

**VISUALISASI PERGERAKAN KAWANAN BURUNG
DENGAN ALGORITMA BOIDS MENGGUNAKAN
GAME ENGINE UNITY
SKRIPSI**



diajukan oleh
MUH. FAUZZI RAMADHANY SUROPATY
18.11.1971

Kepada
**PROGRAM SARJANA
PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2022**

**VISUALISASI PERGERAKAN KAWANAN BURUNG
DENGAN ALGORITMA BOIDS MENGGUNAKAN
GAME ENGINE UNITY**

SKRIPSI

untuk memenuhi salah satu syarat mencapai derajat Sarjana
Program Studi Informatika



diajukan oleh

MUH. FAUZZI RAMADHANY SUROPATY
18.11.1971

Kepada

PROGRAM SARJANA
PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2022

HALAMAN PERSETUJUAN

SKRIPSI

**VISUALISASI PERGERAKAN KAWANAN BURUNG
DENGAN ALGORITMA BOIDS MENGGUNAKAN
GAME ENGINE UNITY**

yang disusun dan diajukan oleh

Muh. Fauzzi Ramadhany Suropaty

18.11.1971

telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 30 Oktober 2021

Dosen Pembimbing

Bayu Setiaji M.Kom

NIK. 190302216

HALAMAN PENGESAHAN

SKRIPSI

**VISUALISASI PERGERAKAN KAWANAN BURUNG
DENGAN ALGORITMA BOIDS MENGGUNAKAN
GAME ENGINE UNITY**

yang disusun dan diajukan oleh

Muh. Fauzzi Ramadhany Suropaty

18.11.1971

Telah dipertahankan di depan Dewan Penguji
pada tanggal 20 Juni 2022

Susunan Dewan Penguji

Nama Penguji

Tanda Tangan

Bayu Setiaji, M.Kom

NIK. 190302216

Ika Nur Fajri, M.Kom

NIK. 190302268

Anggit Ferdita Nugraha, S.T., M. Eng

NIK. 190302480

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer
Tanggal 20 Juni 2022

DEKAN FAKULTAS ILMU KOMPUTER

Hanif Al Fatta, S.Kom., M.Kom.

NIK. 190302096

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Yang bertandatangan di bawah ini,

Nama mahasiswa : Muh. Fauzzi Ramadhany Suropaty
NIM : 18.11.1971

Menyatakan bahwa Skripsi dengan judul berikut:

Visualisasi Pergerakan Kawanan Burung dengan Algoritma Boids menggunakan Game Engine Unity

Dosen Pembimbing : Bayu Setiaji, M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 20 Juni 2022

Yang Menyatakan,



Muh. Fauzzi Ramadhany Suropaty

HALAMAN PERSEMBAHAN

Alhamdulillah skripsi ini dapat diselesaikan berkat bantuan dari berbagai pihak yang turut serta berbagi doa dan dukungan. Untuk itu semua, saya persembahkan skripsi ini dan berterima kasih kepada :

1. Segala Puji dan semua puji – pujian hanya untuk dan milik Allah Subhanahu Wa Ta'Ala semata yang memberikan rahmat, petunjuk serta kemudahan.
2. Untuk Ummi, Abi, kakak, dan dua adik kandung saya, semoga Allah selalu melindungi mereka semua dalam hidayah dan rahmatnya.
3. Untuk keluarga di Delanggu (Bukde, Pakde, Mbak Uwik, Mbak Yuyun, Teddy, Hendra, Mbah Kung, Bulek Watik) yang sudah banyak saya repotkan dan berkenan menampung saya selama studi, Terima kasih banyak semoga Allah membalas semua kebaikan kalian.
4. Dosen pembimbing Bapak Bayu Setiaji M.Kom yang memberikan arahan dan menyempatkan waktunya untuk membimbing saya.
5. Untuk Reza dan keluarga yang membolehkan saya menyewa salah satu kamar dirumahnya untuk saya tinggal selama studi dan membantu saya dalam mengerjakan skripsi ini, terima kasih banyak, semoga Allah membalas kebaikan kalian.
6. Untuk Arya yang selalu membantu dalam pengerjaan skripsi ini, terima kasih banyak.
7. Untuk Rizki, Tegar, Dimas, Ilmi, Agus dan teman – teman kelas 18-IF3 lainnya yang membantu dan mendukung pengerjaan skripsi ini.
8. Untuk Arman, Lihin, Egan, Randy, Jihan, Iki, Rafly, terima kasih banyak atas dukungan dan do'anya.
9. Terima kasih banyak kepada semua pihak yang membantu dan mendukung saya selama ini, Semoga Allah memberikan kebaikan untuk kalian semua.

