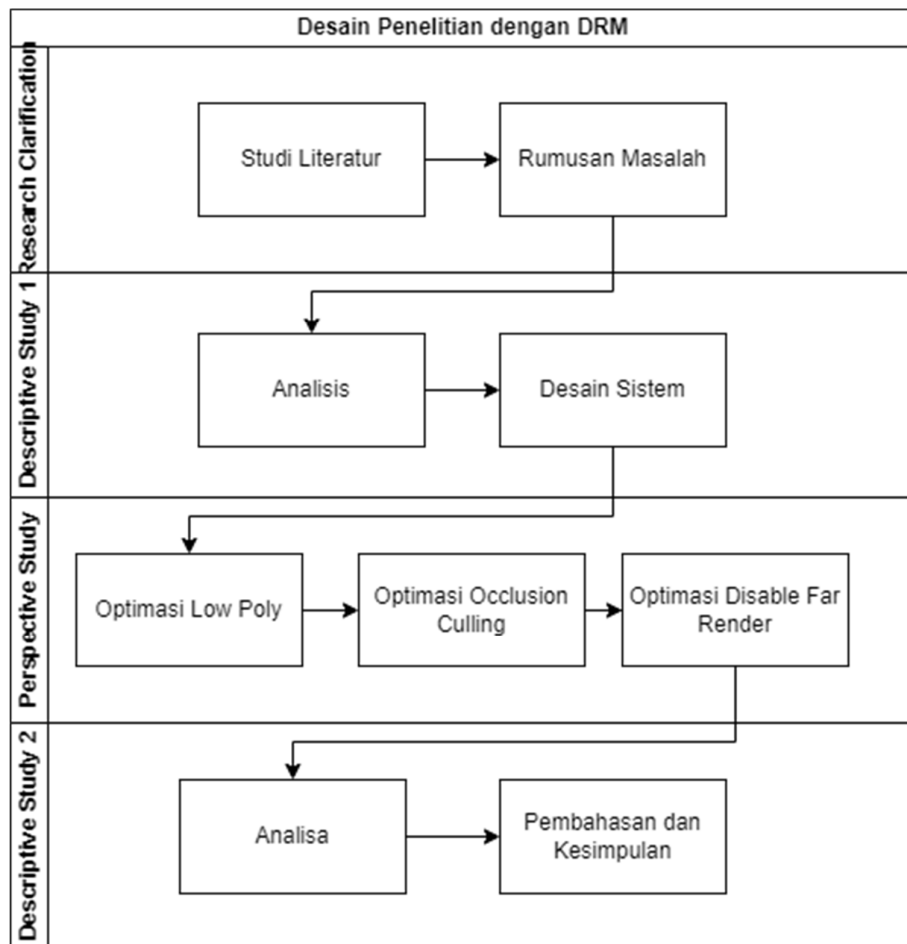


## BAB III METODOLOGI PENELITIAN

### 3.1. Desain Penelitian

Desain penelitian yang digunakan pada penelitian ini adalah penelitian dengan metode *Design Research Methodology* (DRM). DRM adalah metode penelitian dengan langkah sistematis yang biasanya digunakan pada penelitian perancangan, yang mana perancangan yang dimaksud adalah pembangunan aplikasi dan validasi secara sistematis (Blessing dan Chakrabarti, 2009). Berikut adalah gambaran dari susunan langkah metode DRM tersebut.



Gambar 3.1 Gambaran metode penelitian

Berikut ini adalah penjelasan dari gambar 3.1 terhadap metode penelitian yang dilakukan:

### **3.1.1. Research Clarification**

Pada tahap ini, tema yang akan diteliti ditentukan. Topik yang diambil kali ini adalah tentang optimasi pada sistem gim *Virtual Biotope* dengan menggunakan metode *Basic Optimization*. Peneliti melakukan studi literatur dan uji performa awal terhadap apa saja yang dirasa penting untuk dipelajari demi persiapan pengembangan dan optimasi sistem, seperti metrik apa saja yang menyebabkan masalah performa terhadap gim ini. Setelah semua data yang diperlukan didapat, maka perumusan masalah akan dimulai. Pada tahap perumusan masalah, peneliti akan merumuskan setiap masalah yang ada, untuk dicari solusinya pada penelitian.

