

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Dasar

##### 2.1.1 *Game*

Dalam penelitian yang dilakukan oleh (Kaurie, Purwanto, and Minarni 2020) menyebutkan *video game* merupakan sebuah antarmuka interaktif antara pengguna dan perangkat yang menghasilkan umpan balik visual. Pada awalnya, istilah “video” pada permainan video tradisional hanya mengacu pada perangkat tampilan *raster*. Namun, seiring perkembangan teknologi, istilah “*video game*” kini dapat digunakan untuk merujuk pada permainan yang dimainkan pada berbagai jenis perangkat layar. *Platform* merupakan sistem perangkat elektronik yang digunakan sebagai tempat bermain *video game* seperti komputer pribadi dan konsol *video game*. Jenis *platform video game* dapat bervariasi, mulai dari yang besar seperti komputer *mainframe* hingga yang kecil seperti perangkat *mobile*.

Sementara itu, menurut (Asmiatun and Novita Putri 2017) dalam sebuah buku, terdapat beberapa pengertian mengenai *game* yang telah disampaikan oleh berbagai sumber buku dan penelitian, antara lain:

1. “Menurut Adams dalam bukunya yang berjudul *Fundamental Of Game Design, 2nd Edition* yang terbit pada tahun 2010 menyatakan pandangannya tentang *game*”, sebagai berikut:
  - a. “*Game* harus menghadirkan pengalaman yang imajinatif dan koheren, sehingga desainer harus memiliki visi.”

- b. “*Game* harus memperhatikan kepuasan konsumen dalam pemasarannya, sehingga desainer harus mempertimbangkan selera konsumen.”
  - c. “*Game* yang dilengkapi lisensi harus membayar biaya lisensi kembali, sehingga desainer harus memahami keuntungan lisensi dan menggunakannya sebagai keunggulan.”
  - d. “*Game* harus memberikan tantangan intelektual dan pengalaman yang lancar, sehingga desainer harus memahami teknologi.”
2. “Menurut IGDA (*International Game Developers Association*) *Education* SID, *game* merupakan suatu kegiatan yang memiliki beberapa aturan atau *rules*. Sebagai bentuk permainan, *game* seringkali (meskipun tidak selalu) melibatkan konflik, baik dengan pemain lain, sistem edukasi, atau dengan acak/nasib/keberuntungan. Sebagian besar *game* memiliki tujuan, namun tidak semua seperti *The Sims* dan *SimCity*. Selain itu, kebanyakan *game* juga telah menentukan titik awal dan akhir, tetapi tidak semua *game* seperti *World of Warcraft*.”
  3. “J. Huizinga dalam karyanya “*Homo Ludens*” (1938) memberikan definisi *game* sebagai aktivitas yang terjadi dalam batasan waktu dan ruang tertentu, dengan urutan yang jelas berdasarkan aturan yang diterima secara bebas tanpa terikat oleh kebutuhan.”

Berikut merupakan daftar jenis-jenis *game* di kemukakan (Pro 2022) dan sumber penelitian lain, sebagai berikut:

### 1. FPS Games

*Game FPS* (dikenal sebagai *First Person Shooters*) adalah jenis *game* di mana pemain melihat karakter yang dimainkan dari sudut pandang tersebut. Dalam *game* ini, pemain tidak dapat melihat karakter mereka sendiri, namun hanya melihat apa yang ada di depan karakter seperti lengan karakter, senjata yang mereka pegang, dan lingkungan sekitarnya.

Selain itu, jenis *game* ini dapat dimainkan di berbagai *platform* seperti PC, konsol, atau ponsel karena popularitasnya yang tinggi. Salah satu contoh *game FPS* terbaik yang ada saat ini adalah *DOOM Eternal*, yang merupakan rilisan terbaru di *Xbox*. Selain itu, *DOOM* merupakan *game FPS* pertama yang telah membuka jalan bagi popularitas *game FPS* saat ini dan masa depan.

Beberapa *game FPS* memberikan opsi orang ketiga, tergantung pada preferensi pemain. Salah satu keuntungan utama dari *game FPS* adalah pemain dapat bermain dengan teman-temannya, terutama dengan adanya mode *multiplayer online* yang tersedia saat ini. Contoh *game FPS* yang populer antara lain *Call of Duty*, *PUBG*, dan *Fortnite*.

## **2. RPG**

*RPG* (dikenal sebagai *Role Playing Games*) adalah *game* fantasi di mana pemain memainkan sebuah cerita, berinteraksi dengan NPC (*Non Playable Character*) dan mengeksplorasi dunia dengan jumlah *lore* yang banyak. Selain itu, beberapa *RPG* membutuhkan *grinding* yang cukup banyak agar pemain dapat meningkatkan kekuatan, kemampuan karakter mereka dan menemukan tujuan berikutnya dalam dunia *game*.

Demikian pula, sebagian besar RPG memiliki unsur-unsur yang serupa seperti dungeon yang diisi oleh karakter utama dengan latar belakang yang menarik, keahlian yang unik, banyak karakter pendukung, dan pertempuran melawan bos yang menantang. RPG memiliki banyak *sub-genre* seperti *open world*, *sandbox*, JRPG, *action RPG*, dan jenis *game* lain yang mirip dengan *adventure games*. Tujuan utama dalam RPG adalah menyelesaikan cerita utama, serta beberapa *quest* sampingan yang melibatkan mengalahkan bos dan meningkatkan kekuatan karakter utama.

### 3. **Game Pertualangan (*Adventure Games*)**

*Game* pertualangan merupakan jenis *video game* yang berbeda dengan RPG, meskipun seringkali disalahartikan. *Game* pertualangan memiliki pendekatan yang lebih linear, di mana pemain harus menyelesaikan level dan naik sampai mengalahkan *game*.

Demikian juga, beberapa *game* ini memiliki elemen *platforming*, tahap rahasia dan peti harta tersembunyi yang mendorong rasa ingin tahu, dan eksplorasi. Selain itu, terdapat 2 jenis *game* pertualangan, antara lain pertualangan teks yang memerlukan imajinasi yang sangat banyak dan pertualangan grafis, contohnya seperti *Stars Wars: Jedi Fallen Order*. Meskipun *game* RPG sering memberi fleksibilitas yang lebih banyak, *game* pertualangan biasanya memiliki alur cerita yang lebih ketat dan jauh lebih sedikit *grinding*.

### 4. **Game Strategi**

*Game* strategi merupakan jenis *game* yang mengharuskan pemain menggunakan strategi cerdas untuk mencapai tujuan yang ditetapkan. Baik itu dengan mengambil alih markas musuh atau menggunakan strategi cerdas dalam *game* catur yang menekankan pada keputusan yang diambil oleh pemain.

Salah satu contoh *game* strategi yang sudah ada sejak lama adalah *game* catur yang telah ada sejak abad ke 6 Masehi. Selain itu, terdapat *game* strategi yang baru seperti *Age of Empires 3* dan *StarCraft* yang menggabungkan prinsip-prinsip berpikir yang tinggi, dan menghadirkan pengalaman yang lebih baru. Dalam konsep ini, *game* RTS (*Real Time Strategy*) menjadi salah satu jenis *game* yang menarik karena membutuhkan pemikiran kritis untuk selalu berada satu langkah di depan lawan hingga mencapai kemenangan.

## **5. *Game* Olahraga dan Kebugaran**

Dalam dunia *game*, terdapat *game* simulasi olahraga yang memberikan kesempatan bagian pemain untuk menghadapi AI dalam bentuk tim olahraga favorit atau karakter dari *game* olahraga yang populer seperti basket, sepak bola, bisbol, permainan *golf*, dan olahraga ekstrem seperti *snowboarding* atau *skateboarding*.

Selain itu, ada kesempatan bagi pemain untuk bermain bersama teman-teman menggunakan pertama yang memiliki layar yang sama atau bermain dengan sekelompok pemain dan pemain acak lainnya secara daring.

