Pengembangan Game Colony Simulation *"The Old Ages"* Berbasis *Unity DOTS*

I Made Chrisana Dharmawan[1)], I Gede Suardika[2)], Erma Sulistyo Rini[3)]
Program Studi Sistem Informasi
Institut Teknologi dan Bisnis STIKOM BALI
Denpasar, Indonesia
e-mail: 180030099@stikom-bali.ac.id[1)], suardika@stikom-bali.ac.id[2)], erma@stikom-bali.ac.id[3)]

Abstrak

The Old Ages adalah proyek game berbasis Unity yang dikembangkan dengan Data-Oriented Technology Stack (DOTS) untuk mensimulasikan cara bertahan hidup dan pembangunan pemukiman pedesaan secara interaktif dan efisien. Pemain mengumpulkan sumber daya, membangun desa, dan mengelola masyarakat dalam dunia dinamis dengan ekosistem hidup, cuaca berubah, serta ekonomi alami. Setiap keputusan memengaruhi perkembangan dan kelangsungan pemukiman. Dengan perspektif orang pertama, pemain berinteraksi langsung dengan lingkungan, seperti menebang kayu, membangun bangunan, dan menjelajah. Game ini memanfaatkan DOTS untuk pemrosesan entitas berkinerja tinggi, memungkinkan simulasi Artificial Intelligence berskala besar. Penduduk desa diprogram untuk memiliki kebutuhan dan perilaku dinamis yang dikelola secara efisien. ECS (Entity Component System) meningkatkan skalabilitas, sementara Jobs dan Burst Compiler mengoptimalkan performa untuk gameplay yang lancar. Menggabungkan strategi dan imersi, The Old Ages menawarkan pengalaman bermain dengan suasana pedesaan yang kaya dengan mekanisme bertahan hidup dan pembangunan yang meluas.

Kata kunci*: Game Development, Unity, Colony Simulation, DOTS.*

1. Pendahuluan

The Old Ages adalah proyek game yang dikembangkan menggunakan Unity, memanfaatkan *Data-Oriented Technology Stack (DOTS)* untuk menciptakan game simulasi pembangunan peradaban yang dinamis dan interaktif. Dalam game ini, pemain bertugas untuk mengumpulkan sumber daya, membangun desa, dan mengelola pertumbuhan serta kesejahteraan komunitas mereka di dunia yang berkembang berdasarkan keputusan yang diambil. Berlatarkan suasana pedesaan yang natural, game ini bertujuan untuk memberikan pengalaman bertahan hidup dan pembangunan pemukiman yang realistis, dengan fokus pada manajemen sumber daya, interaksi yang dikendalikan *AI*.

Data-Oriented Technology Stack (DOTS) dalam Unity adalah sekumpulan *tools dan teknologi* yang memungkinkan pengembang game untuk membangun game menggunakan desain berorientasi data. *Pengembangan* game tradisional sering kali bergantung pada pemrograman berbasis objek, yang dapat menyebabkan keterbatasan performa saat mensimulasikan lingkungan dan objek dengan nilai kuantitafif yang besar[1]. DOTS memungkinkan simulasi yang sangat berkinerja tinggi dan skalabilitas besar yang diperlukan untuk menangani sistem gameplay yang kompleks secara real-time[2]. Dengan memanfaatkan *Entity Component System (ECS)* memungkinkan pengoptimalan penggunaan memori dan daya pemrosesan game dalam pengelolaan entitas penduduk desa dalam jumlah besar yang dikendalikan AI dan perubahan lingkungan yang dinamis secara halus[3]. *Burst Compiler* dan *Jobs System* lebih lanjut meningkatkan performa kompilasi dan proses yang dikerjakan secara pararel[4].

