

PENGEMBANGAN *GAME* PERTANIAN MENGGUNAKAN *FINITE STATE MACHINE* DAN *FUZZY LOGIC*

SKRIPSI

**Oleh:
ACH MASHDUQ WA'DAN
16650123**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

PENGEMBANGAN GAME PERTANIAN MENGGUNAKAN *FINITE STATE MACHINE* DAN *FUZZY LOGIC*

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
ACH MASHDUQ WA'DAN
NIM. 16650123

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

HALAMAN PERSETUJUAN

PENGEMBANGAN GAME PERTANIAN MENGGUNAKAN *FINITE STATE MACHINE* DAN *FUZZY LOGIC*

SKRIPSI

Oleh:
ACH MASHDUQ WA'DAN
NIM. 16650123

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 26 Juni 2023

Pembimbing I



Dr. Fachrul Kurniawan, M.MT., IPM
NIP. 19771020 200912 1 001


Pembimbing II



Dr. Yunifa Miftachul Arif, M.T
NIP. 19830616 201101 1 004

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrul Kurniawan ST., M.MT., IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

PENGEMBANGAN GAME PERTANIAN MENGGUNAKAN *FINITE STATE MACHINE* DAN *FUZZY LOGIC*


SKRIPSI

Oleh:
ACH MASHDUQ WA'DAN
NIM. 16650123

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 26 Juni 2023

Susunan Dewan Penguji

- Ketua Penguji : Dr. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007
- Anggota penguji I : Dr. Fresy Nugroho, M.T
NIP. 19710722 201101 1 001
- Anggota Penguji II : Dr. Fachrul Kurniawan, M.MT., IPM
NIP. 19771020 200912 1 001
- Anggota Penguji III : Dr. Yunifa Miftachul Arif, M. T
NIP. 19830616 201101 1 004



Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Fachrul Kurniawan ST., M.MT., IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Ach Mashduq Wa'dan

NIM : 16650123

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Skripsi : Pengembangan *Game* Pertanian Menggunakan *Finite State Machine* dan *Fuzzy Logic*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-banar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 26 Juni 2023

Yang membuat pernyataan,



Ach Mashduq Wa'dan

NIM. 16650123

MOTTO

Telat bukan berarti lambat, gagal bukan berarti bodoh, tak bergerak bukan berarti malas. Hidup tak harus sama, lakukan yang terbaik dengan caramu sendiri. Jangan biarkan pencapaian orang lain menghentikan pencapaianmu.

HALAMAN PERSEMBAHAN

Skripsi ini saya persembahkan untuk orang tua, istri dan anak saya tercinta yang tiada lelah memberikan motivasi dan kasih sayangnya kepada saya. Terutama kepada ibu tercinta alm. Husnul Hotimah. Semoga Allah subhanahu wa ta'ala senantiasa memberikan Rahmat dan Maghfiroh-Nya kepada ibu di alam sana.

Inshaallah, kita akan berjumpa kembali di surga-Nya kelak. Aamiin.

Terima kasih

KATA PENGANTAR

Syukur alhamdulillah penulis ucapkan kehadiran Allah SUBHANAHU WA TA'ALA yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan Skripsi ini dengan baik.

Selanjutnya penulis mengucapkan terima kasih kepada semua pihak yang telah membantu terselesaikannya Skripsi ini. Ucapan terima kasih ini penulis sampaikan kepada:

1. Prof. Dr. HM. Zainuddin MA, selaku rektor UIN Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT., selaku Kepala Prodi Teknik Informatika sekaligus dosen pembimbing yang senantiasa mendorong dan memberikan solusi di setiap permasalahan mahasiswanya.
4. Dr. Yunifa Miftchul Arif, M.T, Dr. Muhammad Faisal, M.T dan Dr. Fresy Nugroho, M.T selaku dosen pembimbing dan penguji, yang telah memberikan ilmu pengetahuan, pengarahan, dan motivasi dalam penyelesaian Skripsi ini.
5. Seluruh dosen Teknik Informatika yang telah memberikan ilmu dan pengalaman yang berharga selama masa perkuliahan.

6. Seluruh staf Teknik Informatika yang telah membantu penulis dalam hal administrasi.
7. Orangtua penulis yang senantiasa memberikan kasih sayang, doa, dan dukungan berupa moril maupun materil kepada penulis.
8. Istri dan anak-anak penulis tercinta, Dian Listiyana Ulva, Muhammad Avicenna Zadid Taqwa, dan juga calon buah hati yang lagi di dalam kandungan. Terimakasih telah menemani penulis, memberikan doa, semangat dan keceriaan dalam hidup penulis.
9. Keluarga besar Bani Damhuri, yang selalu memberikan motivasi dan makna pentingnya pendidikan.
10. Teman-teman Teknik Informatika 2016 (Andromeda) yang senantiasa saling memberi semangat dan berjuang Bersama.
11. Semua pihak yang ikut membantu dalam menyelesaikan Skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan Skripsi ini masih terdapat kekurangan dan penulis berharap semoga Skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. Amin Ya Rabbal Alamin.

Wassalamu'alaikum Wr. Wb.

Malang, Juni 2023

Penulis

DAFTAR ISI

| | |
|---|---------------------------------------|
| HALAMAN PENGAJUAN | ii |
| HALAMAN PERSETUJUAN | Kesalahan! Bookmark tidak ditentukan. |
| HALAMAN PENGESAHAN | Kesalahan! Bookmark tidak ditentukan. |
| PERNYATAAN KEASLIAN TULISAN | Kesalahan! Bookmark tidak ditentukan. |
| MOTTO | vi |
| HALAMAN PERSEMBAHAN | vii |
| KATA PENGANTAR | viii |
| DAFTAR ISI | x |
| DAFTAR GAMBAR | xii |
| DAFTAR TABEL | xiv |
| ABSTRAK | xv |
| ABSTRACT | xvi |
| المخلص | Kesalahan! Bookmark tidak ditentukan. |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 4 |
| 1.3 Tujuan Penelitian | 5 |
| 1.4 Manfaat Penelitian | 5 |
| 1.5 Batasan Masalah | 5 |
| 1.6 Sistematika Penulisan | 6 |
| BAB II LANDASAN TEORI | 8 |
| 2.1 <i>Game</i> | 8 |
| 2.1.1 Pengertian <i>Game</i> | 8 |
| 2.1.2 Jenis-Jenis <i>Game</i> | 9 |
| 2.2 Pertanian | 17 |
| 2.3 Software Unity 3D | 18 |
| 2.4 Finite State Machine (FSM) | 20 |
| 2.5 Fuzzy Logic | 21 |
| BAB III DESAIN DAN IMPLEMENTASI | 24 |
| 3.1 Desain <i>Game</i> | 24 |
| 3.1.1 Jenis <i>Game</i> | 24 |
| 3.1.2 <i>Storyline</i> | 24 |
| 3.1.3 Deskripsi Karakter dan Objek | 25 |
| 3.1.4 <i>Storyboard</i> | 27 |
| 3.2 Perancangan Metode | 35 |
| 3.2.1 Perancangan FSM dalam <i>Game</i> | 35 |
| 3.2.2 Perancangan Metode <i>Fuzzy Logic</i> | 38 |
| 3.2 Contoh Perhitungan <i>Fuzzy Sugeno</i> | 49 |
| 3.2.1 Perhitungan Manual | 49 |
| 3.2.2 Perhitungan Matlab | 55 |
| BAB IV HASIL DAN PEMBAHASAN | 61 |
| 4.1 Implementasi Desain Sistem | 61 |

| | |
|--|-----------|
| 4.1.1 Spesifikasi Perangkat Keras (<i>Hardware</i>) | 61 |
| 4.1.2 Spesifikasi Perangkat Lunak (<i>Software</i>)..... | 62 |
| 4.2 Implementasi Permainan..... | 62 |
| 4.2.1 Tampilan Menu utama | 62 |
| 4.2.2 Tampilan Cerita Awal Permainan..... | 64 |
| 4.2.3 Tampilan <i>Gameplay</i> | 64 |
| 4.3 Uji Coba | 70 |
| 4.3.1 Uji Coba FSM Pada NPC | 71 |
| 4.3.2 Uji Coba <i>Fuzzy Sugeno</i> Pada Sistem Tanaman | 76 |
| 4.4 Integrasi Islam..... | 85 |
| BAB V..... | 90 |
| 5.1 Kesimpulan | 90 |
| 5.2 Saran..... | 90 |
| DAFTAR PUSTAKA | |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Contoh diagram state sederhana..... | 20 |
| Gambar 3.1 Storyboard menu utama | 28 |
| Gambar 3.2 Storyboard menu cerita awal <i>game</i> | 30 |
| Gambar 3.3 <i>Storyboard</i> tampilan percakapan NPC kakek Oland dan karakter utama Avice..... | 31 |
| Gambar 3.4 <i>Storyboard</i> tampilan lahan pertanian. | 32 |
| Gambar 3.5 <i>Storyboard</i> tampilan lahan peternakan..... | 33 |
| Gambar 3.6 <i>Storyboard</i> tampilan <i>market</i> | 34 |
| Gambar 3.7 FSM NPC Ayam. | 36 |
| Gambar 3.8 Rancangan FSM NPC Kakek Oland. | 37 |
| Gambar 3.9 Rancangan FSM NPC Om Giant. | 38 |
| Gambar 3.10 Proses <i>fuzzy</i> | 39 |
| Gambar 3.11 Himpunan <i>fuzzy</i> untuk variabel Jenis Bibit. | 40 |
| Gambar 3.12 Himpunan <i>fuzzy</i> untuk variabel Pemupukan. | 41 |
| Gambar 3.13 Himpunan <i>fuzzy</i> untuk variabel Penyiraman. | 43 |
| Gambar 3.14 Matlab: Variabel <i>input</i> dan <i>output</i> pada proses <i>fuzzy</i> | 55 |
| Gambar 3.15 Matlab: Fungsi keanggotaan variabel Jenis Bibit. | 56 |
| Gambar 3.16 Matlab: Fungsi keanggotaan variabel Pemupukan..... | 57 |
| Gambar 3.17 Matlab: Fungsi keanggotaan variabel Penyiraman. | 58 |
| Gambar 3.18 Matlab: Fungsi keanggotaan variabel Hasil Tanaman. | 59 |
| Gambar 3.19 Matlab: <i>Fuzzy rules</i> | 59 |
| Gambar 3.20 Matlab: Hasil perhitungan. | 60 |
| Gambar 3.21 Matlab: Grafik permukaan <i>level</i> | 60 |
| Gambar 4.1 Tampilan Menu utama | 63 |
| Gambar 4.2 Tampilan cerita awal permainan. | 64 |
| Gambar 4.3 Tampilan <i>scene</i> dialog NPC Kakek Oland dan karakter utama..... | 65 |
| Gambar 4.4 Tampilan misi pemain. | 65 |
| Gambar 4.5 Tampilan <i>scene</i> pemilihan lahan dan benih. | 66 |
| Gambar 4.6 Tampilan <i>scene</i> fase penyemprotan pupuk. | 67 |
| Gambar 4.7 Tampilan <i>scene</i> fase penyiraman. | 67 |
| Gambar 4.8 Tampilan <i>scene</i> menunggu tanaman siap panen. | 68 |
| Gambar 4.9 Tampilan <i>scene</i> tanaman siap panen. | 68 |
| Gambar 4.10 Tampilan panel hasil tanaman..... | 69 |
| Gambar 4.11 Tampilan lahan peternakan. | 69 |
| Gambar 4.12 Tampilan ayam sudah menghasilkan telur..... | 70 |
| Gambar 4.13 Tampilan panel dapatkan telur. | 70 |
| Gambar 4.14 <i>Unity: Console</i> perubahan <i>idle state</i> ke <i>wander state</i> | 71 |
| Gambar 4.15 <i>Unity: Console</i> perubahan <i>wander state</i> ke <i>running away state</i> | 72 |
| Gambar 4.16 <i>Unity: Console</i> perubahan <i>Idle State</i> ke <i>RunningAway State</i> | 72 |
| Gambar 4.17 <i>Unity: Console</i> perubahan <i>wander state</i> ke <i>laying-egg state</i> | 72 |
| Gambar 4.18 <i>Unity: Console</i> perubahan <i>state</i> ketika telur dipanen..... | 73 |
| Gambar 4.19 <i>Unity: Console</i> perubahan <i>state</i> ketika <i>player on interaction's collider</i> | 73 |