## **6. *Fighting Games***

Pada saat ini, *game-game* ini telah menghadirkan lebih banyak mekanik karena adanya *game* pertarungan 3D seperti *game Naruto: Ultimate Ninja*

*Storm* dan game *Dragon Ball Z*. Meskipun demikian, tujuan utama dari game tersebut tetap tidak berubah, pemain harus berhasil mengalahkan musuh dengan mengurangi HP (*Health Points* atau *Hit Points*) karakter yang dimainkan hingga habis.

Seiring berjalannya waktu, Jenis *fighting games* telah mengalami perkembangan yang signifikan. Mekanik baru telah ditambahkan dan ketergantungan pada penggunaan tombol secara acak telah dikurangi. Selain itu, fokus *fighting games* modern lebih mengutamakan strategi daripada kekuatan atau kecepatan secara langsung dan terdapat juga jenis *fighting games* yang populer seperti *Street Fighter*, *The King of Fighters*, *Tekken*, *Soul Calibur*, *Mortal Kombat*, dan lain-lain.

## **7. Game Platformer**

*Game platformer* adalah jenis game yang menampilkan lingkungan baik secara vertikal maupun horizontal yang akan meminta pemain untuk mengalahkan musuh, berlari, melompati rintangan, berenang, dan menghindari halangan seperti yang terdapat dalam game *Super Mario* dan *Donkey King*.

*Game platform* modern, seperti *Prince of Persia*, *Tomb Raider*, dan *Uncharted*, menampilkan unsur-unsur yang lebih sering ditemukan dalam *action RPG*. Selain itu, terdapat juga beberapa game seperti *Ghost of Tsushima* yang memiliki unsur-unsur *platform*.

## **8. Game Survival dan Horror**

Dalam permainan dengan jenis game *survival* dan *horror*, fokus utamanya adalah pada kemampuan pemain untuk bertahan hidup, menciptakan

ketegangan, dan menghadirkan adegan yang penuh dengan darah. *Game horror* yang terbaik adalah *game* yang mampu menjaga keseimbangan antara elemen-elemen tersebut untuk memberikan pengalaman yang unik dan tidak terlupakan.

Kebanyakan *game* dengan jenis *game survival* dan *horror* yang menggabungkan unsur-unsur permainan *action* RPG atau *adventure games* dengan aspek bertahan hidup seperti pengelolaan *inventaris*, dan senjata dengan sumber daya yang terbatas, dan berkontribusi pada kelangsungan hidup. *Game* ini menantang karena membutuhkan strategi untuk tetap bertahan dengan keterbatasan persediaan yang ada.

## 9. *Stealth*

*Stealth* merupakan jenis *video game* yang memerlukan pemain untuk bergerak dengan diam-diam dan melumpuhkan musuh atau menyelesaikan misi tanpa terdeteksi. Sebagian besar jenis *video game* yang melibatkan tindakan membunuh musuh akan mencakup berbagai elemen *stealth* di dalamnya.

Dalam mengklasifikasikan *game* tersebut, penggunaan taktik menyelinap yang lebih diutamakan daripada pertarungan langsung merupakan hal yang penting untuk memenuhi kriteria permainan *stealth* yang murni. Apabila *game* tersebut tidak mengedepankan taktik menyelinap, maka besar kemungkinan *game* tersebut akan digolongkan sebagai *Action Role Playing Game* atau *First Person Shooter*.

## 10. *Interactive Movie*

*Bandersnatch* dari *Netflix* telah membawa gaya film interaktif menjadi terkenal. Permainan *interactive movie* merupakan sebuah film yang

memungkinkan pemain mengambil keputusan untuk karakter dalam cerita, yang akan menghasilkan hasil yang berbeda. Selain itu, permainan ini memberikan pengalaman yang unik di mana pemain dapat menggabungkan tontonan film dengan bermain *game*, meskipun terdapat batasan dalam jumlah pilihan keputusan yang tersedia untuk melihat perkembangan ceritanya.

### **11. *Game* Deduksi Sosial**

*Game* deduksi sosial merupakan jenis permainan yang dimainkan secara langsung antara beberapa orang seperti *game Cluedo* dan *game* yang paling populer adalah *Among Us*. Dalam permainan ini, interaksi dengan pemain lain sangat penting untuk mengungkap peran atau afiliasi tim yang tersembunyi. Oleh karena itu, pemain perlu menggunakan logika dan deduksi untuk mengeliminasi kemungkinan, dan mencari tahu identitas setiap individu.

### **12. *Game* AR**

*Game* AR (dikenal sebagai *Augmented Reality*) merupakan jenis *game* di mana dunia nyata menjadi elemen utama dalam *game*. Contohnya adalah *Pokemon Go*, yang saat ini menjadi salah satu *game* AR yang paling populer. Dalam *Pokemon Go*, pemain diharuskan untuk menjelajahi lingkungan sekitar untuk mengunjungi *Poke Centers*, melawan *gym leaders*, dan menangkap Pokemon sampai pemain berhasil menangkap semua yang ada.

### **13. *Game Side Scrolling***

*Game side scrolling* merupakan *game* yang di mana pemain dapat bergerak dari kiri ke kanan untuk melanjutkan tujuan permainan dan terkadang dapat bergerak secara horizontal. *Game* yang memiliki unsur *side scrolling* ini



sangat populer, dan masih dapat dimainkan hingga saat ini seperti *game Shovel Knight* dan *Mario Bros* (Solihin, Hidayat, and Aldya 2019).

### 2.1.2 *Game* edukasi

*Game* edukasi adalah jenis permainan yang secara khusus dirancang dengan tujuan edukasi, atau memiliki nilai edukasi yang terjadi secara tidak langsung atau sekunder. Beberapa sumber menggambarkan *game* edukasi sebagai metode pengajaran yang memperkuat pengetahuan yang telah dipelajari dan mengulanginya dalam lingkungan yang lebih menyenangkan. Definisi ini umumnya digunakan oleh MoNE (MEB) (Küpeli 2019).

Sementara itu, berdasarkan penelusuran peneliti terdapat beberapa pengertian mengenai *game* edukasi yang telah disampaikan oleh berbagai sumber penelitian, antara lain:

1. Dalam penelitian yang di lakukan (Ariffin, Aszemi, and Mazlan 2021), ditemukan bahwa tahun 1980-an, ketika dilakukan penelitian tentang perkembangan kognitif, pembelajaran, dan efek motivasi saat para pemain bermain *video game*. Pada masa tersebut, *game* edukasi diperkenalkan untuk pertama kalinya. Selain itu, pengalaman pendidikan pemain dapat secara positif dipengaruhi oleh media pendidikan interaktif yang menggabungkan berbagai elemen. Sejak saat itu, berbagai penelitian telah dilakukan untuk meneliti dampak permainan terhadap pemain. Penelitian-penelitian tersebut berhasil mengidentifikasi hingga 16 manfaat pembelajaran yang diperoleh ketika pemain terlibat dalam *video game*. Selain itu, perlu dicatat bahwa generasi saat

ini menunjukkan minat yang lebih besar terhadap pembelajaran digital dibandingkan metode tradisional. Hal ini menunjukkan perlunya dilakukan perubahan dalam sistem pendidikan guna memenuhi kebutuhan generasi baru yang semakin kompleks. Permintaan yang semakin tinggi terhadap *video game* edukasi juga mendorong timbulnya persaingan di antara perusahaan-perusahaan untuk menciptakan *video game* edukasi yang komprehensif dan lebih sesuai dengan kebutuhan pemain.

