayakan game dan untuk MDLC itu hanya sebatas uji coba.

Tabel 2.1 Penelitian Terdahulu

No	judul	*Comparing*	*Contrasting*	*Criticize*	*Synthesize*	*Summarize*
1	*Game* Edukasi Pengenalan Hewan Langka Berbasis Android Menggunakan Construct 2[2]	Penelitian ini menghasilkan *Game* Edukasi Pengenalan Hewan Langka Berbasis Android Menggunakan Construct 2. Bertujuan untuk memperkenalkan hewan langka, dan membuat aplikasi yang dapat menampilkan gambar, suara disertai permainan puzzle dan kuis berupa tebak nama hewan	Menggunakan tahapan dengan model pengembangan sistem *agiledan* model UML (*Unified Modelling Language*).	Hewan yang di perkenalkan pada aplikasinya masih kurang	Peneliti melakukan tinjauan pustaka, wawancara dan observasi untuk mengumpulkan dataset	Berdasarkan pengujian aplikasi *game* edukasi pengenalan hewan langka ini dapat membantu memperkenalkan hewan langka kepada masyarakat khususnya pada anak-anak, dengan perolehan presentasi penilaian pengujian sebesar 93,21%.
2	Aplikasi Multimedia Berbasis *Game* Edukasi Menggunakan Construct 2 Untuk Pengenalan Tempat Wisata Budaya Jawa Barat Pada Anak Usia Dini[3]	Menyediakan sebuah informasi mengenai tempat wisata seni, dan budaya *game*	Metode Luther-Sutopo, metode ini merupakan salah satu metode pengembang multimedia	Pada proses pengumpulan data kurang dijelaskan bagaimana cara mendapatkan data-data	Pengumpulan bahan data mulai dari *text, image* dan *animation.* Pada image menggunakan contoh *background* pada situs https://pixabay.com	Aplikasi ini memberikan sebuah pendidikan kepada pengguna dengan cara yang menyenangkan dan tidak membosankan serta. Memberikan sebuah pengetahuan mengenai wisata seni dan budaya yang ada di 4 kota di jawa barat.

3	Aplikasi Petani Pintar Dalam Monitoring Dan Pembelajaran Budidaya Padi Berbasis Android[4]	Aplikasi yang dibuat ini terdapat dua jenis konten penting, yaitu pembelajaran, dan monitoring.	Menggunakan android studio dan bahasa pemrograman java	Pada hasil hanya membahas tentang tampilaan aplikasinya saja tanpa melakukan pengujian	Teknik pengumpulan data dari penelitian ini dilakukan dengan beberapa metode ini: Observasi Kuesioner Wawancara *Library Research*	Pada penelitian ini menyajikan informasi penting seperti teknik budidaya, penyakit, hama, dan manfaat yang berkaitan pada tanaman padi
4	*Game* Gandrung Strories Untuk Edukasi Kebudayaan Menggunakan Metode GDLC[5]	Menggunakan GDLC (*GameDevelopment Life Cycle*) dalam penelititan	Metode GDLC memudahkan peneliti dalam melakukan penelitian dan membuat *game*	Pada proses peneilitian tidak dijelaskan secara detail	Pada tahap inisiasi peneliti memilih target penggunanya generasi muda (anak remaja)	Output dari penelitian ini menghasilkan *game* yang dapat memberikan wawasan budaya yang ada di banyuwangi.
5	Perancangan *Game* Edukasi Pengembangan Kemampuan Logika Berbasis Android[6]	MDLC (*Multimedia Development Life Cycle*) sebagai metode pengembang yang digunakan peneliti	Aplikasi yang digunakan construct2 dalam merancang *game*	Proses peneliti hanya membatasi sampai hasil uji saja dan tidak menjelaskan proses testing	Pengumpulan data mengunakan tahap observasi dan wawancara	Setelah dilakukan proses uji *game* sudah diinstal dan *game* juga termasuk *user friendly*

2.2 Dasar Teori

Kajian atau penjelasan yang membahas dasar beberapa teori yang digunakan peneliti pada penelitian sebagai berikut:

2.2.1 Rekayasa Perangkat Lunak (*Software Engineering*)

2.2.1.1 Definisi Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya berkerja secara *efisien* menggunakan mesin. Rekayasa perangkat lunak lebih focus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat pada pelanggan (*costumer*). Adapun ilmu *computer* lebih *focus* pada teori dan konsep dasar perangkat computer[7]:

Rekayasa perangkat lunak lebih *focus* pada bagaimana membuat perangkat lunak yang memenuhi kriteria.

