

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Beberapa penelitian terkait yang dijadikan referensi dalam penelitian ini dapat dilihat pada tabel berikut:

Tabel 1. Tabel Penelitian Terkait

No	Peneliti (Tahun)	Keterangan
(1)	(2)	(3)
1.	Krisna Awalludin, Suhendro Busono, Uce Indahyanti, Yunianita Rahmawati (2025)	Judul : Pengembangan Game 2D "Cat Collection Coin" Berbasis <i>Single Player</i> Menggunakan Metode <i>Finite State Machine</i> untuk Manajemen State Karakter Metode : <i>Finite State Machine</i> (FSM). Hasil : Implementasi FSM pada <i>game</i> "Cat Collection Coin" terbukti efektif dalam mengelola kompleksitas perilaku karakter (seperti <i>state Idle</i> , <i>Move</i> , <i>Enter Portal</i> , <i>Next Stage</i> , dan <i>Collected Coin</i>) secara sistematis dan terstruktur.
2.	Bima Wicaksono, Yuma Akbar (2025)	Judul : Implementasi <i>Finite State Machine</i> (FSM) pada <i>Game</i> Edukasi 2D untuk Pengenalan Fasilitas Umum Anak Usia Dini Metode : <i>Finite State Machine</i> (FSM). Hasil: Berdasarkan hasil penelitian dapat disimpulkan bahwa pengembangan <i>game</i> edukasi 2D berbasis <i>Roblox Studio</i> dengan metode FSM telah berhasil diimplementasikan. Implementasi ini terbukti efektif dalam meningkatkan pemahaman anak usia dini terhadap pengenalan fasilitas umum. Pengujian ini dilakukan kepada 10 responden yang menunjukkan peningkatan rata-rata 30,88 % dengan tingkatan kepuasan pengguna yang tinggi.
3.	Muhammad Aksal Alshal N. Surianto, Dedy Atmajayab, Syahrul Mubarak Abdullahc (2025)	Judul : Implementasi Kecerdasan Buatan dalam Mengatur Mekanik Musuh pada <i>Game Lily's Dreamland: Chaos in the Storybook</i> Metode : <i>Finite State Machine</i> (FSM). Hasil : Hasil pengujian menunjukkan bahwa implementasi FSM pada NPC musuh dalam <i>game</i> ini berfungsi dengan baik dalam menciptakan perilaku NPC yang dinamis dan adaptif, sehingga meningkatkan kualitas <i>gameplay</i> dan kepuasan pemain.
4.	Asrianda, Zulfadli (2022)	Judul : Konsep <i>Finite State Machine</i> dan Implementasinya pada <i>Game</i> Metode : <i>Finite State Machine</i> (FSM). Hasil : Berdasarkan hasil penelitian dapat disimpulkan bahwa penerapan <i>Finite State Machine</i>

