

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Penelitian Sebelumnya / Kajian Pustaka**

Pada penelitian-penelitian sebelumnya yang membahas pengembangan *game* edukasi sebagai salah satu cara untuk mendukung pembelajaran melalui *game*. Tujuan dari *game* edukasi itu sendiri adalah untuk meningkatkan minat anak dalam belajar melalui bermain, sehingga diharapkan anak lebih mudah memahami topik yang disajikan.

Beberapa penelitian yang sudah dilakukan mengenai *game* edukasi pada tahun 2022, mereka menerapkan metode ADDIE untuk mengembangkan *game* edukasi Pengenalan Aksara Jawa untuk Menambah Pengetahuan Dengan Menggunakan Metode ADDIE. *Game* ini digunakan sebagai media dalam edukasi pengenalan Aksara Jawa[6]. Selanjutnya penelitian pada tahun 2022 dengan menghasilkan sebuah *game quiz* yang bertujuan mengasah otak dengan menyelesaikan teka-teki yang ada pada *game* yang dikembangkan[7]. Berikutnya penelitian yang menghasilkan permainan edukasi bahasa Inggris dengan menggunakan metode ADDIE. Hasil pengujian dari penelitian ini adalah *game* yang dibuat telah memenuhi target untuk bisa dijadikan alternatif edukasi untuk mata pelajaran bahasa inggris [8].Kemudian pada peneltian yang bertujuan untuk meningkatkan ketrampilan membaca dan menulis aksara jawa. Tetapi dalam penelitian ini pemain tidak dapat bermain secara *online* [9]. Selanjutnya penelitian yang membuat *game* dengan menggunakan metode *Prototype*. Pada penelitian ini *quiz game* yang dibuat dapat membantu peserta didik dapat mempelajari bahasa Inggris dengan cepat [10].

Tabel 2.1 Penelitian terdahulu/kajian pustaka.

No	Judul Paper	Metode yang digunakan	Permasalahan yang diselesaikan	Kekurangan	Perbedaan
1	Game Edukasi Pengenalan Aksara Jawa untuk Menambah Pengetahuan Dengan Menggunakan Metode ADDIE [6]	<i>Analysis, Design, Development, Implementation dan Evaluation</i>	Game edukasi aksara Jawa dikembangkan sebagai media pengenalan huruf Jawa bagi siswa-siswi Mts Al-Washliyah Talun Cirebon. Penelitian membuktikan bahwa <i>game</i> ini efektif dalam meningkatkan pengetahuan mereka.	Kuis yang diberikan sedikit yaitu hanya 3 soal	Pada penelitian milik Kafit Mustofa, Nana Suarna, dan Arif Rinaldi Dikananda menggunakan metode ADDIE sedangkan penulis menggunakan GDLC
2	Perancangan <i>Game Puzzle</i> Labirin Menggunakan Metode <i>Game Development Life Cycle (GDLC)</i> Berbasis <i>Unreal Engine</i> , 2022. [7]	<i>Game Development Life Cycle</i>	Tujuan dari pengembangan <i>game</i> ini adalah untuk melatih otak dengan cara menyelesaikan <i>puzzle</i> yang ada di dalamnya.	Pergerakan karakter yang masih terlalu kaku.	Pada penelitian milik Suwahyudi Syarifa, Tasrif Hasanuddina, dan Mardiyah Hasnawia <i>game</i> dikembangkan dengan <i>Unreal Engine</i> sedangkan penulis menggunakan <i>Construct2</i>
3	Penerapan Model ADDIE Dalam Pembuatan Permainan Edukasi Bahasa Inggris Berbasis <i>Android</i> , 2022[8].	Analysis, Design, Development, Implementation dan Evaluation	<i>Game</i> pendidikan bahasa Inggris yang telah dikembangkan berhasil mencapai targetnya sebagai sarana pengajaran	Tidak ada kejelasan dalam proses pengembangannya	Pada penelitian milik Muhamad Iqbal Maulana dan Erfian Junianto <i>game</i> dikembangkan dengan <i>software Unity 3D</i> sedangkan penulis menggunakan <i>Construct2</i>

