IMPLEMENTASI ALGORITMA *PERLIN NOISE* PADA SIMULASI PERMUKAAN AIR *REALTIME* DENGAN PEMROGRAMAN *MULTI-THREADING* MENGGUNAKAN UNITY DOTS

Skripsi

untuk memenuhi sebagian persyaratan mencapai derajat Sarjana S-1

Program Studi Teknik Informatika

Diajukan oleh:

Naufal Asyhab

15650057

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2022

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-1823/Un.02/DST/PP.00.9/08/2022

Tugas Akhir dengan judul : IMPLEMENTASI ALGORITMA PERLIN NOISE PADA SIMULASI PERMUKAAN AIR REALTIME DENGAN PEMROGRAMAN MULTITHREADING MENGGUNAKAN UNITY DOTS

yang dipersiapkan dan disusun oleh:

Nama : NAUFAL ASYHAB
Nomor Induk Mahasiswa : 15650057
Telah diujikan pada : Jumat, 19 Agustus 2022
Nilai ujian Tugas Akhir : A-

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR

Ketua Sidang

Ir. Muhammad Taufiq Nuruzzaman, S.T. M.Eng., Ph.D.
SIGNED

Valid ID: 63021661e743b

Penguji I

Ir. Sumarsono, S.T., M.Kom.
SIGNED

Valid ID: 62f17cb6d844b

Penguji II

Ir. Aulia Faqih Rifa'i, M.Kom.
SIGNED

Valid ID: 6301aa3e6fbb2

Yogyakarta, 19 Agustus 2022
UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi

Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

Valid ID: 6303104451647

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi
Lamp : -

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Naufal Asyhab
NIM : 15650057
Judul Skripsi : Implementasi Algoritma *Perlin Noise* Pada Simulasi Permukaan Air *Realtime* Dengan Pemrograman *Multithreading* Menggunakan Unity DOTS

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Teknik Informatika.

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 08 Agustus 2022
Pembimbing

Ir. M. Taufiq Nuruzzaman, Ph.D
NIP. 19710209 200501 1 003

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini:

Nama :Naufal Asyhab

NIM :15650057

Jurusan :Teknik Informatika

Fakultas :Sains dan Komunikasi

Menyatakan bahwa skripsi yang berjudul "**Implementasi Algoritma *Perlin Noise* pada Simulasi Permukaan Air *Realtime* dengan Pemrograman *Multithreading* menggunakan Unity DOTS**" tidak terdapat pada karya yang pernah di ajukan untuk memperoleh gelar sarjana di suatu Perguruan Tinggi, dan sepengetahuan penulis tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan di sebutkan dalam daftar pustaka.

Yogyakarta, 08 Agustus 2022

Yang menyatakan

Naufal Asyhab

NIM.15650057

KATA PENGANTAR

Alhamdulillahi Robbil 'Alamin. Segala puji syukur bagi Allah SWT yang telah memberikan pertolongan dalam setiap kesulitan yang ada selama penelitian dan penelitian skripsi. Sholawat serta salam semoga tetap tercurah kepada baginda Nabi Muhammad *Sholallahu' alaihi wa sallam*. Atas berkat rahmat-NYA peneliti dapat menyelesaikan penelitian tugas akhir yang berjudul "IMPLEMENTASI ALGORITMA *PERLIN NOISE* PADA SIMULASI PERMUKAAN AIR *REALTIME* DENGAN PEMROGRAMAN *MULTI-THREADING* MENGGUNAKAN UNITY DOTS".