No	Peneliti (Tahun)	Keterangan
		(FSM) pada <i>game</i> efektif dalam menghasilkan respons yang dinamis dan meningkatkan pemahaman pemain terhadap alur cerita. Selain itu, metode ini mampu menciptakan perilaku NPC yang tidak monoton melalui variasi respons yang sulit diprediksi, serta mengatur transisi misi atau <i>level</i> secara sistematis dan bertahap.
5.	Michael Hendra Nata, Joseph Dedy Irawan, Hani Zulfia Zahro' (2024)	Judul : Perancangan <i>Game</i> Android Bring Back the Scroll Menggunakan Metode <i>Finite State Machine</i> (FSM) Metode : <i>Finite State Machine</i> (FSM). Hasil: Hasil dari penelitian ini adalah terciptanya <i>game</i> "Bring Back The Scroll" yang menerapkan metode <i>Finite State Machine</i> (FSM) untuk mengontrol respons <i>Non-Player Character</i> (NPC) terhadap pemain. Berdasarkan pengujian kecerdasan buatan yang telah dilakukan, dapat disimpulkan bahwa implementasi metode FSM pada NPC berhasil dengan tingkat keberhasilan 100%, dan seluruh fungsi berjalan dengan baik sesuai rancangan.
6.	Latief Khanan Alqutbhi, Agung Panji Sasmito, Deddy Rudhistiar (2025)	Judul : Implementasi <i>Finite State Machine</i> (FSM) dan <i>Line of Sight</i> (LoS) dalam Perancangan <i>Game</i> 2D Petualangan Raden Panji Inu Kertapati Metode : <i>Finite State Machine</i> (FSM) dan <i>Line of Sight</i> (LoS). Hasil: Berdasarkan hasil pengujian FSM digunakan untuk mengatur <i>state</i> dan transisi perilaku NPC, LoS mengatur kemampuan NPC dalam mendeteksi keberadaan pemain berdasarkan garis pandang. Seluruh fitur telah berfungsi dengan tingkat keberhasilan 100%. Pengujian implementasi metode <i>Finite State Machine</i> (FSM) dan <i>Line of Sight</i> (LoS) menunjukkan hasil 100% sesuai dengan rancangan sistem yang ditetapkan. Selanjutnya, pengujian kepuasan pengguna terhadap 15 responden menghasilkan persentase sebesar 84,44%, yang mengindikasikan bahwa mayoritas pemain menilai <i>game</i> berjalan dengan baik.
7.	Sumaia A. Eltomi, Rasheed S, Mottaleb (2024)	Judul : Video Game Development on Roblox Platform using Lua Programming Language Metode : Metodologi Pengembangan Game Praktis. Hasil: Penelitian ini membuktikan bahwa dengan penguasaan bahasa Lua dan pemanfaatan fitur Roblox Studio, pengembang dapat menciptakan <i>game</i> berkualitas tinggi yang mencakup berbagai mekanisme kompleks tanpa memerlukan sumber daya perangkat keras yang besar, sehingga membuka peluang luas bagi siapa saja untuk mempublikasikan karya mereka secara global.

No	Peneliti (Tahun)	Keterangan
8.	Moh. Fahrizal Afni Romadhan, Krisna Raditya Pratama, M. Syaiful Anam, Muhammad Alfis Sholikhin, Tegar Wicaksono Hermawan (2025)	Judul : Penerapan <i>Model Project-Based Learning</i> (PJBL) dalam Pengembangan Modul Pembelajaran Pembuatan <i>Game</i> Obby Berbasis Roblox Studio untuk Siswa Sekolah Dasar Metode : <i>Project-Based Learning (PjBL)</i> dengan fokus pada desain visual dan pemrograman bahasa Lua. Hasil: Berdasarkan hasil pengujian FSM digunakan untuk mengatur state dan transisi perilaku NPC, LoS mengatur kemampuan NPC dalam mendeteksi keberadaan pemain berdasarkan garis pandang. Seluruh fitur telah berfungsi dengan tingkat keberhasilan 100%. Pengujian implementasi metode <i>Finite State Machine</i> (FSM) dan <i>Line of Sight</i> (LoS) menunjukkan hasil 100% sesuai dengan rancangan sistem yang ditetapkan. Selanjutnya, pengujian kepuasan pengguna terhadap 15 responden menghasilkan persentase sebesar 84,44%, yang mengindikasikan bahwa mayoritas pemain menilai <i>game</i> berjalan dengan baik.
9.	Lily Brown, Andy Friesen, dan Alan Jeffrey (2021)	Judul : <i>Position Paper: Goals of the Lua Type System</i> Metode : Sistem tipe statis. Hasil: Lua merupakan sistem tipe yang dirancang untuk mendukung komunitas pengembang Roblox yang heterogen melalui mekanisme <i>inferensi tipe</i> (<i>type inference</i>), yang memungkinkan adopsi secara bertahap tanpa memerlukan anotasi manual. Sistem ini bersifat fleksibel dengan menyediakan tipe <i>infallible</i> yang tetap mampu memberikan bantuan editor bahkan pada kode yang belum lengkap. Lua menawarkan keseimbangan antara keamanan dan kemudahan penggunaan melalui dua mode operasi, yaitu mode <i>strict</i> untuk deteksi kesalahan secara mendalam dan mode <i>non-strict</i> untuk meminimalkan peringatan yang dapat mengganggu pemula.
10.	Ibnu Bantan Auliah, Febriana Santi, Hani Zulfia (2025)	Judul : Implementasi Metode <i>Finite State Machine</i> (FSM) pada <i>Game Adventure</i> 3D “Jaka Tingkir : Pendekar Pajang” Metode : <i>Finite State Machine</i> Hasil: Hasil pengujian menunjukkan bahwa FSM berhasil mengatur perilaku NPC musuh melalui lima <i>state</i>, yakni <i>Idle</i>, <i>Patrol</i>, <i>Chase</i>, <i>Attack</i>, dan <i>Dead</i>, dengan transisi yang dipicu oleh jarak pemain dan kondisi kesehatan musuh. Uji AI dan kontrol pemain masing-masing mencatat keberhasilan 100%, sementara uji performa pada lima perangkat berbeda menunjukkan hasil 98%–100%. Adapun tingkat kepuasan pengguna mencapai 88,52%, mencerminkan respons positif terhadap visual, kontrol, dan