No	Judul Paper	Metode yang digunakan	Permasalahan yang diselesaikan	Kekurangan	Perbedaan
			alternatif untuk mata pelajaran bahasa Inggris.		
4	Metode Pembelajaran Berbasis <i>Game</i> Untuk Meningkatkan Ketrampilan Membaca Dan Menulis Aksara Jawa, 2020[9].	Metode <i>Scramble</i>	Penggunaan media permainan dalam penerapan pembelajaran aksara Jawa dapat secara signifikan meningkatkan kemampuan peserta dalam membaca dan menulis aksara Jawa, seperti yang terbukti dari hasil <i>Pra-Tes</i> dan <i>Post-Test</i> .	Fitur yang tersedia terbilang sedikit.	Pada penelitian milik Whinny Qori Fatima, Livia Khairunisa, dan Budi Prihatminingtyas menggunakan metode <i>scramble</i> sedangkan penulis akan menggunakan metode <i>GDLC</i> .
5	Rancang Bangun Aplikasi Game Edukasi Interaktif Dengan Menggunakan Metode <i>Game Design Document</i> , 2019. [10]	Menggunakan metode <i>Game Design Document</i>	Pemanfaatan game sebagai alat pembelajaran dapat membantu peserta didik dalam mempelajari bahasa Inggris dengan cara yang menarik dan mudah dipahami, sehingga peserta didik dapat memahami pelajaran dengan cepat.	Materi yang disampaikan tidak lengkap.	Pada penelitian milik Sri Lestari Rahayu dan Fujiati menggunakan metode <i>Game Design Document</i> sedangkan penulis menggunakan metode <i>GDLC</i> .

Setelah penulis melakukan tinjauan pustaka terhadap penelitian terdahulu maka penulis memutuskan untuk membuat *game* edukasi berbasis *Android* dengan menggunakan metode GDLC dan menggunakan *software Construc2*. Penulis memutuskan hal ini karena setelah menurut penelitian terdahulu dapat disimpulkan bahwa *game* edukasi merupakan salah satu cara yang cukup efektif untuk menunjang belajar para siswa sehingga dapat membantu para siswa untuk dapat memahami materi yang diajarkan lebih baik lagi. *Android* sendiri dipilih karena menurut penelitian terdahulu dapat disimpulkan bahwa *Android* lebih efektif untuk digunakan karena aksesibilitasnya yang dapat diakses oleh banyak orang.

## **2.2. Dasar Teori**

### **2.2.1 Aksara Jawa**

Aksara Jawa merupakan sebuah sistem tulisan yang unik dalam bahasa Jawa, yang tidak terbatas pada penggunaan huruf Latin seperti dalam kebanyakan sistem tulisan yang umum digunakan. Aksara Jawa memiliki kekhasan tersendiri dan biasanya ditemukan dalam beberapa manuskrip kuno dengan variasi gaya penulisan yang berbeda, mencerminkan kekayaan dan kompleksitas budaya Jawa.

Salah satu ciri khas dari aksara Jawa adalah adanya beragam gaya penulisan yang menjadi bagian integral dari identitasnya. Sebagai contoh, terdapat gaya penulisan ngetumbar yang cenderung memiliki bentuk lebih bulat dan mengalir, memberikan kesan estetik yang khas. Kemudian, terdapat pula gaya penulisan Yogya, yang cenderung miring dan memiliki keanggunan tersendiri dalam penyajiannya. Di samping itu, gaya penulisan Solo juga menonjolkan keunikan tersendiri dengan ciri-ciri tertentu yang membedakannya dari gaya-gaya lainnya. Tak ketinggalan, gaya penulisan jejeg panjang-panjang yang lebih dikenal sebagai gaya penulisan Bali juga merupakan salah satu variasi aksara Jawa yang menarik dan menunjukkan keanekaragaman dalam dunia tulis-menulis.

Keberadaan berbagai gaya penulisan aksara Jawa ini mencerminkan kompleksitas dan keberagaman budaya serta seni yang ada dalam masyarakat Jawa. Seiring berjalannya waktu, aksara Jawa terus berkembang dan beradaptasi dengan perubahan zaman, tetapi tetap mempertahankan esensi dan nilai-nilai budaya yang

berharga. Melalui manuskrip-manuskrip kuno, warisan pengetahuan tentang aksara Jawa dan kekayaan budaya Jawa pun tetap terjaga dan dapat diteruskan ke generasi selanjutnya.

Sebagai simbol identitas dan tradisi, aksara Jawa menjadi bagian penting dari warisan budaya Indonesia. Keberadaannya memberikan peluang untuk memahami lebih dalam tentang sejarah, bahasa, dan kearifan lokal yang terkandung dalam bahasa Jawa. Melalui upaya pelestarian dan pengenalan aksara Jawa kepada generasi muda, kita dapat memastikan bahwa keberagaman dan kekayaan budaya Nusantara terus hidup dan berkembang dalam dunia yang semakin modern dan global ini[25].