2. Dalam penelitian yang dilakukan (Fayanto et al. 2022) menyebutkan bahwa *game* edukasi merupakan jenis *game* yang dirancang dengan tujuan memberikan pengajaran kepada pemain mengenai subjek dan objek tertentu serta memberi mereka bekal keterampilan yang diperlukan. Selain itu, *game* edukasi juga dirancang dengan tujuan untuk mendukung proses pembelajaran individu terkait subjek tertentu. Tujuan tersebut meliputi memperluas konsep, memperkuat pemahaman, memahami peristiwa sejarah atau budaya serta membantu pemain mengembangkan keterampilan yang diperlukan. Dengan demikian, *game* edukasi dapat dianggap sebagai metode pembelajaran yang spesifik dalam meningkatkan motivasi dan minat dalam proses belajar. Selanjutnya, *game* tersebut juga mampu memfasilitasi kolaborasi yang mencakup aspek-aspek *pedagogi* dan materi pembelajaran.

Selain itu, *game* edukasi juga memiliki peran yang lebih luas daripada sekadar memperkuat pengetahuan yang telah dipelajari sebelumnya. *Game* tersebut juga digunakan oleh pemain untuk memperoleh pengetahuan dan keterampilan baru, mengembangkan kemampuan mentransfer informasi yang telah dipelajari ke dalam

kehidupan sehari-hari, serta meningkatkan perkembangan pribadi dan berbagai tujuan lainnya (Küpeli 2019).

Demikian pula, *game* edukasi memiliki beragam tujuan yang dapat dikemukakan, sehingga tidak semua tujuan tersebut harus berkaitan dengan pencapaian edukasi secara langsung. Terkadang, manfaat edukasi dijadikan sebagai tujuan kedua atau ketiga untuk meningkatkan tingkat hiburan dan kesenangan yang menjadi prioritas utama. Dengan demikian, pemain dapat memperoleh pembelajaran sambil bermain dan menikmati kesenangan yang dihadirkan (Küpeli 2019).

Dalam merancang sebuah *game* edukasi, perlu diperhatikan hal-hal seperti melakukan analisis terhadap para pemain yang akan bermain, mengidentifikasi kebutuhan mereka, dan menjaga tingkat kompetisi pada tingkat yang optimal. Selain itu, hal ini bertujuan untuk mendorong kerjasama yang baik antara pemain. Apabila tingkat kompetisi terlalu tinggi, terdapat kemungkinan bahwa beberapa pemain akan menyerah atau menjadi pemain pasif. Oleh karena itu, sangat penting untuk melakukan analisis yang baik terhadap tipe-tipe pemain yang ada (Küpeli 2019).

Berikut ini terdapat beberapa jenis *game* edukasi yang dikembangkan untuk pembelajaran para pemain yang telah disampaikan oleh (Pro 2022) dan sumber penelitian, sebagai berikut:

### **1. *Puzzlers* dan *Party Games***

*Puzzle game* adalah jenis permainan teka-teki yang menawarkan kesenangan bagi pemain yang menikmati tantangan dan menguji kemampuan

berpikir. Beberapa *game puzzle* yang populer seperti *Candy Crush*, *Bejeweled*, *Portal*, *Tetris*, *Braid*, dan lain-lain.

Demikian juga, dalam *game* pesta seringkali terdapat *puzzle* atau *mini game* yang dapat dimainkan oleh pemain bersama teman-teman. Selain itu, *game puzzle* juga memiliki *sub-genre* yang beragam seperti *game* logika, *game* pemrograman, *game* fisika, *game* uji coba, dan sebagainya.

## 2. *Game* Simulasi

Meskipun tidak dapat dipastikan bahwa *game* simulasi dapat mempersiapkan pemain untuk mengemudi atau terbang di dunia nyata, terdapat beberapa jenis *game* simulasi seperti permainan mobil, helikopter, atau pesawat yang memberikan tingkat kesenangan yang tertentu. Tujuan utama dari *game* simulasi adalah untuk mereplikasi pengalaman yang terjadi dalam kehidupan nyata.

Dalam penentuan kualitas suatu simulasi ditentukan oleh mekanika dan grafik yang digunakan. Selain itu, penggunaan dana pengembangan *game* juga berperan penting dalam hal ini. Setiap detail, mulai dari *respons* yang akurat pada setir hingga keakuratan cuaca saat terbang pesawat memiliki peran penting. Meskipun mayoritas *game* simulasi memiliki tingkat realisme yang tinggi dan penting untuk diingat bahwa kemampuan mengendalikan pesawat dalam *game* simulasi tidak secara otomatis mempersiapkan seseorang untuk menerbangkan pesawat di dunia nyata.

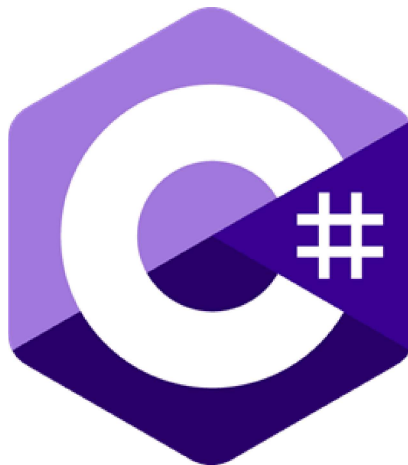
## 3. Permainan Kuis

Permainan kuis dirancang untuk merangsang berpikir, meningkatkan konsentrasi dan kemampuan pemecahan masalah. Permainan kuis digunakan sebagai salah satu media yang efektif untuk memberikan pengajaran, serta melalui pendekatan media yang unik dan menarik. Dengan pendekatan praktik langsung (*learning by doing*), para pemain harus belajar melalui permainan untuk menyelesaikan masalah yang dihadapi (Fayanto et al. 2022).

Salah satu keunggulan permainan kuis adalah adanya animasi yang dapat meningkatkan daya ingat, sehingga anak-anak dapat menyimpan materi pelajaran dalam jangka waktu yang lebih lama dibandingkan dengan metode pengajaran tradisional (Fayanto et al. 2022).

## 2.2 Teori Khusus

### 2.2.1 Bahasa pemrograman C#



**Gambar 2.1** Logo C#  
**Sumber:** seeklogo.com

Menurut penelitian (Stefano Mongi, Lumenta, and Sambul 2018) C# atau dikenal sebagai *C Sharp* adalah sebuah bahasa pemrograman berorientasi objek

yang dikembangkan oleh *Microsoft* sebagai bagian dari inisiatif *.NET Framework*. Selain itu, Bahasa pemrograman ini memiliki dasar dari C++ dan telah terpengaruh oleh fitur-fitur dari bahasa pemrograman lain seperti *Java*, *Delphi*, *Visual Basic*, dan sebagainya, dengan beberapa penyederhanaan.

Dalam penelitian (Uzunbayir and Kurtel 2019) menyebutkan bahwa C# merupakan sebuah pemrograman modern yang menggunakan tipe data yang kuat, deklaratif, dan berorientasi objek. Selain itu, C# juga terdapat pemeriksaan tipe data yang sangat ketat seperti mendeteksi kesalahan dan cacat secara dini, baik saat kompilasi maupun saat *runtime*.

Berdasarkan sumber yang diambil dari (Cogentx 2023), bahasa pemrograman C# digunakan secara luas dalam berbagai konteks, termasuk:

1. Pengembangan aplikasi web berbasis server menggunakan *.NET*.
2. Pembuatan aplikasi *desktop* untuk sistem operasi *Windows* dan *macOS*.
3. Pengembangan aplikasi *mobile native* untuk *platform* *iOS* dan *Android* menggunakan *Xamarin*.
4. Implementasi *microservices* menggunakan *Docker Containers*.
5. Layanan berbasis *cloud*.
6. Penerapan teknologi *Machine Learning*.
7. *Internet of Things* (IoT).
8. Pengembangan *game* untuk komputer, Mac, konsol, perangkat *mobile*, dan *platform* VR/AR dengan menggunakan *Unity*.

Bahasa pemrograman C# juga dapat dijalankan secara *offline* pada komputer dan digunakan sebagai bahasa pemrograman yang berfungsi untuk pengembangan

*game*. Selain itu, bahasa pemrograman ini juga dapat digunakan dalam *Unity* untuk membuat model *game* 2D dan 3D. Oleh karena itu, bahasa pemrograman C# dapat diintegrasikan dengan *Unity* untuk menciptakan *game* yang menekankan pada simulasi, dan bangunan arsitektur yang dirancang untuk pemodelan dan *rendering* dalam *Unity* (Stefano Mongi et al. 2018).