Dalam kesempatan ini peneliti menyampaikan terima kasih yang sebesar-besarnya kepada:

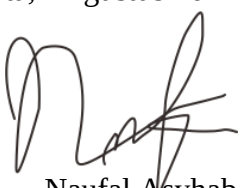
1. Bapak Prof. Dr.Phil. Al Makin, S.Ag., M.A., selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Ibu Dr. Dra. Hj. Khurul Wardati, M.Si., selaku Dekan Fakultas Sains dan Teknologi.
3. Ibu Ir. Maria Ulfah Siregar, S.Kom., MIT., Ph.D., selaku Kepala Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta.
4. Bapak Dr. Bambang Sugiantoro, M.T., selaku dosen pembimbing akademik.
5. Ir. Muhammad Taufiq Nuruzzaman, S.T. M.Eng., Ph.D., selaku dosen pembimbing skripsi yang telah membimbing dan memberikan

pengarahan, nasihat dan masukan kepada peneliti dalam menyusun skripsi ini.

6. Seluruh Bapak/Ibu Dosen Teknik Informatika UIN Sunan Kalijaga Yogyakarta atas ilmu yang diberikan selama masa perkuliahan kepada peneliti.
7. Kedua orangtua, kedua adik dan keluarga besar peneliti yang selalu memberikan *support* kepada peneliti.
8. Teman-teman dari komunitas Gamedev Gamelan Jogja, teman – teman dari MMTC dan teman-teman sesama Game Developer yang tidak dapat disebutkan satu persatu.
9. Terimakasih untuk teman – teman SAMAWA khususnya sahabat – sahabat saya Denny, Rendre, Faqih, Argo, Fardha serta Habibie, Brian, Fikri yang telah memberikan support baik secara langsung maupun tidak sehingga peneliti mampu menyelesaikan skripsi ini.

Peneliti menyadari tentu saja masih banyak kekurangan dalam penelitian laporan skripsi ini, sehingga kritik serta saran dari pembaca sangat peneliti harapkan. Semoga dapat dijadikan sebagai dasar penyempurnaan penelitan selanjutnya.

Yogyakarta, Agustus 2022



Naufal Asyhab

HALAMAN PERSEMBAHAN

Karya skripsi ini saya persembahkan kepada:

1. Kedua orang tua, nenek dan adik – adik yang selalu memberikan dukungan, restu, beserta doanya.
2. Kepada seorang hamba Allah yang tidak bisa saya sebutkan namanya, karena jasa beliau saya mampu melanjutkan kuliah di UIN SUKA, bahkan meskipun saya hampir mengecewakan beliau, beliau tetap percaya dan membantu saya.
3. Kepada keluarga besar yang kerap memberikan dukungan dan bantuan.
4. Kepada sahabat – sahabat SAMAWA yang memberikan saya tempat di kala saya kehilangan tempat dan arah serta banyak memberikan dukungan moral baik secara langsung maupun tidak, sehingga saya bisa perlahan mengatasi rasa trauma, insecure dan bersalah pada diri saya.
5. Kepada sahabat terdekat saya Denny Elang yang tidak capek – capeknya menanyakan kapan sidang dan menjadi teman berbagi serta bercerita soal lika liku kehidupan dan gamedev.
6. Kepada sahabat saya sejak kecil Zullivan yang kerap menjadi teman berbagi dan bercerita ketika di Ngawi dan sesama pejuang Skripsi.
7. Kepada Habibie, Brian, Fitree, Fikri yang telah menjadi teman bagi saya.
8. Kepada Vivi yang kerap mau saya repotkan dengan meminjam laptopnya.
9. Kepada Deni, Samad dan semua jajaran pegawai PTIPD dan anak – anak PKL ketika dulu saya magang.
10. Kepada teman-teman KKN Angkatan 96 UIN Sunan Kalijaga Yogyakarta.

11. Kepada teman – teman Facebook baik sesama Game Developer maupun tidak yang saling berbagi dan memberikan kebahagiaan.
12. Beberapa dari teman-teman Teknik Informatika angkatan 2015, yang tidak berpaling dari saya ketika saya membutuhkan bantuan.
13. Kepada Grandis Hanif Avian yang telah memberikan kenangan dan pengalaman berharga bagi saya.