1. Dapat terus dipelihara setelah peraangkat lunak selesai dibuat seiring waktu berkembangnya teknologi dan lingkungan (*maintainability*).
2. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*dependability*).
3. Efisien dari segi sumber daya dan pengguna.
4. Kemampuan untuk dipakai sesuai dengan kebutuhan (*usability*).

Dari kriteria diatas maka perangkat lunak yang baik adalah perngkat lunak yang dapat memenuhi kebutuhan pelanggan (*customer*) atau *user* (pemakai perangkat lunak) atau berorientasi pada pelanggan atau pemakai perangkat lunak, bukan berorientasi pada pembuat atau pengembang perangkat lunak[7].

2.2.1.2 Tujuan Rekayasa Perangkat Lunak

Secara umum tujuan rekayasa perangkat lunak tidak berbeda dengan rekayasa yang lain. Mengungkapkan bahwa tujuan dari rekayasa perangkat lunak adalah sebagai berikut[7]:

1. Memperoleh biaya produksi perangkat lunak yang rendah.
2. Menghasilkan perangkat lunak yang kinerjanya tinggi, andal dan tepat waktu.
3. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis *paltform*
4. Menghasilkan perangkat lunak yang biaya perawatannya rendah

2.2.2 UML

UML adalah Bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari *system* perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corps.* UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan [8].

2.2.2.1 Definisi *Unified Modeling Languange* (UML)

Unified Modeling Language (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah system dengan menggunakan diagram dan teks-teks pendukung, UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataanya UML paling banyak digunakan pada metodologi berorientasi objek[9].

2.2.2.2 Sejarah UML

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemograman ini kurang berkembang, namun dengan

kemunculanya telah memberikan sumbangan yang besar pada *developer* pengembang bahasa pemrograman berorientasi objek selanjutnya[9].

Perkembangan aktif dari pemograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemograman *smaltalk* pada awal 1980-an yang kemudian diikuti dengan dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eifeel dan CLOS. Sekitar 5 tahun setelah *Smalltalk* banyak pengembangan metode berorientasi objek[9].

Karena Banyaknya metodelogi yang berkembang pesat saat itu, maka muncul ide untuk membuat bahasa yang dapat dimengerti semua orang. Usaha penyatuan ini banyak mengambil dari metodologi yang berkembang saat itu. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti *Modeling Technique* (OMT) dari Rumbaugh dan Booch (1991), konsep The *Classes, Responsibilities, Collaborators* (CRC) dari Rebecca Wirfs-Brock (1990) dan konsep-konsep lainya[9].

Pada 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standardisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusi yang cukup besar didalam metodologi berorientasi objek dan hal-hal yang terkait didalamnya[9].

2.2.2.3 Tujuan Penggunaan UML

Tujuan dari penggunaan UML (*Unified Modeling Language*) adalah sebagai berikut[9]:

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahsa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

4. UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi secara detail tentang *coding* program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (*reserve enginering*).

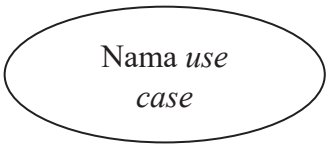
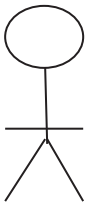
2.2.2.4 Diagram-Diagram Dalam UML

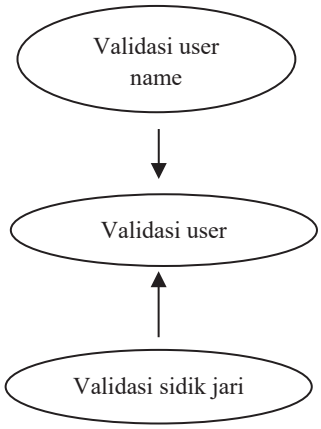
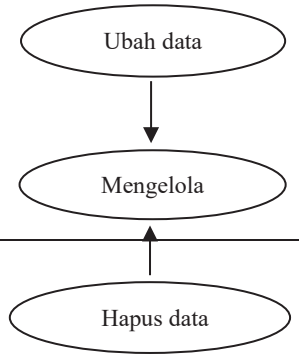
Berikut ini diagram-diagram yang ada dalam UML:

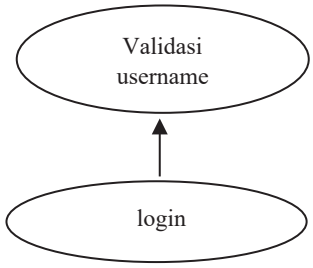
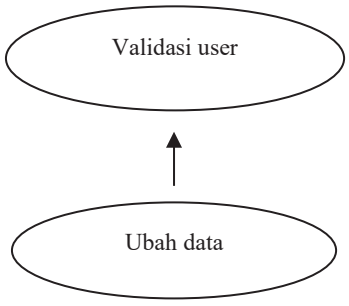
2.2.2.4.1 Use Case Diagram

Use case atau diagram use case merupakan pemodelan untuk kelakuan (*behavoir*) sistem informasi yang akan dibuat. *Use case* mendiskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang dibuat. Berikut adalah simbol-simbol yang ada pada diagram *use case*[9]:

Tabel 2.2 Simbol-Simbol *Use Case Diagram*

Simbol	Deskripsi
Use case  	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal *frase* nama *use case.*
Aktor / *actor*  Nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal *frase* nama aktor.