AKSARA JAWA				
𑀓 ha	𑀕 na	𑀖 ca	𑀗 ra	𑀘 ka
𑀙 da	𑀚 ta	𑀛 sa	𑀜 wa	𑀝 la
𑀞 pa	𑀟 dha	𑀠 ja	𑀡 ya	𑀢 nya
𑀣 ma	𑀤 ga	𑀥 ba	𑀦 tha	𑀧 nga

Gambar 2.1 Aksara Jawa [25].

Abjad Jawa adalah sistem penulisan untuk bahasa Jawa yang terdiri dari 20 aksara utama yang bersifat silabi. Aksara-aksara ini dikenal sebagai "aksara nglegena", karena mereka digunakan dalam bentuk suku kata terbuka atau "aksara wuda", karena mereka belum memiliki tanda baca seperti sandhangan swara atau sandhangan sigeg. [25].

### 2.2.2 Multimedia Interaktif

Multimedia interaktif mewakili sebuah bentuk yang kompleks dan beragam dari presentasi informasi yang mencakup berbagai elemen media yang berbeda.

Meliputi unsur-unsur seperti teks, gambar, grafik, animasi, audio, dan video, multimedia interaktif memberikan pengalaman yang lebih dinamis dan menarik bagi para penggunanya. Perbedaannya terletak pada cara penyajian yang interaktif, di mana pengguna tidak hanya menjadi pemirsa pasif, tetapi juga memiliki peran aktif dalam mengelola, menavigasi, dan berpartisipasi dalam pengalaman multimedia yang disajikan.

Dalam konteks multimedia interaktif, berbagai elemen media tersebut diintegrasikan secara sinergis dan dirancang untuk memberikan pengalaman yang lebih menyeluruh. Teks digunakan untuk menyampaikan informasi dalam bentuk tulisan, gambar memberikan dukungan visual yang kuat, grafik memberikan representasi visual data, animasi menambahkan unsur gerak dan dinamika, audio menyediakan pengalaman mendengar, dan video menawarkan pengalaman audio-visual yang lebih mendalam.

Melalui interaktivitasnya, pengguna diberi kebebasan untuk memilih jalur yang ingin mereka eksplorasi dalam presentasi multimedia, berpindah antar elemen media, menyesuaikan tampilan sesuai keinginan, dan bahkan berpartisipasi dalam aktivitas yang memungkinkan mereka berkreasi dan berinteraksi dengan isi yang disajikan.

Konsep multimedia interaktif telah mengubah cara kita menghadapi informasi dan hiburan. Pengguna tidak lagi menjadi penerima pasif, tetapi menjadi bagian dari proses eksplorasi dan penemuan. Interaktivitas ini juga membuka peluang baru dalam pembelajaran, di mana pengguna dapat terlibat lebih aktif dalam proses pembelajaran dan memperdalam pemahaman mereka melalui pengalaman langsung.

Dengan teknologi terus berkembang, multimedia interaktif terus mengalami perkembangan dan peningkatan kemampuan. Dengan berbagai inovasi dan fitur yang semakin canggih, multimedia interaktif terus memberikan pengalaman yang semakin menarik, mendalam, dan berkesan bagi penggunanya[6].

### **2.2.3 Game**

*Game* merupakan sesuatu aktivitas yang diperuntukan sebagai media hiburan serta kesenangan guna menyingkirkan rasa jenuh. Tetapi, seiring pertumbuhan masa serta teknologi yang terus menjadi modern, *game* disaat ini juga dikembangkan sebagai media pembelajaran.

*Game* dibagi menjadi beberapa jenis, penjelasan atas masing-masing jenis *game* adalah sebagai berikut.

#### 1. *Action Game*

*Game* bergenre aksi yang menantang pemainnya untuk terampil seperti dalam pertempuran. Secara umum, pemain dapat melakukan hal-hal seperti menghindar, meninju, dan menendang musuh. Keakuratan permainan ini harus cukup dan tidak boleh ada delay atau jeda. Semoga para pengembang dapat mengoptimalkan *game* semacam ini.

#### 2. *Adventure Game*

*Game adventure* di mana pemain didorong untuk menjelajahi berbagai lokasi seperti hutan, gua, kastil, dan lainnya. Tujuan utamanya adalah mendapatkan item tertentu, menemukan lokasi baru atau menyelesaikan tugas, dan lain sebagainya.