HALAMAN MOTTO

Betrayal, that pain that feels like it'll eat you from the inside out,

Can either break you or forge you into something Greater

-Silco, Arcane League of Legends-

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

IMPLEMENTASI ALGORITMA *PERLIN NOISE* PADA SIMULASI PERMUKAAN AIR *REALTIME* DENGAN PEMROGRAMAN *MULTI-THREADING* MENGGUNAKAN *UNITY DOTS*

Naufal Asyhab

15650057

INTISARI

Air merupakan elemen kehidupan yang sangat penting sehingga simulasi air memainkan peranan yang besar dan krusial sebagai objek penelitian grafik komputer sejak lama. Selaras dengan itu metode pengembangan simulasi air serta penggunaannya dalam industri seperti film dan gim semakin marak digunakan.

Salah satu algoritma pengembangan simulasi air yang populer adalah *Perlin Noise*. *Perlin Noise* merupakan algoritma PCG berbasis gradien yang memilih titik – titik pada bidang secara acak kemudian menginterpolasi dan memperhalusnya sehingga menghasilkan noise. Dan untuk membuat simulasi air yang terkesan realistis maka disini peneliti menambahkan Sistem AI yang bertugas untuk memproduksi dan mengontrol pergerakan ikan – ikan yang berenang. Prosedur untuk memproduksi noise dan mengontrol pergerakan ikan – ikan akan terus dijalankan untuk membuat simulasi terkesan hidup. Dengan begitu banyaknya proses paralel yang dijalankan secara bersamaan akan membebani kinerja CPU sehingga butuh suatu metode untuk mengoptimisasinya, yaitu dengan pemrograman *multithreading*. Keunggulan pemrograman *multithreading* adalah mampu memecah *thread* yang dan menjalankannya pada *cores* terpisah pada CPU sehingga mampu meningkatkan performa program secara signifikan. Disisi lain, terdapat beberapa *tradeoff* seperti penulisan kode program yang sulit dan kerentanan akan *race condition*. Unity DOTS mampu mengatasi masalah – masalah tersebut.

Hasil penelitian ini menunjukan bahwa algoritma *Perlin Noise* dan sistem AI yang diterapkan membuat simulasi air menjadi semakin realistis, serta dengan diterapkannya *multithreading* dengan DOTS program menjadi optimal serta performanya meningkat dengan pesat dibuktikan dengan berkurangnya rata – rata CPU load time sebesar 60.5% dan meningkatnya rata – rata fps sebesar 82.6%.

Kata Kunci: Algoritma, *Perlin Noise*, Simulasi Permukaan Air *Realtime*, *AI*, *Multithreading,* Unity, Unity DOTS.

PERLIN NOISE ALGORITHM IMPLEMENTATION ON REALTIME WATER SURFACE SIMULATION & FISH AI SYSTEM WITH MULTI-THREADING PROGRAMMING USING UNITY DOTS

ABSTRACT

Water is a very important element of life so that water simulation plays a big and crucial role as the object of computer graphics research for a long time. In line with this, water simulation development methods and their use in industries such as films and games are increasingly being used.

One of the popular water simulation development algorithms is Perlin Noise. Perlin Noise is a gradient-based PCG algorithm that generates noise by randomly selecting points on the plane and interpolating and refining them. To create realistic water simulations, the researchers include an AI system whose job it is to generate and control the movement of the swimming fish. Procedures for producing noise and controlling the movement of fish will continue to be carried out to make the simulation seem alive. With so many parallel processes running concurrently, it will burden CPU performance, so it takes a method to optimize it, namely multithreading programming. The advantage of multithread programming is that it is able to break threads and run them on separate cores on the CPU so as to significantly improve program performance. On the other hand, there are some tradeoffs such as writing difficult program code and vulnerability to race conditions. Unity DOTS is able to overcome these problems.

The results of this study indicate that the Perlin Noise algorithm and the AI system applied make water simulations more realistic, and with the implementation of multithreading with DOTS the program becomes optimal and its performance increases rapidly as evidenced by the decrease in CPU load time by 60.5% and an increase in average CPU load time of 60.5%. the average fps is 82.6%.