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah Subhanallahu Wa Ta'ala atas limpahan rahmat dan hidayah nya sehingga penulis dapat menyelesaikan skripsi yang berjudul

“VISUALISASI PERGERAKAN KAWANAN BURUNG DENGAN ALGORITMA BOIDS MENGGUNAKAN GAME ENGINE UNITY”

Di dalam tulisan ini disajikan pokok – pokok pembahasan yang meliputi pendahuluan, landasan teori, perancangan sistem, implementasi, dan pengujian sistem. Peneliti menyadari sepenuhnya bahwa dalam penulisan skripsi ini masih banyak kekurangan dan keterbatasan. Oleh karena itu peneliti mengharapkan saran yang membangun agar tulisan ini bermanfaat bagii perkembangan ilmu pengetahuan ke depan.

DAFTAR ISI

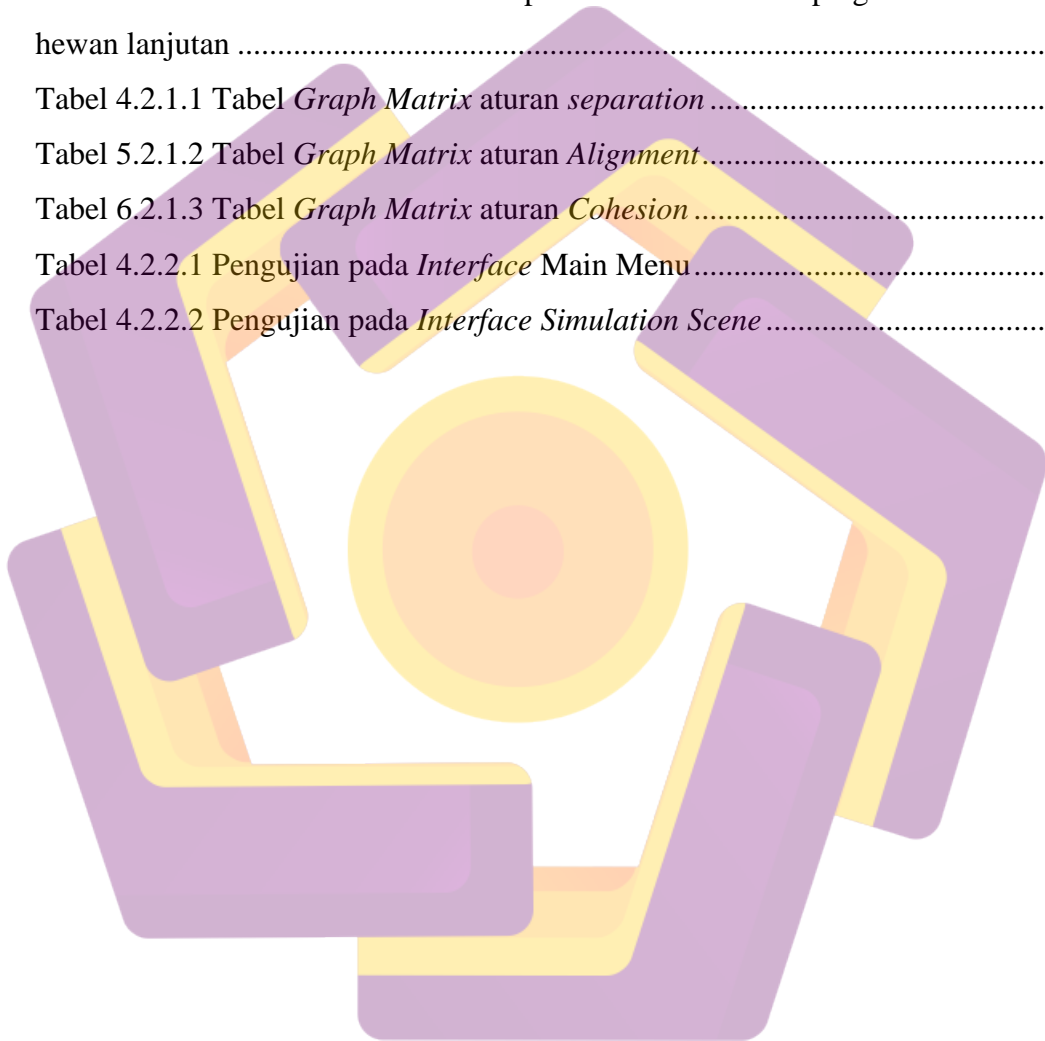
HALAMAN PERSETUJUAN SKRIPSI.....	i
HALAMAN PENGESAHAN SKRIPSI.....	ii
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMBANG DAN SINGKATAN	xiii
DAFTAR ISTILAH	xiv
INTISARI.....	xvii
ABSTRACT.....	xviii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	1
1.3. Batasan Masalah.....	2
1.4. Maksud dan Tujuan Penelitian.....	2
1.5. Manfaat Penelitian.....	2
1.6. Metode Penelitian.....	3
1.6.1. Metode Pengumpulan Data.....	3
1.6.2. Metode Analisis	3
1.6.3. Metode Perancangan	3
1.6.4. Metode Implementasi.....	3
1.6.5. Metode Pengujian.....	3