No	Peneliti (Tahun)	Keterangan
		pengalaman bermain. Secara keseluruhan, implementasi FSM pada <i>game</i> ini mencapai tingkat keberhasilan 99%.

Sejumlah penelitian terdahulu telah menunjukkan bahwa metode *Finite State Machine* (FSM) merupakan pendekatan yang umum digunakan dalam industri pengembangan *game* untuk mengelola perilaku karakter dan *Non-Player Character* (NPC) secara sistematis dan terstruktur.

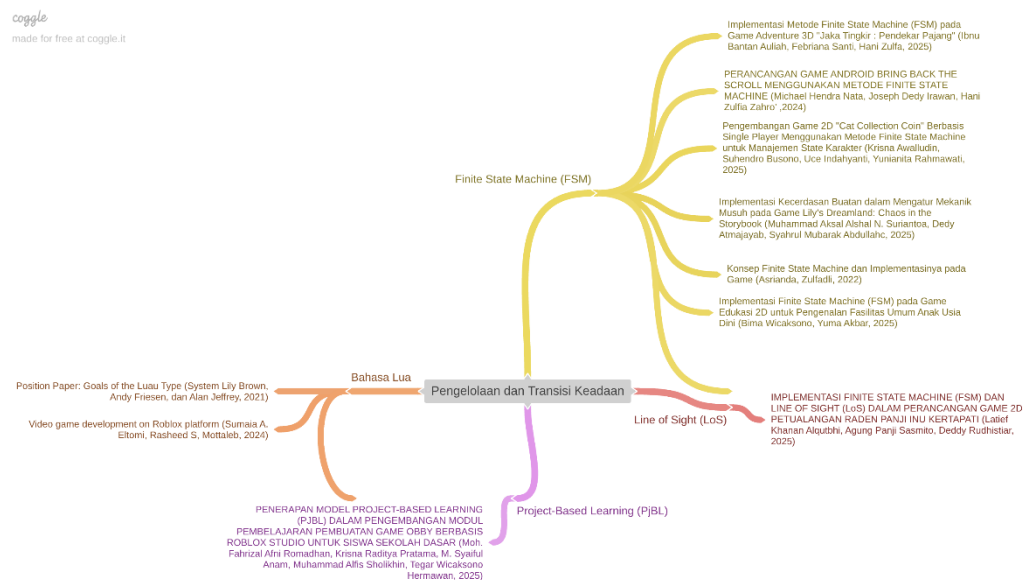
Krisna Awaluddin, Suhendro Busono, Uce Indahyanti, dan Yunianita Rahmawati pada tahun 2025 dalam penelitiannya membuktikan bahwa implementasi FSM pada *game* Cat Collection Coin efektif dalam mengelola berbagai kondisi karakter seperti *idle*, *move*, dan *enter portal* secara terorganisir, yang berkontribusi pada peningkatan pengelolaan kompleksitas perilaku dalam permainan[20]. Sementara itu, Bima Wicaksono dan Yuma Akbar pada tahun 2025 melalui penelitiannya tentang implementasi FSM pada *game* edukasi 2D berbasis *Roblox Studio* mengungkapkan bahwa penerapan FSM mampu mengatur logika permainan dengan baik dan meningkatkan pemahaman pemain terhadap mekanisme yang disajikan[21].