Keywords: *Algorithm, Perlin Noise, Realtime Water Surface Simulation, AI, Multithreading,* Unity, Unity DOTS.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Air merupakan sumber daya alam yang vital bagi kehidupan di bumi dan sekitar 71% dari bumi terdiri dari lautan. Sejak ditemukannya grafis computer, simulasi permukaan air telah mendapatkan banyak perhatian. Simulasi permukaan air telah banyak memainkan peranan penting dan diimplementasikan dalam berbagai sektor industri seperti *VR (Virtual Reality)*, Film, Animasi 3D, Gim dan lain sebagainya. Sebagai contohnya dalam Film seperti *Titanic, Finding Nemo* dan *Pacific Rim*, Atau dalam gim *action survival* seperti *Tomb Rider, Uncharted* dan gim petualangan bawah laut seperti *Abzu*.

Perbedaan utama antar simulasi air ini adalah *trade-off* antara kecepatan komputasi dan tingkat realisme. Atas kategori *trade-off* ini simulasi air dibedakan menjadi 2, yaitu *Pre-rendered* dan *Realtime*. Simulasi air *Pre-rendered* digunakan dalam Film dan simulasi-simulasi sains, simulasi *Pre-rendered* ini membutuhkan proses komputasi rendering yang memakan waktu berhari – hari atau bahkan berminggu – minggu. Tidak seperti simulasi air *Pre-rendered*, simulasi air *Realtime* berjalan dengan terus mengupdate sekian banyak kali dalam setiap detiknya. Simulasi air *Realtime* ini banyak digunakan dalam gim – gim guna untuk meningkatkan realisme dan imersivitas dalam bermain.

Seiring dengan berkembangnya teknologi dan berbagai macam penelitian mengenai simulasi air, berbagai metode untuk mensimulasikan air secara *Realtime* pun ditemukan. Metode yang paling umum digunakan adalah dengan melibatkan manipulasi pada bidang ketinggian dari 2D *mesh* yang terdiri dari *vertices* yang saling berhubungan, seperti persamaan Navier-Stokes, FFT (*Fast Fourier Transform*) dan *Perlin Noise*.

Perlin Noise merupakan algoritma PCG (*Procedural Generated Content*) yang dikembangkan oleh Ken Perlin (1981) dengan menspesifikkan nilai dan gradien pada ruang secara pseudo-random pada masing – masing titik dari 8 *vertices* terdekat pada kisi kubus bilangan bulat (*Integer Cubic Lattice*) dan kemudian dilakukan interpolasi spline. *Perlin Noise* banyak digunakan untuk efek visual grafik komputer yang terlihat acak dan natural misalkan api, asap, awan, kabut, medan/*terrain*. Dalam bidang gim sendiri, algoritma *Perlin Noise* banyak digunakan untuk menghasilkan object gim dan *Terrain*. Kelebihan dari Perlin Noise selain daripada familiaritasnya yang banyak digunakan, parameter – parameter yang mudah dikendalikan seperti frekuensi, oktaf, *lacunarity* dan persistensi sepanjang arah sumbu axis manapun dan pola acak yang dihasilkan tetap terlihat natural.

Untuk menghasilkan simulasi air secara *realtime*, maka *Perlin Noise* memanipulasi height map pada mesh secara kontinu/terus menerus sehingga menghasilkan visual air yang realis. Proses yang dijalankan secara kontinu dan parallel seperti pada umumnya algoritma PCG memakan sumber daya komputasi yang berat dan beresiko menurunkan performa, jika diimplementasikan dalam gim

maka akan berakibat pada menurunnya pengalaman bermain (*User Experience*) dan lebih buruknya terjadi crash pada program. Maka dari itu perlukan teknik optimisasi tertentu agar proses yang berjalan secara paralel dan kontinu tersebut dapat dijalankan dengan efisien dan performan. Sehingga peneliti mempunyai solusi untuk membangun sistem simulasi air menggunakan metode *Multithread Programming*.

