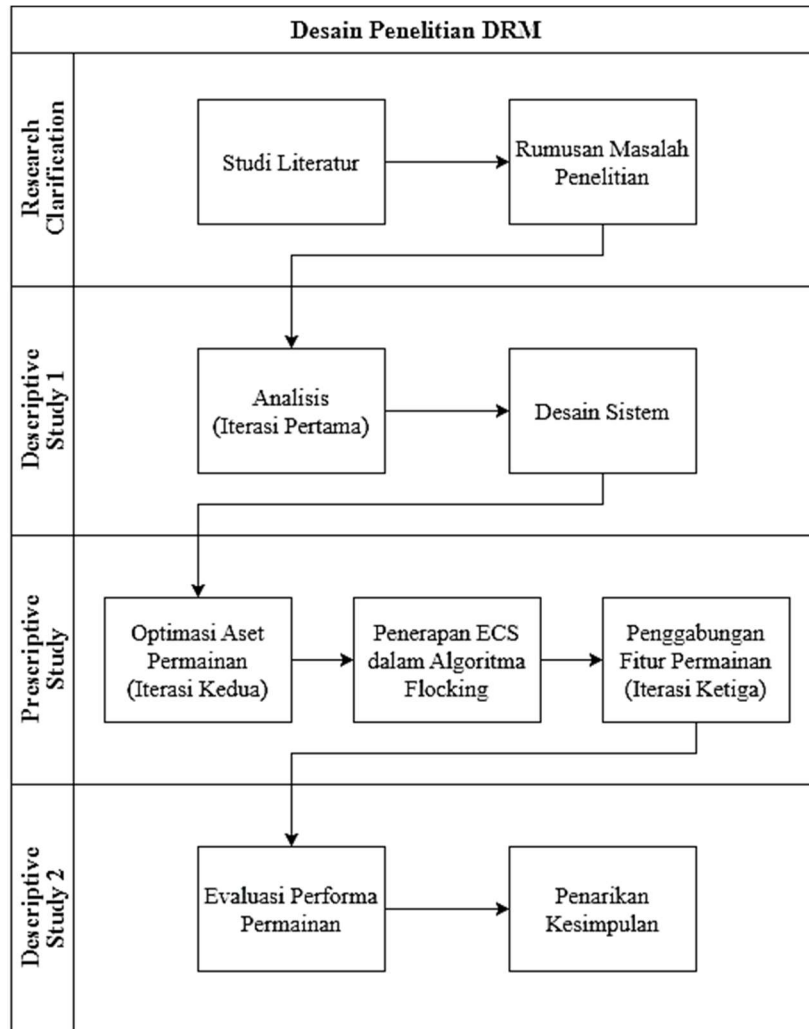


## BAB III METODOLOGI PENELITIAN

### 3.1. Desain Penelitian

Penelitian ini menggunakan metode *Design Research Methodology* (DRM). DRM adalah metode penelitian dengan fokus pendekatan pada perancangan, pengembangan, dan pengujian solusi desain. Perancangan yang dimaksud dalam konteks pengembangan aplikasi dan validasi secara sistematis (Blessing dan Chakrabarti, 2009). Alur penelitian dengan metode DRM terlihat pada gambar berikut.



Gambar 3.1 Alur Metode Penelitian

Penjelasan setiap tahap dari Gambar 3.1 terkait metode penelitian ini adalah sebagai berikut:

### **3.1.1. *Research Clarification***

Pada tahapan ini, topik yang diambil adalah tentang optimasi performa. Optimasi yang dimaksud terhadap sistem gim Virtual Biotope, melalui penerapan pendekatan *data-oriented design* dengan menggunakan Unity DOTS. Dari tahapan ini, dilakukan studi literatur dan pengujian performa gim untuk mempersiapkan pengembangan dan optimasi sistem lebih lanjut. Setelah menemukan data yang diperlukan seperti metrik dan bagian yang dapat menyebabkan masalah performa gim, maka peneliti mulai merumuskan masalah sebagai fokus untuk pencarian solusi pada penelitian.

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

Setelah dibuatnya rumusan masalah, maka data yang ada akan dimulai analisis. Data tersebut dapat berupa data dari penelitian terkait, dokumentasi dari Unity dan Unity DOTS, serta segala artikel dan jurnal relevan yang ditemukan. Sistem gim akan dirancang kembali dan dikembangkan sesuai kebutuhan optimasi, dan setelah dilakukan analisis menuju tahapan desain sistem.