| | | |
|-------------|--|----|
| Gambar 4.20 | <i>Unity: Console perubahan state ketika NPC on player's collider. .</i> | 74 |
| Gambar 4.21 | <i>Unity: Console perubahan state ketika player out interaction's collider.</i> | 74 |
| Gambar 4.22 | <i>Unity: Console perubahan state ketika player on market's collider.</i> | 74 |
| Gambar 4.23 | <i>Unity: Console perubahan state ketika NPC on player's collider. ...</i> | 75 |
| Gambar 4.24 | <i>Unity: Console perubahan state ketika player out market's collider</i> | 75 |
| Gambar 4.25 | <i>Unity: Console output hasil tanaman percobaan ke-1.</i> | 77 |
| Gambar 4.26 | <i>Unity: Console output hasil tanaman percobaan ke-2.</i> | 77 |
| Gambar 4.27 | <i>Unity: Console output hasil tanaman percobaan ke-3.</i> | 78 |
| Gambar 4.28 | <i>Unity: Console output hasil tanaman percobaan ke-4.</i> | 78 |
| Gambar 4.29 | <i>Unity: Console output hasil tanaman percobaan ke-5.</i> | 79 |
| Gambar 4.30 | <i>Unity: Console output hasil tanaman percobaan ke-6.</i> | 79 |
| Gambar 4.31 | <i>Unity: Console output hasil tanaman percobaan ke-7.</i> | 79 |
| Gambar 4.32 | <i>Unity: Console output hasil tanaman percobaan ke-8.</i> | 80 |
| Gambar 4.33 | <i>Unity: Console output hasil tanaman percobaan ke-9.</i> | 80 |
| Gambar 4.34 | <i>Unity: Console output hasil tanaman percobaan ke-10.</i> | 80 |
| Gambar 4.35 | <i>Unity: Console output hasil tanaman percobaan ke-11.</i> | 81 |
| Gambar 4.36 | <i>Unity: Console output hasil tanaman percobaan ke-12.</i> | 81 |
| Gambar 4.37 | <i>Unity: Console output hasil tanaman percobaan ke-13.</i> | 82 |
| Gambar 4.38 | <i>Unity: Console output hasil tanaman percobaan ke-14.</i> | 82 |
| Gambar 4.39 | <i>Unity: Console output hasil tanaman percobaan ke-15.</i> | 82 |
| Gambar 4.40 | <i>Unity: Console Output Hasil Tanaman Percobaan ke-16</i> | 83 |
| Gambar 4.41 | <i>Unity: Console output hasil tanaman percobaan ke-17.</i> | 83 |
| Gambar 4.42 | <i>Unity: Console output hasil tanaman percobaan ke-18.</i> | 84 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 3.1 Nilai linguistik variabel <i>fuzzy</i> : <i>input</i> dan <i>output</i> | 38 |
| Tabel 3.2 Domain himpunan keanggotaan Jenis Bibit..... | 40 |
| Tabel 3.3 Domain himpunan keanggotaan Pemupukan..... | 41 |
| Tabel 3.4 Domain himpunan keanggotaan Penyiraman | 43 |
| Tabel 4.1 Kebutuhan perangkat keras | 61 |
| Tabel 4.2 Kebutuhan perangkat lunak | 62 |
| Tabel 4.3 Keterangan menu pada Menu utama..... | 63 |
| Tabel 4.4 Hasil uji coba FSM pada NPC | 75 |
| Tabel 4.5 Hasil pengujian algoritme <i>fuzzy logic</i> | 84 |

ABSTRAK

Wa'dan, Ach Mashduq. 2023. **Pengembangan Game Pertanian Menggunakan Finite State Machine dan Fuzzy Logic**. Skripsi. Program Studi Teknik Informatika, Fakultas Sains Dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (I) Dr. Fachrul Kurniawan, M.MT., IPM, Pembimbing (II) Dr. Yunifa Miftachul Arif, M.T

Kata Kunci: *Pertanian, Game, FSM, Fuzzy Logic*.

Pertumbuhan generasi muda dalam sektor pertanian dari tahun 2013 sampai 2018 semakin menurun, hal ini menunjukkan tingkat minat anak muda dalam sektor pertanian sangat rendah. Perlu adanya media pengenalan serta pendekatan yang menghibur dan mudah diakses dalam penyuluhan pertanian, sehingga ke depannya komposisi petani di Indonesia bisa didominasi oleh anak muda. Perkembangan media teknologi saat ini berkembang sangat pesat, salah satunya media hiburan atau *game*. Maka melalui penelitian ini, dikembangkan sebuah *game* pertanian yang interaktif dengan menerapkan *sisitem* kecerdasan buatan di dalamnya menggunakan metode *finite state machine* untuk mendesain perilaku NPC dan *fuzzy logic* untuk mengklasifikasi hasil tanaman. Pada penelitian ini diperoleh hasil yang sangat baik untuk penerapan metode *finite state machine* dan *fuzzy logic* dengan hasil validasi 100% pada uji coba penerapan metode di dalam *game*.

ABSTRACT

Wa'dan, Ach Mashduq. 2023. **Development of Farm Game Using Finite State Machine and Fuzzy Logic**. Theses. Department of Informatics Engineering, Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University of Malang. Advisors: (I) Dr. Fachrul Kurniawan, M.MT., IPM (II) Dr. Yunifa Miftachul Arif, M.T

The growth of the young generation in the agricultural sector from 2013 to 2018 is decreasing, this shows that the level of interest of young people in the agricultural sector is very low. There is a need for introductory media as well as an entertaining and accessible approach in agricultural counseling so that in the future the composition of farmers in Indonesia can be dominated by young people. The development of technological media is currently growing very rapidly, one of which is entertainment media or *games*. So through this research, an interactive agricultural *game* was developed by implementing an artificial intelligence system in it using the finite state machine method to design NPC behavior and *fuzzy* logic to classify crop yields. In this study, very good results were obtained for the application of the finite state machine and *fuzzy* logic methods with 100% validation results in the trial application of the method in the *game*.

Key-words: *Farm, Game, FSM, Fuzzy Logic*

الملخص

وعدان، احمد مصدوق. ٢٠٢٣. تطوير لعبة الزراعة باستخدام FSM و منطق فوزي قسم الهندسة المعلوماتية. كلية العلوم والتكنولوجيا. جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (١) فهروول كرنياون، الماجستير، (٢) يونيفا مفتاح العارف الماجستير.

الكلمات المفتاحية : الزراعة، ألعاب، FSM، المنطق الفوزي

انخفض نمو جيل الشباب في القطاع الزراعي من 2013 إلى 2018، وهذا يدل على أن مستوى اهتمام الشباب بالقطاع الزراعي منخفض للغاية. يجب أن تكون هناك وسائط تمهيدية بالإضافة إلى مناهج مسلية ويمكن الوصول إليها بسهولة في الإرشاد الزراعي، بحيث يمكن للشباب في المستقبل أن يهيمن على تركيبة المزارعين في إندونيسيا. ينمو تطوير الوسائط التكنولوجية حالياً بسرعة كبيرة، أحدها هو الترفيه أو وسائط الألعاب. لذلك من خلال هذا البحث، تم تطوير لعبة زراعية تفاعلية من خلال تطبيق نظام ذكاء اصطناعي فيه باستخدام طريقة FSM لتصميم سلوك NPC ومنطق فوزي لتصنيف غلة المحاصيل. في هذه الدراسة، تم الحصول على نتائج جيدة جداً لتطبيق FSM وطرق المنطق الفوزي مع نتائج التحقق بنسبة 100٪ في تجارب تنفيذ الطريقة في الألعاب.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan negara dengan sumber daya alam yang sangat melimpah, terutama pada sektor pertanian. Secara khusus, sektor pertanian memberikan kontribusi terhadap Produk Domestik Bruto (PDB), menjamin pasokan pangan dalam negeri dan menciptakan lapangan kerja, menjadikannya sebagai sektor kritis yang sangat menentukan keberadaan masyarakat dan perekonomian nasional (Julius, 2015). Namun, pertanian di Indonesia masih memiliki masalah yang perlu dibenahi salah satunya di bidang tenaga kerja. Di sisi tenaga kerja, generasi yang lebih tua masih menjadi mayoritas petani Indonesia, dengan rata-rata usia di atas 45 tahun. Berdasarkan data SUTAS (2018), proporsi petani muda (berusia antara 25 dan 44 tahun) di Indonesia secara keseluruhan populasi petani telah menurun sekitar 3% antara tahun 2013 dan 2018. Sementara itu, persentase petani lanjut usia (yang berusia antara 45 dan 65 tahun) meningkat hampir 3% antara tahun 2013 dan 2018.

Hal ini menunjukkan bahwa masih rendahnya minat anak muda terhadap sektor pertanian sehingga menjadi permasalahan yang perlu dibenahi dalam hal regenerasi. Program penyuluhan pertanian yang dilakukan pemerintah masih sebatas materi serta pembinaan formal yang tidak interaktif sehingga kurang menarik perhatian minat anak muda. Perlu adanya media pengenalan serta pendekatan yang menghibur dan mudah diakses dalam penyuluhan pertanian,

sehingga ke depannya komposisi petani di Indonesia bisa didominasi oleh anak muda.

Perkembangan media teknologi saat ini berkembang dengan pesat, di antaranya yaitu media hiburan atau *game*. Dunia virtual baru yang terus berkembang menyerupai kehidupan nyata disediakan oleh industri *game* yang semakin canggih. (Rahadian, Suyatno, & Maharani, 2016). Hampir semua sektor kehidupan dapat disimulasikan dalam bentuk visual digital, termasuk sektor pertanian yang dapat disimulasikan dalam bentuk *game*. Salah satu aplikasi yang menawarkan hiburan berupa tantangan dan kenikmatan adalah *game*. Banyak orang, terutama yang menggunakan *Android*, memiliki akses mudah ke *game*. Pengguna yang mengonsumsi hiburan di perangkat *Android* dapat menjadi lebih imajinatif dan terdorong untuk mempertimbangkan cara memecahkan masalah guna maju ke *level* berikutnya. (Rifandi, 2017).

Dari banyaknya jenis *game* yang telah dikembangkan, *game* dapat dikelompokkan berdasarkan *genre*-nya salah satunya yaitu *game* simulasi. *Game* simulasi biasanya dirancang untuk menyimulasikan kehidupan secara nyata di mana *player* secara bebas mengendalikan suatu objek/karakter dengan tujuan melakukan hal-hal tertentu seperti menyelesaikan tugas dan misi. Dengan *game* simulasi ini, *player* dapat berfantasi seakan-akan dia hidup di dalam *game* sehingga *player* mendapatkan pengalaman seperti di dunia nyata. *Game* simulasi dengan tema pertanian yang terkenal sampai saat ini yaitu “Harvest Moon”, sebuah *game* konsol yang dikembangkan oleh Marvelous Interactive bersama Natsume dengan

serial simulasi pertanian di mana tujuan utamanya adalah mengelola sebuah perkebunan dengan cara menggembala ternak dan bercocok tanam.