Sistem komputasi single-threaded menerima satu instruksi pada satu waktu dan mengeluarkan satu hasil pada satu waktu. Jumlah pekerjaan yang diperlukan untuk dilakukan *CPU* menentukan berapa lama waktu yang dibutuhkan untuk memuat dan menyelesaikan program.

Sedangkan *Multithreading* adalah teknik pemrograman yang memanfaatkan kapasitas *CPU* untuk mengoperasikan beberapa *thread* sekaligus di berbagai *core* dari *CPU*. Dibandingkan dengan instruksi yang dieksekusi satu per-satu, mereka dieksekusi secara bersamaan. Secara default, satu *thread* berjalan di awal program. Inilah yang disebut sebagai "*main thread*". Untuk menangani tugas, *main thread* meluncurkan *thread* baru. *Thread* tambahan ini bekerja secara paralel dengan *main thread* dan, setelah selesai akan menyinkronkan hasilnya dengan *main thread*.

Jika hanya terdapat beberapa *task* yang sudah berjalan lama, metode multithreading ini akan bekerja secara efektif. Di sisi lain dalam kode pengembangan gim isu dari penggunaan metode *multithreading* bersifat signifikan, dimana tugas seperti *rendering* grafis yang berat diproses oleh *GPU,* kemudian di sisi program bertugas untuk memproses waktu yang dibutuhkan untuk menjalankan

algoritma seperti *path finding*, *object tracking* (seperti objek peluru, trajektori, *collision* dll), *Artificial Intelligence*, memuat *resources* dari *hardware* penyimpanan dsb. Tugas – tugas ini sering berisi sejumlah besar instruksi kecil yang semuanya harus dieksekusi pada saat yang bersamaan. Jika membuat *thread* untuk masing-masing, maka akan didapatkan banyak dari mereka masing-masing memiliki umur yang terbatas. Ini dapat membebani kemampuan pemrosesan *CPU* dan sistem operasi dan berakibat pada performa aplikasi/gim yang berjalan. Dengan menggunakan metode *multithreading* penugasan paralel ini dapat diorganisir dengan membagi instruksi – instruksi tersebut kedalam bentuk *thread* yang lebih kecil untuk dialokasikan pada semua *core* dari *processor* yang aktif dan menjalankan mereka secara simultan, bukan secara linier sehingga performa program dapat meningkat dengan pesat.

Tetapi terdapat beberapa kerugian dari pemrograman multi-threading, seperti halnya tingkat kesulitan penulisan kode dibandingkan dengan metode *single-threading*, kesulitan dalam mereplikasi error dalam aplikasi *multi-threaded* atau *multi-context* dibandingkan dengan aplikasi yg dihasilkan dari *singlethreading*, dan hasilnya akan lebih sulit untuk mengidentifikasi akar masalah ketika terjadi error, kemudian memanajemen pemrosesan konkurensi diantara *threads* merupakan pekerjaan yang sulit dan kompleks serta berpotensi untuk menimbulkan error baru pada program seperti *race condition*. *Race condition* ini adalah kondisi ketika dua pemrosesan saling memperebutkan sumber daya yang sama. Ini berarti keseluruhan program menjadi tidak dapat diprediksi dan karena satu proses dapat mengubah *value* sebelum proses lain membaca *value* tersebut.

Dalam pengembangan gim sendiri, Unity merupakan *game engine* yang sudah lama menjadi standar industri dan jamak digunakan oleh para Pengembang Gim, sehingga terdapat banyak sumber daya pengembangan seperti *SDK* & *Library* untuk berbagai solusi permasalahan pengembangan gim seperti *Physics & Collission, Path Finding, Tiling Map, Terrain, Artificial Intelligence* dll, termasuk untuk pemrograman *multithreading* ini, yaitu Unity DOTS.