1.6.6.	Penulisan Laporan	3
1.7.	Sistematika Penulisan	4
1.7.1.	BAB I PENDAHULUAN	4
1.7.2.	BAB II LANDASAN TEORI	4
1.7.3.	BAB III ANALISIS DAN PERANCANGAN	4
1.7.4.	BAB IV IMPLEMENTASI DAN PEMBAHASAN	4
1.7.5.	BAB V PENUTUP	4
BAB II	LANDASAN TEORI	5
2.1	Tinjauan Pustaka	5
2.2	Dasar Teori	6
2.2.1	<i>Artificial Life</i>	6
2.2.2	Algoritma Boid	6
2.2.3	<i>Game Engine Unity</i>	8
2.2.4	Bahasa Pemrograman C#	8
2.2.5	Simulasi	8
2.2.6	<i>Swarm Intelligence</i>	9
2.2.7	<i>Artificial Intelligence</i>	9
BAB III	ANALISIS DAN PERANCANGAN	10
3.1.	Gambaran Umum	10
3.2.	Analisis Kebutuhan Fungsional	10
3.3.	Analisis Kebutuhan Non-Fungsional	10
3.4.	Pemodelan untuk Simulasi	11
3.4.1.	<i>Model Behaviour</i>	11
3.4.2.	UML (<i>Unified Manipulation Language</i>)	14
3.4.3.	Rancangan <i>Interface</i>	42

BAB IV IMPLEMENTASI DAN PEMBAHASAN	44
4.1 Implementasi Sistem Simulasi	44
4.1.1 Implementasi Algoritma.....	44
4.1.2 Implementasi Interface.....	46
4.2 Pengujian Sistem Simulasi	47
4.2.1 Pengujian <i>Algoritma</i>	47
4.2.2 Pengujian <i>Interface</i>	53
4.2.3 Pengujian Algoritma <i>Boids</i> terhadap pergerakan burung	54
BAB V PENUTUP.....	57
5.1. Kesimpulan.....	57
5.2. Saran	57
DAFTAR PUSTAKA	58

DAFTAR TABEL

Tabel 2.1.1 Perbandingan Penelitian.....	6
Tabel 3.4.2.1.2.1 Definisi Use Case pada sistem simulasi pergerakan kawanan hewan	15
Tabel 3.4.2.1.2.2 Definisi Use Case pada sistem simulasi pergerakan kawanan hewan lanjutan	16
Tabel 4.2.1.1 Tabel <i>Graph Matrix</i> aturan <i>separation</i>	49
Tabel 5.2.1.2 Tabel <i>Graph Matrix</i> aturan <i>Alignment</i>	51
Tabel 6.2.1.3 Tabel <i>Graph Matrix</i> aturan <i>Cohesion</i>	53
Tabel 4.2.2.1 Pengujian pada <i>Interface</i> Main Menu.....	53
Tabel 4.2.2.2 Pengujian pada <i>Interface Simulation Scene</i>	54