#### 3. *Sports Game*

Jenis permainan ini bersifat bersaing dan dapat dimainkan oleh beberapa orang secara bersamaan, baik sendiri maupun sebagai sebuah tim. Kebanyakan permainan dalam genre ini merupakan permainan olahraga.

#### 4. *Role Playing Game*

*Game* jenis ini dimana pemain memiliki peran yang dimainkan dalam bentuk pertumbuhan atau perkembangan karakter utama. Sebagian besar jenis *game* tersebut adalah *game online*, meskipun beberapa juga dapat dimainkan secara *offline*.

#### 5. *Simulation Game*

*Simulation game* tujuannya adalah untuk mensimulasikan kegiatan nyata. Misalnya simulasi mengendarai pesawat atau truk. Sebagian besar *game* ini membutuhkan instruksi membaca agar pemain memahami cara memainkannya[21].

## 6. *Education Game*

*Game* memiliki jenis yang lain yaitu *puzzle*. *Puzzle* adalah permainan *puzzle* atau tebak-tebakan. *Puzzle* juga bisa disebut permainan edukatif karena dapat melatih kemampuan otak. Teka-teki adalah permainan yang paling umum digunakan dan termasuk permainan pendidikan yang paling sederhana untuk digunakan di sekolah.

## 7. *Puzzle game*

*Game puzzle* adalah jenis permainan yang melibatkan pemecahan teka-teki atau masalah tertentu dengan menggunakan keterampilan kognitif, pemecahan masalah, dan logika. Biasanya, permainan ini melibatkan penggabungan atau penyusunan elemen-elemen *puzzle* untuk mencapai tujuan tertentu[7].

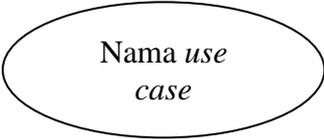
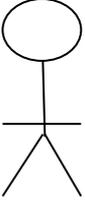
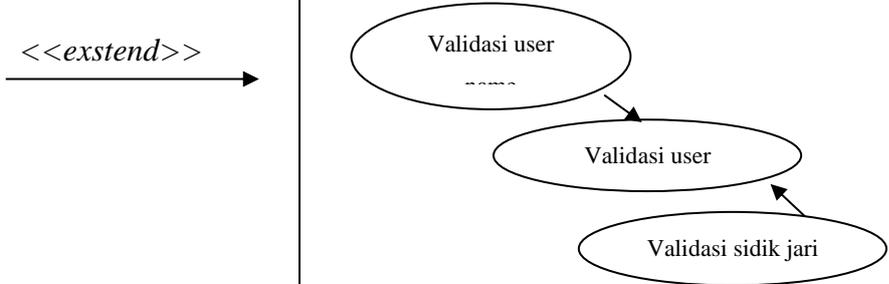
### 2.2.4 UML

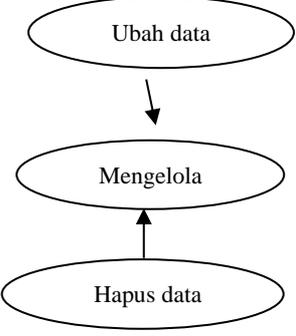
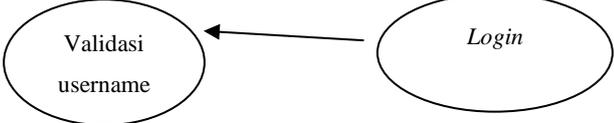
UML atau *Unified Modelling Language* adalah sebuah model yang sering digunakan untuk merancang pengembangan perangkat lunak berbasis objek. UML memberikan standar dalam merancang blueprint atau rencana dalam sistem, termasuk proses bisnis, pembuatan kelas yang dibutuhkan dalam bahasa pemrograman tertentu, skema basis data, dan berbagai komponen yang dibutuhkan dalam pembuatan aplikasi dari sistem tersebut. [13]. Berikut ini diagram-diagram yang ada dalam UML:

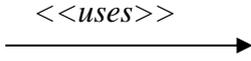
#### 1. *Use Case Diagram*

Diagram *Use Case* adalah gambaran tentang bagaimana sistem informasi berperilaku dalam interaksinya dengan satu atau lebih aktor. Menggunakan simbol-simbol tertentu, diagram *Use Case* menjelaskan interaksi itu secara rinci. [22]:

Tabel 2.2 Simbol-Simbol *Use Case Diagram*.