DOTS (*Data Oriented Tech Stack*) merupakan teknologi baru dalam Unity yang saling bersinergi untuk mendeliver pendekatan DOD (*Data Oriented Design*) yang berkebalikan dengan OOP (*Object Oriented Programming*). DOTS memungkinkan pengembang gim untuk mengoptimalkan kinerja project Unity dengan menggunakan *multicore processing* dalam pemrosesan data. DOTS terdiri dari beberapa elemen yaitu ECS (*Entity Component System*), C# *Job System, Burst Compiler* dan *Native Containers*. Disini peneliti menggunakan *Job System, Burst Compiler* dan *Native Containers* dari DOTS untuk membangun Simulasi Air & Sistem AI ini.

Sehingga peneliti bertujuan untuk membuktikan apakah dengan diimplementasikannya Algoritma *Perlin Noise* mampu memberikan efek visual Air yang natural dan realis serta dengan pemrograman multithreading menggunakan Unity DOTS ini mampu mengoptimisasi performa *Perlin Noise* serta program yang dibuat dengan menggunakan Unity dengan efisien dan aman, tanpa harus mengkhawatirkan isu – isu yang timbul pada pemrograman multithreading tanpa menggunakan Unity DOTS.

Dengan demikian, penelitian "Implementasi Algoritma Perlin Noise pada Simulasi Permukaan Air *Realtime* dengan Pemrograman Multithreading menggunakan Unity DOTS" ini dapat dilakukan. Diharapkan dengan dilakukannya penelitian ini dapat membuktikan dengan diimplementasikannya algoritma *Perlin Noise* mampu membuat simulasi Air yang terlihat natural dan realis secara *realtime*, serta dengan menggunakan pemrograman *multithreading* dapat membuat eksekusi *Perlin Noise* menjadi performan, serta membuktikan bahwa Unity DOTS mampu membuat penulisan kode program *multithreading* menjadi jauh lebih efisien dan aman sehingga terhindar dari isu – isu yang dapat ditimbulkan pemrograman *multithreading.*

1.2 Rumusan Masalah

Berdasar latar belakang di atas, masalah penelitian ini adalah bagaimana implementasi algoritma *Perlin Noise* dan membangun program simulasi air *realtime* menggunakan metode pemrograman *multithreading* yang efisien, performan dan aman dari isu – isu yang umumnya dapat ditimbulkan pemrograman *multithreading* menggunakan Unity DOTS.

1.3 Batasan Masalah

Agar pembahasan tidak melebar dan terfokus pada tujuan yang diinginkan maka dijelaskan maka dijelaskan ruang lingkup dari skripsi ini. Masalah – masalah yang akan dibahas yaitu sebagai berikut:

1. Algoritma yang digunakan untuk *Wave Generator* adalah *Perlin Noise*
2. *Game Engine* yang digunakan untuk mengembangkan sistem adalah Unity
3. Testing dijalankan pada Unity Editor
4. Bahasa pemrograman yang digunakan adalah C#
5. SDK yang digunakan untuk pemrograman *multithreading* adalah Unity DOTS

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan algoritma *Perlin Noise* untuk menciptakan efek visual permukaan air
2. Menerapkan pemrograman *multithreading* pada program yang dikembangkan
3. Menggunakan Unity DOTS sebagai basis pemrograman *multithreading* untuk sistem tersebut, serta menguji kompatibilitas algoritma *Perlin Noise* dengan DOTS. Diharapkan dengan menggunakan metode *multithreading* dengan DOTS tersebut program yg dibangun dengan *multithreading* menjadi efisien, performan dan aman dari isu – isu pemrograman *multithreading*

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan memiliki manfaat penelitian sebagai berikut:

1. Dapat mengetahui bagaimana efek visual air yang dihasilkan dengan menerapkan Algoritma *Perlin Noise*
2. Dapat mengetahui bagaimana dampak dari diimplementasikannya metode pemrograman *multithreading* pada program dengan banyaknya proses yang dijalankan secara paralel
3. Dapat mengetahui bagaimana kompatibilitas algoritma *Perlin Noise* dengan DOTS serta dampak yang dihasilkan dengan menggunakan DOTS pada *Perlin Noise*
4. Dapat membuktikan bahwa pengembangan program *multithreading* dengan menggunakan Unity DOTS membuat kode program menjadi jauh lebih mudah serta program menjadi efisien, performan dan aman dari isu – isu yang dapat ditimbulkan oleh pemrograman *multithreading* biasa