DAFTAR GAMBAR

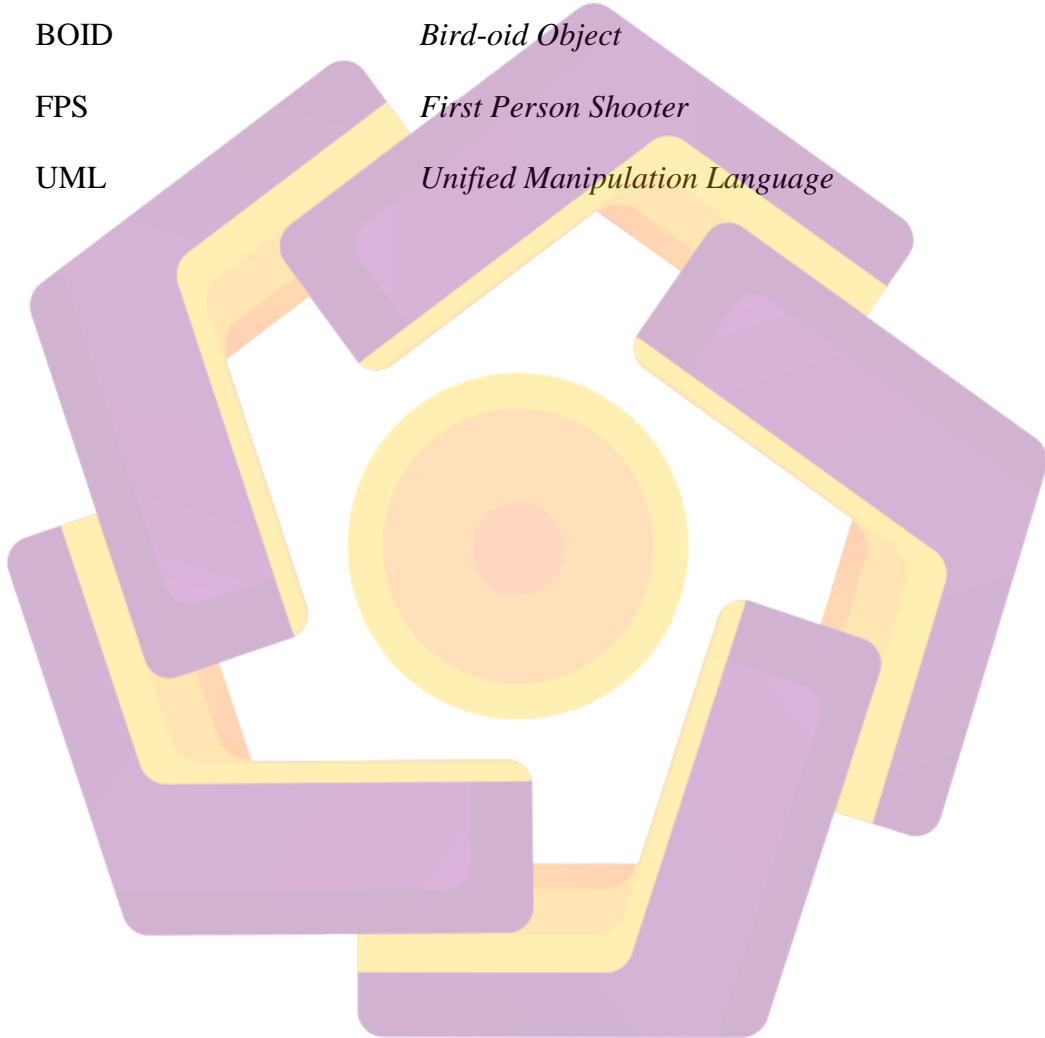
Gambar 2.2.2.1 Gambaran Kohesi.....	7
Gambar 2.2.2.2 Gambaran Pemisahan.....	7
Gambar 2.2.2.3 Gambaran Penjajaran	7
Gambar 3.4.2.1.4 Usecase Diagram Sistem Simulasi Pergerakan Kawanan Hewan	14
Gambar 3.4.2.2.1 Class Diagram Sistem Simulasi Pergerakan Kawanan Hewan	17
Gambar 3.4.2.3.1 <i>Activity Diagram</i> pengguna melakukan simulasi.....	18
Gambar 3.4.2.3.2 <i>Activity Diagram</i> pengguna mengubah nilai kecepatan minimum	18
Gambar 3.4.2.3.3 <i>Activity Diagram</i> pengguna mengubah nilai kecepatan maksimum.....	19
Gambar 3.4.2.3.4 <i>Activity Diagram</i> pengguna mengubah nilai kecepatan maksimum.....	19
Gambar 3.4.2.3.5 <i>Activity Diagram</i> pengguna mengubah nilai kekuatan kemudi maksimum.....	20
Gambar 3.4.2.3.6 <i>Activity Diagram</i> pengguna mengubah nilai radius kawanan..	20
Gambar 3.4.2.3.7 <i>Activity Diagram</i> pengguna mengubah nilai jarak pandang	21
Gambar 3.4.2.3.8 <i>Activity Diagram</i> pengguna mengubah nilai berat kohesi	21
Gambar 3.4.2.3.9 <i>Activity Diagram</i> pengguna mengubah nilai berat penjajaran .	22
Gambar 3.4.2.3.10 <i>Activity Diagram</i> pengguna mengubah nilai berat pemisahan	22
Gambar 3.4.2.3.11 <i>Activity Diagram</i> pengguna mengubah nilai berat menghindari tabrakan.....	23
Gambar 3.4.2.3.12 <i>Activity Diagram</i> pengguna mengubah nilai berat menghindari tabrakan.....	23
Gambar 3.4.2.3.13 <i>Activity Diagram</i> pengguna mengubah nilai berat menghindari tabrakan.....	24
Gambar 3.4.2.3.14 <i>Activity Diagram</i> pengguna mengubah jumlah entitas yang akan dimunculkan.....	24
Gambar 3.4.2.3.15 <i>Activity Diagram</i> pengguna mengubah warna entitas.....	25