Real-Time Strategy (RTS) adalah salah satu genre game strategi dimana pemain melakukan aksi secara serentak (*real-time*)[5]. Dilengkapi dengan fitur perspektif orang pertama, pemain dapat berinteraksi langsung *dengan* lingkungan menebang kayu, membangun bangunan, dan menjelajahi lanskap luas, sambil mengelola berbagai kebutuhan penduduk desa. Penduduk desa dikendalikan menggunakan kecerdasan buatan yang diprogram untuk mengikuti rutinitas dan perilaku layaknya masyarakat di pedesaan, serta dipengaruhi oleh perubahan cuaca, siklus waktu , dan ketersediaan sumber daya[6]. Hal ini dapat melatih pemain dalam melatih kapasitas pemprosesan visual otak[7].

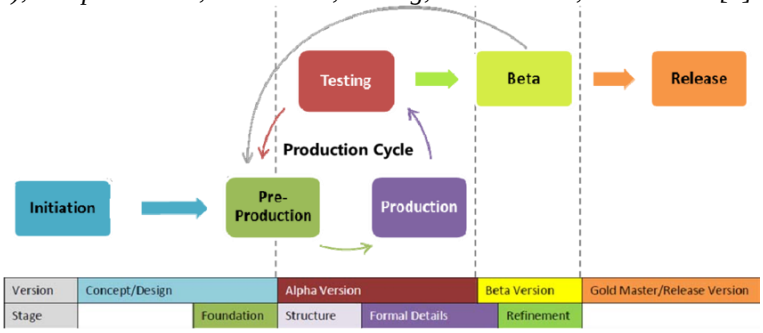
Penelitian ini membahas teknis dasar dari pengembangan game simulasi koloni *The Old Ages*, dan membuktikan bagaimana penggunaan modul *DOTS* dan *Game Engine Unity* untuk menciptakan game simulasi koloni berlatar pedesaan yang imersif dan dioptimalkan untuk mengatur objek yang banyak. Kombinasi perspektif orang ketiga dan latar suasana pedesaan yang kaya menawarkan pengalaman baru dalam gameplay bertahan hidup dan pembangunan pemukiman[8]. Melalui penelitian ini, penulis akan

menganalisa bagaimana kemampuan *Game Engine Unity* dalam memanfaatkan kemampuan komputer untuk menciptakan simulasi lingkungan.

2. Metode Penelitian

2.1 Metode *Game Developemnt Life Cycle*

Dalam pengembangan game simulasi koloni *"The Old Ages"*, metode pengembangan yang dipakai adalah *Game Development Life Cycle* (GDLC). Metode GDLC memiliki 6 tahapan, yaitu *Initiation(Concept), Pre-production, Production, Testing, Beta Release, dan Release[9].*



Gambar 1. *Game Development Life Cycle*

2.2 Metode *Data-Oriented Technology Stack*

Penelitian ini menggunakan pendekatan eksperimental untuk mengevaluasi efektivitas *Data-Oriented* Technology Stack (DOTS) dalam menangani simulasi entitas berskala besar. Fokus utama penelitian ini adalah menganalisis peningkatan performa, manajemen memori, dan skalabilitas saat menggunakan *Entity Component System (ECS)* dibandingkan dengan arsitektur pemprograman tradisional berbasis *Object Oriented*[10].

3. Hasil dan Pembahasan

3.1 *Initiation* (Inisialisasi)

Initiation adalah tahap awal dalam metode mengembangan GDLC, pada tahap ini melibatkan penjabaran perencanaan awal, ide dan gagasan yang akan menjadi konsep dasar game tersebut dijabarkan, mulai dari penentuan alur, tema, dan rancangan objektif dalam game. Skema tersebut akan menjadi panutan dasar dalam pelaksanaan pengembangan game. *The Old Ages* dirancang sebagai game *Real-Time Strategy survival* dan *city builder* bertema pedesaan dengan perspektif orang ketiga. Konsepnya berfokus pada realisme, simulasi *AI* penduduk desa, dan lingkungan yang dinamis. Teknologi yang digunakan adalah Unity DOTS (*ECS, Job System*, dan *Burst Compiler*).