Simbol	Deskripsi
<p data-bbox="448 409 571 443"><i>Use case</i></p> 	<p data-bbox="710 409 1335 611">Sistem ini memiliki kemampuan untuk mengirim pesan antara unit atau aktor yang digambarkan dalam kasus penggunaan dengan kata kerja pada bagian awal nama kasus penggunaan.</p>
<p data-bbox="427 663 596 696">Aktor / <i>actor</i></p>  <p data-bbox="432 936 587 969">Nama aktor</p>	<p data-bbox="710 640 1335 999">Suatu proses atau sistem lain yang saling berhubungan dengan sistem informasi akan diidentifikasi sebagai aktor. Walau ikon agen digunakan untuk mewakili individu, aktor mungkin terdiri dari hal lain selain manusia dan biasanya dikenal dengan nama yang diawali pada deskripsi aktor tersebut.</p>
<p data-bbox="368 1077 651 1111"><i>Asosiasi / association</i></p> 	<p data-bbox="710 1025 1335 1167">Aktor dan <i>use case</i> berkomunikasi satu sama lain dan berkontribusi dalam sebuah <i>use case</i> atau melakukan interaksi dengan aktor.</p>
<p data-bbox="392 1429 628 1462"><i>Ekstensi / extend</i></p> 	<p data-bbox="710 1193 1335 1552">Hubungan Use case tambahan dapat berjalan secara independen meskipun hubungannya dengan use case utama, keduanya memiliki kesamaan. Dalam pemrograman berbasis objek, use case tambahan biasanya memiliki nama serupa dengan use case utamanya karena diterapkan warisan umum, misalnya:</p>

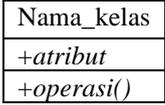
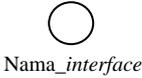
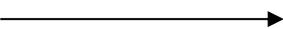
Simbol	Deskripsi
	<p>Panah mengarahkan ke kasus penggunaan tambahan; pada dasarnya, kasus penggunaan yang diperluas memiliki jenis yang sama dengan kasus penggunaan induknya.</p>
<p>Generalisasi/ <i>generalization</i></p> 	<p>Sebagai contoh, terdapat hubungan antara generalisasi dan spesialisasi antara dua use case, di mana salah satu memiliki fungsi yang lebih umum daripada yang lain.</p>  <p>Arah panah mengarah pada <i>use case</i> yang lebih umum.</p>
<p>Menggunakan/ <i>include/ use case</i></p> <p>&lt;&lt;<i>includee</i>&gt;&gt;</p> 	<p>Hubungan antara use case tambahan dan use case terkait dengan adanya ketergantungan dimana use case tambahan membutuhkan use case utama dalam rangka menjalankan tugasnya. Terdapat dua pendapat utama terkait implementasi use case tambahan: inklusif, yang berarti use case tambahan akan selalu dipanggil saat use case utama dijalankan, seperti yang terjadi dalam contoh berikut.masuknya <i>use case</i>:</p>  <p>Inklusi berarti bahwa use case tambahan selalu akan memeriksa apakah use case yang</p>

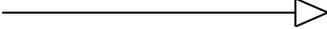
Simbol	Deskripsi
	<p>ditambahkan sudah diimplementasikan sebelum menjalankan fungsinya. Contohnya dalam kasus seperti itu.</p>  <p>Dua interpretasi terkait dengan hubungan antara <i>use case</i> tambahan dan <i>use case</i> dapat diambil menurut pertimbangan dan interpretasi yang diperlukan.</p>

## 2. Class Diagram

*Class Diagram* yang akan digunakan dalam pembangunan sistem digambarkan dalam diagram kelas, yang menggambarkan struktur sistem. Setiap kelas memiliki atribut dan metode atau operasi yang terkait.[22].

Tabel 2.3 Simbol-Simbol *Class Diagram*.

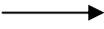
Simbol	Deskripsi
<p>Kelas</p> 	<p>Kelas yang berkaitan dengan struktur sistem.</p>
<p>Antarmuka / <i>interface</i></p> 	<p>Mirip dengan konsep antarmuka dalam pemrograman berorientasi objek.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Dalam pengertian umum, asosiasi seringkali disertai dengan polisemi dalam hubungan antar kelas.</p>
<p>Asosiasi berarah / <i>directed association</i></p> 	<p>Hubungan antar kelas berarti bahwa satu kelas menggunakan kelas lain. Asosiasi biasanya dikaitkan dengan polisemi.</p>

Simbol	Deskripsi
<i>Generalisasi</i> 	Hubungan kelas-kelas dalam konteks generalisasi-spesialisasi.
Kebergantungan / <i>dependency</i> 	Ketergantungan antara kelas adalah dasar hubungan antar kelas.
Agregasi / <i>aggregation</i> 	Hubungan antar kelas penuh makna ( <i>whole-part</i> ).