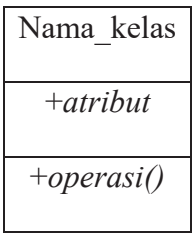
Asosiasi / *association*	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
Ekstensi / *exstend* <<*exstend*>>	Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri meski tanpa *use case* tambahan itu mirip dengan prinsip. *Inheritance* pada pemogram berorientasi objek, biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, missalnya:  Arah panah mengarah pada *use case* yang ditambahkan; biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya.
Generalisasi/*generalization*	Hubungan *generalisasi* dan *spesialisasi* (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya, misalnya; 

	Arah panah mengarah pada *use case* yang menjadi generalisasinya.
Menggunakan/*include/use case* *<<includee>>* *<<uses>>*	Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini, ada dua sudut pandang yang cukup besar mengenai *include* di *use case* : *Include* berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, misal pada kasus berikut :  *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case*. tambahan dijalankan, misal pada kasus berikut;  Kedua interprestasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interprestasi yang dibutuhkan.

2.2.2.4.2 Class Diagram

Classs Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi[9].

Tabel 2.3 Simbol-Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antarmuka / *interface*  Nama_*interface*	Sama dengan konsep *interface* dalam pemograman beroriantasi objek.
Asosiasi / *association*	Relasi antar kelas dengan makna umum, *asosiasi* biasanya juga disertai dengan *multiplicity.*
Asosiasi berarah / *directed association*	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, *asosiasi* biasanya juga disertai dengan *multiplicity.*
Generalisasi 	Relasi antar kelas dengan makna *generalisasi-spesialisasi* (umum-khusus).
Kebergantungan / *dependency* 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / *aggregation* 	Relasi antar kelas dengan makna semua-bagian (*whole-part*).

2.2.2.4.3 Statechart Diagram

Statechart diagram disebut diagram mesin status digunakan untuk menggambarkan perubahan status atau transaksi status dari sebuah mesin atau sistem perubahan tersebuh digambarkan dalam bentuk *graf* berarah. *Statechart diagram* cocok digunakan untuk menggambarkan alur interaksi pengguna dengan sistem. Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *statechart* diagram, dapat kita lihat pada table berikut ini[9]:

Tabel 2.4 Simbol-Simbol *Statechart Diagram*

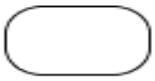
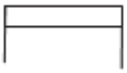
Simbol	Deskripsi
Start / status awal (*initial state*) 	*Start* atau initial *state* adalah *state* atau keadaan awal pada saat sistem mulai hidup.
End / status akhir (*final state*) 	*End* atau *final state* adalah *state* keadaan akhir dari daur hidup suatu sistem.
Event 	*Event* adalah kegiatan yang menyebabkan berubahnya status mesin.
State 	*State* atau status adalah keadaan sistem pada waktu tertentu.

2.2.2.4.4 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis.Berikut ini simbol-simbol yang

sering digunakan pada saat pembuatan *activity* diagram, dapat kita lihat pada tabel berikut ini[9]:

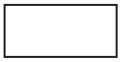
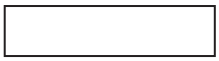
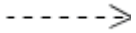
Tabel 2.5 Simbol-Simbol *Activity Diagram*

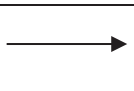
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabung menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang ada.

2.2.2.4.5 Squence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sisem (termasuk pengguna, *display*, dan sebagainya) berupa message yang digambarkan terhadap waktu. Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *sequence diagram*, dapat kita lihat pada tabel berikut ini[9]:

Tabel 2.6 Simbol-Simbol *Sequence Diagram*

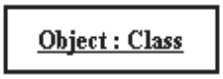
Simbol	Deskripsi
Aktor  atau 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem.
Lifeline	Menyatakan kehidupan suatu objek.
Objek 	Merupakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
Pesan tipe *create* 	Menyatakan suatu objek memuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe *call* 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain.
Pesan tipe *send* 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya.
Pesan tipe *return* 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembali ke objek tertentu.
Pesan tipe destroy	Menyatakan suatu objek mengakhiri

	hidup objek yang lain.