Pada *Unity*, C# digunakan sebagai fitur *scripting* yang memudahkan dalam melakukan rotasi dan perubahan skala objek hanya dengan beberapa baris kode. Selanjutnya, C# juga memungkinkan untuk melakukan duplikasi, penghapusan, dan perubahan properti, dan memudahkan bahasa pemrograman C# untuk mendefinisikan variabel properti visual melalui skrip yang dapat dieksekusi dalam aplikasi *Unity*. Bahasa pemrograman ini berdasarkan *.NET Framework* yang mendukung *open source*, sehingga program dapat dijalankan menggunakan sumber terbuka (Stefano Mongi et al. 2018).

### 2.2.2 Pembelajaran



**Gambar 2.2 Logo C**  
**Sumber:** [academy.alterra.id](http://academy.alterra.id)

Penelitian ini didukung oleh variabel bahasa pemrograman C. Menurut (A S 2023), C merupakan bahasa pemrograman prosedural dengan sistem statis yang menyediakan fungsi pemrograman terstruktur, rekursi, dan skop variabel leksikal. C diciptakan dengan konstruksi yang memungkinkan transfer yang baik ke instruksi perangkat keras umum. Bahasa pemrograman C telah lama digunakan dalam program-program yang sebelumnya ditulis dalam bahasa rakitan (*assembly language*).

Bahasa pemrograman C merupakan bahasa pemrograman yang bersifat *machine-independent*, digunakan secara luas untuk menciptakan berbagai jenis aplikasi dan sistem operasi seperti *Windows*, serta program-program kompleks seperti basis data *Oracle*, *Git*, *interpreter Python*, dan permainan (*games*). Bahasa C juga dianggap sebagai dasar pemrograman yang penting dalam proses pembelajaran pemrograman lainnya. Aplikasi-aplikasi yang menggunakan bahasa pemrograman C meliputi sistem operasi dan beragam perangkat lunak aplikasi untuk arsitektur komputer mulai dari superkomputer hingga PLC dan sistem (A S 2023).

Menurut (A S 2023), bahasa pemrograman C adalah bahasa pemrograman yang memiliki keunggulan dan beberapa manfaat dibandingkan dengan bahasa pemrograman lainnya. Berikut ini adalah manfaatnya dari bahasa pemrograman, sebagai berikut:

1. Bahasa C merupakan bahasa pemrograman yang serbaguna dan dapat digunakan untuk berbagai jenis aplikasi.



2. Bahasa C memiliki tingkat efisiensi yang tinggi, sehingga memungkinkan penulisan kode yang cepat dan dapat diandalkan.
3. Bahasa C bersifat portabel, artinya program yang ditulis dalam bahasa C dapat dengan mudah dikompilasi dan dijalankan di berbagai *platform*.
4. Bahasa C merupakan bahasa pemrograman yang telah mapan dengan komunitas pengembang yang besar dan aktif. Selain itu, komunitas ini terus bekerja untuk meningkatkan bahasa C, menciptakan alat-alat, dan perpustakaan baru.

Dengan keunggulan dan manfaat tersebut, bahasa pemrograman C menjadi pilihan yang kuat dan penting dalam proses pembelajaran pemrograman serta dalam pengembangan berbagai aplikasi, sistem operasi, dan perangkat lunak lainnya.

### **2.2.3 UML (*Unified Modeling Language*)**

*Unified Modeling Language* adalah bahasa standar berorientasi objek yang bertujuan untuk mempermudah pengguna dalam mengembangkan sistem. UML berfungsi sebagai sarana pemodelan yang dapat disesuaikan dengan keinginan dan kebutuhan pengguna tanpa harus mengikuti metodologi tertentu (Alfisyahri and Simanjuntak 2020).

Selain itu, UML juga digunakan sebagai bahasa visual oleh insinyur perangkat lunak dan pengembang untuk merancang, mendokumentasikan, dan memvisualisasikan sistem perangkat lunak. Meskipun UML bukan bahasa pemrograman, namun UML menyediakan representasi visual yang membantu pengembang perangkat lunak dalam memahami hasil atau potensi kesalahan dalam

program. Dalam hal ini, penggunaan UML dapat menghemat waktu dan meningkatkan kolaborasi antara anggota tim pengembang (Editorial Team 2023).

Berikut ini terdapat 3 manfaat utama dari UML menurut (Editorial Team 2023), adalah sebagai berikut:

1. Meningkatkan komunikasi: UML memungkinkan para pengembang untuk berkomunikasi mengenai topik yang melibatkan sistem dan kebutuhannya. Dengan demikian, UML dapat membantu para profesional dalam berinteraksi dengan rekan kerja dan antar tim yang berbeda.
2. Menghasilkan model sistem yang jelas: Sistem model UML menggunakan representasi subjek dan menggabungkannya dengan serangkaian konsep untuk memperkaya komunikasi visual.
3. Menyediakan standarisasi: UML mengintegrasikan teknologi dan sistem informasi serta menerapkan teknik khusus dalam pengembangan sistem yang berhasil dan terstandarisasi.

Selain itu, pengembang juga menggunakan model ini untuk mendokumentasikan langkah-langkah dan tindakan yang terlibat dalam proses bisnis, seperti:

1. Membedakan peran dan keterlibatan orang-orang dalam suatu proses tertentu.
2. Merangkum perjalanan organisasi.
3. Mengungkapkan bagaimana interaksi antar individu terjadi selama aktivitas bisnis berlangsung.

Dalam hal ini, penggunaan UML memberikan manfaat yang signifikan dalam pemahaman dan dokumentasi sistem, serta memfasilitasi komunikasi yang lebih baik antara anggota tim dan lintas tim.



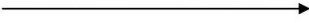

Dalam penelitian (Hutabri and Dasa Putri 2019) menjelaskan beberapa diagram pada UML adalah sebagai berikut:

### 1. *Use Case Diagram*

Diagram Kasus Pengguna adalah diagram yang digunakan untuk memodelkan perilaku suatu sistem dengan menggambarkan interaksi antara 1 atau lebih aktor yang akan menggunakan sistem.

**Tabel 2.1** *Use Case Diagram* (Diagram Kasus Pengguna)

| No. | Simbol   | Deskripsi   |
|-----|--|---|
| 1   | <p><i>Use Case</i></p>                        | Adalah fungsi atau kegunaan dari sistem untuk melakukan pertukaran pesan antara unit dan aktor.                                   |
| 2   | <p><i>Actor/ Aktor</i></p>  <p>nama aktor</p> | Merupakan aktor atau entitas, sistem, atau proses yang berperan dan memiliki hubungan dengan sistem informasi yang sedang dibuat. |
| 3   | <p><i>Association/ Asosiasi</i></p>           | Merupakan interaksi antara aktor dengan <i>Use Case</i> atau komunikasi antara aktor dan <i>Use Case</i> .                        |

