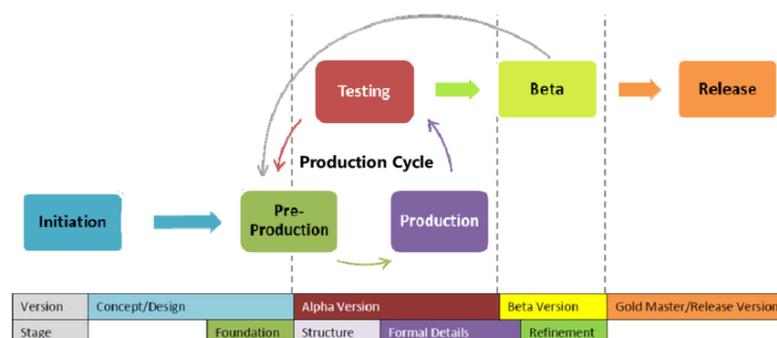


## BAB III METODE PENELITIAN

### 3.1 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian *Research and Development* (R&D). Berdasarkan pendapat Sukmadinata (dalam Rosvita & Anugraheni, 2021), R&D dipilih karena sejalan dengan tujuan peneliti yaitu menghasilkan produk baru atau menyempurnakan produk yang telah ada lalu menguji produk tersebut. Tahapan dari jenis penelitian ini menurut Borg & Gall (dalam Gustiani, 2019) meliputi *Research and information collecting, Planning, Develop preliminary form of product, Preliminary field testing, Main product revision, Main field testing, Operational product revision, Operational field testing, Final product revision, dan Dissemination and implementation*. Penelitian ini mengacu pada model yang diadaptasi dari Sarwono (2022), yang menyederhanakan tahapan Borg & Gall menjadi tiga tahap utama, yaitu tahap studi pendahuluan, tahap penyusunan draft produk, dan tahap pengembangan produk.

Pada penyusunan draft dan pengembangan produk, penelitian ini menggunakan pendekatan *Game Development Life Cycle* (GDLC) sebagai metode pengembangan *game*. Metode ini merupakan metode pengembangan *game* yang menggunakan pendekatan interaktif sehingga memungkinkan peneliti melakukan perubahan keperluan dan dinamika proyek secara tepat (Wahyu dkk., 2025). Metode ini juga fleksibel terhadap perubahan dan dibuat agar dapat mengatasi kriteria kualitas di setiap tahapan guna mempertahankan kualitas produk akhir (Ramadan & Widyani, 2013).

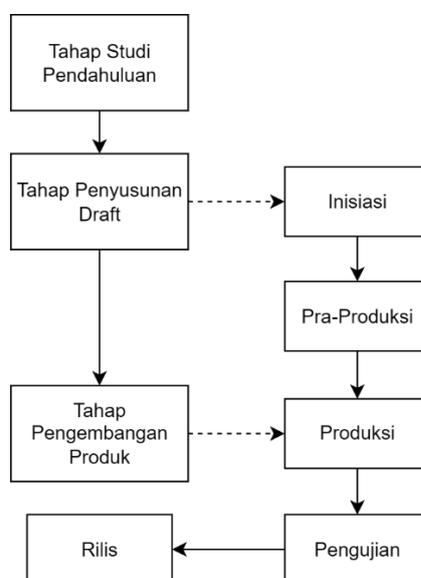


Gambar 3.1 Tahapan GDLC  
(Sumber: Ramadhan & Widyani, 2013)

Tahapan yang dalam metode pengembangan *Game Development Life Cycle* (GDLC) terdiri dari Inisiasi, Pra-Produksi, Produksi, Pengujian, dan Rilis. Mengutip dari artikel Ramadan & Widyani (2013), setiap tahapannya akan dijelaskan secara rinci pada pembahasan selanjutnya.

### 3.2 Prosedur Penelitian

Penelitian ini menghasilkan produk utama, yaitu *game* berbasis *rhythm* bernama “FingerXibility” yang berfokus pada melatih keterampilan jari sebagai dasar bermain piano. Dengan menggunakan metode penelitian *Research and Development* (R&D) dan menerapkan metode pengembangan *Game Development Life Cycle* (GDLC) sebagai penyusunan draft dan pengembangan produk, prosedur dari penelitian ini adalah sebagai berikut.



Gambar 3.2 Prosedur Penelitian

Setiap tahapan dalam prosedur penelitian tersebut meliputi tahap studi pendahuluan, tahap penyusunan produk (inisiasi dan pra-produksi), dan tahap pengembangan produk (produksi, *testing*, dan rilis) akan dijelaskan dengan rinci pada pembahasan selanjutnya.