Berdasarkan permasalahan di atas, penulis melihat perlu adanya penelitian pengembangan game simulasi pertanian berbasis android sebagai pengenalan dan pendekatan yang dapat dijadikan sebagai media alternatif untuk meningkatkan minat pemuda di bidang pertanian. Pemberian informasi dan sosialisasi dengan menggunakan *game* dapat memberikan pengalaman dan wawasan secara interaktif serta mudah diakses. Selain memberikan informasi simulasi pertanian, pengguna juga bisa belajar tentang perkembangan teknologi informasi dan keseimbangan antara otak kanan dan kiri. Agar *game* lebih menarik dan tidak mudah membosankan penulis menggunakan kecerdasan buatan dengan metode *finite state machine* (FSM) untuk mendesain perilaku NPC, dan metode logika *fuzzy* digunakan untuk mengklasifikasi hasil tanaman yang telah ditanam oleh *player*, dengan parameter jenis bibit, ketersediaan air dan ketersediaan pupuk pada saat menanam.

Dengan dikembangkannya *game* ini, penulis berharap tingkat minat anak muda untuk memajukan pertanian di Indonesia semakin meningkat. Islam sangat memuliakan profesi petani. Selain sebagai ladang penghasilan, bertani juga merupakan sebuah ibadah. Hal ini sejalan dengan sabda Nabi yang diriwayatkan oleh Imam Bukhari. Rasulullah saw bersabda:

مَا مِنْ مُسْلِمٍ يَغْرِسُ غَرْسًا، أَوْ يَزْرَعُ زَرْعًا، فَيَأْكُلُ مِنْهُ طَيْرٌ أَوْ إِنْسَانٌ أَوْ بَهِيمَةٌ، إِلَّا كَانَ لَهُ بِهِ صَدَقَةٌ

“Tidaklah seorang muslim yang menanam tanaman atau bertani kemudian burung, manusia atau pun binatang ternak memakan hasilnya, kecuali semua itu merupakan sedekah baginya.” (H.R. Al-Bukhari:2320).

Allah subhanahu wa ta'ala telah menganugerahkan bumi yang di dalamnya terdapat berbagai macam tumbuhan yang bisa dinikmati. Allah subhanahu wa ta'ala berfirman dalam Al-Quran surah Ar-Rahman ayat 10-13:

وَالْأَرْضَ وَضَعَهَا لِلْأَنَامِ (١٠) فِيهَا فَاكِهَةٌ وَالنَّخْلُ ذَاتُ الْأَكْمَامِ (١١) وَالْحَبُّ ذُو الْعَصْفِ وَالرَّيْحَانُ (١٢)
فَبِأَيِّ آلَاءِ رَبِّكُمَا تُكَذِّبَانِ (١٣)

“Dan bumi telah dibentangkan-Nya untuk makhluk(-Nya). Di dalamnya ada buah-buahan dan pohon kurma yang mempunyai kelopak mayang. Dan biji-bijian yang berkulit dan bunga-bunga yang harum baunya. Maka nikmat Tuhanmu yang manakah yang kamu dustakan?” (Q.S. Ar-Rahman 10-13).

Di ayat lainnya Allah subhanahu wa ta'ala juga memberikan nikmat untuk memudahkan manusia dalam bercocok tanam, sebagaimana diterangkan dalam surah Al-An'am ayat ke-99, Allah subhanahu wa ta'ala berfirman:

وَهُوَ الَّذِي أَنْزَلَ مِنَ السَّمَاءِ مَاءً فَأَخْرَجْنَا بِهِ نَبَاتَ كُلِّ شَيْءٍ فَأَخْرَجْنَا مِنْهُ خَضِرًا نُخْرِجُ مِنْهُ حَبًّا مُتَرَاكِبًا

“Dan Dialah yang menurunkan air dan langit, lalu Kami tumbuhkan dengan air itu segala macam tumbuh-tumbuhan, maka Kami keluarkan dari tumbuh-tumbuhan itu tanaman yang menghijau, Kami keluarkan dari tanaman yang menghijau itu butir yang banyak.” (QS. Al-An'am 99)

Dari ayat dan hadits di atas, merupakan rahmat serta petunjuk yang diberikan oleh Allah subhanahu wa ta'ala untuk kebutuhan dalam bercocok tanam.

1.2 Rumusan Masalah

Berdasarkan penjelasan dari latar belakang, maka dapat dirumuskan permasalahan pada penelitian ini adalah sebagai berikut:

1. Seberapa valid penerapan *finite state machine* pada *game* pertanian dalam mendesain perilaku NPC?

2. Seberapa valid penerapan *fuzzy logic* pada *game* pertanian dalam menentukan klasifikasi hasil tanaman?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan dari penelitian ini adalah sebagai berikut:

1. Menciptakan perilaku NPC yang adaptif dengan metode *finite state machine*.
2. Menciptakan klasifikasi hasil tanaman dengan *fuzzy logic*

1.4 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Menjadi salah satu media alternatif bagi generasi muda untuk mengenal bidang pertanian.
2. Menjadi solusi berbasis teknologi dalam penyuluhan bidang pertanian.
3. Menghasilkan *game* pertanian berbasis *android* yang interaktif.

1.5 Batasan Masalah

Berdasarkan identifikasi masalah di atas, maka penelitian ini memiliki batasan-batasan sebagai berikut:

1. *Game* ini dimainkan secara *single player* (pemain tunggal)
2. *Software* yang digunakan dalam pengembangan *game* ini adalah *game engine Unity 3D*.

3. Hasil klasifikasi pada tanaman menggunakan 3 variabel input yaitu jenis bibit, penyiraman dan pemupukan pada saat menanam dan 1 variabel output yaitu hasil tanaman?

1.6 Sistematika Penulisan

Sistematika yang menjadi kerangka dan kaidah penulisan skripsi sangat diperlukan agar memudahkan dalam membaca dan memahami pembahasan dalam skripsi ini secara keseluruhan. Proses penulisan diatur sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini membahas mengenai Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Manfaat Penelitian serta Batasan Masalah pada penelitian.

2. BAB II LANDASAN TEORI

Landasan teori ini meliputi kajian-kajian mengenai teori yang berhubungan dengan penelitian ini.

3. BAB III DESAIN DAN IMPLEMENTASI

Pada bab ini penulis merancang sebuah desain untuk diterapkan pada *game* yang akan dikembangkan serta mengimplementasikan metode-metode yang diusulkan.

4. BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas mengenai implementasi dari rancangan penelitian dan metode penelitian serta analisa dan hasil uji coba penelitian.

5. BAB V PENUTUP

Bab ini berisi kesimpulan dan saran dari semua penelitian yang telah dilakukan. Kesimpulan datang dalam bentuk deskripsi masalah dan hasil studi objektif. Saran berupa perbaikan atau solusi atas permasalahan atau kekurangan penelitian.

BAB II

LANDASAN TEORI

2.1 *Game*

2.1.1 Pengertian *Game*

Kata "*game*" berasal dari bahasa Inggris yang berarti permainan. Greg Costikyan (2013) memaparkan bahwa *game* merupakan semacam karya seni yang pesertanya disebut sebagai pemain, memilih cara mengelola sumber daya mereka dengan menggunakan objek *game* untuk mencapai tujuan. Media elektronik digunakan dalam *game* untuk membuatnya lebih menarik dan memberi pemain rasa kepuasan batin. *Game* adalah hiburan dalam bentuk multimedia. Bermain video *game* adalah salah satu cara untuk belajar. (Aulianti, Karim, & Riska, 2021).

Pengembang *game* biasanya memiliki tujuan tertentu saat membuat *game* yang sesuai dengan keinginan dan preferensi mereka. Tujuan *game* yaitu antara lain:

1. Sebagai hiburan (*Entertainment*),
2. Melatih ketangkasan (*Expand Skill*),
3. Mendidik (*Education*),
4. Menyampaikan pesan (*Embed Messages*).

Jenni Novak (2012) mencantumkan sejumlah *goals* saat menentukan tujuan pembuatan *game*, seperti menggunakannya sebagai alat untuk hiburan, mengajarkan keterampilan interaksi sosial, mencapai tujuan pendidikan melalui pengajaran yang menghibur, dan membina kebugaran fisik dan mental yang dapat dikembangkan. digunakan dalam terapi psikologis dan rehabilitasi. Beberapa *video*

game juga diproduksi oleh programmer yang ingin menyampaikan apresiasi artistik atau orisinalitasnya (Rianingtias, 2019).

2.1.2 Jenis-Jenis Game

Game dapat dikelompokkan berdasarkan kategorinya. *Game* dapat dikategorikan menjadi dua kelompok yaitu *game* 2 dimensi (2D) dan 3 dimensi (3D), tergantung bagaimana *game* tersebut direpresentasikan. Kamera *game* 2D hanya menggambarkan visual *game* yang mungkin dilihat pemain karena secara matematis kamera pada *game* 2D memiliki dua elemen koordinat Kartesius, yaitu x dan y . Pada *game* 3D konsep kameranya mirip dengan konsep kamera pada kehidupan nyata karena *game* 3D menggunakan elemen x , y , dan z dalam perhitungannya. (Nurdin, 2013).

Berdasarkan *platform* atau alat yang digunakan dalam penerapannya menurut Sulistyو dalam skripsi Auditya Gita Perdana (2019), jenis *game* dapat dibagi menjadi 5 jenis, antara lain:

1. *Arcade Game*, di Indonesia biasa dikenal dengan *ding-dong*, biasanya ditempatkan di area khusus dengan mesin yang dirancang khusus untuk jenis *video game* tertentu, sehingga membuat pemain merasa lebih terlibat. dan lebih menikmati permainan. Seperti permainan dengan tambahan fitur kursi khusus, senjata khusus, sensor gerak, sensor hentakan dan setir, dan transmisi roda gigi, dan lainnya.
2. *PC Game*, yaitu *game* yang dimainkan di *platform Personal Computer*.
3. *Console Game*, yaitu *game* yang dimainkan pada konsol tertentu, seperti *XBOX 360*, *PlayStation 2*, *PlayStation 3*, *Nintendo Wii*.

4. *Handheld Game*, yaitu *game* yang dimainkan di konsol khusus konsol khusus yang bisa dibawa ke mana-mana, seperti *Nintendo DS*, *Sony PSP*.
5. *Mobile Game*, yaitu *game* yang di desain khusus untuk dimainkan di *mobile phone* atau PDA.

Berdasarkan *genre*-nya, menurut Rio Caesar (2015), *game* dikelompokkan menjadi beberapa kelompok antara lain:

1. *Action Adventure Game*, *game* yang memadukan *gameplay* antara *action game* dan *adventure game*.
 - a. *Metroidvania*, *sub-genre game* yang memadukan gaya permainan antara aksi-petualangan dan/atau *platformer* yang berfokus pada *non-linear* terpandu serta eksplorasi dan progres yang terjaga keamanannya.
 - b. *Survival Horror*, *game* dengan gaya permainan yang mengharuskan pemain mengelola sumber dayanya untuk menyelesaikan permainan di bawah tekanan lingkungan *game*.
 - c. *Stealth-Based Game*, *game* yang mengharuskan pemain untuk bersembunyi diam-diam di lingkungan *game* sambil menyelesaikan misi untuk menyelesaikan permainan.
2. *Action Game*, *game* dengan gaya permainan yang berfokus pada pertempuran
 - a. *Platform Game (Platformer)*, *sub-genre* dari *action game* di mana tujuan utamanya adalah untuk memindahkan karakter pemain di

antara titik-titik dalam suatu lingkungan. *Game platform* dicirikan oleh level yang terdiri dari medan yang tidak rata dan platform yang ditangguhkan dengan ketinggian bervariasi yang membutuhkan lompatan dan pendakian untuk dilintasi.