2.2.2.4.6 Collaboration Diagram

Collabration diagram menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan. Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *collaboration* diagram, dapat kita lihat pada tabel berikut ini[9]

Tabel 2.7 Simbol-Simbol *Collaboration Diagram*

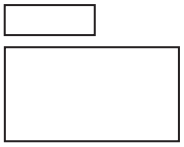
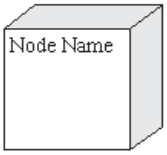
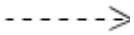
Simbol	Deskripsi
Objek 	Objek yang melakukan interaksi pesan.
Link	Relasi antara objek yang menghubungkan objek satu dengan lainnya atau dengan diri sendiri.
Stimulus	Arah pesan yang terjadi, jika pada suati link ada dua arah pesan yang berbeda maka arah juga digambarkan dua arah pada dua sisi link.

2.2.2.4.7 Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Berikut ini simbol-simbol yang sering digunakan

pada saat pembuatan *deployment diagram*, dapat kita lihat pada tabel berikut ini[9]:

Tabel 2.8 Simbol-Simbol *Deployment Diagram*

Simbol	Deskripsi
package 	*Package* merupakan sebuah bungkusan dari satu atau lebih node.
Node 	*Node* adalah sumber daya fisik yang menjalankan kode komponen.
Dependency 	Ketergantungan antara node, arah panah mengarah pada node yang dipakai.
link	Relasi antara node.

2.2.3 Game

Game dalam pengertian secara umum merupakan sebuah media penghibur yang dapat dilakukan setiap kalangan juga berguna sebagai penghilang rasa jenuh. Beberapa manfaat dari game yang lain seperti halnya melatih memecahkan masalah, menambahkan konsentrasi, pengembangan otak, melatih kecepatan maupun ketepatan dan lain sebagainya. Beberapa jenis dari *genre game* yang umum di kalangan masyarakat yaitu *adventure, action, role playing, strategy,*

vehicle simulations, sport, construction and management simulations, artificial life and puzzle games[10].

Saat ini teknologi yang berkembang pesat apalagi dipergunakan untuk bermain game adalah *mobilephone* atau yang kini dapat dikenal dalam sebutan *smartphone*. Perkembangan *smartphone* ini merujuk pada banyaknya penggunanya didunia. Perkembangan *smartphone* yang tinggi didunia ini, menjadikan alasan yang kuat guna membuat, merancang yang diikuti dengan pengembangkan aplikasi pada *smartphone*. Umumnya, *smartphone* lebih sering digunakan sebagai aktivitas untuk mempermudah pekerjaan sehari-hari dikarenakan *smartphone* dapat dibawa kemana saja atau ringan dilengkapi dengan fitur yang tidak kalah dengan komputer[10].

2.2.3.1 Game Edukasi

Game edukasi merupakan sebuah permainan yang dibuat dan dirancang khusus untuk dijadikan sebuah media yang digunakan untuk mengajar orang melalui materi yang berisikan suara, teks, gambar, vidio, dan animasi, yang materinya membahas suatu subjek tertentu, yang memiliki tujuan untuk memperluas konsep, memberikan pemahaman yang lebih baik dari materi yang mengajarkan sebuah peristiwa sejarah maupun budaya, dan dapat pula mengajarkan pengguna *game* edukasi ini agar menjadi lebih baik. Hal tersebut dikarenkan mereka dapat bermain sambil belajar dengan mudah[3].

Game edukasi adalah *game* yang dirancang atau dibuat untuk berpikir dan termasuk meningkatkan konsentrasi serta memecahkan masalah. *Game* edukasi merupakan salah satu media yang digunakan untuk mengajarkan, meningkatkan pemahaman penggunanya melalui media yang unik dan menarik. *Game* edukasi adalah jenis permainan edukatif yang biasanya diperlihatkan kepada anak-anak, jadi permainan warna sangat penting di sini, bukan karena kesulitan yang penting. Dengan demikian, *game* edukasi merupakan salah satu bentuk dari *game* yang dapat bermanfaat untuk mendukung proses belajar mengajar dengan cara yang

lebih menyenangkan dan kreatif, serta digunakan untuk memberikan edukasi atau meningkatkan pengetahuan pengguna melalui sarana yang menarik[11]. Permainan dijalankan pada perangkat komputer yang dirancang dan diproduksi khusus untuk digunakan sebagai media yang digunakan untuk mengajar orang melalui materi yang berisi suara, teks, gambar, video, dan animasi. Topik utama yang terkait dengan objek tertentu bertujuan untuk dapat memperluas wawasan konsep, pemahaman serta memudahkan penyajian materi dalam proses belajar mengajar[12].