#### 3.2.1 Studi Pendahuluan

Tahap pertama yang dilakukan dalam penelitian ini yaitu studi pendahuluan, dimana peneliti melakukan analisis masalah mencakup pengumpulan dan analisis informasi, pendefinisian masalah (Gustiani, 2019) lalu melakukan kajian terhadap

konsep atau teori terkait (Sarwono, 2022). Proses tersebut dapat dilakukan melalui studi literatur yang berkaitan dengan permasalahan yang dikaji (Haryati, 2012) ke berbagai sumber seperti buku, jurnal ilmiah, artikel daring, pendapat para ahli, ataupun melalui penelitian sebelumnya. Hal ini memberikan gambaran tentang hasil penelitian sebelumnya yang nantinya digunakan sebagai bahan referensi (Okpatrioka, 2023) serta memperoleh pemahaman yang mendalam mengenai beberapa aspek penting yang menjadi dasar dalam merancang dan pengembangan produk.

Fokus utama dalam penelitian ini adalah mengkaji perkembangan *game* khususnya pada *genre rhythm game* dan penggunaannya sebagai media edukasi. Selain itu, peneliti juga mendalami kajian mengenai keterampilan dasar bermain piano, dengan penekanan pada *fingering* atau teknik penjarian sebagai salah satu keterampilan teknis paling mendasar dalam permainan piano. *Fingering* dipilih sebagai fokus karena memiliki peran krusial dalam membentuk efisiensi, fleksibilitas, dan koordinasi jari, terutama bagi pemula. Meskipun penelitian ini belum menguji secara langsung tingkat efektivitas edukatifnya, informasi dari studi pendahuluan menjadi panduan utama dalam menentukan arah pengembangan konten dan fitur *game* agar selaras dengan tujuan pembelajaran *fingering* pada piano.

### 3.2.2 Penyusunan Draft Produk

Tahap penyusunan draft produk merupakan tahap rancangan awal dari *game* edukasi yang disusun berdasarkan hasil studi pendahuluan yang telah dilakukan sebelumnya (Sarwono, 2022). Penyusunan ini bertujuan untuk memformulasikan ide awal menjadi bentuk yang lebih konkret melalui perencanaan konsep, desain sistem, dan pembuatan *prototype game*. Selain itu, tahap ini juga memastikan agar setiap komponen yang dirancang memiliki keterkaitan yang jelas dengan tujuan utama penelitian dan penggunaan produk (Okpatrioka, 2023), yaitu mengembangkan *game* edukasi yang dapat melatih keterampilan *fingering* sebagai dasar dalam bermain piano.

Pada penelitian ini, proses penyusunan draft produk dilakukan dengan mengacu pada metode *Game Development Life Cycle (GDLC)*. Berikut tahapan

dalam penyusunan draft produk.

### 3.2.2.1 Inisiasi (*Initiation*)

Pada tahap inisiasi, peneliti mulai merumuskan ide dan konsep *game*, menentukan tujuan pembelajaran, memilih teknologi yang akan digunakan, serta mengidentifikasi kebutuhan perangkat keras dan perangkat lunak yang mendukung (Junaidi, 2024). Tahap ini menghasilkan sebuah konsep kasar dari *game* yang akan dikembangkan beserta deskripsi sederhananya (Ramadan & Widyani, 2013).

### 3.2.2.2 Pra-Produksi (*Pre-Production*)

Pada tahap pra-produksi, dilakukan proses perancangan dari konsep *game* yang telah dikembangkan menjadi bentuk visual dan teknis. Tahap ini berfokus pada perancangan sistem dan pembuatan *prototype* (Ramadan & Widyani, 2013), meliputi pembuatan *flowchart*, *use case diagram*, *activity diagram*, dan desain antarmuka, mulai dari tampilan menu, mode permainan, hingga tampilan hasil.

#### 1) *Flowchart* (Diagram Alir)

*Flowchart* atau bisa disebut juga dengan diagram alir adalah alat pemetaan sederhana yang digunakan untuk menjelaskan alur kerja dari sistem yang akan dibangun (Ridlo, 2017). *Flowchart* menjadi pondasi penting dalam pemahaman proses logika pemrograman karena membantu pengembang dalam merancang, menganalisis, dan memahami logika dan urutan proses sistem (F. I. Kusumawati dkk., 2024). *Flowchart* terdiri dari berbagai macam simbol yang memiliki fungsinya masing-masing lalu dihubungkan dengan panah, proses membangunnya disebut dengan *Flowcharting* (Khairunnisa dkk., 2023).