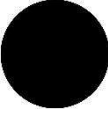

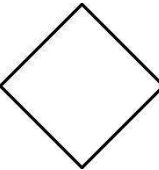

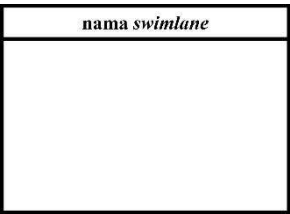
| No. | Simbol   | Deskripsi  |
|-----|--|--|
| 4   | <p data-bbox="512 387 735 421"><i>Extend/ Ekstensi</i></p> <p data-bbox="544 461 692 488">&lt;&lt;extend&gt;&gt;</p>    | <p data-bbox="831 387 1359 786">Merupakan relasi tambahan dari <i>Use Case</i> terhadap <i>Use Case</i> lain. Selain itu, <i>Use Case</i> ini dapat berdiri sendiri tanpa <i>Use Case</i> yang ditambahkan <i>Use Case</i> dapat berdiri. Arah panah menunjukkan arah ke <i>Use Case</i> yang ditambahkan.</p>               |
| 5   | <p data-bbox="520 835 727 936"><i>Generalization/</i><br/>Generalisasi</p>   | <p data-bbox="831 835 1359 1301">Merupakan hubungan umum – khusus (generalisasi dan spesialisasi) untuk 2 <i>Use Case</i>. Satu <i>Use Case</i> memiliki fungsi yang lebih umum, sementara <i>Use Case</i> lainnya lebih spesifik. Arah panah menunjukkan arah ke <i>Use Case</i> yang umum.</p>                             |
| 6   | <p data-bbox="507 1350 743 1384"><i>Uses atau Include</i></p> <p data-bbox="539 1424 695 1451">&lt;&lt;include&gt;&gt;</p>  <p data-bbox="555 1491 679 1518">&lt;&lt;uses&gt;&gt;</p>  | <p data-bbox="831 1350 1359 1816">Merupakan relasi antara 2 <i>Use Case</i>, dimana <i>Use Case</i> yang ditambahkan menggunakan <i>Use Case</i> tambahan. <i>Use case</i> yang ditambahkan memerlukan <i>Use Case</i> tambahan untuk menjalankan fungsionalitasnya sebagai syarat menjalankan <i>Use Case</i> tersebut.</p> |

**Sumber:** (Hutabri and Dasa Putri 2019)

## 2. Activity Diagram

Diagram Aktivitas merupakan diagram yang mendeskripsikan alur kerja atau aktivitas dalam sistem atau proses bisnis yang terdapat di dalam sistem atau perangkat lunak.

**Tabel 2.2 Activity Diagram (Diagram Aktivitas)**




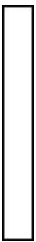
| No. | Simbol  | Deskripsi  |
|-----|---|--|
| 1   | Status awal/ akhir<br>             | Merupakan keadaan awal atau akhir keadaan dari sistem, setiap diagram aktivitas memiliki satu status awal. |
| 2   | Aktivitas<br>                    | Merupakan tindakan atau kegiatan yang dilakukan oleh sistem, sering diawali dengan kata kerja.             |
| 3   | <i>Decision/ Percabangan</i><br> | Merupakan percabangan dalam diagram aktivitas untuk keputusan yang memiliki lebih dari 1 pilihan.          |
| 4   | <i>Join/ Penggabungan</i><br>    | Merupakan penggabungan aktivitas-aktivitas yang terpisah menjadi 1.  |
| 5   | <i>Swimlane</i><br>              | Merupakan pembagian berdasarkan organisasi bisnis yang bertanggung jawab atas aktivitas yang terjadi.      |


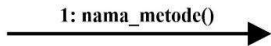
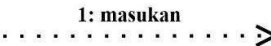
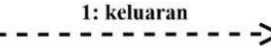
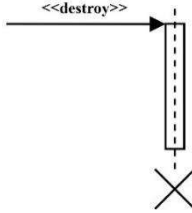
**Sumber:** (Hutabri and Dasa Putri 2019)

### 3. *Sequence Diagram*

Diagram Urutan menjelaskan perilaku objek pada *Use Case* dengan menggambarkan alur waktu hidup dari objek dan pesan yang diterima dan dikirim antar objek.

**Tabel 2.3** *Sequence Diagram* (Diagram Urutan)

| No. | Simbol  | Deskripsi  |
|-----|---|--|
| 1   | Aktor nama aktor atau tanpa waktu aktif<br> | Merupakan pihak eksternal, seperti orang atau sistem lain, yang terhubung dengan sistem yang dibuat. Aktor bisa berupa simbol bergambar orang atau simbol tanpa waktu aktif. |
| 2   | <i>Lifeline</i> /Garis hidup<br>           | Merupakan representasi kehidupan objek dalam diagram, menggambarkan waktu objek berinteraksi.  |
| 3   | <i>Objek</i><br>                           | Merupakan entitas yang terlibat dalam pertukaran pesan pada diagram.   |
| 4   | Waktu aktif<br>                            | Merupakan simbol yang menunjukkan bahwa objek sedang berinteraksi dan berada dalam keadaan aktif. Semua tahapan yang terjadi selama keadaan aktif akan dijelaskan.           |

| No. | Simbol   | Deskripsi  |
|-----|--|--|
| 5   | Pesan tipe <i>create</i><br>    | Merupakan pernyataan bahwa suatu objek menciptakan objek lain.   |
| 6   | Pesan tipe <i>call</i><br>      | Merupakan pernyataan bahwa suatu objek memanggil metode atau operasi pada objek lain atau diri sendiri.                                    |
| 7   | Pesan tipe <i>send</i><br>      | Merupakan pernyataan bahwa objek mengirimkan informasi, masukan, atau data ke objek lain.  |
| 8   | Pesan tipe <i>return</i><br>  | Merupakan pernyataan bahwa suatu objek menjalankan metode atau operasi yang memberi hasil pengembalian atau keluaran ke objek tertentu.    |
| 9   | Pesan tipe <i>destroy</i><br> | Merupakan pernyataan bahwa suatu objek mengakhiri hidup objek lain. Idealnya, jika ada <i>create</i> , sebaiknya juga ada <i>destroy</i> . |

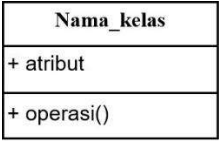
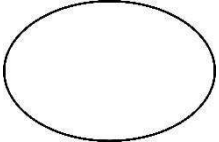




Sumber: (Hutabri and Dasa Putri 2019)

#### 4. Class Diagram


Diagram Kelas digunakan untuk membuat sistem dengan mendeskripsikan struktur sistem dari sisi pendefinisian kelas yang dibuat. Kelas

memiliki atribut (variabel) dan metode (fungsi) yang memungkinkan kelas berinteraksi dalam sistem.

**Tabel 2.4** *Class Diagram* (Diagram Kelas)

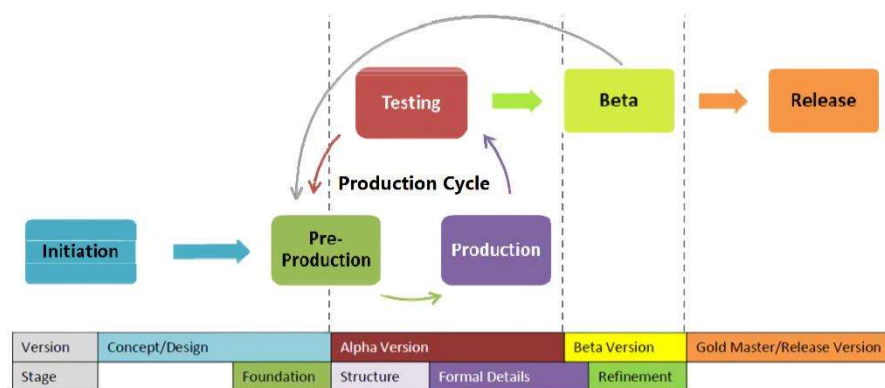
| No. | Simbol   | Deskripsi   |
|-----|--|---|
| 1   | <p style="text-align: center;">Kelas</p>    | Merupakan struktur dalam sistem yang memiliki atribut dan operasi.  |
| 2   | <p style="text-align: center;">Antarmuka</p>  <p style="text-align: center;"><i>nama_interface</i></p> | Merupakan struktur yang mirip dengan kelas, namun hanya memiliki metode yang di deklarasikan tanpa isi dan atribut.                             |
| 3   | <p style="text-align: center;">Asosiasi</p>   | Merupakan hubungan antarkelas yang bersifat umum, biasanya dilengkapi dengan <i>multiplicity</i> .  |
| 4   | <p style="text-align: center;">Asosiasi bearah</p>    | Merupakan hubungan antara kelas yang memiliki arti, di mana 1 kelas digunakan oleh kelas lain, biasanya dilengkapi dengan <i>multiplicity</i> . |
| 5   | <p style="text-align: center;">Generalisasi</p>   | Merupakan hubungan antarkelas yang menunjukkan relasi umum ke khusus.   |
| 6   | <p style="text-align: center;">Kebergantungan</p>   | Merupakan hubungan kebergantungan antara kelas.   |



| No. | Simbol  | Deskripsi  |
|-----|---|--|
| 7   | Agregasi<br> | Merupakan hubungan antarkelas yang menyatakan hubungan semua-bagian ( <i>whole-part</i> ). |