Gambar 3.4.2.3.16 <i>Activity Diagram</i> pengguna mengubah nilai radius tetangga.	25
Gambar 3.4.2.3.17 <i>Activity Diagram</i> pengguna menampilkan visualisasi kekuatan pemisahan.....	26
Gambar 3.4.2.3.18 <i>Activity Diagram</i> pengguna menampilkan visualisasi kekuatan penjaran	27
Gambar 3.4.2.3.19 <i>Activity Diagram</i> pengguna menampilkan visualisasi kekuatan kohesi	27
Gambar 3.4.2.3.20 <i>Activity Diagram</i> pengguna menampilkan visualisasi Neighbours	28
Gambar 3.4.2.4.1 <i>Sequence diagram</i> pengguna memulai simulasi	29
Gambar 3.4.2.4.2 <i>Sequence diagram</i> pengguna mengubah kecepatan minimum	29
Gambar 3.4.2.4.3 <i>Sequence diagram</i> pengguna mengubah kecepatan maksimum	30
Gambar 3.4.2.4.4 <i>Sequence diagram</i> pengguna mengubah jarak pemisah.....	31
Gambar 3.4.2.4.5 <i>Sequence diagram</i> pengguna memulai simulasi	31
Gambar 3.4.2.4.6 <i>Sequence diagram</i> pengguna mengubah radius tetangga.....	32
Gambar 3.4.2.4.7 <i>Sequence diagram</i> pengguna mengubah jarak pandang	33
Gambar 3.4.2.4.8 <i>Sequence diagram</i> pengguna mengubah radius tetangga.....	33
Gambar 3.4.2.4.9 <i>Sequence diagram</i> pengguna mengubah berat pemisahan.....	34
Gambar 3.4.2.4.10 <i>Sequence diagram</i> pengguna mengubah berat pemisahan	35
Gambar 3.4.2.4.11 <i>Sequence diagram</i> pengguna mengubah berat menghindari tabrakan	35
Gambar 3.4.2.4.12 <i>Sequence diagram</i> pengguna mengubah jarak raycast.....	36
Gambar 3.4.2.4.13 <i>Sequence diagram</i> pengguna mengubah radius tetangga.....	37
Gambar 3.4.2.4.14 <i>Sequence diagram</i> pengguna mengisi nilai entitas	37
Gambar 3.4.2.4.15 <i>Sequence diagram</i> pengguna mengubah warna entitas.....	38
Gambar 3.4.2.4.16 <i>Sequence diagram</i> pengguna mengubah radius entitas.....	39
Gambar 3.4.2.4.17 <i>Sequence diagram</i> pengguna menampilkan visualisasi kekuatan pemisahan.....	39
Gambar 3.4.2.4.18 <i>Sequence diagram</i> pengguna menampilkan visualisasi kekuatan penjaran	40