*Flowchart* merupakan serangkaian operasi sederhana yang melibatkan penerimaan input (Chaudhuri, 2005), sehingga dapat dimanfaatkan dalam menjelaskan suatu proses dari jenis pekerjaan apapun. Kelebihan lainnya selain dapat digunakan oleh jenis pekerjaan apapun, diantaranya (Khairunnisa dkk., 2023):

- a) Cara terbaik untuk mengkomunikasikan logika dari sistem kepada berbagai pihak.
- b) Proses menganalisis masalah lebih efektif.
- c) Dapat digunakan sebagai dokumentasi program untuk berbagai tujuan.

- d) Berperan sebagai panduan atau *blueprint* selama fase analisis sistem dan pengembangan program sehingga menjadi efisien.
- e) Membantu dalam proses *debugging* dan pemeliharaan program.  
Saat proses *flowcharting*, ada lima aturan yang harus diperhatikan dan diikuti (Chaudhuri, 2005). Lima aturan tersebut diantaranya:
  - a) Hanya simbol standar yang boleh digunakan (Lampiran 10).
  - b) Logika program harus menggambarkan alur dari atas ke bawah dan dari kiri ke kanan.
  - c) Setiap simbol yang digunakan hanya boleh berisi satu titik masuk dan satu titik keluar, kecuali simbol keputusan.
  - d) Operasi yang ditampilkan harus ditulis tanpa bergantung pada bahasa pemrograman tertentu.
  - e) Semua cabang keputusan harus diberi label dengan jelas.

## 2) *Use Case Diagram*

*Use case diagram* merupakan salah satu diagram UML yang didefinisikan sebagai model hasil analisis perancangan sistem yang bertujuan untuk mendeskripsikan kebutuhan sistem (Ramdany, 2024). *Use case diagram* mengidentifikasi fungsionalitas yang dimiliki sistem serta interaksi dan keterhubungan antara pengguna (sebagai aktor) dengan fungsionalitas sistem (Arifin & Hendro HS., 2017) dalam perspektif pengguna (Setiyani, 2021). Dengan kata lain, *use case diagram* digunakan sebagai gambaran fungsi yang tersedia dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Khaerudin dkk., 2021).

Ada beberapa panduan yang dapat diperhatikan agar *use case diagram* yang dibuat lebih efisien (Mule dkk., 2015). Beberapa panduan tersebut, diantaranya:

- a) Nama *use case* harus dapat mengidentifikasi fungsionalitas yang dijalankan.
- b) Berikan nama yang sesuai untuk aktor.
- c) Tunjukkan hubungan dan ketergantungan dengan jelas.
- d) Jangan masukkan semua jenis hubungan. Tujuannya hanya mengidentifikasi kebutuhan.
- e) Gunakan catatan kapan pun untuk mengklarifikasi beberapa poin penting.

### 3) **Activity Diagram**

*Activity diagram* termasuk salah satu diagram UML yang digunakan untuk menggambarkan alur aktivitas (*workflow*) dalam suatu sistem (Arianti dkk., 2022). *Activity diagram* adalah *flowchart* yang diperluas (Arifin & Hendro HS., 2017) serta berisi operasi-operasi dan aktivitas-aktivitas yang ada pada *use case diagram* (Arifin & Hendro HS., 2017).

Dikutip dari situs Onero (2025), ada beberapa langkah yang dapat diikuti agar *activity diagram* yang dihasilkan rapi dan informatif. Langkah-langkah tersebut diantaranya:

- a) Tentukan titik awal diagram (*Start Point*) dengan *initial node*.
- b) Berikan aksi untuk setiap langkah utama yang ada pada *use case diagram*.
- c) Hubungkan atau transisikan antar aksi dengan menggunakan panah (*directional flow*).
- d) Jika terdapat proses bersyarat atau pilihan, gunakan *desicion node*.
- e) Jika terdapat proses paralel, manfaatkan simbol *fork & join*. *Fork* digunakan untuk memecah satu *flow* menjadi beberapa aksi paralel, sedangkan *Join* digunakan untuk menggabungkan kembali *flow* paralel.
- f) Jika perlu, gunakan *Swimlane* untuk memisahkan tugas berdasarkan aktor.
- g) Jika proses telah selesai, tandai menggunakan *final node*.
- h) Cek kembali agar tidak terjadi kesalahan dalam penggunaan simbol (Lampiran 12).