Sumber: (Hutabri and Dasa Putri 2019)

#### 2.2.4 Metode pengembangan *game*



**Gambar 2.3** Tahap Metode GDLC

Sumber: (Andriyat Krisdiawan and Darsanto 2019)

Metode pengembangan *game* yang digunakan dalam penelitian adalah *Game Development Life Cycle* (GDLC), yang berasal dari *Software Development Life Cycle* (SDLC) (Roedavan et al. 2021). Selain itu, menurut penelitian (Andriyat Krisdiawan and Darsanto 2019) *Game Development Life Cycle* merupakan suatu metode pengembangan *game* yang mengimplementasikan pendekatan berulang yang terdiri dari 6 tahap pengembangan, yang dimulai dengan tahap inisialisasi, pra

produksi, produksi, pengujian, beta, dan rilis. Secara keseluruhan, terdapat 6 tahap yang dapat dikelompokkan menjadi 3 proses utama, yaitu:

1. Tahap inisialisasi yang meliputi konseptualisasi dan perancangan.
2. Tahap produksi yang mencakup pra produksi, produksi, dan pengujian (*Alpha* dan *Beta*).
3. Rilis (*Release*).

Dalam penelitian (Andriyat Krisdiawan and Darsanto 2019) menjelaskan tahap *Game Development Life Cycle* sebagai metode pengembangan *game*, sebagai berikut:

#### **1. *Initiation***

*Initiation* atau Inisiasi merupakan langkah pertama yang melibatkan pemahaman konsep dasar permainan, termasuk penentuan jenis *game* yang akan dikembangkan, identifikasi tren terkini, topik yang relevan dan target pengguna permainan. Tahap ini menghasilkan konsep permainan dan deskripsi yang bersifat ringkas.

#### **2. *Pre Production***

Tahap Pra produksi merupakan bagian penting dalam siklus produksi suatu *game*. Tahap ini meliputi pembuatan dan penyesuaian desain *game* serta pembuatan prototipe permainan. Desain *game* fokus pada definisi jenis *game*, cara bermain, mekanisme/*gameplay*, alur cerita, karakter, tantangan, faktor kepuasan, aspek teknis, dan dokumentasi elemennya dalam Dokumen Desain *Game* (GDD). Dalam tahap pra produksi akan berakhir ketika revisi atau perubahan desain *game* telah disetujui dan terdokumentasikan dalam GDD.

### 3. *Production*

Tahap Produksi merupakan proses inti yang melibatkan penciptaan aset dan pembuatan kode sumber serta integrasi keduanya. Prototipe yang terkait dalam tahap ini berfokus pada perincian dan penyempurnaan formal.

### 4. *Pengujian (Testing)*

Pada tahap Pengujian (*Testing*), dilakukan pengujian internal (*Alpha Testing*) dan eksternal (*Beta Testing*) untuk menguji efektivitas *game* tersebut. Selain itu, terdapat metode pengujian yang spesifik untuk setiap tahap prototipe.

### 5. *Rilis (Release)*

Setelah menyelesaikan proses pengembang *game* dan berhasil melewati tahap *beta testing*, peneliti telah mencapai tahap rilis *game* yang telah dibuat.

## 2.2.5 *Software pendukung*

### 1. *Unity*



**Gambar 2.4** *Unity*  
**Sumber:** unity.com

Dalam penelitian (Coogan and He 2018), *Unity* dijelaskan sebagai salah satu mesin *game* yang sangat populer dengan lebih dari 700 juta pengguna.

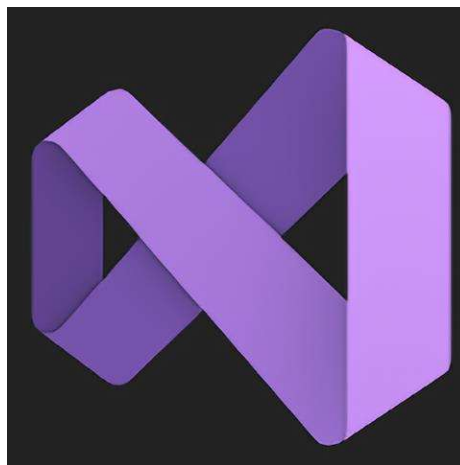
Mesin ini digunakan secara luas dalam pembuatan *video game* untuk berbagai *platform*, termasuk perangkat *mobile*, browser internet, PC, dan Mac. Selain itu, *Unity* juga digunakan dalam pembuatan video, materi edukatif, dan aplikasi neurosains. Penerapan *Unity* tidak hanya terbatas pada desain *game* tradisional, tetapi juga telah mengintegrasikan dukungan untuk realitas virtual dan *augmented reality* yang memungkinkan interaksi dengan berbagai perangkat keras VR.

Dalam penelitian ini (Hussain et al. 2020) menjelaskan bahwa *Unity* memberikan kontribusi dalam berbagai aspek pengembangan *game* seperti pemrosesan, pelacakan aset, penulisan aset, penulisan skrip, dan pengaturan fisika. Fitur-fitur ini dalam pengembangan *game Unity* dapat mengurangi biaya dan waktu pengembangan serta memberikan fleksibilitas saat mengimplementasikan proyek pada berbagai *platform*. Mesin pengembangan *game Unity* ini juga mendukung pengembangan aplikasi *game* untuk 27 *platform* dan perangkat yang berbeda dengan lingkungan pengembangan yang ramah pengguna. *Unity* menyediakan berbagai sumber daya seperti elemen siap pakai, alat intuitif, tutorial, dokumentasi yang jelas, dan komunitas *online*, yang memungkinkan pembuat *game* untuk menciptakan konten 2D dan 3D yang menakjubkan secara gratis. Selain itu, pendekatan *Unity* yang didasarkan pada metodologi *agile* memungkinkan rilis yang konstan, dan *prototyping* yang cepat serta membantu mempercepat proses pengembangan *game*.

Selain itu, *Unity* juga menyediakan lingkungan perkembangan *game* dalam berbagai aspek. IDE *Unity* mendukung bahasa pemrograman C# dan

*JavaScript* untuk *scripting*, serta menawarkan fitur-fitur penting yang cocok untuk pengembangan *game*. Dalam pengembangan *game* menggunakan *Unity*, ditemukan bahwa mesin ini juga mendukung efek video dan audio berkualitas tinggi yang memungkinkan pengembangan *game* yang cerdas dan efektif. Salah satu keunggulan pengembangan *game* melalui *Unity* adalah kemudahan dalam penyesuaian dan *debugging*. Seluruh unit dan variabel *game* dapat digambarkan melalui *gameplay* yang memudahkan pengembang untuk melakukan *debugging* prosedur saat *runtime*. Hal ini mempercepat proses pengembangan *game* dengan adanya kemampuan *debugging* yang memadai (Hussain et al. 2020).