Arsianda dan Zulfadli pada tahun 2022 dalam studinya menjelaskan bahwa konsep FSM mampu menciptakan respons NPC yang lebih dinamis dan tidak monoton, karena setiap karakter memiliki kemampuan untuk bertransisi antar *state* berdasarkan kondisi yang terjadi dalam permainan[6]. Lebih lanjut, Michael Hendra Nata, Joseph Dedy Irawan, dan Hani Zulfia Zahro pada tahun 2024 dalam penelitiannya mengenai *game Android* Bring Back The Scroll menunjukkan bahwa penerapan FSM berhasil mengatur respons NPC terhadap pemain dengan tingkat keberhasilan implementasi yang mencapai 100%, memastikan seluruh fungsi permainan berjalan sesuai rancangan sistem[22].

Latief Khanan Alquthbi, Agung Panji Sasmito, dan Deddy Rudhistiar pada tahun 2025 mengembangkan pendekatan yang lebih kompleks dengan mengombinasikan metode *Finite State Machine* (FSM) dengan *Line of Sight* (LoS), yang bertujuan untuk meningkatkan kemampuan NPC dalam mendeteksi keberadaan pemain dan menentukan perilaku yang tepat berdasarkan jarak pandang karakter[9]. Di sisi lain, Sumaia A. Eltomi dan Rasheed S. Mottaleb pada tahun 2024 dalam penelitian mereka mengenai pengembangan *game* pada platform Roblox menunjukkan bahwa bahasa pemrograman Lua pada *Roblox Studio* memberikan fleksibilitas bagi pengembang untuk menciptakan mekanisme permainan yang kompleks dengan sumber daya yang relatif sederhana,

mengindikasikan potensi besar *Roblox Studio* sebagai platform pengembangan *game* interaktif[23].

Meskipun demikian, dari beberapa penelitian yang telah dikaji, masih ditemukan keterbatasan dalam penerapan FSM, terutama pada konteks pengembangan *game platformer* berbasis Roblox yang mengatur perilaku musuh atau NPC secara lebih interaktif dalam berhadapan dengan pemain. Berdasarkan celah penelitian tersebut, penelitian ini bertujuan untuk mengimplementasikan metode *Finite State Machine* (FSM) pada perilaku NPC dalam *game platformer* berbasis *Roblox Studio*, dengan harapan dapat menghasilkan mekanisme permainan yang lebih dinamis dan meningkatkan pengalaman bermain (*gameplay experience*) bagi pemain selain itu peneliti juga menguji implementasinya menggunakan *Harness Test* dan *Logging* kemudian mengevaluasi pengalaman bermain menggunakan *GEQ Core Module* serta analisis statistik.



Gambar 1 *Mind Map* Penelitian

B. Hipotesis Penelitian

Penelitian ini menerapkan metode FSM untuk mengendalikan *state Idle*, *Patrol*, *Chase*, *Attack*, dan *Dead*. Melalui penerapan FSM, perilaku NPC yang terbentuk dapat menghadirkan *gameplay* yang lebih dinamis, sehingga dapat meningkatkan pengalaman *player* saat bermain. *Game Experience Questionnaire* (GEQ) digunakan untuk mengukur pengalaman bermain yang mencakup komponen *Immersion*, *Competence*, *Flow*, *Tension*, *Challenge*, *Negative Affect*, dan *Positive Affect*. Data hasil pengukuran GEQ dimanfaatkan untuk menilai pengaruh penerapan FSM terhadap *gameplay*. Hipotesis penelitian dirumuskan sebagai berikut:

Tabel 2. Hipotesis Penelitian

Hipotesis Nol	Hipotesis Alternatif
Skor GEQ pada <i>game</i> yang menerapkan FSM tidak lebih tinggi secara signifikan dibandingkan nilai tengah skala	Skor GEQ pada <i>game</i> yang menerapkan FSM lebih tinggi secara signifikan dibandingkan nilai tengah skala

C. Landasan Teori

1. *Game*

Game adalah sistem terstruktur yang melibatkan pemain dalam konflik buatan yang diatur oleh seperangkat aturan spesifik, dan dirancang untuk menghasilkan hasil (*outcome*) yang terukur dan dapat dievaluasi. Menurut Agustinus Nilwan, *game* diartikan sebagai suatu aktivitas terstruktur yang dapat digunakan sebagai alat pembelajaran. Sebuah permainan dapat dikarakteristikan melalui beberapa aspek, yaitu peralatan yang digunakan dalam *game*, peraturan yang berlaku, keterampilan (*skill*) yang dibutuhkan, serta jumlah pemain yang terlibat seperti *Single Player* (pemain tunggal) dan *Multiplayer* (lebih dari satu pemain)[24].

Industri *game* merupakan salah satu sektor yang mengalami perkembangan pesat dalam bidang ilmu komputer[25]. *Game* atau permainan digital merupakan sarana hiburan yang banyak diminati oleh berbagai kalangan, mulai dari anak-anak hingga orang dewasa[26].

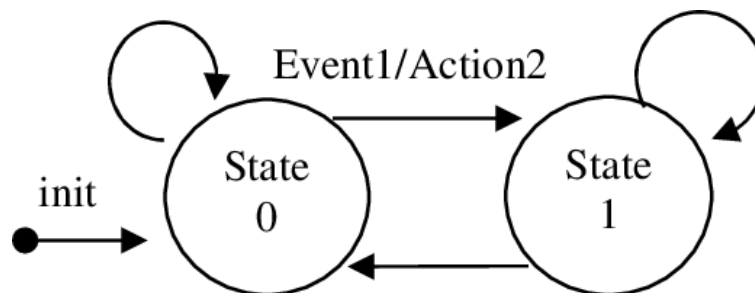
2. Platformer

Platform permainan merupakan jenis *genre* yang terkenal dan telah mengalami kemajuan yang cukup besar. Sebelumnya, *game platformer* hanya bisa dimainkan lewat komputer dan konsol *game*, akan tetapi perkembangan teknologi, membuat *genre* ini bisa dimainkan tanpa terbatas waktu dan tempat menggunakan perangkat seluler. *Game platformer* populer di semua usia, karena desainnya yang menyenangkan dan tantangan dalam cara bermainnya[27].

3. *Finite State Machine* (FSM)

Finite State Machine (FSM) atau Mesin Keadaan Hingga merupakan model komputasi matematis yang digunakan untuk merepresentasikan sistem dengan sejumlah keadaan (*state*) terbatas. Karakteristik utama dari FSM adalah bahwa sistem hanya dapat berada dalam satu *state* aktif pada satu waktu tertentu, dan transisi antar *state* terjadi berdasarkan *input*, *event*, atau kondisi (*condition*) yang telah didefinisikan sebelumnya. FSM diterapkan pada *Non-Player Character* (NPC), yaitu karakter yang dikendalikan oleh sistem kecerdasan buatan (*artificial intelligence*) untuk mendukung dinamika dan interaktivitas permainan[28].

Menurut Michael Sipser dalam bukunya *Introduction to the Theory of Computation*, FSM didefinisikan sebagai model formal yang digunakan dalam teori automata untuk merepresentasikan sistem diskrit yang memiliki jumlah *state* terbatas dengan aturan *transisi* yang eksplisit dan deterministik[29]. Pada penelitian ini, definisi Sipser digunakan sebagai landasan konseptual, sedangkan implementasinya mengacu pada paradigma *game* sehingga setiap *state* direpresentasikan melalui skrip Lua yaitu *Idle*, *Patrol*, *Chase*, *Attack*, dan *Dead*. Model FSM telah diaplikasikan dalam berbagai domain, antara lain perancangan sistem digital, pengembangan *compiler* dan *parser*, serta implementasi kecerdasan buatan pada *game*, khususnya untuk mengelola perilaku *Non-Player Character*(NPC). FSM memiliki dua *state* dan dua input serta empat aksi output yang berbeda. Ketika sistem dinyalakan, sistem akan berpindah ke *state0*. Pada *state* ini, sistem akan menghasilkan aksi pertama jika terjadi *input Event0*. Jika terjadi *input Event1*, maka aksi kedua akan dijalankan, dan sistem akan berpindah ke *state1*, serta proses ini terus berlanjut.