### 4) **Desain Antarmuka**

Menurut Roth (Wibawanto & Nugrahani, 2018), desain antarmuka adalah seperangkat elemen yang digunakan untuk memanipulasi objek digital. Dalam proses pengembangan perangkat lunak, tahap pengembangan desain antarmuka penting dilakukan sebagai produk multimedia, aspek visual, dan menjadi aspek pertama yang dilihat oleh pengguna (Wibawanto & Nugrahani, 2018). Dikutip dari situs Telkom University yang ditulis oleh Kifansa (2025) ada beberapa prinsip dasar yang harus diperhatikan agar dapat menciptakan desain antarmuka yang efektif. Beberapa prinsip tersebut, diantaranya:

- a) Konsisten dalam menggunakan elemen yang seragam untuk menghindari

kebingungan.

- b) Pastikan antarmuka tetap sederhana dan fokus pada hal-hal penting.
- c) Berikan umpan balik (*feedback*) yang jelas saat pengguna melakukan sebuah tindakan, seperti suara klik tombol atau pesan konfirmasi.
- d) Pastikan informasi penting mudah ditemukan dan tidak tersembunyi.
- e) Pastikan desain yang dikembangkan dapat digunakan oleh semua orang, termasuk pengguna dengan keterbatasan.

Keempat alat bantu pemodelan sistem yang telah dibahas, yaitu *flowchart*, *use case diagram*, *activity diagram*, dan desain antarmuka, memiliki peran yang saling melengkapi dalam proses pengembangan perangkat lunak. *Flowchart* memberikan gambaran logis dan urutan proses secara menyeluruh, yang menjadi fondasi awal dalam memahami alur kerja sistem. Kemudian *use case diagram* mengidentifikasi fungsionalitas sistem dari sudut pandang pengguna, sehingga dapat dipastikan bahwa sistem memenuhi kebutuhan aktor yang terlibat. Aktivitas yang tercantum dalam *use case* tersebut lalu dijabarkan lebih rinci melalui *activity diagram* untuk menggambarkan aliran kerja sistem secara dinamis dan mendalam. Keseluruhan proses ini ditutup dengan perancangan desain antarmuka yang tidak hanya berfokus pada aspek visual, tetapi juga memperhatikan prinsip kemudahan penggunaan, konsistensi, serta aksesibilitas bagi seluruh pengguna. Oleh karena itu, pemahaman dan penerapan keempatnya menjadi langkah strategis dalam menciptakan sistem yang efektif, efisien, serta berorientasi pada kebutuhan pengguna.

### **3.2.3 Pengembangan Produk**

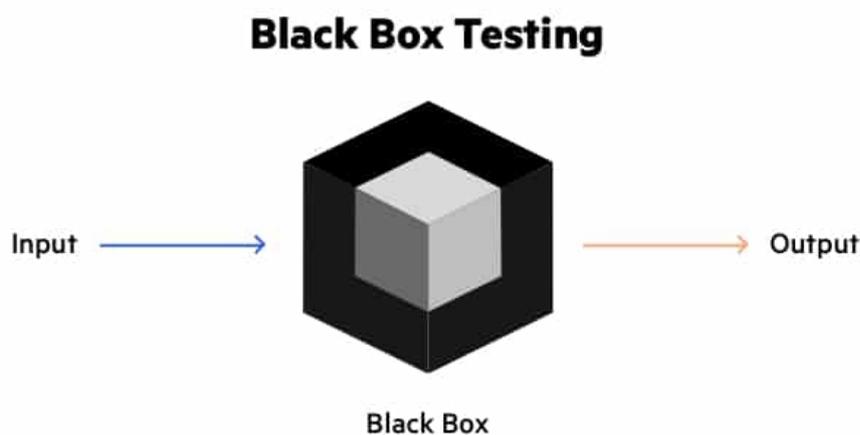
#### **3.2.3.1 Produksi (*Production*)**

Setelah proses pembuatan desain dan *prototype* selesai, hasilnya diimplementasikan pada tahap produksi menggunakan *game engine* yang telah ditentukan pada tahap inisiasi. Tahap ini berputar antara pembuatan aset, pembuatan *source code*, dan mengintegrasikan kedua elemen tersebut dengan perubahan kecil yang diizinkan demi penyempurnaan permainan (Ramadan & Widayani, 2013). Tahapan yang dilakukan dalam penelitian ini mencakup pengumpulan dan pembuatan aset visual seperti ikon, karakter, tombol, dan latar;

penyusunan dan integrasi elemen musik seperti lagu, efek suara, dan *metronome*; pengkodean fitur interaktif seperti tombol, skor, pengaturan tingkat kesulitan, dan sistem level; serta penerapan *gameplay* yang terdiri dari mode "Tangga Nada dan Akor" dan "Musik".