## 2. *Visual Studio Community*



**Gambar 2.5 *Visual Studio***  
**Sumber:** [visualstudio.microsoft.com](https://visualstudio.microsoft.com)

Menurut (Tiushka 2023) *Visual Studio Community* merupakan sebuah *Integrated Development Environment (IDE)* yang disediakan oleh *Microsoft*. IDE ini dirancang untuk menyediakan pengembang dengan *platform* yang kuat untuk membangun aplikasi yang dinamis. Selain itu, *Visual Studio Community* dapat digunakan secara gratis, dan memiliki kemampuan untuk diperluas sesuai

kebutuhan pengembang serta IDE yang mendukung pengembang aplikasi untuk berbagai *platform*, termasuk *Android*, *iOS*, *Windows*, dan *web*. *Visual Studio Community* juga terdapat fitur-fitur yang lengkap dan fleksibel, sehingga menjadi pilihan yang populer bagi para pengembang dalam mengembangkan aplikasi, baik mereka yang baru memulai maupun yang sudah berpengalaman.

Demikian pula, IDE ini dilengkapi dengan beragam alat dan fitur yang penting untuk mempermudah proses penulisan kode. Berawal dari editor kode yang handal, fasilitas *debugging* yang komprehensif, integrasi dengan *repositori Git* yang sudah ada hingga kemampuan kolaborasi yang lancar serta ekosistem yang terus berkembang dengan adanya berbagai ekstensi. Dalam segala kelebihan tersebut, *Visual Studio Community* memang menjadi solusi yang lengkap dan menawarkan segalanya kepada para pengembang (Tiushka 2023).

*Visual Studio Community* menonjol dengan fitur utamanya, yakni mendukung berbagai bahasa pemrograman, termasuk *C#*, *C++*, *JavaScript*, *Python*, dan lain-lain. Selain itu, kemampuannya dalam mengakomodasi beragam bahasa pemrograman membuatnya menjadi IDE pilihan bagi para pengembang yang beroperasi di berbagai lingkungan pemrograman (Tiushka 2023).

IDE ini juga dilengkapi dengan server internal yang memungkinkan para pengembang untuk menjalankan dan melakukan *debugging* aplikasi web mereka secara langsung dalam lingkungan tersebut. Fitur ini juga disertai dengan adanya *IntelliSense*, sebuah alat penyelesaian kode yang responsif

terhadap konteks, berperan dalam mempercepat proses penulisan kode, membantu meningkatkan produktivitas pengguna, dan memberikan pengalaman penulisan kode yang lancar (Tiushka 2023).

### 2.3 Penelitian Terdahulu

Dalam **Tabel 2.5** ini terdapat penelitian-penelitian terdahulu yang dilakukan oleh peneliti sebelumnya yang relevan dengan penelitian yang dilakukan oleh peneliti. Tabel ini memuat informasi mengenai penulis, judul penelitian, metode, dan kesimpulan.

**Tabel 2.5** Penelitian Terdahulu

| No. | Penulis  | Judul Penelitian  | Kesimpulan  |
|-----|--|---|---|
| 1   | Selvarajah Mohanarajah dan Thambithurai Sritharan    | <p>“<i>SHOOT2LEARN: FIX-AND-PLAY EDUCATIONAL GAME FOR LEARNING PROGRAMMING; ENCHANGING STUDENT ENGAGEMENT BY MIXING GAME PLAYING AND GAME PROGRAMMING</i>”/ ISSN 1547-9714/ Vol. 21/ 2022/ <i>Journal of Information Technology Education: Research.</i>”</p> <p>Metode yang digunakan dalam penelitian ini adalah <i>Fix-and-play gaming approach.</i></p> | <p>Penelitian ini menghadapi masalah yang signifikan dalam pembelajaran bahasa pemrograman dan berhasil mengusulkan pendekatan baru menggunakan permainan <i>fix-and-play</i> sebagai solusi. Hasil penelitian menunjukkan bahwa mahasiswa CS1 yang mengikuti <i>game Shoot2Learn</i> mengalami peningkatan yang signifikan dalam pembelajaran <i>conditional branching</i>. Pendekatan <i>fix-and-play game</i> ini berhasil meningkatkan keterlibatan dan pembelajaran mahasiswa. Penelitian ini memberikan kontribusi penting dalam pemahaman dan perbaikan pembelajaran bahasa pemrograman.</p> |
| 2   | Ade Solihin, Eka Wahyu Hidayat, dan Aldy Putra Aldya | <p>“<i>Application of the Finite State Machine Algorithm on 2D Platformer Rabbit Games vs Zombies</i>”/ ISSN 2528-1682/ Vol. 4/ No. 1/ 2019/ <i>Jurnal Online Informatika.</i>”</p> <p>Metode yang digunakan dalam penelitian ini adalah <i>Multimedia Development Life Cycle (MDLC).</i></p>   | <p>1. Penelitian ini berhasil mengembangkan sebuah <i>game survival 2D platformer</i> dengan menggunakan konsep kecerdasan buatan (AI) yang diimplementasikan melalui algoritma <i>Finite State Machine</i>. Selain itu, <i>game</i> ini dapat dijalankan pada <i>smartphone Android</i> dan memberikan pengalaman bermain yang baru dan berbeda dengan karakter yang tidak biasa.</p>  |

| No. | Penulis   | Judul Penelitian   | Kesimpulan  |
|-----|---|--|---|
|     |   |  | <p>2. Penerapan algoritma pada karakter NPC dalam <i>game</i> ini berhasil menghasilkan pergerakan dan <i>respons</i> yang tepat terhadap gerakan karakter pemain.</p> <p>3. Pengujian <i>beta test</i> menunjukkan bahwa aplikasi <i>game Rabbit vs Zombies</i> termasuk dalam kategori yang layak digunakan, namun masih memerlukan beberapa penyempurnaan berdasarkan masukan dari pengguna.</p>   |
| 3   | Rio Andriyat<br>Krisdiawan dan<br>Darsanto              | <p>“PENERAPAN MODEL PENGEMBANGAN GAME GDLC (<i>GAME DEVELOPMENT LIFE CYCLE</i>) DALAM MEMBANGUN <i>GAME PLATFORM</i> BERBASIS <i>MOBILE</i>/ ISSN 2621-8070/ Vol. 2/ No. 1/ 2019/ TEKNOKOM: Jurnal Teknologi dan Rekayasa Sistem Komputer.”</p> <p>Metode yang digunakan dalam penelitian ini adalah <i>Game Development Life Cycle</i> (GDLC).</p>  | <p>Penelitian ini menunjukkan bahwa penggunaan <i>Game Development Life Cycle</i> (GDLC) dengan pendekatan iteratif yang terdiri dari 6 fase pengembangan merupakan teknik yang efektif dalam mengembangkan <i>platform game</i> berbasis <i>mobile</i>. Selain itu, model perancangan sistem berbasis UML, pengujian <i>alfa</i> dan <i>beta</i> juga berperan penting dalam pengembangan <i>game</i>.</p>   |
| 4   | Fazriyanor<br>Kaurie, Agung<br>Purwanto, dan<br>Minarni | <p>“Pengembangan Teknologi <i>Game</i> Indonesia untuk <i>Game</i> 2D “<i>HEROES OF INDONESIA</i>” Menggunakan <i>Unity</i> 2D <i>Engine</i> Berbasis <i>Mobile</i>/ ISSN 2087-3921/ Vol. XI/ No. 2/ 2020/ Jurnal Ilmu Komputer dan Bisnis.”</p> <p>Metode yang digunakan dalam penelitian ini terdiri dari tahap Analisis Data, Desain Sistem, Pembuatan Program, Pengujian, Implementasi dan Evaluasi.</p> | <p>Dalam penelitian ini mengkaji masalah kurangnya minat generasi muda anak bangsa terhadap sejarah kemerdekaan negara Indonesia dan upaya memanfaatkan teknologi informasi, terutama <i>smartphone</i>, untuk meningkatkan minat dan pemahaman mereka terhadap sejarah tersebut. Dalam penelitian ini, dilakukan penyusunan aplikasi <i>mobile game</i> “<i>Heroes of Indonesia</i>” yang berhasil merancang sebuah <i>game</i> yang sederhana namun menarik. <i>Game</i> ini dilengkapi dengan fitur-fitur yang memberikan pengalaman bermain yang menyenangkan sekaligus memberikan edukasi melalui kuis seputar sejarah kemerdekaan negara Indonesia. Selain itu, <i>game</i> ini dapat memberikan nilai pendidikan yang signifikan kepada pemainnya dengan penggunaan elemen-elemen sejarah seperti penggunaan <i>background</i> bendera Indonesia dengan pulau-pulau di dalamnya, penggunaan senjata yang</p> |