- 1) *Cinematic Platformer*, sebuah *subgenre* dari *platform game* dengan mekanisme *game* dan gaya visualnya yang memiliki tingkat realisme yang lebih tinggi dari *game platform* lainnya.
 - 2) *Elimination Platformer*, jenis *platform game* dimana pemain bertujuan untuk mengeliminasi semua musuh dari permainan dengan berlari, melompat, dan menggunakan serangan khusus lainnya pada musuh seperti di kebanyakan *platformer* konvensional.
 - 3) *Puzzle Platformer*, *gameplay Puzzle Platformer* yang umum adalah membuat pemain mencoba mengatasi rintangan dan keluar dari serangkaian ruangan, sering kali dengan menggunakan mekanisme permainan yang unik.
- b. *Beat 'em Up*, jenis permainan dimana pemain bertarung dengan tangan kosong melawan musuh yang banyak.
- 1) *Hack and Slash*, *video game* dengan gaya permainan *Beat 'em Up* di mana pemain bertarung menggunakan senjata jarak dekat.

- 2) *Spectacle Fighter*, video game dengan gaya permainan *Beat 'em Up* atau *Hack and Slash* tetapi focus pada efek visual yang menarik.
- c. *Fighting Game*, salah satu sub-game *action* di mana poin utamanya adalah pertarungan satu lawan satu.
- 1) *Platform Fighter*, subgenre video game dari game pertarungan yang mencakup game yang menekankan pertarungan di atas panggung dengan gerakan 2D, mirip dengan *game platform*
 - 2) *Mascot Fighter*, game yang jarang seri itu sendiri, dan biasanya didasarkan pada game atau serial anime lain, atau merupakan *Massive Multiplayer Crossover*.
- d. *First-Person Shooter (FPS)*, video game menembak yang berpusat pada senjata dan pertempuran berbasis senjata lainnya dalam sudut pandang orang pertama, dengan pemain mengalami aksi melalui mata protagonis atau antagonis yang bersenjata, dan kemudian mengendalikan karakter pemain di sebuah ruang tiga dimensi.
- e. *Third-Person Shooter (3PS/TPS)*, genre dari *action game 3D* yang menggunakan sudut pandang orang ketiga (*third person perspective*) di mana karakter utama terlihat dalam layar, dan *gameplay*-nya terutama terdiri dari pengambilan gambar.
- f. *Rail Shooter*, sejenis video game berbasis aksi. Dalam *rail shooter*, kontrol pemain terbatas pada mengarahkan ke mana harus menembakkan senjata virtual atau memindahkan avatar mereka di

sekitar layar; pemain tidak mengontrol jalur yang diambil avatar mereka dari awal hingga akhir (walaupun mereka mungkin dapat menjeda gerakan itu).

- g. *Light Gun Game*, *video game* di mana pemain menggunakan *controller* berupa senjata mainan yang ditembakkan ke arah objek musuh yang ada di layar.
- h. *Shoot 'em Up (Shmups)*, gaya bermainnya yaitu karakternya bergulir sambil menembak.
 - 1) *Bullet Hell*, gaaya permainannya di mana musuh menembakkan ratusan proyektil secara vertikal ke karakter pemain.
 - 2) *Cute 'em Up*, *game* tembak-menembak dengan estetika kartun yang lucu.
 - 3) *Horizontal Scrolling Shooter*, bentuk *Shoot Em Up* yang bergerak secara horizontal dan secara otomatis bergerak dan karakter menembak musuh tergantung pada kontrol pemain
 - 4) *Vertical Scrolling Shooter*, *game Shoot Em Up* yang bergerak secara otomatis dan kontrol menembaknya dilakukan secara vertikal
- i. *Vehicular Combat*, gaya permainan di mana pemain mengoperasikan kendaraan yang dilengkapi dengan senjata untuk menyerang musuh.

3. *Adventure Game*, sebuah *genre video game* di mana pemain berperan sebagai protagonis dalam sebuah cerita interaktif, didorong oleh eksplorasi dan/atau pemecahan teka-teki.
 - a. *Dating Simulator*, sebuah *game* di mana pengguna mencoba menemukan keinginan karakter untuk mengencani mereka.
 - b. *Environmental Narrative Game*, jenis *game* yang pemainnya menyelesaikan *game* sambil dipandu oleh sebuah cerita.
 - c. *Full Motion Video (Interactive Movie)*, sejenis *game* berdasarkan *clip*, memungkinkan pemain diinstruksikan untuk menekan kombinasi tombol tertentu saat video diputar di latar belakang dan merespons kombinasi tombol yang mereka masukkan.
 - d. *Interactive Fiction*, petualangan yang hanya menggunakan kontrol berbasis teks
 - e. *Point-and-Click Game*, digunakan untuk mengeklik petunjuk di layar dalam permainan tunjuk-dan-klik.
 - f. *Visual Novel*, meskipun sama dengan *Dating Sim*, *Visual Novel* ini lebih fokus pada *genre* petualangan.
4. *Art Game*, *video game* dengan gaya permainan di mana pemain menciptakan karya seni dalam permainannya.
5. *Card Battle Game*, gaya bermain di mana pemain membawa setumpuk kartu mereka sendiri dan bermain.
6. *Casual Video Game*, *video game* dengan gaya permainan yang mudah untuk dimainkan dan dipelajari.

7. *Edutainment Game*, sebuah *game* yang menekankan aspek pendidikan dan hiburan.
8. *Endless Running Game*, sebuah permainan yang membutuhkan kesabaran untuk terus bermain selama mungkin.
9. *Idle Game*, permainan yang berkembang tanpa interaksi dari pemain.
10. *Maze Game*, sebuah format *game* di mana labirin adalah dasar dari *game* tersebut.
11. *Mecha Game*, *game* yang menggunakan karakter robot raksasa.
12. *Mons Series*, *game* di mana monster bergiliran melawan karakter lain.
13. *Multiplayer Online Battle Arena* (MOBA), *game* di mana kelompok pemain bersaing dengan kelompok pemain lain. Berbeda dengan RTS, di sini pemain hanya memainkan satu karakter miliknya, yang lain dimainkan oleh pemain lain dan komputer.
14. *Racing Game*, *game* di mana pemain bersaing adu kecepatan dengan pemain lainnya atau komputer.
15. *Puzzle Game*, jenis permainan yang membutuhkan keterampilan mental daripada akurasi atau kecepatan.
16. *Role-Playing Game* (RPG), *game* yang mana pemiannya berperan sebagai karakter, umumnya dalam latar fantasi atau fiksi ilmiah, yang dapat berinteraksi dalam dunia imajiner permainan.
17. *Simulation Game* (Sim), suatu bentuk permainan yang mensimulasikan sebagian atau seluruh aspek kehidupan manusia.

18. *Rhythm Game*, *game* yang pemainnya harus menyesuaikan ritme sesuai dengan perintah *game*.
19. *Sports Game*, sebuah *video game* yang berkaitan dengan olahraga.
20. *Strategy Game*, sebuah *game* yang membutuhkan pemikiran strategis dan taktis pemain untuk meraih kemenangan.
 - a. *4X (4X)*, *game* dengan tujuan sederhana yaitu menjelajah (*explore*), memperluas (*expand*), mengeksploitasi (*exploit*), dan memberantas (*exterminate*).
 - b. *Artillery Game*, *genre game* di mana pemain harus membidik objek yang jauh dari senjatanya.
 - c. *Real-Time Strategy (RTS)*, *video game* dengan gaya permainan yang berfokus pada pemain yang membangun dan mengendalikan pasukan selama pertempuran.
 - d. *Time Management Game*, *game* dimana pemain harus bisa menyelesaikan misi dalam waktu yang telah ditentukan.
 - e. *Tower Defense*, bentuk permainan di mana pemain menyiapkan formasi pertahanan untuk kemudian melawan kekuatan tentara musuh.
 - f. *Turn-Based Strategy (TBS)*, *video game* dengan gaya permainan yang secara random beralih antara satu pemain dan pemain lainnya untuk mengendalikan perang dan menyusun strategi.

Pada pengembangan proyek *game* penelitian ini, penulis fokus pada pembuatan *game* dengan *genre Simulation Game (Sim)*.

2.2 Pertanian

Menurut Lia Widya (2020), pertanian adalah sumber daya hayati yang digunakan oleh aktivitas manusia untuk menjaga lingkungan, menyediakan makanan, sumber energi atau bahan baku industri. Pertanian, menurut Van Aarsten (1953), adalah penggunaan aktivitas manusia untuk menghasilkan produk yang berasal dari hewan dan/atau tumbuhan yang diproduksi untuk menggunakan semua peluang yang disediakan oleh alam untuk mengembangbiakkan. Beberapa orang mendefinisikan pertanian sebagai aktivitas manusia yang memerlukan pembukaan lahan, menghasilkan berbagai tanaman, termasuk tanaman tahunan dan tahunan, tanaman pangan dan non-pangan, dan menggunakan tanaman ini untuk memelihara ikan dan ternak (Suratiah, 2006).

Menurut Deloitte (2012), berdasarkan perspektif petani, siklus pertanian biasanya terbagi menjadi tiga tahapan utama yaitu:

1. *Pre-cultivation* (persiapan lahan), yaitu meliputi pemilihan benih, persiapan lahan, jadwal pertanian, akses ke sumber, pendanaan kredit, dan sebagainya.
2. *Crop cultivation* dan *harvesting* (penanaman dan panen), meliputi persiapan tanah dan penaburan benih, pengelolaan sumber daya, pengelolaan air dan kesuburan, pengendalian hama, dll.
3. *Post-harvest* (pasca panen), meliputi transportasi, distribusi, pengemasan, pengolahan makanan, dll.

2.3 Software Unity 3D

Unity merupakan *game engine* yang dibuat oleh *Unity Technology*. *Game Engine* adalah perangkat lunak yang memproses visual, suara, input, dan elemen lain untuk membuat *game*. Mampu membuat *game* lintas platform yang dapat dipublikasikan untuk standalone (.exe), berbasis web, Android, iOS, XBOX, atau PS3 dengan catatan mendapatkan lisensi memberikan keunggulan *Unity* dibandingkan *game engine* lainnya. Versi gratisnya, sementara itu, terbatas pada web dan publikasi mandiri. Versi premium dan versi gratis adalah dua versi *Unity*. Beberapa kemampuan, seperti ketidakmampuan mentransfer *game* ke konsol, tidak tersedia dalam edisi gratis (Siti Mahmudah, 2017). *Unity* sangat mudah digunakan dalam pembuatan *game* 2D maupun 3D

Game engine Unity 3D sendiri bukanlah penemuan baru. *Unity Technologies* didirikan pada tahun 2004. *Unity* menawarkan fitur pengguna gratis, mudah digunakan, dan terintegrasi di berbagai *platform* merupakan salah satu faktor yang berkontribusi terhadap pertumbuhan dan popularitasnya yang cepat. (Septian R.H., 2018).