Gambar 2 Diagram *State* Sederhana

4. Lua

Lua merupakan bahasa pemrograman skrip (*scripting language*) yang memiliki karakteristik ringan, cepat, dan mudah dipelajari. Sebagai bahasa yang bersifat *interpreted* (ditafsirkan), kode Lua dieksekusi secara langsung oleh *interpreter* tanpa memerlukan tahap kompilasi seperti yang diperlukan pada bahasa pemrograman kompilasi seperti C atau C++.

Bahasa pemrograman Lua diciptakan pada tahun 1993 oleh tiga peneliti, yaitu Roberto Ierusalimsky, Luiz Henrique de Figueiredo, dan Waldemar Celes. Ketiga peneliti tersebut merupakan anggota dari Kelompok Teknologi Grafis Komputer (*Computer Graphics Technology Group* atau *Tecgraf*) yang berada di bawah Universitas Katolik Kepausan Rio de Janeiro, Brasil. Dalam perkembangannya, platform Roblox mengadopsi versi modifikasi dari Lua 5.1 sebagai bahasa skrip utamanya[30].

Bahasa ini mendapat inspirasi dari SOL (*Simple Object Language*) dan penamaanya dari kata dalam bahasa Portugis yang berarti ‘bulan’. Lua digunakan berbagai aplikasi industri dan menjadi salah satu bahasa pemrograman skrip terkemuka dalam bidang pengembangan *game*[31]. Lua adalah bahasa pemrograman berbasis skrip yang bersifat dinamis, dirancang secara ringkas namun tetap fleksibel karena semantiknya dapat diperluas atau dikembangkan sesuai kebutuhan. Sebagai bahasa skrip, Lua menyediakan API dalam bahasa C yang lebih sederhana dibandingkan dengan bahasa skrip lainnya[32].

5. *Non Player Character* (NPC)

Aksi penyerangan pada *Non Player Character* (NPC) merupakan salah satu elemen penting yang perlu diperhatikan dalam pembuatan *game*[33]. Menurut Bartle, *Non-Player Character* (NPC) merupakan elemen esensial dalam dunia virtual yang menjalankan beberapa fungsi fundamental. Pertama, NPC berfungsi untuk mengisi dan menghidupkan dunia *game*. Kedua, NPC menyediakan sarana interaksi sosial bagi pemain. Ketiga, NPC berperan dalam mendukung dan memperkuat narasi permainan. Melalui fungsi-fungsi tersebut, NPC berkontribusi dalam menciptakan lingkungan virtual yang imersif (*immersive virtual environment*)[34].

6. Roblox

Roblox adalah platform permainan daring (*online gaming platform*) yang sangat populer di kalangan anak-anak dan remaja. Sistem pembuatan permainan (*game creation system*) yang menyediakan ruang digital bagi pengguna untuk mengekspresikan kreativitas, melakukan interaksi sosial, dan membangun identitas digital mereka[35][36].

Roblox merupakan platform digital berbasis *user-generated content* yang memfasilitasi pengguna untuk merancang dan memprogram permainan mereka sendiri melalui *Roblox Studio* dengan menggunakan bahasa pemrograman Lua, yang merupakan turunan dari Lua. Permainan yang telah dikembangkan dapat dipublikasikan dalam platform dan diakses oleh pengguna lain secara daring (*online*).