Permainan edukatif adalah permainan yang membangkitkan minat belajar dan bermain siswa, sehingga dengan emosi yang gembira diharapkan dapat lebih mudah memahami materi yang disajikan. Seorang ahli menjelaskan bahwa *game* edukasi merupakan salah satu tema *game* yang berusaha memberikan nilai edukasi pada *game*, sehingga pada awalnya *game* hanya berfungsi sebagai sarana hiburan, pada akhirnya dapat juga digunakan sebagai sarana pembelajaran atau pelatihan [13]. Beberapa fungsi penting dari permainan edukatif, diantaranya adalah sebagai berikut:

a. Memberikan pengetahuan kepada anak melalui proses belajar dengan bermain *game*.
b. Merangsang perkembangan daya pikir, daya cipta, dan kebahasaan, guna menumbuhkan sikap, kecerdasan, dan moral yang baik.
c. Ciptakan lingkungan permainan yang menarik, berikan rasa aman dan kesenangan.
d. Meningkatkan kualitas pembelajaran anak[14].

2.2.3.2 Kriteria *Game*

Game juga mempunyai kiriteria yaitu :

a. Setiap user pasti memiliki ketertarikan.
b. Setiap user harus mengetahui aturan permainan.
c. Setiap user pasti mempunyai strategi yang berbeda-beda.
d. Hasil permainan atau pemenang dipengaruhi dari persetujuan / pilihan-pilihan yang telah dibuat.

2.2.3.3 Jenis-Jenis *Game*

Jenis *(Genre) game* merupakan *style* dari suatu *game*. Dibawah ini terdapat jenis-jenis game, yaitu:

1. *Sports Game*

   Permainan jenis ini sering bersifat kompetitif dan sering dapat dimainkan oleh banyak pemain pada saat yang bersamaan, secara individu atau dalam tim. Seperti namanya, sebagian besar *game* dalam kategori ini adalah *game* olahraga[11].

2. *Massively Multiplayer Online Firstperson shooter games* (MMOFPS)

   Jenis *game online* ini menggunakan sudut pandang orang pertama sehingga pemain seolah-olah berada dalam permainan dari sudut pandang karakter yang dimainkan, dimana setiap karakter memiliki kemampuan yang berbeda dalam hal akurasi, refleks dan kemampuan lainnya. *Game* ini dapat melibatkan banyak orang dan biasanya merupakan *game* yang berlatar perang dengan senjata militer. Contoh *game* tersebut termasuk *Counter Strike, Call of Duty, Point Blank, Quake, Blood, Unreal.*

3. *Massively Multiplayer Online Realtime strategy games* (MMORTS)

   Jenis permainan ini menekankan pada strategi hebat pemain. *Game* ini memiliki fitur dimana pemain harus menentukan strategi permainan. Dalam RTS, tema permainan dapat berupa sejarah (misalnya *seri Age of Empires*), fantasi (misalnya *Warcraft*) dan fiksi ilmiah (misalnya *Star Wars*).

4. *Cross-platform online play*

   Jenis ini dapat dimainkan secara *online* dengan perangkat yang berbeda Ketika mesin *game* konsol ini mulai berkembang menjadi

komputer dengan jaringan *open source*, seperti *Dreamcast, PlayStation 2* dan *Xbox* memiliki fungsi online. Contoh: *Need for Speed Underground*, dapat dialirkan dari *PC* atau *Xbox 360*.

5. *Massively Multiplayer Online Browser Game*

Jenis permainan ini dimainkan di browser seperti *Mozilla Firefox, Opera* atau *Internet Explorer*. *Game* pemain tunggal sederhana dapat dimainkan di browser menggunakan teknologi pengkodean HTML dan HTML (JavaScript, ASP, PHP, MySQL). Perkembangan teknologi grafis berbasis web seperti Flash dan Java memunculkan *game-game* yang dikenal dengan nama "*Flash games*" atau "*Game Java*" yang menjadi sangat populer. *Game* sederhana seperti *PacMan* bahkan telah dibuat ulang menggunakan plugin di situs web. *game browser* baru menggunakan teknologi web seperti multipemain interaktif. Ajax mengaktifkan halaman web. *Game browser* baru yang menggunakan teknologi web seperti interaksi multipemain.

6. *Simulation games*

Jenis permainan ini bertujuan untuk memberikan pengalaman melalui simulasi. Ada beberapa jenis *game* simulasi, antara lain *game* simulasi kehidupan, *game* simulasi konstruksi dan manajemen, dan simulasi kendaraan. Dalam *game* simulasi kehidupan, pemain bertanggung jawab atas karakter atau karakter dan memenuhi kebutuhan karakter seperti di kehidupan nyata, tetapi di dunia virtual. Karakter memiliki kebutuhan dan kehidupan seperti manusia, seperti bekerja, bertukar, makan, berbelanja, dll. Biasanya karakter-karakter tersebut hidup di dunia maya yang penuh dengan karakter yang diperankan oleh pemain lain. Contoh permainannya adalah *Second Life*.