Banyak aspek dalam *Unity* sendiri membuatnya lebih mudah digunakan oleh konsumen. Berikut fitur-fitur *Unity* antara lain:

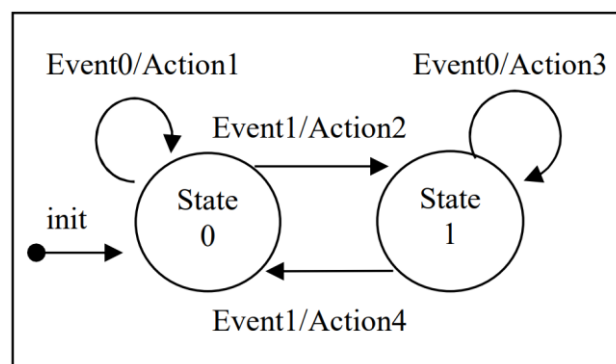
- a. *Rendering*, *Unity* dapat menerima format desain seperti 3ds Max, Maya, Softimage, Blender, Adobe Photoshop dan Adobe Fireworks. Aset ini dapat ditambahkan ke proyek *game* Anda dan dikelola melalui antarmuka pengguna grafis *Unity*. Ini juga memiliki kemampuan untuk melakukan pemetaan *bump*, pemetaan *paralaks*, pemetaan refleksi,

bayangan dinamis dengan peta bayangan, layer oklusi ruang ambien (SSAO), *rendering* ke tekstur, dan efek pasca-pemrosesan layar penuh.

- b. *Scripting, Script* pada *Unity* dibuat dengan Mono 2.6, Implementasi *open source* dari *.NET Framework*. Pemrogram dapat menggunakan *UnityScript* (bahasa khusus yang terinspirasi oleh sintaks *ECMAScript*-nya dalam bentuk *JavaScript*), *C#*, atau *Boo* (terinspirasi oleh sintaks bahasa pemrograman *Python*).
- c. *Asset Tracking, Unity* juga memuat *Server Unity Asset* yaitu sebuah solusi terkontrol untuk pengembangan *game asset* dan *script* berbasis *Server*.
- d. *Asset Store*, diluncurkan November 2010, *Unity Asset Store* adalah sebuah *resource* yang terdapat di *Unity editor*. *Asset store* terdiri dari kumpulan lebih dari 4,400 *asset packages*, *3D models*, *textures* dan *materials*, *sistem particle*, *music* dan *sound effects*, *tutorial* dan *project*, *scripting package*, *editor extensions* dan *service online*. Semua bisa ditambahkan ke proyek pengguna dari *Asset Store*.
- e. *Physics, Unity* juga mendukung fitur *PhysX physics engine* (sejak *Unity 3.0*) dari *Nvidia* (sebelumnya *Ageia*) dengan fitur tambahan untuk simulasi pakaian dan lain sebagainya.
- f. *Platforms, Unity* mendukung pengembangan ke berbagai *platform*. Di dalam *project*, *developer* dapat mengontrol pengiriman ke perangkat *desktop*, *mobile*, *console*, dan *web browser* (Septian R. H., 2018).

2.4 Finite State Machine (FSM)

Metode FSM sangat terkenal karena kehandalannya ketika diterapkan dalam *game*. Proses desain *Finite State Machine* (FSM) untuk sistem kontrol menggunakan istilah status, kejadian, dan tindakan untuk mendefinisikan perilaku atau prinsip pengoperasian sistem. Sistem akan berada dalam salah satu mode aktif di beberapa titik selama jangka waktu yang lama. Jika sistem menerima masukan atau kejadian tertentu dari perangkat eksternal atau komponen sistem dapat beralih atau bertransisi ke keadaan lain. Biasanya, aktivitas sistem sebagai respons terhadap *input* juga hadir selama perubahan status ini. Kegiatan yang dilakukan mungkin langsung atau terdiri dari beberapa langkah terkait. Contoh diagram *state* sederhana ditunjukkan pada Gambar 2.1



Gambar 2.1 Contoh diagram *state* sederhana

Diagram menggambarkan *finite state machine* (FSM) dengan 2 keadaan, 2 masukan, dan 4 tindakan keluaran yang berbeda. Seperti terlihat pada gambar, sistem memasuki *Event 0* ketika dimulai dan menghasilkan *Action 1* jika terjadi pada *Event* ini. *State Input 0*; namun demikian, jika *State1* terjadi, *Action2* akan diambil, sistem kemudian akan beralih ke *Event1*, dan seterusnya. Memanfaatkan

pemrograman berorientasi objek, atau OOP, adalah cara berbeda untuk membuat FSM. (Rahadian, Suyatno, & Maharani, 2017). Tingkat fleksibilitas OOP yang tinggi dan kemudahan perawatan pada sistem dasar dan rumit merupakan keuntungan saat digunakan di FSM. Selain itu, ia memperoleh salah satu manfaat OOP, yaitu kemampuan untuk menggunakan kembali kode yang telah diketik sebelumnya (*code reusability*), yang mengurangi jumlah waktu yang dihabiskan untuk mengetik kode. (Rahadian, Suyatno, & Maharani, 2017).

2.5 Fuzzy Logic

Kesulitan ketidakpastian ditangani dengan menggunakan logika *fuzzy*. Ketidakpastian mengacu pada situasi yang penuh dengan pertanyaan, kesalahan, kesenjangan dalam pengetahuan, dan nilai-nilai parsial. Logika *fuzzy* sebenarnya telah menjadi topik pembahasan sejak lama, khususnya sejak zaman para filosof Yunani kuno. Plato adalah filsuf pertama yang meletakkan prinsip dasar logika *fuzzy* dalam hal ini. Selain benar dan salah, menurut Plato, ada kategori ketiga (Arif, Yunifa M., M. Hardi, & Supeno Mardi, 2012).

Logika *fuzzy* merupakan salah satu komponen pendukung *soft computing*. Prof. Lotfi A. Zadeh pertama kali membahas logika *fuzzy* pada tahun 1965. Teori himpunan *fuzzy* berfungsi sebagai landasan logika *fuzzy*. Tingkat keanggotaan memainkan peran penting dalam menentukan ada atau tidaknya unsur-unsur dalam suatu himpunan dalam teori himpunan *fuzzy*. Fitur utama penalaran logika *fuzzy* adalah nilai keanggotaan, derajat keanggotaan, atau fungsi keanggotaan. Logika *fuzzy* sering digunakan untuk memetakan masalah dari *input* ke *output* yang diharapkan. Memilih himpunan *fuzzy* yang sesuai adalah langkah pertama dalam

mengembangkan sistem *fuzzy*. Nilai keanggotaan suatu *item* x dalam himpunan A , yang sering dinyatakan sebagai $A(x)$, memiliki dua pilihan karena himpunan *fuzzy* merupakan himpunan tegas., yaitu (Nurmuslimah, 2020):

1. Satu (1), menunjukkan bahwa suatu *item* menjadi anggota dalam suatu himpunan, atau
2. Nol (0), menunjukkan bahwa suatu *item* tidak menjadi anggota dalam suatu himpunan.

Himpunan *fuzzy* memiliki 2 atribut, yaitu:

1. Linguistik, yaitu pemberian nama pada suatu kelompok yang menjelaskan suatu kondisi atau keadaan tertentu menggunakan bahasa umum, seperti: DINGIN, HANGAT dan PANAS.
2. Numeris, yaitu sebuah nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 10, 15, 20, dsb.

Pada umumnya ada 3 metode sistem inferensi *fuzzy* yang digunakan dalam logika *fuzzy*, yaitu: Metode Tsukamoto, Mamdani, dan Sugeno.

- a. Metode Tsukamoto, pendekatan penalaran monoton diperluas dengan metode Tsukamoto. Aturan *IF-THEN* harus memiliki himpunan *fuzzy* dengan fungsi keanggotaan yang tetap untuk mewakili setiap hasil. Mengingat *-predikat, output* dari hasil inferensi dari setiap aturan diberi tegas (*crisp*). Rata-rata yang tertimbang digunakan untuk menentukan hasilnya.
- b. Metode Mamdani, karena himpunan aturan bersifat independen (tidak saling bergantung), maka setiap aturan dalam metode ini yang

berbentuk implikasi (“sebab akibat”) memiliki anteseden berupa konjungsi (*AND*) dengan nilai keanggotaan dalam bentuk minimum (*MIN*), dan akibat gabungan berupa maksimum (*MAX*).

- c. Metode Sugeno, merupakan metode inferensi fuzzy untuk aturan yang dinyatakan dalam bentuk IF – THEN dan. Keluaran (hasil) dari sistem tidak berupa himpunan fuzzy tetapi berupa konstanta atau persamaan linier. Metode ini dikenal dengan metode TSK karena diperkenalkan Takagi-Sugeno Kang pada tahun 1985.

Menurut Cox (1994), Metode TSK terdiri dari 2 jenis yaitu:

1. Model *Fuzzy Sugeno Orde-Nol* Jika (x_1 adalah A_1) o (x_2 adalah A_2) o (x_3 adalah A_3) o o (x_N adalah A_N) maka $z=k$
 Dengan A_i adalah himpunan *fuzzy* ke- i sebagai *anteseden*, dan k adalah suatu konstanta (tegas) sebagai konsekuensi.
2. Model *Fuzzy Sugeno Orde-Satu* Jika (x_1 adalah A_1) ... o (x_N adalah A_N) maka $z = p_1*x_1 + \dots + p_N*x_N + q$ Dengan A_i adalah himpunan *fuzzy* ke- i sebagai *anteseden*, dan p_i adalah suatu konstanta (tegas) ke- i dan q juga merupakan konstanta dalam konsekuensi. Apabila komposisi aturan menggunakan metode Sugeno maka *defuzzifikasi* dilakukan dengan cara mencari nilai rata-ratanya.

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Desain *Game*

Desain *game* merupakan tahap analisis pada *game* dan proses mengidentifikasi dan mengevaluasi terhadap *game* sejenis atau *game* yang akan dibangun. Pada tahap ini, persyaratan fungsional didefinisikan, persiapan desain implementasi dibuat, dan desain permainan dijelaskan, yang dapat berupa perencanaan, sketsa, dan gambar.

3.1.1 Jenis *Game*

Pada penelitian ini *game* yang akan dibuat yaitu permainan berbasis simulasi dan *adventure*. *Game* ini merupakan *game single player* (pemain tunggal) yang mana pemain akan mengendalikan satu karakter utama yang dapat melakukan aktifitas seperti petani pada umumnya. Pemain dapat memilih bibit, menanam, menyiram, memanen, melakukan transaksi serta melakukan interaksi dengan penduduk sekitar. Tujuan pengembangan *game* ini adalah menerapkan efek simulasi pertanian agar *game* tidak membosankan dan tetap terus bisa dimainkan.

3.1.2 *Storyline*

Nama *game* ini yaitu “Saung Sawah” yang merupakan *game* simulasi dengan *setting* pedesaan. Cerita yang diangkat pada *game* ini adalah mengenai seorang pemuda berumur 24 tahun yang baru menyelesaikan studinya dan diminta untuk meneruskan pertanian kakeknya yang akan pensiun. Pada *game* ini pemain akan dilatih agar menjadi petani yang handal, dimana pemain harus menyelesaikan

tugas yang akan diarahkan langsung oleh kakek. Dasar dari permainan ini yaitu pemain melakukan aktifitas seperti menanam, menyiram tanaman, memberi pupuk pada tanaman, dan memanennya. Dan setelah panen pemain bisa menjual hasil panennya kepada penjual. Dimana yang perlu diperhatikan pada saat proses menanam harus memperhatikan kesehatan tanaman sehingga hasil ketika tanaman dipanen bisa mendapatkan kualitas yang bagus dan ketika dijual akan mendapatkan harga yang lebih mahal.