Gambar 3.4.2.4.19 <i>Sequence diagram</i> pengguna menampilkan visualisasi kekuatan kohesi	41
Gambar 3.4.2.4.20 <i>Sequence diagram</i> pengguna menampilkan visualisasi kawanannya	41
Gambar 3.4.3.1 Rancangan antarmuka <i>scene</i> Main Menu.....	42
Gambar 3.4.3.2 Rancangan antarmuka <i>Simulation Scene</i> dengan panel menu terbuka.....	43
Gambar 3.4.3.3 Rancangan antarmuka <i>Simulation Scene</i> dengan panel menu tertutup	43
Gambar 4.1.1.1 Fungsi aturan <i>separation</i> / pemisahan.....	44
Gambar 4.1.1.2 Fungsi aturan <i>alignment</i> / penjajaran	45
Gambar 4.1.1.3 Fungsi aturan <i>cohesion</i> / kohesi	45
Gambar 4.1.2.1 <i>Interface</i> pada Main Menu <i>Scene</i>	46
Gambar 4.1.2.2 <i>Interface</i> pada <i>Simulation Scene</i>	47
Gambar 4.2.1.1 <i>Flow Graph</i> aturan <i>separation</i>	47
Gambar 4.2.1.2 Penyederhanaan <i>Flow Graph</i> aturan <i>separation</i>	48
Gambar 4.2.1.3 <i>Flow Graph</i> aturan <i>alignment</i>	50
Gambar 4.2.1.4 <i>Flow Graph</i> aturan <i>cohesion</i>	51
Gambar 4.2.3.1 Pengujian Aturan <i>Separation</i>	55
Gambar 4.2.3.2 Pengujian aturan <i>alignment</i>	55
Gambar 4.2.3.3 Pengujian aturan <i>cohesion</i>	56

DAFTAR LAMBANG DAN SINGKATAN

AI	<i>Artificial Intelligence</i>
CPU	<i>Central Processing Unit</i>
GPU	<i>Graphics Processing Unit</i>
BOID	<i>Bird-oid Object</i>
FPS	<i>First Person Shooter</i>
UML	<i>Unified Manipulation Language</i>



DAFTAR ISTILAH

AI (*Artificial Intelligence*)

Kecerdasan buatan yang dibuat oleh manusia yang digunakan untuk menyelesaikan suatu masalah.

Algoritma

Prosedur sistematis untuk memecahkan masalah matematis dalam langkah-langkah terbatas.

Simulasi

Metode pelatihan yang meragakan sesuatu dalam bentuk tiruan yang mirip dengan keadaan yang sesungguhnya

Swarm Intelligence

Sejenis kemampuan pemecahan masalah yang muncul dalam interaksi unit pemrosesan informasi sederhana

Game Engine

Perangkat lunak yang terutama dirancang untuk pengembangan video game, dan umumnya mencakup *library* dan program pendukung yang relevan.

Flocking

Kumpulan orang (binatang) yang berkawan.

Model Behaviour

Penggambaran perilaku keseluruhan sistem.

UML (*Unified Manipulation Language*)

Bahasa visual yang menyediakan cara bagi pengembang perangkat lunak untuk membangun, mendokumentasikan dan memvisualisasikan sistem perangkat lunak.

Interface / Antarmuka

Perangkat lunak yang memungkinkan program untuk bekerja dengan pengguna

Use Case Diagram

Diagram ini menunjukkan bagian-bagian dari sistem atau fungsionalitas suatu sistem dan bagaimana mereka berhubungan satu sama lain. Ini memberi pengembang pemahaman yang jelas tentang bagaimana berbagai hal berfungsi tanpa perlu melihat detail implementasi.