7. *Massively multiplayer online games (MMOG)*

Pemain dimainkan di dunia skala besar (>100 pemain) di mana setiap pemain dapat berinteraksi langsung seperti dunia nyata.

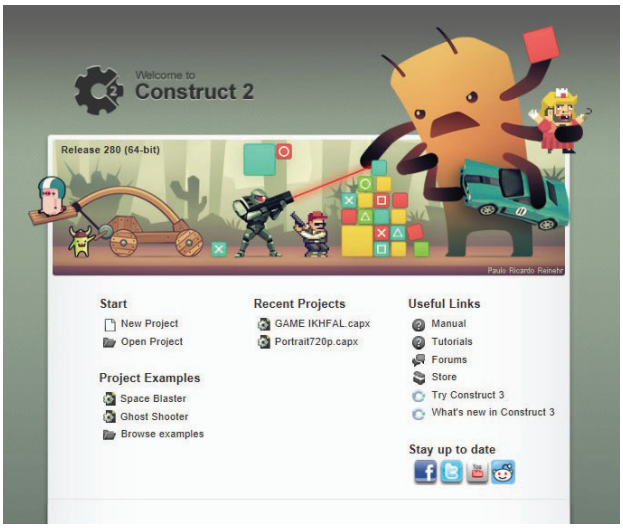
MMOG muncul dengan pertumbuhan akses internet berkecepatan tinggi di negara-negara maju, memungkinkan ratusan, bahkan ribuan pemain untuk bermain bersama. MMOG sendiri juga memiliki banyak jenis seperti MMORPG, MMORTS, MMOFPS, MMOSG[15].

8. *Massively Multiplayer Online Role-playing games* (MMORPG)

Massively Multiplayer Online Role-playing games (MMORPG) adalah jenis *game* dengan dunia *virtual* tiga dimensi berkelanjutan dan lingkungan simulasi yang dapat diakses melalui *web browser* atau *platform* lainnya. MMORPG memungkinkan pemain untuk terhubung satu sama lain dan melakukan aktivitas bersama di dunia *game*[16]. .

2.2.4 Construct 2

Construct 2 adalah sebuah *tools* atau *software* pembuat *game* berbasis HTML5 yang dikhususkan untuk *platform* 2D dan *Scirra* sebagai pengembang software ini. Pada *software* construct 2 user tidak perlu menguasi bahasa pemrograman khusus, karena *software* ini tidak menggunakan bahasa pemrogaraman khusus dan semua perintah yang digunakan pada ada *Event* dan *Action* diatur pada *EventSheet*. Serta *software* ini memiliki dalam *Powerfull Event System* [18].



Gambar 2.1 Tampilan Awal Construct 2

Construct 2 adalah mesin *game* 2D yang dikembangkan oleh *Scirra Ltd*. Minimal *platform* HTML 5. Menggunakan alat Construct 2 memudahkan pengembang *game* yang masih pemula dan memiliki sedikit pengetahuan coding. *Game* yang dibuat dengan Construct 2 dapat dipublikasikan di berbagai platform, termasuk desktop (*PC*, *Mac* atau *Linux*), serta di *platform* seluler (Android, iOS, *Blackberry*, *WindowsPhone* 8.0, *Tizen*) dan *platform* lainnya), serta di HTML 5 halaman web. Jika Anda memiliki hak pengembang, itu juga dapat dirilis di *Nintendo WiiU*. Construct2 juga menyediakan berbagai efek visual menggunakan mesin web GL dan perilaku penyisipan plugin yang dapat membantu pengembang membuat aplikasi yang menarik dan interaktif. Panggil hanya fungsi yang terdapat dalam Construct 2 menggunakan parameter *event* yang disediakan.

Bagian ruang kerja pada Struktur construct 2 dibedakan sebagai berikut:

1. *Build* 2 *Work Area*, untuk mendeskripsikan berbagai objek yang dihasilkan, seperti objek *sprite*, objek *background*, dan lain-lain.
2. *Build* 2 menu *Properties*, untuk mengatur kebutuhan objek yang dibuat, seperti layout warna, ukuran objek sprite, dan lain-lain.
3. Menu Projek dan Layer, projek untuk memilih projek serta layer digunakan untuk membuat layer dalam satu tampilan.
4. Menu *Library* adalah lokasi penyimpanan beberapa objek yang sudah dibuat.
5. *Event Sheet*, adalah ruang kerja dari Construct 2 seperti pengkodean setiap *event* agar bisa berfungsi.