3.1.3 Deskripsi Karakter dan Objek

Karakter utama adalah karakter yang dikendalikan oleh *player*. Pada *game Saung Sawah* karakter utama nya adalah seorang pemuda berusia 24 tahun yang baru menyelesaikan pendidikan, dan diminta untuk meneruskan pertanian kakeknya yang akan pensiun.

1. Karakter Utama Avice

Karakter utama dari game ini adalah Avice, seorang pemuda cerdas dengan keinginan kuat untuk mempelajari hal-hal baru. Avice juga sangat antusias dan bertekad, dan dia ingin melanjutkan pertanian kakeknya agar bisa terus berkembang.

2. Karakter NPC Kakek Oland

Kakek Oland adalah kakek dari karakter utama yaitu Avice, ia telah mengelola pertanian miliknya selama 35 tahun, ia telah memiliki banyak pengalaman dan pengetahuan mengenai pertanian dan siap mengajarkan Avice sang cucu agar bisa menjadi petani yang handal seperti dia.

3. Karakter NPC Om Giant

Om Giant merupakan distributor pertanian, dia memiliki banyak relasi dengan penjual dan konsumen pertanian. Dimana dia menjual segala macam barang mulai dari bibit, pupuk, dan alat-alat yang digunakan di dalam pertanian. Om Giant akan memberikanmu *cashback* pembelian jika kamu membeli barang dengan pembelian barang mencapai target yang diberikan oleh Om Giant.

4. Karakter NPC Ayam

Ayam merupakan hewan ternak yang dimiliki oleh karakter utama. Dalam waktu tertentu dia akan menghasilkan telur yang dapat diambil kemudian dijual.

5. Objek Tanaman

Pada *game* ini terdapat 5 jenis tanaman yang bisa kamu tanam yaitu Padi, Jagung, Tomat, Wortel dan Kentang. Dan masing-masing tanaman memiliki harga dan kualitas bibit.

6. Objek Sumur

Sumur berfungsi untuk mengambil air ketika persediaan air habis. Sumur akan otomatis beregenerasi sesuai dengan waktu yang telah ditetapkan. Kamu bisa mengambil air selama regenerasi sumur telah terisi.

7. Koin

Tampilan koin akan ditampilkan di bar koin. Koin berfungsi untuk melakukan pembelian lahan, bibit atau barang lainnya di dalam

permainan. Pemain bisa memperoleh koin dengan menjual hasil tanamannya atau barang lainnya.

8. Energi

Energi pemain akan ditampilkan di bar energi. Energi ini berfungsi untuk melakukan aksi memanen. Ketika energi pemain tidak cukup, makan pemain harus mengisi energinya dengan cara memakan sesuatu atau beristirahat.

9. Air

Air akan ditampilkan di tombol penyiraman. Air berfungsi untuk menyiram tanaman.

10. Pupuk

Pupuk akan ditampilkan di tombol penyemprotan pupuk. Pupuk berfungsi untuk memberikan nutrisi pada tanaman.

3.1.4 Storyboard

3.1.4.1 Storyboard Level 1 Game

Pada level ini, pemain akan berinteraksi dengan tampilan menu utama yang memiliki beberapa pilihan dan fitur.

1. Tampilan Menu utama

Tampilan menu utama adalah tampilan pertama yang muncul saat pemain memulai *game*. Tujuan dari tampilan menu utama adalah sebagai berikut:

- a. Memberikan pilihan akses ke fitur utama *game*

Pada tampilan menu utama, pemain akan diberikan beberapa pilihan untuk memilih fitur utama yang ingin diakses. Misalnya, pemain dapat memilih opsi "*Start*", "*Setting*", atau "*Exit*". Setiap pilihan akan membawa pemain ke tampilan yang sesuai dengan fitur yang dipilih.

b. Menampilkan informasi penting

Tampilan menu utama juga dapat digunakan untuk menampilkan informasi penting kepada pemain, seperti nama permainan, logo permainan, dan instruksi singkat tentang cara bermain. Hal ini bertujuan untuk memberikan pemain gambaran awal tentang permainan dan menjelaskan tata cara bermain secara singkat.

c. Membangun suasana permainan

Tampilan menu utama juga dapat digunakan untuk menciptakan suasana permainan yang menarik dan sesuai dengan tema *game* pertanian. Misalnya, menggunakan ilustrasi karakter, pertanian, dan elemen alam yang dapat menggambarkan suasana petualangan di dunia pertanian.



Gambar 3.1 *Storyboard* menu utama

2. Alur interaksi pengguna dengan menu utama dapat dijelaskan sebagai berikut:

a. Start

Pemain dapat memilih opsi "*Start*" untuk memulai permainan dari awal. Setelah memilih opsi ini, pemain akan dialihkan ke tampilan intro dalam *gameplay*.

b. Setting

Opsi "*Setting*" akan membawa pemain ke tampilan pengaturan *game*. Di sini, pemain dapat mengatur preferensi mereka terkait suara dll.

c. Exit

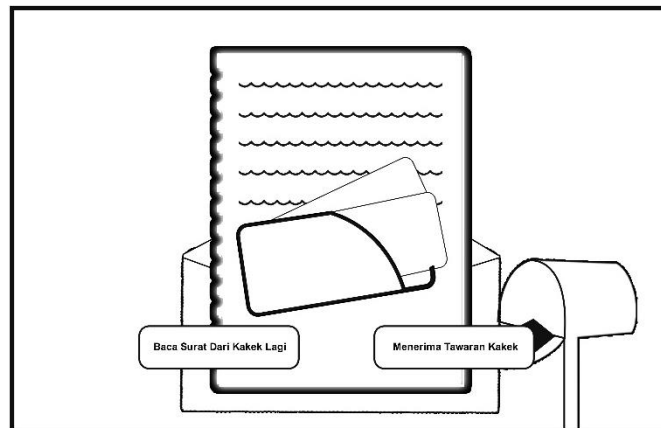
Opsi "*Exit*" akan keluar dari permainan dan mengakhiri sesi permainan.

3.1.4.2 Storyboard Level 2 Game

Pada level ini, pemain akan berinteraksi dengan menu awal cerita pada *game* yang memiliki beberapa pilihan dan fitur.

1. Menu tampilan awal cerita pada *game*

Setelah pemain memilih opsi "*Start*" dari menu utama, mereka akan diarahkan ke tampilan cerita awal dalam *game*. Tampilan ini dirancang untuk memungkinkan pemain merasakan jalan cerita di dalam *game* sehingga *game* menjadi lebih menarik.



Gambar 3.2 *Storyboard* menu cerita awal *game*

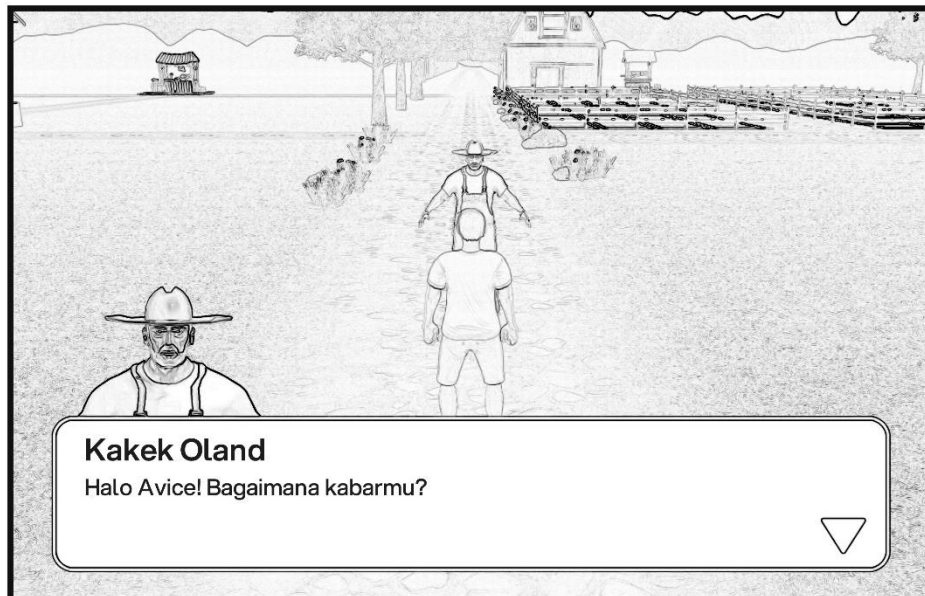
2. Alur interaksi pengguna dengan menu cerita awal *game* dapat dijelaskan sebagai berikut:
 - a. Pemain akan disuruh membuka surat yang berada di kotak pos.
 - b. Pemain membaca surat dari kakek
 - c. Pemain memilih tombol “Baca Surat Dari Kakek Lagi” atau “Menerima Tawaran Kakek”.
 - d. Jika pemain menekan tombol “Baca Surat Dari Kakek Lagi”, maka surat akan ditampilkan kembali, dan jika pemain menekan tombol “Menerima Tawaran Kakek”, maka pemain akan diarahkan ke dalam *gameplay*.

3.1.4.2 *Storyboard Level 3 Game*

Pada storyboard level 3 terdapat tampilan *gameplay* permainan. Setelah pemain memilih tombol “Menerima Tawaran dari Kakek” dalam *Scene 2*, mereka akan masuk ke tampilan *gameplay*. Tampilan ini dirancang untuk menampilkan

permainan dimana pemain mulai memainkan *game* dengan menyelesaikan misi-misi di dalamnya.

1. Tampilan Awal Permainan:
 - a. Tampilan ini akan menampilkan percakapan antara karakter utama dan juga NPC kakek Oland.
 - b. NPC kakek Oland akan memberikan petunjuk awal tata cara bermain kepada pemain.
 - c. Pemain mengikuti petunjuk awal tata cara bermain.
 - d. Setelah petunjuk tata cara bermain selesai, NPC kakek Oland akan memberikan misi kepada pemain.
 - e. Pemain berusaha menyelesaikan misi yang ada.



Gambar 3.3 *Storyboard* tampilan percakapan NPC kakek Oland dan karakter utama.

2. Alur Interaksi *Player* dalam Tampilan Awal *Gameplay*

- a. Pemain mengikuti arahan cara bermain dari NPC kakek Oland
- b. Pemain mendapatkan misi dari NPC kakek Oland.

3. Tampilan Lahan Pertanian

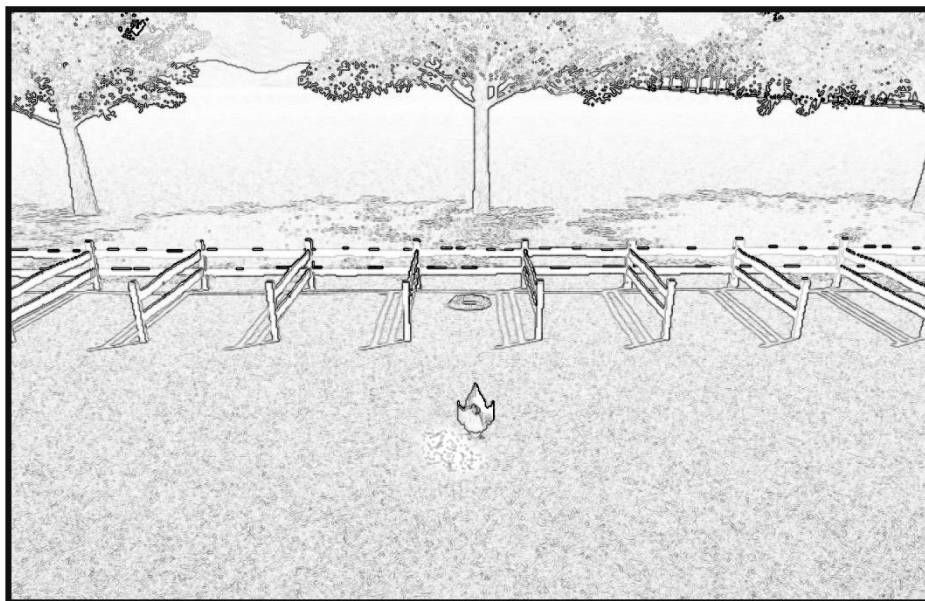
- a. Tampilan ini menampilkan sebuah lahan yang bisa ditanami oleh pemain.
- b. Ketika pemain sudah menanam tanaman, maka lahan akan menampilkan tanaman mulai dari tampilan bibit, tanaman kecil, tanaman dewasa kemudian tanaman siap panen.