Class Diagram

Diagram kelas adalah prinsip dasar dari sistem perangkat lunak berorientasi objek. Kelas-kelas dalam suatu sistem atau operasi menggambarkan strukturnya, yang membantu pengembang perangkat lunak mengidentifikasi hubungan antara setiap objek.

Activity Diagram

Diagram ini menggambarkan aliran kontrol dalam suatu sistem. Pengembang perangkat lunak menggunakan diagram aktivitas sebagai referensi untuk mengikuti langkah-langkah saat menjalankan *Use Case Diagram*. Diagram aktivitas juga dapat menggambarkan penyebab dari suatu peristiwa tertentu.

Sequence Diagram

Diagram ini menunjukkan bagaimana objek berkomunikasi satu sama lain secara berurutan. Ini biasanya digunakan oleh pengembang perangkat lunak untuk mendokumentasikan dan memahami persyaratan yang diperlukan untuk membangun sistem baru atau mendapatkan lebih banyak pengetahuan tentang sistem yang ada.

Whitebox Test

Pengujian yang dilakukan untuk menguji perangkat lunak dengan cara menganalisa dan meneliti struktur internal dan kode dari perangkat lunak.

Blackbox Test

Pengujian yang dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik.

Sistem

Perangkat unsur yang secara teratur saling berkaitan sehingga membentuk suatu totalitas.

Implementasi

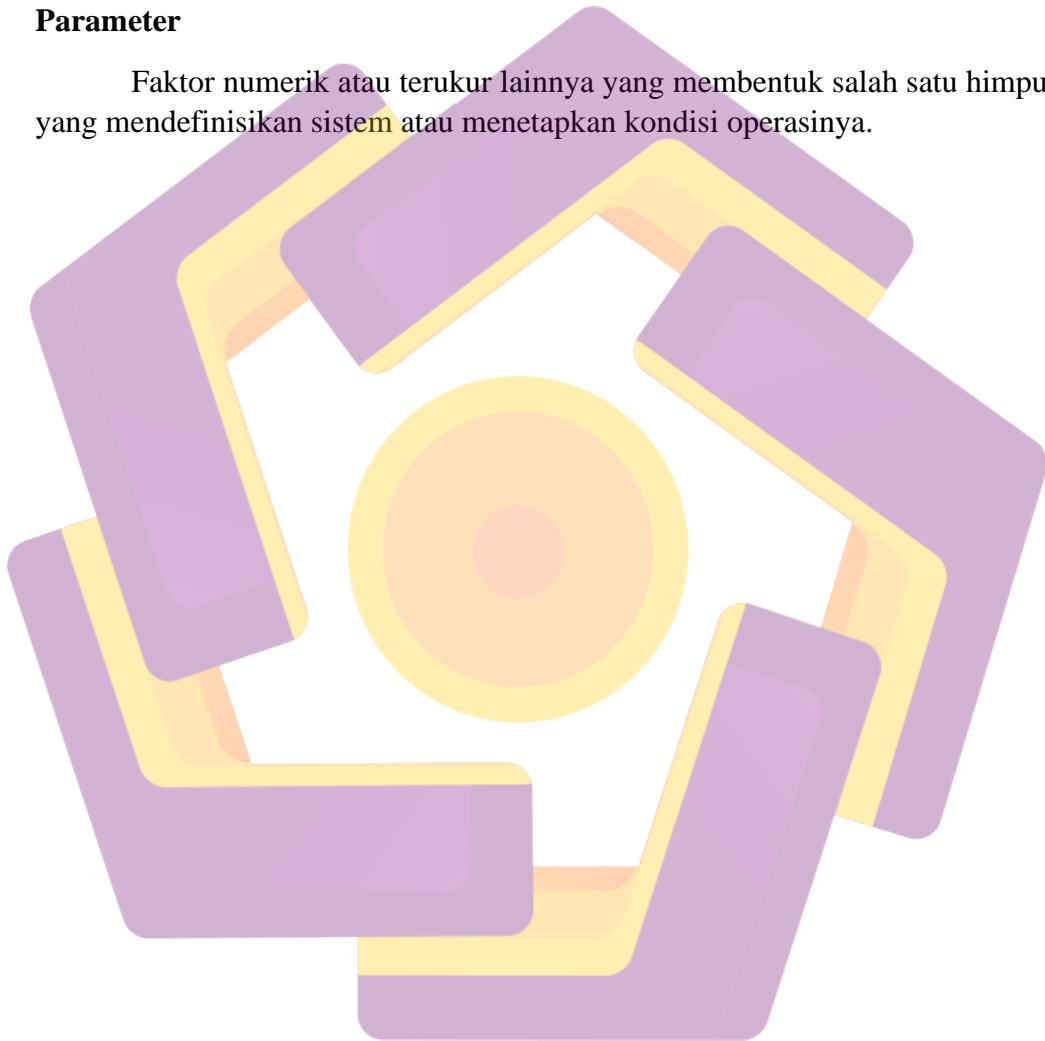
Pengembangan versi kerja sistem dari desain yang diberikan.