### 3.2.3.2 Pengujian (*Testing*)

Pada tahap pengujian, baik pengujian *Alpha* maupun pengujian *Beta*, metode *BlackBox Testing* digunakan untuk menguji fungsionalitas dari *game* yang dikembangkan. Dikutip dari situs Imperva, *BlackBox* adalah “metode pengujian sistem tanpa mempedulikan bagaimana sistem dikodekan atau diarsitektur”, karena hanya melibatkan proses input yang dilakukan pengguna dan output yang diberikan oleh sistem. Tujuan utama dari pengujian *blackbox* adalah menganalisis kerja sistem seperti antarmuka, kinerja, fungsi fitur, dan lainnya untuk menghindari kesalahan (Parlika dkk., 2020).



Gambar 3.3 Proses Pengujian dengan Metode *BlackBox*

(Sumber: [imperva.com](http://imperva.com))

Penggunaan metode *blackbox* dalam pengujian perangkat lunak memiliki sisi positif dan negatif. Sisi positifnya, yaitu tes dilakukan dari sudut pandang pengguna, sehingga tidak memerlukan akses ke sumber kode, serta prosesnya cepat dan spesifik. Karena itu, selama proses pengujian, penguji tidak diharuskan memiliki pengetahuan terkait pemrograman dan arsitektur perangkat lunak

(Dhaifullah dkk., 2022). Sedangkan sisi negatifnya yaitu tidak semua produk perangkat lunak dapat diuji menggunakan metode *blackbox* dan penyebab kegagalan sulit ditemukan karena cakupan kode akses terbatas (Parlika dkk., 2020).

### 5) Pengujian *Alpha*

Tahap pengujian *Alpha* dilakukan oleh pengembang guna mengevaluasi kinerja dan proses sistem berjalan, yang bertujuan untuk memastikan dan mengevaluasi sistem *game* berjalan secara fungsional tanpa adanya potensi *bug* atau *error* (Junaidi, 2024) sebelum dilanjutkan pada pengujian selanjutnya yang melibatkan pengguna secara langsung.

### 6) Pengujian *Beta*

Tahap pengujian *Beta* dilakukan oleh pengguna guna menguji fungsionalitas dan non-fungsionalitas dari *game* yang telah dikembangkan sebelumnya. Pengujian non-fungsional bertujuan untuk memeriksa aspek tambahan diluar fitur dan fungsionalitas sistem. Contohnya kemudahan dalam menggunakan *game*, kompatibel dengan perangkat, ukuran layar, atau sistem operasi yang relevan. Tujuan lainnya yaitu untuk mendapatkan umpan balik secara langsung dan mendalam terkait *game* yang telah dikembangkan melalui penerapan metode *one-on-one validation* dimana pengembang berinteraksi dengan pengguna secara individu. Metode tersebut memberikan umpan balik berupa pengalaman, kebutuhan, serta masukan dan harapan pengguna untuk pengembangan selanjutnya (Junaidi, 2024).

#### 3.2.4 Rilis (*Release*)

Setelah *game* berhasil mencapai tahap pengujian *Beta* dan tidak ditemukan *bug* ataupun *error* pada sistem *game* selama pengujian berlangsung, maka dapat melakukan tahap selanjutnya, yaitu perilsan *game* (Widyananda dkk., 2023). Perilsan dapat dilakukan melalui *platform* khusus pengembangan *game* Perilsan dapat dilakukan pada *platform* distributor digital khusus pengembangan dan perilsan *game* seperti Itch.io dan Steam.

### 3.3 Subjek Penelitian

Subjek dari penelitian ini yaitu pengembangan *game* bernama “FingerXibility”, sebuah *game* edukasi berbasis *rhythm* yang dirancang untuk

melatih keterampilan *fingering* sebagai keterampilan dasar bermain piano. Subjek pengujian *beta* akan melibatkan 2 mahasiswa. Kedua mahasiswa tersebut karena dapat mewakili pengguna, dengan 1 mahasiswa adalah seorang *gamer* serta menyukai dan sudah memainkan berbagai macam *rhythm game* yang ada di internet; dan 1 mahasiswa lainnya tidak atau jarang bermain *game*, baik *rhythm game* maupun *game* pada *genre* lainnya. Dengan begitu, proses pengujian tidak hanya sekedar menguji *game* yang berhasil dikembangkan, namun juga mewakili target pengguna baik yang sudah pernah menggunakan aplikasi sejenis maupun yang belum pernah.