Gambar 3.4 *Storyboard* tampilan lahan pertanian.

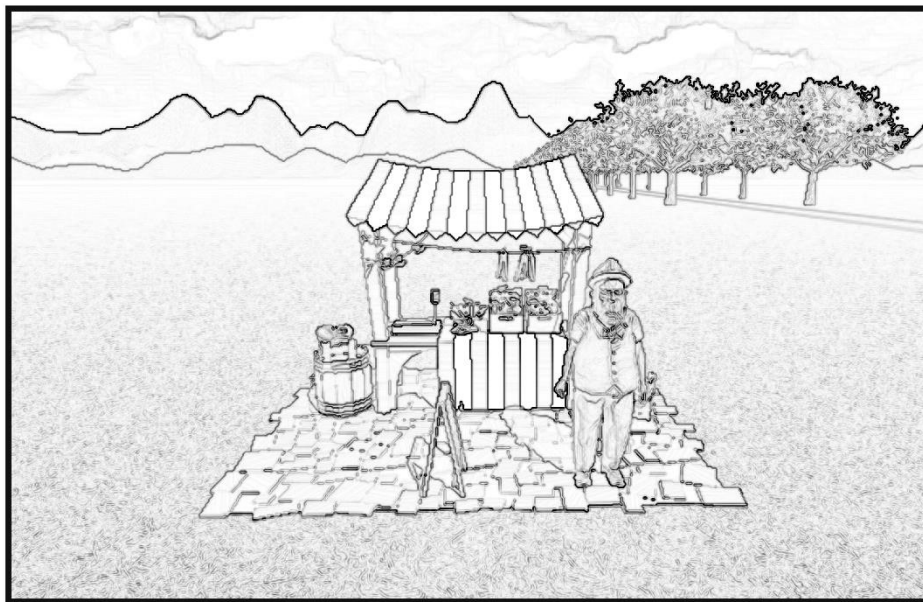
4. Alur Interaksi *Player* Dalam Menanam Benih dan Merawat Tanaman
 - a. Pemain memilih lahan kosong yang masih belum ditanami.

- b. Pemain dapat memilih jenis bibit yang ingin mereka tanam dengan mengklik atau memilih bibit tersebut di *layer* dan menekan tombol tanam.
 - c. Setelah bibit ditanam, pemain akan memasuki tampilan perawatan pertanian.
 - d. Pemain melihat informasi tentang kebutuhan tanaman, mulai dari penyemprotan pupuk dan penyiraman agar tumbuh dengan baik.
 - e. Setelah melewati waktu fase perawatan, makan tanaman akan masuk pada fase panen.
 - f. Pemain memanen tanaman yang sudah siap panen.
5. Tampilan Peternakan
- a. Tampilan ini menampilkan lahan hewan ternak yaitu ayam, dimana di dalamnya ia melakukan aktifitas seperti berjalan, makan dan juga bertelur.



Gambar 3.5 *Storyboard* tampilan lahan peternakan.

6. Alur Interaksi *Player* dan Hewan Ternak:
 - a. NPC Ayam akan berjalan-jalan mengitari lahan peternakan, ketika pemain mendekat maka ayam akan berlari menjauhi pemain.
 - b. Ketika NPC Ayam memasuki fase bertelur, maka NPC Ayam akan ke kandang untuk siap bertelur. Ketika NPC Ayam sudah bertelur, pemain bisa mengambil telurnya.
 - c. NPC Ayam akan kembali ke *state* awal yaitu berjalan mengitari lahan peternakan ketika pemain sudah mengambil telur.
7. Tampilan Transkasi Jual-Beli:
 - a. Tampilan ini menampilkan sebuah *market* dimana pemain bisa melakukan transaksi jual-beli.



Gambar 3.6 *Storyboard* tampilan *market*.

8. Alur Interaksi *Player* Dalam Melakukan Transaksi Jual-Beli
 - a. Pemain memilih barang yang akan dijual kepada NPC Om Giant.
Setelah itu pemain akan mendapatkan koin sesuai dengan barang yang telah dijual.
 - b. Pemain memilih barang yang akan dibeli dari NPC Om Giant.
Setelah itu, pemain membayar dengan koin sesuai dengan barang yang akan dibeli.

3.2 Perancangan Metode

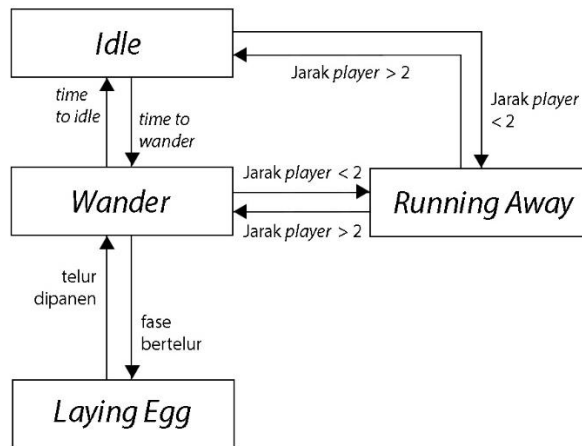
3.2.1 Perancangan FSM dalam *Game*

Metode FSM (*Finite State Machine*) diterapkan pada perancangan desain perilaku NPC Ayam, NPC Kakek Oland dan NPC Om Giant. Masing-masing perilaku NPC dibuat sesuai dengan skenario pada *game*.

3.2.1.1 Perancangan FSM Ayam

Dalam FSM NPC Ayam, pada saat sistem dimulai NPC Ayam berada pada *state idle* ketika memasuki *time to wander* maka NPC Ayam akan bertransisi ke *state wander* ketika memasuki *time to idle*, NPC Ayam akan kembali ke *idle state*. Ketika jarak NPC Ayam dan *player* kurang dari 2, maka NPC Ayam akan bertransisi ke *state running away* dan menjauh dari *player*. Ketika jarak NPC Ayam dan *player* lebih dari 2 maka NPC Ayam akan kembali ke *state idle* atau *state wander* tergantung *state* yang berjalan sebelum transisi ke *state running away*. Ketika memasuki fase bertelur maka NPC Ayam akan menuju kandang dan

bertransisi ke *state laying egg*. Untuk diagram FSM NPC ayam dapat dilihat di Gambar.3.7 berikut.

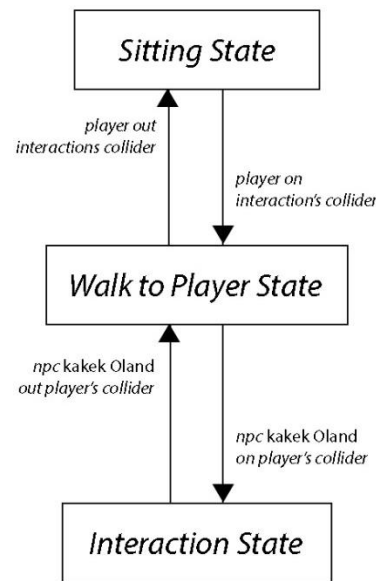


Gambar 3.7 FSM NPC Ayam.

3.2.1.2 Perancangan FSM NPC Kakek Oland

Dalam FSM NPC Kakek Oland, *state* awal yaitu *sitting state*, ketika *player* berada di *interaction's collider* maka NPC Kakek Oland akan melakukan transisi ke *walk to player state*, ketika NPC berada di *player's collider* maka NPC Kakek Oland akan melakukan transisi ke *interaction state* dan akan mengaktifkan *button mission*. Dimana ketika *player* mengklik *button mission* maka akan menampilkan panel misi.

Ketika *player* tidak berada pada *interaction's collider* maka NPC Kakek Oland akan kembali ke *sitting state*. Untuk gambar rancangan FSM NPC Kakek Oland dapat dilihat pada Gambar 3.8 berikut.

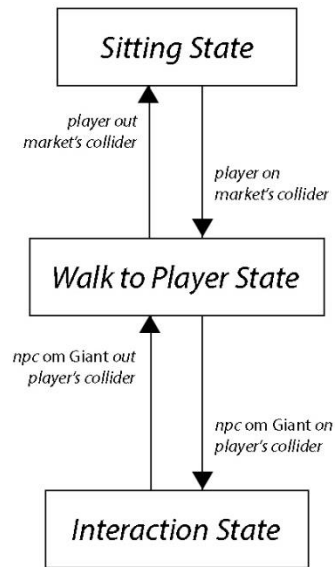


Gambar 3.8 Rancangan FSM NPC Kakek Oland.

3.2.1.3 Perancangan FSM NPC Om Giant

Dalam FSM NPC Om Giant, *state* awal yaitu *sitting state*, ketika *player* berada di *market's collider* maka NPC Om Giant akan melakukan transisi ke *walk to player state*, ketika NPC berada di *player's collider* maka NPC Om Giant akan melakukan transisi ke *interaction state* dan akan mengaktifkan *button market*. Di mana ketika *player* mengklik *button market* maka akan menampilkan panel *market*.

Ketika *player* tidak berada pada *market's collider* maka NPC Om Giant akan kembali ke *sitting state*. Untuk gambar rancangan FSM NPC Om Giant dapat dilihat pada Gambar 3.9 berikut.



Gambar 3.9 Rancangan FSM NPC Om Giant.