### **3.1.2. Descriptive Study 1**

Setelah semua masalah dirumuskan, maka semua data yang ada akan dianalisis. Data-data dari penelitian yang telah ada, dokumentasi dari pihak pengembangan Unity, dan segala materi dari jurnal-jurnal yang ditemukan akan diolah. Dari data yang telah dianalisis tersebut sistem akan dirancang ulang dengan optimasi yang diperlukan. Setelah analisa selesai, maka masuklah ke tahap desain sistem. Analisa optimasi akan dirancang pada tahap ini.

### **3.1.3. Perspective Study**

Pada tahap ini, sistem akan mulai dilakukan optimasi. Sistem akan di optimasi sesuai dengan rancangan yang telah dibuat melalui analisa data dan studi literatur. Pada tahap ini, modifikasi akan dilakukan kepada objek-objek yang ada di dalam proyek Unity, disesuaikan dengan kebutuhan dan spesifikasi yang ditargetkan. Metode optimasi *basic optimization* yang dilakukan pada gim adalah penerapan objek *low poly*, *occlusion culling*, dan *disable far render*. Diharapkan, pada tahap ini telah tercipta sebuah sistem yang dapat memenuhi kebutuhan *frame rate* yang ditargetkan.

### **3.1.4. Descriptive Study 2**

Setelah optimasi sistem selesai dibuat, maka tingkat keberhasilan optimasi tersebut akan diuji dengan membandingkan hasil pengembangan dengan versi sebelumnya. Analisa yang ditentukan pada tahap ini dilakukan demi menguji seberapa sukses sistem tersebut telah berjalan dengan baik. Setelah pengujian dilakukan, peneliti harus dapat menyimpulkan hasil pengujian tersebut, seperti apakah kelebihan optimasi yang telah dibuat, kelemahan dari metode optimasi

yang digunakan, dan masih banyak lainnya. Seluruh hasil akan dijabarkan dengan cara sejas mungkin.

### 3.2. Alat dan Bahan

Berikut ini adalah alat yang akan digunakan untuk membantu penelitian kali ini:

*Software* : Unity versi 2021.3.16f1, Unity Profiler, Blender, Visual Studio 2019

*Hardware*: Mouse, laptop dengan spesifikasi:

- *Processor* : AMD Ryzen 5900
- *GPU* : NVidia GeForce RTX 3060 Laptop GPU
- *VRAM* : 6GB
- *RAM* : 16GB (8GB + 8GB SODIMM 3200MHz)
- *Storage* : 516GB + 1TB
- *Monitor* : 1920x1080 144hz

Dan juga *Android Phone* Xiaomi Poco F4 dengan spesifikasi:

- *OS* : Android 12
- *Chipset* : Qualcomm SM8250-AC Snapdragon 870 5G (7 nm)
- *CPU* : Octa-core (1x3.2 GHz Kryo 585 & 3x2.42 GHz Kryo 585 & 4x1.80 GHz Kryo 585)
- *RAM* : 8GB
- *GPU* : Adreno 650
- *Display* : 1080 x 2400 pixel, 20:9 (~395 ppi density)
- *Storage* : 256GB

Sementara itu, bahan-bahan yang digunakan adalah dokumentasi Unity, jurnal, dan penelitian-penelitian sebelumnya sebagai acuan pengembangan dan optimasi sistem.

### 3.3. Skenario Pengujian

Proses pengujian akan melibatkan evaluasi yang dilakukan dalam beberapa tahap untuk memastikan kinerja yang optimal. Tahap awal melibatkan pengujian di dalam Unity, dengan menggunakan *Unity Profiler* dan *Profile Analyzer*. Alat-alat ini memberikan wawasan berharga tentang kinerja gim, memungkinkan

identifikasi area yang membutuhkan optimasi dengan data grafik dan *frame time* yang diberikan.

Selama proses optimasi, setiap perbaikan individu akan diuji dan diukur dengan teliti, untuk melihat kemajuan yang dicapai. Analisis ini akan disajikan menggunakan metrik *frame time*, yang memungkinkan pemeriksaan terperinci terhadap efisiensi FPS pada gim. Dengan memeriksa data ini dapat diperoleh pemahaman yang lebih mendalam tentang dampak setiap optimasi terhadap kinerja secara keseluruhan.