3.2 *Pre-Production*

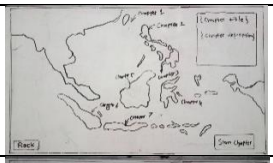
Desain utama dari sebuah game meliputi mekanisme game, alur cerita, karakter, concept art, storyboard, dan gameplay secara keseluruhan. Pembuatan prototype berupa game versi awal untuk menguji mekanisme utama dari game yang dikembangkan, serta merencanakan kebutuhan teknis game tersebut, seperti *game engine, tools, framework*, dan arsitektur secara keseluruhan. Pada tahap ini, desain game diperinci lebih lanjut. Elemen *gameplay* utama ditentukan, termasuk:

Tabel 1. *Pre-production*

Input	Proses	Output	Remarks
Genre	Penentuan genre game melalui preferensi target pemain	Genre *Colony Simulation, Real-Time Strategy, dan City Builder.*	Mengacu pada referensi genre popular seperti *Banished* atau *Dawn of Man*
Gameplay	Merancang mekanisme gameplay, termasuk pengumpulan sumber daya, pembangunan, dan manajemen.	a) Sistem pengumpulan sumber daya, seperti menebang kayu, menambang, berburu. b) Pembangunan dan manajemen desa, seperti struktur bangunan,	Fokus pada kombinasi mekanik strategi real-time dan perspektif orang ketiga.

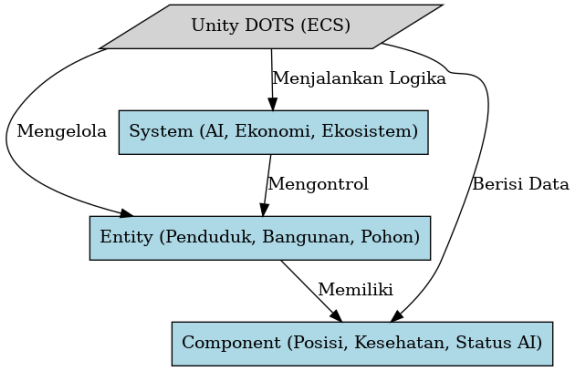
		kebutuhan penduduk, ekonomi. c) *AI* penduduk desa : *pathfinding*	
Skenario	Mengembangkan narasi berdasarkan interaksi sosial dan ekosistem dinamis dalam game.	Naskah cerita dunia game, karakter AI, dan alur utama.	Skenario dengan sistem ekosistem hidup, perubahan waktu.
Game Engine	Menentukan game engine yang akan digunakan untuk membangun game	Unity 2022.3.58f1	Unity dipilih menjadi game engine karena penulis sudah familiar dengan tools yang disediakan dan Bahasa pemprograman yang digunakan, yaitu C#.

Tabel 2. *Storyboard*

No	Sketsa	Event	Keterangan
1		Scene Main menu	Scene Main Menu dengan tombol – tombol untuk memilih mde permainan, seperti Campaign, Freeplay, dan opsi lain, seperti pengaturan, tentang, dan tombol keluar.
2		Scene Campaign	Scene Campaign Map untuk menunjukan pilihan chapter mode kampanye, dan deskripsi chapter kampanye tersebut.
3		Scene Gameplay	Scene dimana player bermain dengan visual 3 dimensi, dilengkapi dengan HUD dan Overlay untuk membantu managemen penduduk dan persediaan.

3.3 Entity Component System Diagram

The Old Ages menggunakan *Unity DOTS (Entity Component System, Job System, Burst Compiler)* yang menggambarkan bagaimana *Entity* (seperti penduduk desa, bangunan, pohon), *Component* (data seperti posisi, kebutuhan, pekerjaan), dan *System* (proses yang berjalan, seperti *AI*, ekonomi, dan ekosistem) bekerja. Berikut merupakan penggambaran diagram *ECS*:



Gambar 2. Diagram Arsitektur *Enity Component System*

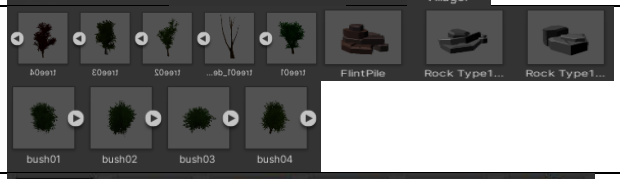
3.4 *Production*

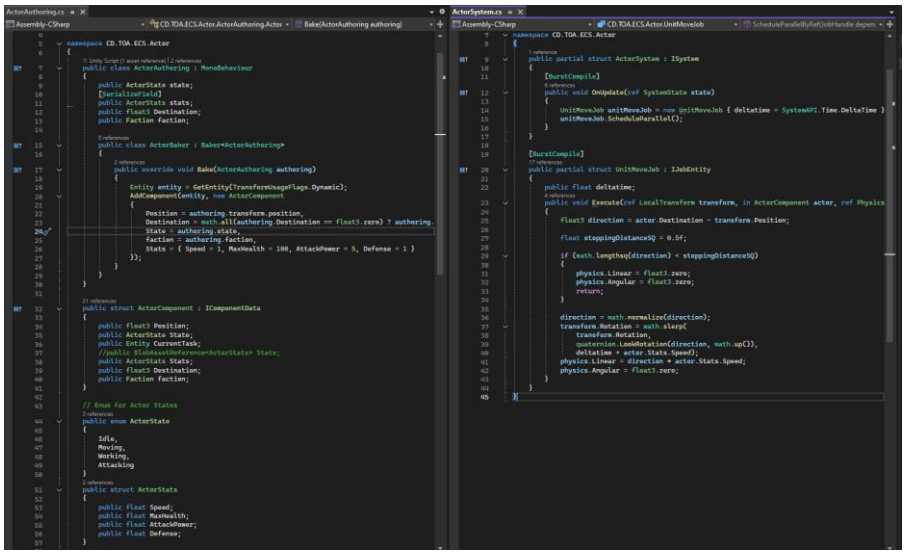
Tahap production mencakup beberapa aktifitas, seperti pembuatan asset, pemprograman, dan integrasi. Asset yang dibuat yaitu berupa sprite, model 3 dimensi, texture, animasi, efek suara, dan musik.

E-ISSN : 3031-9692 Vol. 2 No. 1 2025
Institut Teknologi dan Bisnis STIKOM Bali, 8 Maret 2025

pemprograman logika game, seperti *artificial intelligence* yang akan mengontrol perilaku karakter, dan mekanisme game lainnya dengan cara menulis kode dalam bahasa pemprograman, serta mengintegrasikan aset-aset grafis dan audio dengan code agar dapat berjalan secara serasi didalam game. Component dan System dibuat untuk memanipulasi task (seperti *DestinationTask*), *stats* dan *position* pada Aktor.

Tabel 3. Daftar aset visual

No	Gambar	Keterangan
1		Model 3 dimensi yang mengilustrasikan alat – alat,manusia, dan bangunan.
2		Model 3 dimensi yang mengilustrasikan vegetasi, dan batuan.
5		Ikon 2d yang melambangkan sebuah objek atau entitas didalam game.



Gambar 3. Source-code *ActorSystem* dan *ActorComponent*

3.5 *Testing*

Testing adalah tahap dimana game akan dijalankan dan diuji secara menyeluruh dan mendalam untuk mencari permasalahan, seperti *bugs* dan *glitch*. Testing dilakukan oleh pengembang untuk mengevaluasi pengalaman, dan kenyamanan yang dirasakan dalam bermain game.

Gambar 4. Tampilan Main Menu

Gambar 5. Tampilan Map Campaign

Gambar 6. *Campaign Chapter 1*

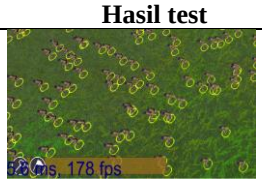
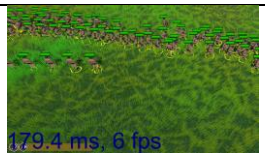
Tabel 4. Tabel *testing scene* game

No	Scene	Ekspektasi	Hasil test	Lulus/Gagal
1	Main Menu	Menampilkan tombol-tombol menu utama dengan latar belakang suasana asri pemukiman,	Tombol tombol ditampilkan dan penampakan suasana pemukiman	Lulus
2	Map Campaign	Menampilkan map campaign berserta chapter 1 yang dapat dipilih.	Map campaign dapat ditampilkan dan pilihan chapter 1 dapat dipilih	Lulus
3	Campaign Chapter 1	Menampilkan UI Overlay dan HUD, menampilkan dunia, serta menjalankan mekanisme game.	Overlay dan HUD tampil sesuai rencana, dunia tampil dan mekanisme game berjalan.	Lulus