7. GDLC

Game Development Life Cycle (GDLC) adalah metodologi pengembangan yang mengatur proses pembuatan *game* secara sistematis dan terstruktur, mencakup seluruh tahapan mulai dari inisiasi ide, perancangan, implementasi, pengujian, hingga rilis dan pemeliharaan produk[37].

a) *Initiation*

Tahap *initiation* merupakan fase pembuka dalam pengembangan permainan yang memusatkan perhatian pada pembentukan ide dan konsep dasar *game*. Fase ini mencakup beberapa aktivitas, yaitu melakukan analisis terhadap kebutuhan sistem, menetapkan tujuan dari pengembangan *game*, serta

merumuskan gambaran umum arsitektur sistem yang akan diimplementasikan.

b) *Pre-production*

Tahap *pre-production* adalah tahap perencanaan yang dilakukan dengan lebih teliti sebelum proses pengembangan benar-benar dimulai. Di tahap ini, desain game dibuat lebih rinci, mencakup pengaturan cara bermain, alur cerita dalam permainan, serta persiapan semua bahan dan sistem yang diperlukan.

c) *Production*

Tahap *production* adalah fase implementasi atau pembuatan *game* secara menyeluruh. Pada tahap ini, dilakukan proses pengkodean, membuat gambar dan suara, serta menggabungkan semua bagian permainan sesuai dengan rencana yang sudah disusun sebelumnya.

d) *Testing*

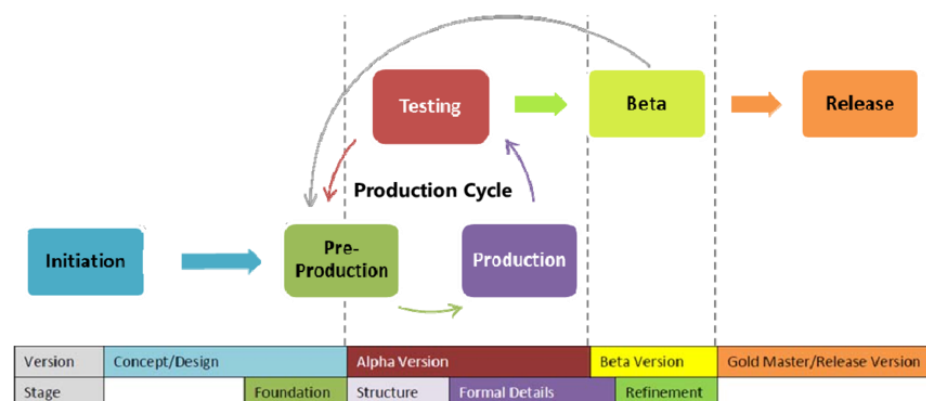
Tujuan dari tahap pengujian adalah untuk menguji kinerja dan fungsionalitas sistem permainan untuk memastikan bahwa *game* berjalan sesuai dengan spesifikasi yang direncanakan. Ini adalah proses untuk menemukan dan memperbaiki kesalahan (*bug*) dalam sistem.

e) *Beta*

Tahap *beta* adalah tahap pengujian berikutnya yang melibatkan pengguna nyata untuk mencoba *game* dan memberikan masukan serta saran tentang pengalaman mereka saat bermain. Fase ini bertujuan untuk mengetahui sejauh mana pengguna menerima *game* tersebut dan menemukan kekurangan-kekurangan sebelum *game* dikeluarkan secara resmi.

f) *Release*

Tahap *release* adalah bagian terakhir dalam proses membuat *game*, *game* itu dipublikasikan dan didistribusikan kepada para pemain. Setelah *game* dirilis, hal tersebut tetap bisa dijadikan bahan untuk dikembangkan dan diperbaiki di tahap setelahnya.