1.6 Keaslian Penelitian

Penelitian yang berkaitan dengan membangun Implementasi Algoritma *Perlin Noise* pada Simulasi Permukaan Air dengan Pemrograman *Multithreading* menggunakan *Unity DOTS* ini sepanjang sepengetahuan peneliti belum pernah dilakukan sebelumnya.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian implementasi algoritma *Perlin Noise* pada simulasi permukaan air dan Sistem AI Ikan dengan pemrograman *multithreading* menggunakan Unity DOTS, maka dapat diambil kesimpulan sebagai berikut:

1. Peneliti telah berhasil mengimplementasikan algoritma *Perlin Noise* untuk membuat efek *vertex displacement shader* permukaan air yang realis dengan tampilan dinamika air natural yang dan Sistem AI Ikan baik menggunakan metode *singlethreading* maupun *multithreading* menggunakan DOTS
2. Peneliti telah berhasil membuktikan dengan diimplementasikannya metode pemrograman *multithreading* menggunakan Unity DOTS (*Job System* & *Burst*) mampu mengoptimalkan dan membuat program yang dibuat dengan Unity yang menggunakan proses – proses berat seperti *vertex displacement shader Perlin Noise* dan manajemen behavior AI yang sangat banyak menjadi sangat performan dan aman, dibuktikan dengan tidak ada *race condition* yang terjadi serta hasil peningkatan persentase load time CPU rata – rata sebesar **60.5%** dan persentase peningkatan *fps* rata – rata sebesar **82.6%**.

6.2 Saran

Berikut ini beberapa saran yang dapat dipertimbangkan untuk peneliti selanjutnya:

1. Simulasi permukaan air ini dapat diterapkan dengan algoritma lain selain *Perlin Noise*, seperti *Simplex Noise* ataupun *Fast Fourier Transform* dan algoritma noise lainnya.
2. Pergerakan AI Ikan dapat diimplmentasikan dengan menggunakan algoritma *Boids* untuk pergerakan yang lebih natural dan realis.
3. Tech stack dari Unity DOTS yang peneliti gunakan untuk mengimplementasikan pemrograman *multithreading* adalah *Job System & Burst Compiler* saja. Diharapkan kedepannya akan lebih baik jika menggunakan ECS (*Entity Component System*) juga dibandingkan dengan GameObject Unity agar simulasi/program yang dijalankan lebih optimal.
4. Unity DOTS yang peneliti gunakan saat ini merupakan versi preview yang masih dikembangkan oleh pihak Unity, diharapkan ketika kedepannya jika sudah terdapat versi rilis maka akan lebih optimal jika diterapkan ke simulasi yang serupa.
5. Simulasi ini masih memiliki kekurangan dalam menampilkan beberapa elemen yang dapat menambahkan realisme seperti halnya mensimulasikan angin, kapal dll sehingga kedepannya dapat ditambahkan.

DAFTAR PUSTAKA

Borufka, Roman. 2020. *Performance Testing Suite for Unity DOTS*. Prague: Charles University

Turpeinen, Max. 2020. *A Performance Comparison for 3D Crowd Rendering using an Object-Oriented System and Unity DOTS with GPU Instancing on Mobile Devices*. Sweden : KTH Royal Institute of Technology

Näykki, Alarik. 2021. *Unity DOTS in Production : DOTS Pathfinding implementation in VR & AR*. Turki: Turku University of Applied Sciences

Perlin, Ken. 2002. *Improving Noise*. New York : New York University

Lagae, A., S. Lefebvre et, al. 1981.*A Survey of Procedural Noise Functions*. Journal Compilation, Vol. 0, Number 0 pp. 1 - 20