Tabel 5. Tabel *testing in-game mechanism*

No	Aksi	Ekspektasi	Hasil test	Lulus/Gagal
1	Actor Spawn	Memunculkan 3 *actor* (penduduk) di awal game		Lulus
2	*Actor Select*	Menampilkan marker di actor dan memilih actor.		Lulus
3	Actor Move	Memberi perinah Kepada actor untuk bergerak ke suatu lokasi di map		Lulus

Tabel 6. Perbandingan performa

No	Tipe	Ekspektasi	Hasil test	FPS Avg
1	*SubScene Entity*	Memunculkan 500 *actor* (penduduk) di awal game dan menggerakannya secara serentak		178
2	GameObject	Memunculkan 500 *actor* (penduduk) di awal game dan menggerakannya secara serentak		6

3.6 Beta

Tahap *Beta* adalah tahap dimana game akan di-compile agar menjadi biner *stand-alone* untuk diuji coba agar dapat berjalan, dan memastikan tidak ada *error* dan *bug*. Tahap ini biasanya dilaksanakan oleh *beta tester atau insider* yang memiliki keterampilan khusus untuk mencari bug dan error dalam program, serta campur tangan pengembang yang menerima laporan dan umpan balik untuk perbaikan aplikasi. Untuk build beta dapat diunduh melalui link berikut : https://drive.google.com/drive/folders/1bNx2MCQvIWPetvs4UI1WQ6WZ5c-uHMb9?usp=sharing

4. Kesimpulan

Pengembangan game *"The Old Ages"* menggunakan *Data-Oriented Technology Stack (DOTS)* memberikan peningkatan performa yang cukup signifikan, terutama dalam dalam memanagemen komponen untuk menampung atribut, serta iterasi pada *job system* actor.

Daftar Pustaka

[1] Robert. Nystrom, *Game programming patterns*. Genever Benning, 2014.
[2] H.-C. Song, "A Study on Optimized Multi-Thread Porgramming : Information and The Use of Unitiy DOTS," *Journal of Digital Contents Society*, vol. 22, no. 10, pp. 1715–1719, Oct. 2021, doi: 10.9728/dcs.2021.22.11.1715.
[3] U. Technologies, *Introduction to the Data-Oriented Technology Stack for Advanced Unity Developers*. Unity, 2022. [Online]. Available: https://unity.com/resources/introduction-to-dots-ebook
[4] R. Eyniyev, "OPTIMIZING PARALLELISM IN UNITY WITH JOB SYSTEM TOOL," *Azerbaijan Journal of High Performance Computing*, vol. 5, no. 2, pp. 183–192, Dec. 2022, doi: 10.32010/26166127.2022.5.2.183.192.
[5] E. Z. Elfeky *et al.*, "A Systematic Review of Coevolution in Real-Time Strategy Games," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3115768.
[6] J. Flick, "C# and Shader Tutorials for the Unity Engine." [Online]. Available: https://catlikecoding.com/unity/tutorials/
[7] Y. Yao *et al.*, "Action Real-Time Strategy Gaming Experience Related to Enhanced Capacity of Visual Working Memory," *Front Hum Neurosci*, vol. 14, 2020, doi: 10.3389/fnhum.2020.00333.
[8] Š. Tichý, "The Last Clan - RTS game in Unity," Charles University, Prague, 2023.
[9] J. Lasmana Putra and C. Kesuma, "Penerapan Game Development Life Cycle Untuk Video Game Dengan Model Role Playing Game." [Online]. Available: http://jurnal.bsi.ac.id/index.php/co-science
[10] J. D. Bayliss, "The Data-Oriented Design Process for Game Development," *Computer (Long Beach Calif)*, vol. 55, no. 5, pp. 31–38, May 2022, doi: 10.1109/MC.2022.3155108.