Gambar 3 *Game Development Life Cycle*

8. Pengujian FSM

Pengujian FSM merupakan proses evaluasi terhadap sistem yang menerapkan model *Finite State Machine*, yang bertujuan untuk memverifikasi bahwa setiap kondisi (*state*) dan perpindahan antar kondisi (transisi) berfungsi sesuai dengan rancangan sistem yang telah ditetapkan. Metode *Finite State Machine* (FSM) di implementasikan untuk mengatur perpindahan antar state pada perilaku NPC yaitu *Idle* (diam), *Patrol* (patroli), *Chase* (mengejar), *Attack* (menyerang), dan *Dead* (Mati). Implementasi ini berhasil menciptakan interaksi musuh yang dinamis dan realistis dalam permainan. Hasil pengujian sistem menunjukkan bahwa implementasi FSM dan mekanisme kontrol permainan berjalan dengan baik dan sesuai dengan fungsi yang dirancang[14].

9. *Game Experience Questionnaire* (GEQ)

Game Experience Questionnaire (GEQ) merupakan instrumen psikometrik yang dirancang untuk mengevaluasi pengalaman pengguna dalam konteks bermain *game* secara valid, reliabel, dan sensitif terhadap perubahan yang dirasakan oleh pemain. Dalam modul ini, GEQ mengevaluasi pengalaman bermain *game* melalui tujuh komponen fundamental, yang meliputi *immersion*, *flow*, *competence*, *positive affect*, *negative affect*, *tension*, dan *challenge*. Instrumen yang digunakan dalam penelitian ini adalah *Game Experience Questionnaire* (GEQ). GEQ dikembangkan melalui proyek riset FUGA (*Fun of Gaming*) dan telah teruji melalui penggunaan yang ekstensif dalam berbagai penelitian yang mengkaji pengalaman pengguna dalam bermain *game*. [38]. Penelitian ini menerapkan metode kuantitatif deskriptif dalam mengevaluasi pengalaman pengguna (*User Experience*) dari *game* "Super Qui".

10. Beta Playtesting

Beta *testing* merupakan metode pengujian yang dilakukan pada tahap akhir pengembangan untuk menguji *game* dalam kondisi penggunaan nyata. Pengujian ini menggunakan instrumen kuesioner untuk memperoleh umpan balik sebagai metode pengumpulan data untuk mengevaluasi aspek fungsionalitas, performa, dan pengalaman pengguna (*user experience*) terhadap *game* yang dikembangkan[39]. *Beta playtesting* adalah tahap pengujian *game* yang mengikutsertakan pemain untuk menguji permainan yang telah dikembangkan, sehingga pengembang dapat memperoleh umpan balik mengenai desain permainan dan pengalaman bermain yang dirasakan pemain. Dalam konteks pengembangan *game*, aktivitas ini dilakukan melalui metode *playtesting* yang berfungsi sebagai pendekatan penelitian pengalaman pengguna untuk mengevaluasi keputusan desain berdasarkan tanggapan pemain dengan tujuan meningkatkan kualitas *gameplay experience*[40].

Pada tahap ini menggunakan metode *closed beta playtesting*, metode ini adalah metode pengujian yang dilakukan dengan melibatkan sejumlah pengguna yang dipilih untuk mencoba produk dan memberikan umpan balik sebelum *game* dirilis secara luas[41]. Untuk memahami pengalaman pemain selama masa pengujian, evaluasi bisa dilakukan dengan menggunakan instrumen berupa

Game Experience Questionnaire (GEQ). GEQ merupakan kuesioner inti yang berfungsi untuk mengukur pikiran dan perasaan pemain selama bermain *game*[42].

11. Uji Statistik Signifikansi

Uji statistik signifikansi merupakan teknik analisis inferensial yang digunakan untuk menguji hipotesis penelitian dan menentukan perbedaan atau pengaruh yang diperoleh dari data yang ada secara signifikan atau hanya karena faktor kebetulan. Dilakukan uji signifikansi dengan menguji normalitas untuk mengetahui data berdistribusi normal dengan menggunakan uji *Shapiro-Wilk*, untuk data yang tidak normal maka menggunakan uji *Wilcoxon Signed-Rank Test*. Pengujian ini dilakukan menggunakan *software* JASP (*jeffreys's Amazing Statistics Program*) yang menyediakan berbagai analisis statistik melalui antarmuka yang mudah digunakan sehingga mempermudah proses pengolahan dan interpretasi data penelitian[43].