Keuntungan penggunaan Construct 2 untuk developer *game* pemula adalah sebagai berikut:

1. Construct 2 tidak menggunakan bahasa pemrograman yang membingungkan dan mudah dimengerti bagi pemula.
2. Penggunaan pemrograman tidak rumit.
3. Banyak efek yang bisa digunakan untuk membuat game ini menarik[19].

2.2.5 Android

Beberapa tahun belakangan ini, dunia dihebohkan dengan adanya *platform* baru yang kian menguasai pasar global. Saat ini pun, lebih dari setengah persen pengguna ponsel dunia telah menggunakan sistem operasi tersebut, apakah itu? Ya, platform tersebut adalah Android. Berbagai macam gadget menggunkan Android sebagai perangkat *platform*nya. Mulai dari ponsel pintar, *tablet, PC*, jam tangan, TV hingga kamera dan perangkat teknologi lainnya. Meski terbilang sebagai *platform* pendatang baru, Android cukup mengejutkan banyak perusahaan teknologi dengan persentasi pengguna yang terus bertambah dan menjadi OS nomor satu hingga saat ini. Terhitung sejak pertengan tahun 2013, 79% *market share* telah dikuasai oleh *platform* yang satu ini. Hal tersebut tidak terlepas dari adanya ikatan kerjasama antara pihak Android dengan berbagai perusahaan teknologi raksasa seperti Samsung, ASUS, Huawei, Xiaomi, Sony, Oppo, dan Vivo yang kini menggandeng Android sebagai *platform* ponsel mereka. Lalu, sebenarnya Android itu apa?

Gambar 2.9 Logo Android

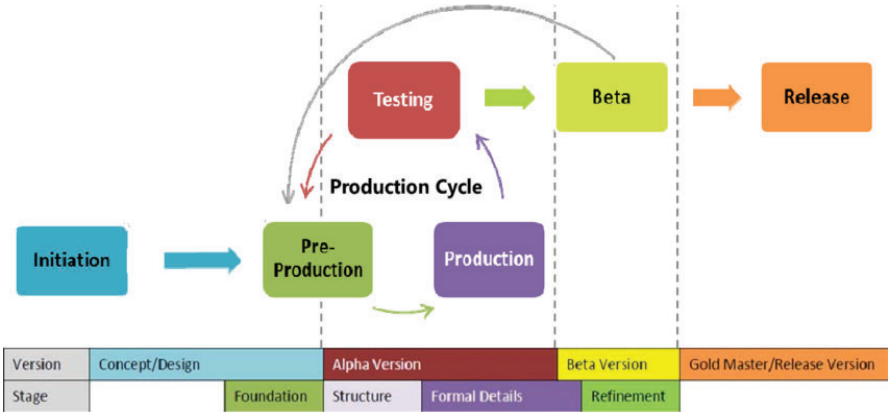
Dalam bahasa inggris istilah Android berarti “Robot yang menyerupai manusia”. hal tersebut dapat terlihat jelas pada icon Android yang menggambarkan sebuah robot berwarna hijau yang memiliki sepasang tangan dan kaki. Sebagai sistem operasi, Android berfugsi sebagai penghubung (*device*)

antara pengguna dan perangkat keras pada *smartphone* atau alat elektronik tertentu. Sehingga, hal tersebut memungkinkan pengguna dapat berinteraksi dengan *device* dan menjalankan berbagai macam aplikasi *mobile.* Lalu, mengapa Android menjadi pilihan utama para pengguna *smartphone* saat ini? Pada bab selanjutnya saya akan membahas hal tersebut lebih mendalam. Namun secara garis besar, daya pikat Android terletak pada *platform opensource* yang membuka banyak peluang besar bagi seluruh pengembang teknologi. Hal tersebut bertujuan dalam membuat dan mengembangkan berbagai fitur aplikasi yang dapat digunakan oleh seluruh pengguna Android. [20].

Android pun memiliki sejarahnya tersendiri. Terbilang sebagai perusahaan *platform* belia, Android baru dirilis pada bulan Oktober 2003 oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White di bawah sebuah perusahaan bernama Android Inc di Palo Antom, California. Sebelum akhirnya diakuisisi oleh Google pada tahun 2005, tujuan awal *platform* yang satu ini adalah untuk mengembangkan sebuah sistem operasi yang lebih canggih bagi kinerja dari sebuah kamera digital. Namun, keberadaan pasar global mengubah arus Andy dan kawan-kawan untuk membawa Android Inc beralih fungsi sebagai perusahaan yang bergerak pada pengembangan sistem operasi *smartphone*. Keputusan tersebut ternyata benar-benar membuahkan hasil. Terbukti, Android dapat menyaingi para pendahulunya yaitu *Symbian* dan *Windows Mobile* dalam menguasai *platform Smarphone* berskala *global*. 5 November 2007 adalah kali pertama Android meluncurkan versi *beta* yang bersamaan dengan berdirinya *Open Handset Alliance* atau OHA. Hal tersebut dijadikan momentum dan ditetapkan sebagai hari Android. Tidak cukup sampai disitu saja, ternyata satu minggu setelah peresmian versi *beta*. Android meluncurkan *Software Development Kit* atau dikenal dengan SDK pada tanggal 12 November 2007. SDK memungkinkan pengguna untuk dapat berkontribusi, membuat dan mengembangkan sendiri aplikasi Android mereka[2].