### 3. Statechart Diagram

*Statechart diagram*, juga dikenal sebagai *state machine diagram*, digunakan untuk menggambarkan perubahan keadaan mesin atau sistem atau keadaan suatu peristiwa, yang direpresentasikan dalam bentuk diagram terarah. *State diagram* berguna untuk menggambarkan bagaimana pengguna berinteraksi dengan sistem. Simbol berikut sering digunakan saat membuat diagram keadaan, seperti yang dapat kita lihat pada tabel di bawah ini [22].

Tabel 2.4 Simbol-Simbol *Statechart Diagram*.

Simbol	Deskripsi
<i>Event</i> 	<i>Event</i> adalah tindakan yang menyebabkan perubahan kondisi mesin
<i>Start / status awal (initial state)</i> 	<i>Start</i> atau <i>initial state</i> adalah keadaan atau keadaan awal saat sistem dijalankan.
<i>End / status akhir (final state)</i> 	<i>End</i> atau <i>final state</i> Keadaan akhir adalah keadaan akhir dari siklus hidup sistem
<i>State</i> 	<i>State</i> atau keadaan adalah keadaan sistem pada saat itu aman

#### 4. Activity Diagram

*Activity Diagram* memperlihatkan urutan aktivitas dalam suatu sistem atau proses bisnis. Simbol-simbol tertentu biasanya digunakan untuk membuat diagram ini, seperti ditunjukkan pada tabel.[22].

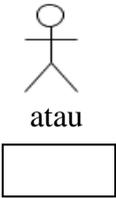
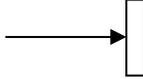
Tabel 2.5 Simbol-Simbol *Activity Diagram*.

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem.
Aktivitas 	Aktivitas yang diambil oleh sistem Tindakan biasanya dimulai dengan sebuah kata Kerja.
Decision 	Asosiasi industri dengan lebih dari satu peluang operasi.
Join 	Asosiasi yang menyatukan lebih dari satu industri satu
Status akhir 	Keadaan terakhir yang dijalankan sistem adalah akhir dari diagram operasi.
Swimlane 	Pisahkan organisasi bisnis yang bertanggung jawab atas aktivitas tersebut apa yang ada

#### 5. Sequence Diagram

*Sequence diagram* menunjukkan bagaimana objek berinteraksi satu sama lain dan dengan lingkungan (termasuk pengguna, layar, dll) dalam bentuk pesan yang dipresentasikan dalam urutan waktu. Beberapa simbol yang umum digunakan dalam membuat diagram urutan ditunjukkan dalam tabel berikutnya. [22].

Tabel 2.6 Simbol-Simbol *Sequence Diagram*.

Simbol	Deskripsi
<p>Aktor</p>  <p>atau</p>	Sistem informasi memiliki orang, proses, atau sistem lain yang berinteraksi dengannya.
<p><i>Lifeline</i></p> <p>.....</p>	Menunjukkan kehidupan suatu hal.
<p>Objek</p> 	Merupakan objek di mana pesan berinteraksi.
<p>Waktu aktif</p> 	Menginformasikan bahwa item aktif dan berinteraksi dengan pesan.
<p>Pesan tipe <i>create</i></p> 	Untuk menunjukkan bahwa objek tersebut memiliki objek lain, arah panah objek yang dibuat
<p>Pesan tipe <i>call</i></p> 	Mendeklarasikan bahwa sebuah objek dapat memanggil operasi atau metode yang ada pada objek lain.
<p>Pesan tipe <i>send</i></p> 	menunjukkan bahwa objek mengirimkan data/input/informasi lainnya.
<p>Pesan tipe <i>return</i></p> 	menunjukkan objek yang melakukan tindakan, atau cara mengembalikan objek yang ditentukan.
<p>Pesan tipe <i>destroy</i></p> 	Untuk mengakhiri penggunaan objek lain, objek harus diberi pernyataan berhenti.

### 2.2.5 Android

*Android* adalah sistem operasi untuk smartphone yang berdasarkan pada Linux. Ini adalah platform yang terbuka yang memungkinkan pengembang untuk membuat aplikasi mereka sendiri untuk digunakan di berbagai perangkat seluler. Pada tahun 2005, *Android* berpindah kepemilikan ketika *Google* membelinya, yang kemudian membuat sistem operasi ini dilisensikan sebagai *open source* di bawah *Apache Software Foundation*. [18].