### **3.1.3. *Prescriptive Study***

Pada tahap ini, proses optimasi mulai dilakukan. Sistem gim di optimasi sesuai dengan data dan rancangan yang telah dilakukan pada tahapan-tahapan sebelumnya. Dimulai dari optimasi aset 3D dengan menganalisis dan memodifikasi aset sesuai dengan target dan kebutuhan menurut dokumentasi mengenai optimasi. Menerapkan *data-oriented design* terhadap *flockbox* melalui Unity DOTS dan memastikan implementasi tersebut dapat berfungsi dengan baik mempertahankan perilaku *flocking* sesuai kebutuhan. Setelah diterapkan, diharapkan pada tahap ini telah menggabungkan hasil implementasi baik dari pemodelan ulang aset dan penerapan Unity DOTS dapat memenuhi kebutuhan optimasi untuk gim dapat berjalan lebih optimal pada perangkat yang digunakan dari segi penggunaan sumber daya yang dimiliki berdasarkan spesifikasi perangkat tersebut.

### 3.1.4. *Descriptive Study 2*

Pada tahapan ini, optimasi yang telah dibuat akan dilakukan uji perbandingan terhadap hasil iterasi sebelumnya. Evaluasi dilakukan untuk menguji tingkat keberhasilan optimasi ketika gim dijalankan. Setelah evaluasi, maka dilakukan penarikan kesimpulan, mulai dari kelebihan dan kelemahan metode optimasi yang telah diterapkan, dan rekomendasi berupa catatan untuk peneliti selanjutnya.

## 3.2. Alat dan Bahan

Alat yang relevan digunakan untuk melaksanakan proses penelitian ini dapat dilihat pada Tabel 3.1 berikut:

Tabel 3.1  
*Hardware dan Software yang digunakan dalam penelitian*

<i>Hardware</i>	
Laptop	Processor: Intel(R) Core(TM) i7-13620H CPU @ (Cache 24 M, hingga 4,90GHz) GPU: NVIDIA GeForce RTX 4050 Laptop GPU 6GB GDDR6-VRAM RAM: 8GB DDR5-4800 SO-DIMM *2 Storage: 512GB SSD Monitor: 15.6", 1920x1080 60-144Hz
<i>Software</i>	
Unity Game Engine	Unity 2021.3.16f1 Unity 2022.3.31f1 Unity Editor Unity Profiler Profile Analyzer
Blender	Blender 4.1
Microsoft Visual Studio	Visual Studio 2022 Visual Studio Code
Github	Github Desktop

Sementara untuk bahan-bahan yang digunakan pada penelitian ini adalah jurnal, penelitian relevan sebelumnya, dan dokumentasi *Unity Engine* serta *Unity Data-Oriented Technology Stack* bersama dengan informasi lainnya yang diambil dari contoh repositorinya sebagai acuan dalam optimasi sistem.

### 3.3. Instrumen Penelitian

Evaluasi performa gim dilakukan dengan menggunakan *Unity Profiler* yang berfungsi untuk memberikan informasi dan statistik dengan luaran *frame time* yang dapat digunakan dalam mengidentifikasi bagian-bagian yang dapat dioptimalkan dengan data yang diberikan dari alat tersebut. Pengujian *frame time* diperlukan sebelum dan sesudah optimasi untuk mengetahui adanya pengaruh dalam metode optimasi yang digunakan. *Frame time* dipilih sebagai metrik dalam pengembangan gim daripada *frames per second* (fps) untuk mengukur seberapa cepat dan konsisten gim dijalankan (Unity Technologies, 2021). *Frame time* ditentukan dengan rumus berikut.

$$ms \text{ per frame} = \frac{1000}{target \text{ fps}} \quad (1)$$

Rumus tersebut menggunakan variabel sebelum dan sesudah yang dibagi dengan hasil rata-rata dari kedua variabel tersebut, kemudian dikalikan dengan seratus untuk mendapatkan hasil perbedaan terhadap dua variabel tersebut. Rumus persentase perbedaan bersifat simetris, artinya hasil dari penggunaan rumus tersebut akan tetap sama terlepas dari urutan kedua variabel yang digunakan dan mengatasi masalah perbedaan persentase konvensional yang tidak bersifat aditif (Cole dan Altman, 2017). Berikut adalah rumus yang digunakan:

$$Persentase \text{ perbedaan} = 100 \times \left| \frac{(\text{Difference})}{(\text{Mean})} \right| \quad (2)$$

Dimana:

*Difference* = nilai perbedaan dari pengurangan nilai sebelum dan nilai sesudah

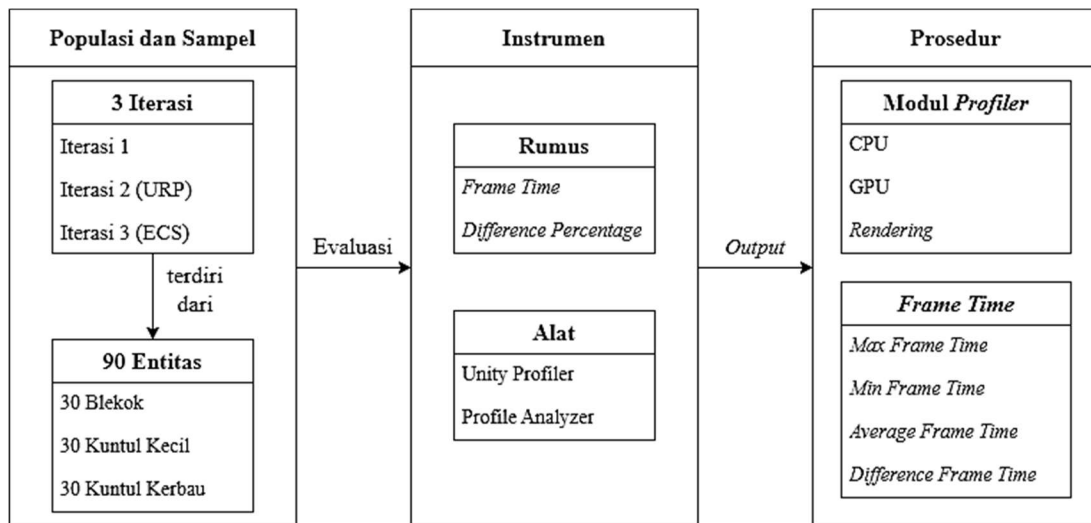
*Mean* = Nilai rata-rata dari penambahan nilai sebelum dan nilai sesudah

Skenario yang digunakan untuk menguji performa gim akan menggunakan 3 macam iterasi dari pengembangan gim Virtual Biotope. Iterasi pertama dari gim Virtual Biotope adalah iterasi yang pertama kali dibuat pada *Unity Engine* versi LTS 2021.3.16f1. Sedangkan untuk iterasi kedua dari gim Virtual Biotope dihasilkan pada *Unity Engine* versi LTS 2021.3.16f1 dengan menggunakan metode *rendering* URP dan level utama dari gim tersebut juga telah dicek kembali dengan aset yang telah dioptimalkan dari segi model dan prinsip desain level. Terakhir, iterasi ketiga dari Virtual Biotope adalah iterasi khusus untuk menguji performa permainan dengan *Unity*

*DOTS* yang mengharuskan menggunakan *Unity Engine* minimal pada versi LTS 2022.3.11f1 dan menggunakan metode *rendering* URP. Maka dari itu, iterasi ketiga dari gim *Virtual Biotope* dibuat dengan *Unity Engine* versi LTS 2022.3.31f1 dan metode *rendering* URP.

### 3.4. Analisis Data

Setiap iterasi akan menggunakan skenario pengujian melalui level utama yang sama dengan kondisi tertentu. Jumlah burung akan ditingkatkan menjadi 30 burung pada setiap spesies yang terdiri dari burung blekok, kuntul kecil, dan kuntul kerbau sehingga totalnya menjadi 90 burung. Kemudian, *flockbox* akan dibuat dengan dimensi 200,100,300 (X,Y,Z) terletak pada daerah awal pemain ditempatkan. Setiap kondisi tersebut berlaku pada semua iterasi untuk memastikan hasil pengujian dari *Unity Profiler* tidak terpengaruh dari faktor lainnya seperti lokasi burung atau *occlusion culling* yang dapat mempengaruhi proses analisis. Mengetahui adanya sistem animasi dalam *flocking* yang telah diterapkan pada gim *Virtual Biotope* tersebut, dilakukan pengujian pada skenario dengan sistem animasi yang diaktifkan dan sistem animasi yang dinonaktifkan untuk mengetahui perbandingan penggunaan pendekatan *data-oriented design* terhadap algoritma *flocking*. Alur skenario pengujian pada penelitian ini dapat dilihat pada Gambar 3.2 berikut.



Gambar 3.2 Alur Skenario Pengujian

Pengujian yang telah dilakukan akan dilakukan perbandingan. Perbandingan tersebut ditentukan dengan melihat *frame time*, penggunaan sumber daya dari CPU dan GPU yang terlihat berdasarkan modul yang aktif dalam *Unity Profiler*, dan skenario yang digunakan untuk perbandingan tersebut dilakukan pada skenario *gameplay* atau *Play Mode*. Untuk menentukan perbedaan dari sebelum dan sesudah implementasi, digunakan rumus persentase perbedaan dari Rumus (2). Berdasarkan rumus tersebut, ditentukan persentase perbedaan terhadap nilai *frame time* yang diambil dari hasil pengukuran pada *Unity Profiler* saat gim dijalankan mengingat salah satu tujuan dari penelitian ini adalah membandingkan performa gim dari segi *flocking* sebelum dan sesudah implementasi pendekatan *data-oriented* untuk mengetahui apakah terdapat peningkatan dari performa gim secara keseluruhan.