Aplikasi

Program komputer atau perangkat lunak yang didesain untuk mengerjakan tugas tertentu

Parameter

Faktor numerik atau terukur lainnya yang membentuk salah satu himpunan yang mendefinisikan sistem atau menetapkan kondisi operasinya.



INTISARI

Artificial Intelligence atau kecerdasan buatan semakin gencar di buat untuk berbagai macam kebutuhan manusia, video game dan film pun terkena dampak ini dimana studio game berlomba membuat game yang semakin realistis untuk mendapatkan efek imersif dan studio animasi film pun berlomba membuat animasi yang semakin realistis dan natural yang dapat dinikmati banyak kalangan. Salah satu contohnya yaitu membuat pergerakan sebuah gerombolan atau kawanan, yang jika dilakukan secara manual akan sangat memakan waktu dan tidak efisien, Salah satu algoritma yang dapat membantu masalah ini yaitu algoritma boids.

Algoritma boids adalah algoritma yang dikembangkan oleh Craig W. Reynolds untuk mensimulasikan pergerakan kawanan burung, boid adalah kepanjangan dari *bird-oid object* atau objek yang seperti burung, algoritma ini memiliki aturan dasar yaitu pemisahan (*separation*), penjajaran (*alignment*), dan Kohesi (*Cohesion*). aturan yang lebih rumit juga dapat ditambahkan seperti menghindari halangan atau mengejar tujuan,

Algoritma kawanan ini sangat berguna sebagai simulasi pergerakan banyak objek secara bersamaan karena dapat memperlihatkan pergerakan yang cukup realistis, baik kawanan burung, ikan, dan lain sebagainya. algoritma ini dapat diterapkan di berbagai media seperti animasi atau film, game, dan program sederhana. Sudah banyak animasi dan game yang menggunakan algoritma ini seperti *Batman Returns* (1992) dan pada game *Half-Life* (1998). Visualisasi pergerakan burung pada studi ini akan menggunakan game engine yaitu Unity yang cukup ramai digunakan untuk berbagai macam kebutuhan, seperti game, simulasi, animasi dan lain sebagainya.

Kata kunci: **Algoritma Boids, Visualisasi Pergerakan, Simulasi, Unity.**

ABSTRACT

AI or artificial intelligence is increasingly being made for various kinds of human needs, video games and films are also affected by this where game studios are competing to make more realistic games to get immersive effects and film animation studios are also competing to make animations that are more realistic and natural. That can be enjoyed by many people. One example is making the movement of a flock or school, which if done manually would be very time consuming and inefficient. One algorithm that can help with this problem is the boids algorithm.

Boid Algorithm developed by Craig W. Reynolds to simulate movements of flock of bird, boid are shortened version of Bird-oid Object or bird-like object, this algorithm have some basic rules such as separation, alignment, and cohesion. More complex rule can be added such as obstacle avoidance or target chasing,

This flocking algorithm is very useful for simulate movements of many objects at once because it can represent quite realistic movements, both flock of birds, school of fish, and so on. This algorithm can be implemented into varies of media, such as animation or film, game, and simple program. There have been many animation and games that use this algorithm, such as Batman Returns (1992) and the game Half-Life (1992). Visualization movements of flock of bird in this study would use a game engine, Unity which quite popular for many uses such as game, simulation, animation, and et cetera.

Keywords: Boids Algorithm, Movements Visualization, Simulation, Unity.