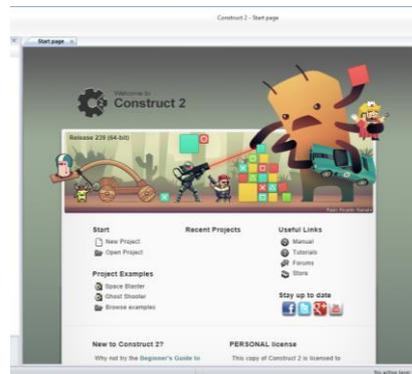


Gambar 2.2 Logo *Android* [18].

*Android* telah mengalami banyak perkembangan dari waktu ke waktu dengan dirilisnya beberapa versi dari *Android*. Pada tahun 2008, *Android* merilis versi 1.1 dan segera disusul dengan *Android* versi 1.5 (*Cupcake*) dan *Android* versi 1.6 (*Donut*). Setelah itu, diluncurkan *Android* versi 2.1 (*Eclair*) dan *Android* versi 2.2 (*Froyo*) pada tahun berikutnya. Kemudian, *Android* merilis versi 2.3 (*Gingerbread*) dan versi 3.0/3.1 (*Honeycomb*) yang didesain khusus untuk tablet pada tahun 2011. Pada akhir tahun yang sama, diluncurkan *Android* versi 4.0 (*Ice Cream Sandwich*), yang kemudian diikuti dengan *Android* versi 4.1 (*Jelly Beans*) dan *Android* versi 4.4 (*Kitkat*) pada tahun 2013. Dalam beberapa tahun terakhir, *Android* merilis versi 5.1 (*Lollipop*), versi 6.0 (*Marshmallow*), dan versi terbaru yaitu versi 7.0 (*Nougat*). [24].

### **2.2.6 Construct 2**

*Construct 2* adalah pembuat *game* dari *Scirra*, salah satu *game* yang diminati oleh pengembang *game* karena *Construct 2* mudah digunakan dan memiliki banyak tutorial dan *template*.

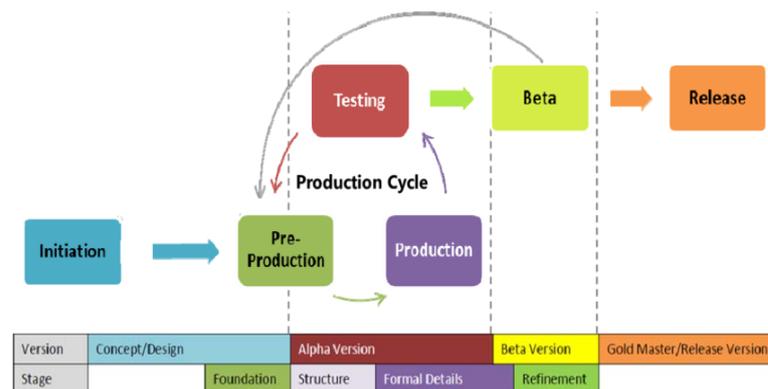


Gambar 2.3 Tampilan awal *Construct 2*.

*Android* merupakan salah satu produk yang dapat dihasilkan dari *device construct 2* menggunakan HTML5. *Construct 2* dapat diunduh secara gratis dengan adegan dan fitur terbatas yang permainannya tidak gratis tetapi tidak seperti adegan berlisensi dan banyak fitur lainnya. [7].

### 2.2.7 Metode Game Development Life Cycle

Penulis dalam perancangan game, penulis menggunakan GDLC atau *Game Development Life Cycle* sebagai pengembangan sistem. Prosedur GDLC terdiri dari beberapa langkah seperti yang ditunjukkan di bawah ini [23].



Gambar 2.4 Alur metode *Game Development Life Cycle* [23].

#### 1. *Initiation*

Selama *Initiation*, penulis menyiapkan *software* dan *tools* atau alat yang akan digunakan untuk membuat *game*.

#### 2. *Pre-Production*

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

### 3. *Production*

Pada *Production* penulis melakukan penyempurnaan dari tahap sebelumnya.