2.2.6 *Game Development Life Cycle* (GDLC)

Peneliti dalam merancang sebuah *game* akan menggunakan metode GDLC sebagai pengembangan sistem. Pada metode GDLC terdapat beberapa langkah-langkah seperti pada gambar dibawah ini[21].



Gambar 2.2 Metodologi GDLC

1. *Initiation*

   Pada *Initiation* peneliti akan mempersiapkan *software* dan alat atau *tools* yang akan digunakan dalam membuat *game*.

2. *Pre-Production*

   Pada *Pre-production* peneliti merencanakan alur *game* dan membuat design *prototype*.

3. *Production*

   Pada *Production* peneliti melakukan penyempurnaan dari tahap sebelumnya.

4. *Testing*

   Pada *Testing* peneliti akan melakukan pengujian *game* untuk mengetahui *game* sudah berfungsi dengan benar atau belum.

5. *Beta*

   Game selesai dibuat dan sudah proses *Testing* itu belum langsung dilakukan proses *release*. Tetapi melakukan proses *Beta* yaitu meminta *feedback* orang lain terhadap *game* yang telah dibuat dengan cara meminta untuk menilai fungsionalitas *game* dengan cara memainkan *game* dan mengisi kusioner kelayakan *game*. Proses ini diluar dari

production cycle dan hasil dari *tester* bisa berdampak untuk mengulangi proses *production cycle*.

6. *Release*

   Pada *Release* peneliti akan mempublish *game* yang sudah dibuat jika semua proses sudah dilakukan dan lulus pada tahap *beta.*

2.2.7 Black Box Testing

Pada penelitian ini nanti akan ada tahap pengujian game dan peneliti memilih pengujian menggunakan *black box* testing. *Black box* testing adalah proses pengujian yang melihat dari percobaan hasil input dan output dari sebuah aplikasi sudah berfungsi dengan benar atau masih ada bug. *Black box* testing ini berada di tahap sesudah *production* dan hasil dari pengujian ini sangat berpengaruh bagi proses pembuatan. Jika hasilnya ada yang *bug / erorr* maka peneliti mengulang kembali tahap *pre-production* dan *production.*

2.2.8 Pendidikan Lingkungan Hidup

Pendidikan Lingkungan Hidup (PLH) merupakan bagian dari kurikulum Sekolah Dasar. Materi PLH diintegrasikan dalam beberapa mata pelajaran dan juga bisa diajarkan sebagai mata pelajaran terpisah. Seperti pada SDN 3 Teluk yang menggunakan pembelajaran kurikulum 2013 dan di ajarkan pada kelas 5, namun kurangnya minat siswa terhadap materi PLH yang ada di kelas menjadikan materi ini kurang diminati. Agar materi pembelajaran PLH lebih di minati siswa terciptalah media game edukasi yang dapat membantu pembelajaran menjadi lebih menarik dan interaktif, diharapkan dengan adanya media pembelajaran seperti ini dapat menarik kembali minat siswa.

2.2.9 Konsep *Game* Pilah Sampah dan Penerapan *Puzzle*

Pengembangan game pilah sampah merupakan respon atas meningkatnya kebutuhan untuk mendidik dan meningkatkan kesadaran tentang pengelolaan sampah. Dengan menyajikan isu lingkungan melalui format interaktif, game ini memanfaatkan teknologi digital sebagai alat pendidikan yang menarik dan inovatif. Game pilah sampah dirancang dengan tujuan utama untuk memberikan

pemahaman mendalam kepada pemain tentang praktik pengelolaan sampah yang berkelanjutan. Dalam konteks ini, permainan bukan hanya menyajikan hiburan, tetapi juga berfungsi sebagai alat pembelajaran yang efektif.

Elemen *puzzle* dalam permainan ini memiliki peran ganda sebagai sarana pembelajaran dan juga sebagai sumber tantangan yang merangsang pemikiran kreatif. Tantangan yang disajikan dalam bentuk teka-teki berfokus pada konsep-konsep memilah sampah, dengan tujuan untuk mendorong pemahaman yang lebih mendalam serta mengajak pemain untuk berinteraksi aktif dengan materi pembelajaran