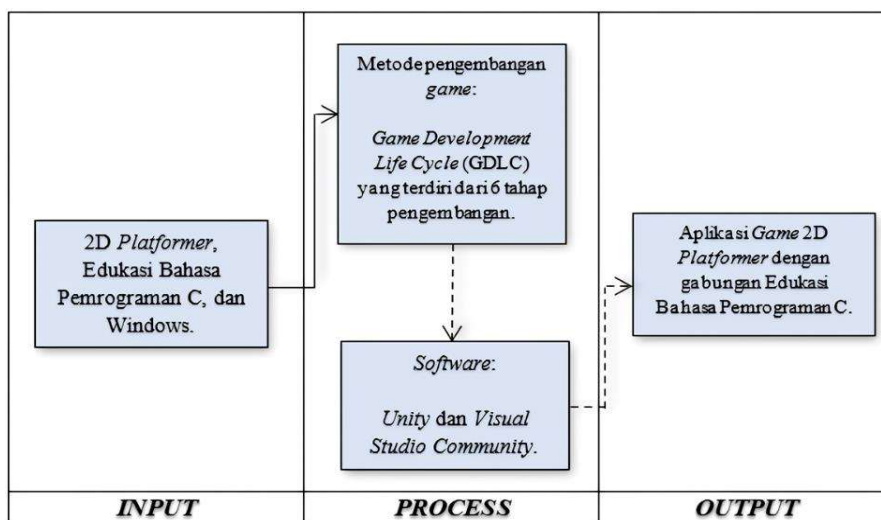


| No. | Penulis  | Judul Penelitian  | Kesimpulan  |
|-----|--|---|---|
|     |  |   | digunakan saat melawan penjajah, penggunaan lagu-lagu nasional Indonesia, serta pengenalan pakaian adat dan hiasan kepala khas Kalimantan Tengah yaitu suku Dayak.  |
| 5   | Grigorios Sideris dan Stelios Xinogalos  | <p>“<i>PY-RATE ADVENTURES: A 2D Platform Serious Game for Learning the Basic Concepts of Programming with Python/ ISSN 1046-8781/ Vol. 50/ No. 6/ 2019/ Simulation and Gaming.</i>”</p> <p>Metode yang digunakan dalam penelitian ini adalah <i>Evaluation Research</i>.</p>  | <p>Dalam penelitian ini, difokuskan pada pemanfaatan <i>serious game</i> sebagai sarana untuk mendorong siswa dalam proses pembelajaran bahasa pemrograman dengan cara yang lebih menarik dan memotivasi. Evaluasi dari partisipasi 31 peserta menunjukkan bahwa <i>game</i> ini memiliki desain yang menarik dan mudah dipahami. Selain itu, peserta memberikan penilaian positif terhadap desain, huruf, dan warna <i>game</i> tersebut. <i>Game</i> ini berhasil meningkatkan kepercayaan diri peserta, menyajikan tantangan yang sesuai, dan mendapatkan kepuasan dari peserta. Peserta juga menganggap <i>game</i> ini relevan dengan pemrograman dan menyatakan bahwa <i>game</i> ini membantu mereka memahami konsep dasar pemrograman dengan <i>Python</i>. Hasil penelitian menunjukkan bahwa <i>game Py-rate Adventures</i> berhasil mencapai tujuan pendidikan yang ditetapkan dan mendapatkan <i>respons</i> positif dari peserta penelitian.</p> |
| 6   | Rickman Roedavan, Bambang Pudjoatmodjo, Yahri siradj, Sazilah Salam, dan BQ Desy Hardianti | <p>“<i>Serious Game Development Model Based on the Game-Based Learning Foundation/ ISSN 2337-5787/ Vol. 15/ No. 3/ 2021/ Journal of ICT Research and Applications.</i>”</p> <p>Penelitian ini menerapkan metode penelitian yang melibatkan model pengembangan <i>Serious Game</i> yang diusulkan. Selanjutnya, metode tersebut mencakup pengujian dan validasi model dengan pendekatan relativis. Selain itu, dilakukan pula uji empiris terhadap hasil <i>output</i> model dengan menggunakan kuesioner pengalaman</p> | <p>Dalam penelitian ini, peneliti mengidentifikasi 2 permasalahan dalam pengembangan <i>Serious Game</i> (SG) adalah kebingungan dalam mendefinisikan <i>Serious Game</i> dan ketidakefektifan model pengembangan yang ada. Penelitian ini berhasil memperjelas definisi SG dengan membatasinya pada permainan digital yang diterapkan dalam konteks serius dan memiliki mekanisme penilaian untuk meningkatkan keterampilan pemain. Selain itu, penelitian ini mengembangkan model pengembangan baru berdasarkan prinsip-prinsip <i>Game-Based Learning</i> (GBL) dengan tujuan memenuhi berbagai konteks serius dan menghasilkan SG dengan karakteristik yang diinginkan.</p> <p>Hasil penelitian menunjukkan bahwa model pengembangan SG yang</p>  |

| No. | Penulis                               | Judul Penelitian  | Kesimpulan   |
|-----|---------------------------------------|---|--|
|     |                                       | pengguna sebagai alat untuk menguji dan mengevaluasi hasil <i>output</i> model.   | diusulkan memiliki validitas teoritis dan memberikan kepuasan kepada pengguna. Aspek daya tarik dan stimulasi dari SG yang dikembangkan berdasarkan model tersebut mendapatkan penilaian positif. Aspek kejelasan, keandalan, dan efisiensi juga ditemukan memiliki nilai yang positif. Namun, terdapat potensi untuk meningkatkan aspek kebaruan dalam pengembangan SG. |
| 7   | Ellbert Hutabri dan Anggia Dasa Putri | <p>“Perancangan Media Pembelajaran Interaktif Berbasis <i>Android</i> pada Mata Pelajaran Ilmu Pengetahuan Sosial untuk Anak Sekolah Dasar/ ISSN 2087-5347/ Vol. 8/ No. 2/ 2019/ Jurnal <i>Sustainable: Jurnal Hasil Penelitian dan Industri Terapan</i>.”</p> <p>Metode yang digunakan dalam penelitian ini adalah RAD (<i>Rapid Application Development</i>).</p> | Penggunaan media pembelajaran interaktif berbasis <i>Android</i> meningkatkan semangat belajar siswa dan memfasilitasi pembelajaran mandiri di rumah. Selain itu, metode pengembangan RAD ( <i>Rapid Application Development</i> ) dan pendekatan berorientasi objek efektif dalam pengembangan media pembelajaran.  |

Sumber: Penelitian (2023)

## 2.4 Kerangka Pemikiran



Gambar 2.6 Kerangka Pemikiran

Sumber: Penelitian (2023)

Berdasarkan **Gambar 2.6**, penelitian ini menggunakan *2D Platformer* dengan gabungan edukasi bahasa pemrograman C dan platform *Windows* sebagai dasar dalam perancangan *game*. Selain itu, perancangan *game 2D Platformer* dengan gabungan edukasi bahasa pemrograman C menggunakan metode GDLC yang terdiri dari 6 tahap, yaitu inisiasi, pra produksi, produksi, pengujian *alpha* (*Alpha Testing*), pengujian *beta* (*Beta Testing*), dan rilis (*release*). Dalam pembuatan aplikasi *game* ini menggunakan *software* pendukung seperti *Unity* dan *Visual Studio Community*. Hasil akhir dari perancangan *game* ini adalah aplikasi *game 2D Platformer* dengan menggabungkan edukasi bahasa pemrograman C.