### 4. *Testing*

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

## 2.2.8 *System Usability Testing*

SUS adalah hasil kegunaan yang melibatkan pengguna akhir dalam proses penyebaran. Jumlah responden dalam peran yang sangat penting ditentukan. *System Usability Scale* (SUS) dalam keefektifan responden bukanlah filosofi yang lengkap, juga tidak ada definisi individual dari teori validitas. Persyaratan ini dikumpulkan dari responden *System Usability Scale* (SUS), yang merupakan pengguna akhir produk, baik *software* maupun *website*. Metode ini memiliki instrumen yang jelas serta hasil perhitungan evaluasi mempunyai nilai yang dapat dipertanggung jawabkan. Skala *likert* dengan 5 tingkat kepuasan pengguna digunakan dalam perhitungan skor *System Usability Scale*. Kelima tingkat kepuasan ini berupa pernyataan “Sangat tidak setuju”, “Tidak Setuju”, “Netral”, “Setuju, dan “Sangat Setuju”.

Terdapat kelebihan dalam penggunaan metode SUS, yaitu proses evaluasi yang dilaksanakan lebih mudah dipahami oleh responden, meski hanya melibatkan sedikit responden, hasil yang diberikan dapat maksimal serta hasil pengujian memberikan hasil yang cukup jelas apakah sistem layak atau tidak layak untuk digunakan [28].

Tabel 2.7 Rentang Nilai *Grade* Pada *System Usability Scale*

<i>Grade</i>	Rentang Nilai
A	Skor > 80.3
B	Skor => 74 dan skor < 80.2
C	Skor => 68 dan skor < 78
D	Skor => 52 dan skor < 67
F	Skor < 52

### 2.2.9 *Black Box Testing*

Black box testing adalah jenis pengujian perangkat lunak yang dilakukan tanpa pengetahuan tentang struktur internal sistem yang diuji. Dalam *black box testing*, fokus utama adalah menguji fungsi-fungsi eksternal sistem serta *input* dan *output* yang terkait. Hal ini dilakukan dengan tujuan memastikan bahwa komponen-komponen sistem bekerja dengan benar, memenuhi persyaratan fungsional, dan sesuai dengan harapan pengguna. Pengujian *black box* memiliki proses utama, yaitu [16]:

- a. Analisis kebutuhan perangkat lunak.
- b. Masukan yang bersifat valid untuk menguji bahwa perangkat lunak bekerja dengan baik. Serta masukan yang bersifat tidak valid untuk menguji bahwa perangkat lunak dapat mendeteksi dan menangani masukan yang tidak valid.
- c. Pengujian dirancang dari beberapa masukan yang dipilih.
- d. Hasil pengujian ditentukan sesuai dengan masukan.
- e. Membandingkan hasil pengujian dengan hasil yang diharapkan.
- f. Keputusan perancangan disesuaikan dengan kelayakan fungsi dari perangkat lunak yang sedang di uji.

### 2.2.10 *Pre-Test dan Post-Test*

Pre-test adalah pengujian atau evaluasi yang dilakukan sebelum dilakukannya suatu intervensi atau perlakuan tertentu pada subjek penelitian. Biasanya, pre-test digunakan untuk mengukur pengetahuan, keterampilan, atau karakteristik awal dari subjek penelitian sebelum mereka terlibat dalam perlakuan atau intervensi yang diteliti. Pre-test bertujuan untuk memperoleh data awal yang nantinya dapat dibandingkan dengan hasil post-test, sehingga dapat melihat perubahan yang terjadi setelah perlakuan atau intervensi diberikan.

Sementara itu, post-test adalah evaluasi yang dilakukan setelah subjek penelitian telah mengalami suatu perlakuan atau intervensi tertentu. Tujuan dilakukannya post-test adalah untuk mengevaluasi perubahan yang terjadi pada subjek penelitian setelah mereka menerima perlakuan atau intervensi tersebut. Dengan membandingkan hasil pre-test dan post-test, peneliti dapat melihat sejauh mana pengaruh perlakuan atau intervensi terhadap subjek penelitian dan apakah perlakuan atau intervensi tersebut efektif dalam mencapai tujuan yang ditetapkan

Proses pre-test dan post-test [3]:

1. Pretest: Sebelum intervensi dilakukan, dilakukan pengumpulan data awal atau baseline untuk mengukur kondisi awal subjek atau variabel yang akan diintervensi.
2. Intervensi: Perlakuan atau intervensi diberikan pada subjek atau variabel yang menjadi fokus penelitian.
3. Posttest: Setelah intervensi dilakukan, dilakukan pengumpulan data kedua untuk mengukur perubahan yang terjadi setelah intervensi.