Nugroho, Fresy, Eko Mulyanto Yuniarno, Mochamad Hariadi. 2021. *An Environmental Domain Awareness for Serious-Game Using Perlin Noise Base Heterogenous Haze Visualization.* International Journal of Intelligent Engineering System, Vol. 15, No. 2

Kopecky II, Kenneth Edward. 2007. *Real-time water simulation and rendering using features of the latest OpenGL-capable graphics hardware*. Master Thesis: Iowa State University, Capstones

Li, Hua, Huamin Yang, Jianphing Zhao. 2017. *Water Reflection, Refraction Simulation Based on Perlin Noise and Ray Tracing*. International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 10, No. 3, pp. 63 - 74

Tian, Li. 2014. *Ocean wave simulation by the mix of FFT and Perlin Noise*. Beijing : Institute of Literature, Chinese Academy of Social Sciences

Cahyani, Berlian Gita. 2020. Implementasi Perlin Noise Pada Simulasi Kabut Heterogen Gunung Kelud. Malang : Universitas Islam Negeri Maulana Malik Ibrahim

Wijaya, Wandi, Abdul Rahman. 2018. Analisis Perbandingan Perlin Noise dan Simplex Noise Untuk Penciptaan Permukaan Daratan Pada Pembuatan Game. Pangkalpinang : STMIK Atma Luhur Pangkalpinang

Mulya, Megah, Abdiansyah. 2013. Penerapan Multi-threading untuk Meningkatkan Kinerja Pengolahan Citra Digital. Jurnal Generic : Vol. 8, No. 2, pp. 230-237

Medvecký-Heretik, Bc. Jakub. 2018. *Real-time Water Simulation in Game Environment*. Master Thesis : Masaryk University, Czechia

Hyttinen, Tuomo. 2017. *Terrain Synthesis Using Noise*.Master Thesis: University

of Tampere, Finland

J. Rusnell, Brennan.*Water Drops on Surfaces*. West Canada: University of Sasketchewan

Tecuci, George. 2012. *Artificial Intelligence*. WIREs Computational Statistics: Vol. 4, Issue 2, pp. 168 - 180

Koulaxidis, Georgios, Stelios Xinogalos. 2022. *Improving Mobile Game Performance with Basic Optimization Techniques in Unity*. MDPI: Vol.3, pp. 201 - 223

Kim, Jung Yoon. 2018. *Comparison of Flocing Algorithm According to Number of Boids in Virtual Reality Environment*. Advance Science and Technology Letters, Vol. 150, pp. 330 - 333

Lindqvist, Sebastian. 2018. *Performance Evaluation of Boids on the GPU and CPU*. Sweden : Blekinge Institute of Technology

Alaliyat, Saleh, Filippo Sanfilippo, Harald Yndestald. 2014. *Optimisation of Boids Swarm Model Based on Genetic Algorithm And Particle Swarm Optimisation Algorithm (Comparative Study)*. Conference: Proceedings of the 28th European Conference on Modelling and Simulation (ECMS)

Code Monkey. (2020, Januari 27). *Pathfinding in Unity DOTS*. Youtube.com. https://www.youtube.com/watch?v=1bO1FdEThnU

Venkat, Ajay. (2020, Agustus 19). *Unity Job System and Burst Compiler: Getting Started. raywenderlich.com. https://www.raywenderlich.com/7880445-unity-job-system-and-burst-compiler-getting-started*

Coding, Catlike. *Noise, being a pseudorandom artist.* catlikecoding.com. https://catlikecoding.com/unity/tutorials/noise/

Unity Technologies. (2018, Maret 27). *C# Job System Overview*. docs.unity3d.com. https://docs.unity3d.com/Manual/JobSystem.html

Unity Technologies. *Burst User* Guide. docs.unity3d.com. https://docs.unity3d.com/Packages/com.unity.burst@0.2/manual/index.html

Unity Technologies. *What is DOTS and Why is it important?*. learn.unity.com. https://learn.unity.com/tutorial/what-is-dots-and-why-is-it-important