Setelah menyelesaikan fase pengujian di Unity, fokus akan beralih ke evaluasi kinerja dalam dunia nyata pada perangkat Android. Metrik utama dalam penilaian ini adalah perbandingan *frame time* dan FPS yang dicapai antara versi permainan yang telah di optimasi dan yang belum di optimasi. Pengukuran ini memberikan indikasi tentang kehalusan permainan pada perangkat gawai.

Semua pengujian akan dibandingkan melalui uji perbandingan. Uji perbandingan yang akan digunakan adalah dengan menggunakan rumus persentase perbedaan. Persentase perbedaan dapat dihitung menggunakan rumus di mana perbedaan antara hasil awal dan akhir dibagi dengan rata-rata nilai dari kedua nilai tersebut. Hasil dari perhitungan tersebut kemudian akan dikalikan dengan angka seratus untuk mendapatkan hasil persentase perbedaan terhadap kedua nilai tersebut. Rumus ini memungkinkan peneliti untuk mendapatkan nilai yang pasti walaupun kedua variabel tersebut ditukar, apabila dibandingkan dengan menggunakan rumus persentase kenaikan ataupun penurunan (Cole dan Altman, 2017). Berikut ini adalah rumus yang akan digunakan:

$$\text{Persen Perbedaan} = \left| 100 \times \frac{(x - y)}{\frac{1}{2}(x + y)} \right| \quad (7)$$

Di mana:

x = nilai sebelum, dan

y = nilai sesudah

Pengujian kali ini akan memasukkan nilai sebelum ke dalam variabel x, dan nilai setelah ke dalam variabel y, dikarenakan tujuan dari optimasi ini adalah untuk mengurangi *frame time* dan meningkatkan FPS pada sistem yang diuji. Diharapkan bahwa dengan melakukan optimasi, waktu yang dibutuhkan untuk

menampilkan satu *frame* pada perangkat tersebut dapat berkurang, sementara FPS dapat meningkat. Perlu diingat bahwa dalam analisis perbandingan ini, nilai-nilai yang lebih rendah pada *frame time* menunjukkan hasil yang lebih baik karena menandakan waktu yang lebih singkat untuk menampilkan satu *frame*, dan dapat menyebabkan peningkatan FPS secara langsung.

Walaupun angka tersebut ditentukan seperti itu, hasil angka yang telah dihitung dari pengujian ini tidak akan berubah walaupun kita menukar kedua variabel tersebut, berdasarkan pendapat Cole & Atman sebelumnya. Sebagai contoh, mari kita misalkan pada tahap sebelum optimasi, gim dapat memproses suatu *frame* dengan kecepatan 20 mili detik, sedangkan setelah optimasi gim dapat menjalankan *frame* tersebut dengan kecepatan 16 mili detik. Dengan menggunakan rumus yang telah disebutkan, dapat dikatakan bahwa variabel  $x$  memiliki nilai berbeda sebesar 22.22% daripada variabel  $y$ . Hasil tersebut dapat dilihat dari perhitungan berikut:

$$\left| 100 \times \frac{(16 - 20)}{\frac{1}{2}(16 + 20)} \right| = 22.22\% \quad (8)$$

Sedangkan, apabila kita tukar kedua angka tersebut, maka hasilnya akan menjadi positif, tanpa mengubah hasil dari perhitungan tersebut. Dapat dijelaskan bahwa variabel  $x$  memiliki nilai berbeda sebesar 22.22% daripada variabel  $y$ . Hasil tersebut dapat dilihat dari perhitungan berikut:

$$\left| 100 \times \frac{(20 - 16)}{\frac{1}{2}(20 + 16)} \right| = 22.22\% \quad (9)$$

Proses pengujian ini bertujuan untuk memastikan bahwa setiap aspek permainan telah melalui pengujian dan optimasi yang cermat. Dengan menggunakan *Unity Profiler* dan *Profile Analyzer*, penulis akan dapat memperoleh pemahaman yang mendalam tentang titik lemah kinerja permainan dan area yang dapat diperbaiki lebih lanjut. Kemudian penggunaan perangkat Android menambah pengujian tingkat lanjut, memastikan bahwa permainan berkinerja optimal di platform target sebenarnya. Pendekatan ini diharapkan dapat menciptakan produk akhir yang dapat mencapai tingkat efisiensi yang tinggi.