3.2.2 Perancangan Metode *Fuzzy Logic*

Pada pengembangan *game* ini, menggunakan metode *fuzzy* Sugeno untuk menentukan hasil tanaman melalui *input* penyemprotan pupuk dan penyiraman dari pemain. Berikut merupakan tahapan-tahapan dalam merancang metode *fuzzy* Sugeno

3.2.2.1 Variabel *Fuzzy*

Dalam permainan ini hasil panen (*output*) diklasifikasikan berdasarkan tiga variabel *input* yaitu variabel Jenis Bibit, variabel Pemupukan, dan variabel Penyiraman. Hasil Tanaman adalah variabel *output*, sementara. Berikut adalah nilai bahasa dari variabel yang digunakan:

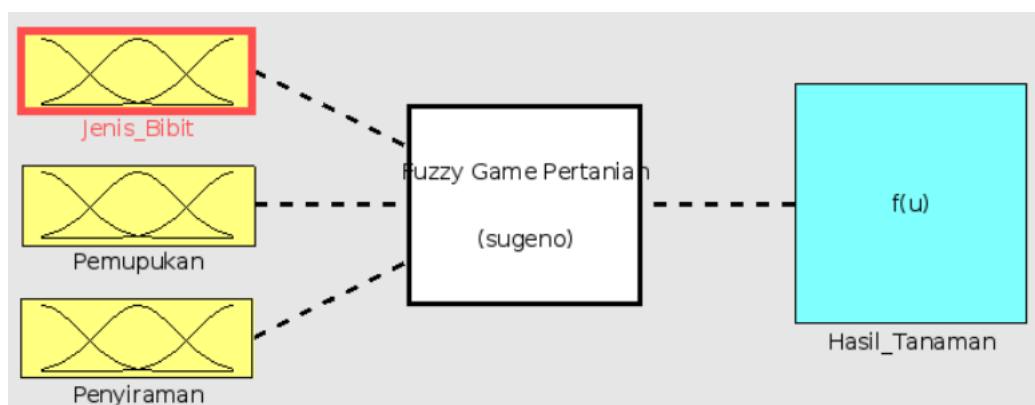
Tabel 3.1 Nilai linguistik variabel *fuzzy*: *input* dan *output*.

| Variabel | | Himpunan <i>Fuzzy</i> |
|--------------|-------------|-----------------------|
| <i>Input</i> | Jenis Bibit | Biasa |
| | | Unggul |

| | | |
|---------------|---------------|---------|
| | Pemupukan | Sedikit |
| | | Sedang |
| | | Banyak |
| | Penyiraman | Kurang |
| | | Sedang |
| | | Penuh |
| <i>Output</i> | Hasil Tanaman | Kering |
| | | Standar |
| | | Bagus |

3.2.2.2 Fuzzyfikasi

Untuk menentukan derajat keanggotaan dalam himpunan *fuzzy*, nilai *crisp* (numerik) dipetakan ke dalam himpunan tersebut melalui proses *fuzzyfikasi*.



Gambar 3.10 Alur proses *fuzzy*

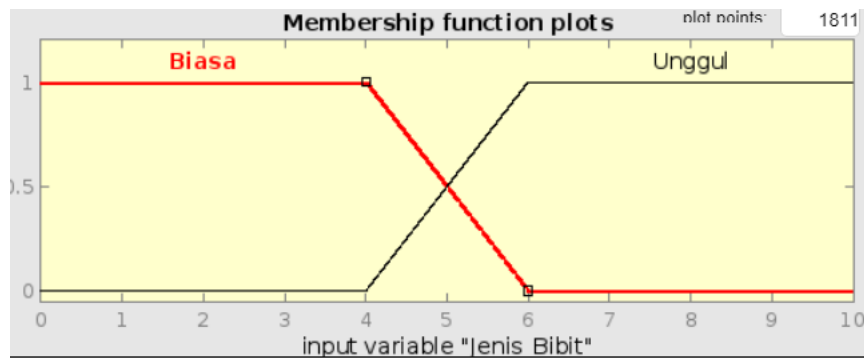
Berdasarkan *Fuzzy Interface System* (FIS) tahap selanjutnya yaitu memetakan himpunan keanggotaan *fuzzy* untuk mendapatkan nilai keanggotaannya. Untuk tahapannya dijelaskan sebagai berikut:

1. Variabel Jenis Bibit

Variabel Jenis Bibit memiliki tiga himpunan *fuzzy*, yaitu Biasa dan Unggul. Nilai semesta pembicaraan untuk variabel Jenis Bibit adalah [0 10]. Dan nilai domain untuk himpunan *fuzzy* adalah sebagai berikut:

Tabel 3.2 Himpunan *fuzzy* variabel Jenis Bibit.

| Himpunan <i>Fuzzy</i> | Domain |
|-----------------------|--------|
| Biasa | 0 - 6 |
| Unggul | 5 - 10 |



Gambar 3.11 Fungsi keanggotaan pada variabel Jenis Bibit.

Pada himpunan Biasa menggunakan fungsi keanggotaan representasi linear turun, sehingga untuk menentukan derajat keanggotaannya menggunakan rumus berikut:

$$\mu_{\text{JenisBibitBiasa}}(x) = \begin{cases} 1; & x \leq 4 \\ \frac{6-x}{6-4}; & 4 < x < 6 \\ 0; & x \geq 6 \end{cases} \quad (3.1)$$

Sedangkan pada himpunan Unggul menggunakan fungsi keanggotaan representasi linear naik, sehingga untuk menentukan derajat keanggotaannya menggunakan rumus berikut:

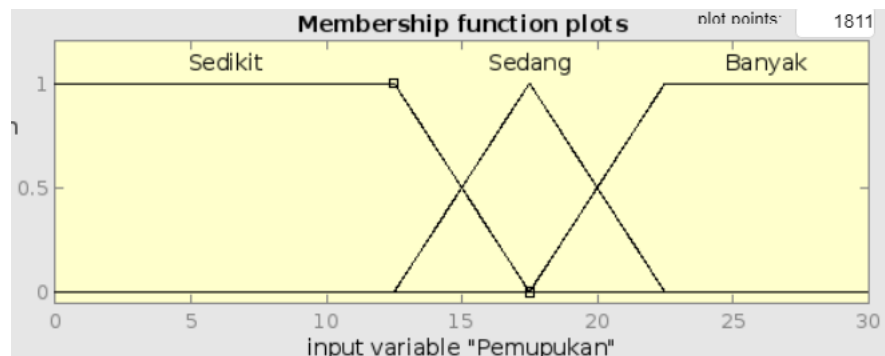
$$\mu_{JenisBibitUnggul}(x) = \begin{cases} 0; & x \leq 4 \\ \frac{x-4}{6-4}; & 4 < x < 6 \\ 1; & x \geq 6 \end{cases} \quad (3.2)$$

2. Variabel Pemupukan

Variabel Pemupukan memiliki tiga himpunan *fuzzy*, yaitu Sedikit, Sedang dan Banyak. Nilai semesta pembicaraan untuk variabel Pemupukan adalah [0 30]. Dan nilai domain untuk himpunan *fuzzy* adalah sebagai berikut:

Tabel 3.3 Himpunan *fuzzy* variabel Pemupukan

| Himpunan <i>Fuzzy</i> | Domain |
|-----------------------|-------------|
| Sedikit | 0 - 17.5 |
| Sedang | 12.5 - 22.5 |
| Banyak | 17.5 - 30 |



Gambar 3.12 Fungsi keanggotaan pada variabel Pemupukan.

Pada himpunan Sedikit menggunakan fungsi keanggotaan representasi linear turun, sehingga untuk menentukan derajat keanggotaannya menggunakan rumus berikut:

$$\mu_{PemupukanSedikit}(x) = \begin{cases} 1; & x \leq 12.5 \\ \frac{17.5-x}{17.5-12.5}; & 12.5 < x < 17.5 \\ 0; & x \geq 17.5 \end{cases} \quad (3.3)$$

Pada himpunan Sedang menggunakan fungsi keanggotaan representasi kurva segitiga, sehingga untuk menentukan derajat keanggotaannya menggunakan rumus berikut:

$$\mu_{PemupukanSedang}(x) = \begin{cases} 0; & x < 12.5 \\ \frac{x-12.5}{17.5-12.5}; & 12.5 \leq x \leq 17.5 \\ \frac{22.5-x}{22.5-17.5}; & 17.5 \leq x \leq 22.5 \\ 1; & x > 22.5 \end{cases} \quad (3.4)$$

Sedangkan pada himpunan Banyak menggunakan fungsi keanggotaan representasi linear naik, sehingga untuk menentukan derajat keanggotaannya menggunakan rumus berikut:

$$\mu_{PemupukanBanyak}(x) = \begin{cases} 0; & x \leq 17.5 \\ \frac{x-17.5}{22.5-17.5}; & 17.5 < x < 22.5 \\ 1; & x \geq 22.5 \end{cases} \quad (3.5)$$

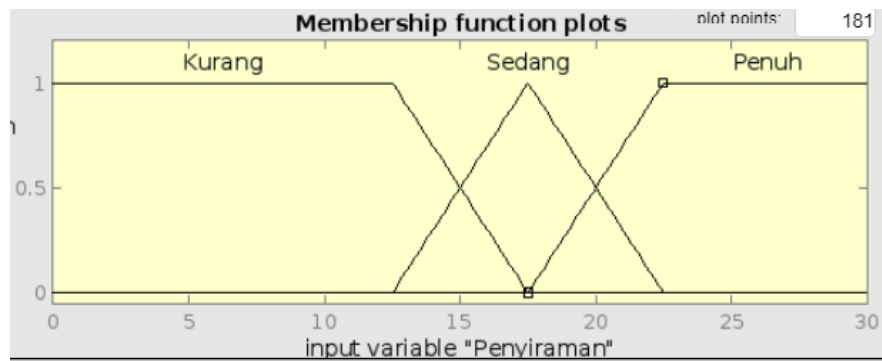
3. Variabel Penyiraman

Variabel Penyiraman memiliki tiga himpunan *fuzzy*, yaitu Kurang, Sedang dan Penuh. Nilai semesta pembicaraan untuk variabel

Penyiraman adalah [0 30]. Dan nilai domain untuk himpunan *fuzzy* adalah sebagai berikut:

Tabel 3.4 Himpunan *fuzzy* variabel Penyiraman

| Himpunan <i>Fuzzy</i> | Domain |
|-----------------------|-------------|
| Kurang | 0 – 17.5 |
| Sedang | 12.5 – 22.5 |
| Penuh | 17.5 - 30 |



Gambar 3.13 Fungsi keanggotaan pada variabel Penyiraman.

Pada himpunan Kurang menggunakan fungsi keanggotaan representasi linear turun, sehingga untuk menentukan derajat keanggotannya menggunakan rumus berikut:

$$\mu_{\text{PenyiramanKurang}}(x) = \begin{cases} 1; & x \leq 12.5 \\ \frac{17.5-x}{17.5-12.5}; & 12.5 < x < 17.5 \\ 0; & x \geq 17.5 \end{cases} \quad (3.6)$$

Pada himpunan Sedang menggunakan fungsi keanggotaan representasi kurva segitiga, sehingga untuk menentukan derajat keanggotaannya menggunakan rumus berikut:

$$\mu_{\text{PenyiramanSedang}}(x) = \begin{cases} 0; & x < 17.5 \\ \frac{x-12.5}{17.5-12.5}; & 12.5 \leq x \leq 17.5 \\ \frac{22.5-x}{22.5-17.5}; & 17.5 \leq x \leq 22.5 \\ 1; & x > 22.5 \end{cases} \quad (3.7)$$

Pada himpunan Penuh menggunakan fungsi keanggotaan representasi linear naik, sehingga untuk menentukan derajat keanggotaannya menggunakan rumus berikut:

$$\mu_{\text{PenyiramanPenuh}}(x) = \begin{cases} 0; & x \leq 17.5 \\ \frac{x-17.5}{22.5-17.5}; & 17.5 < x < 22.5 \\ 1; & x \geq 22.5 \end{cases} \quad (3.8)$$

4. Variabel Hasil Tanaman

Variabel Hasil Tanaman dibagi menjadi tiga himpunan *fuzzy*: Kering, Standar, dan Bagus. Variabel Hasil Tanaman merupakan *output* dalam proses *fuzzy* ini. Pada variabel Hasil Tanaman semua himpunan menggunakan fungsi keanggotaan *singleton*. *Singleton* merupakan sebuah himpunan *fuzzy* di mana fungsi keanggotaan yang pada titik tertentu memiliki nilai dan di luar titik adalah 0.

3.2.2.3 Fuzzy Rules

Model *fuzzy* yang diterapkan adalah *fuzzy* Sugeno *orde-nol* dengan aturan-aturan sebagai berikut:

- (R1) Jika (JENIS_BIBIT adalah BIASA) dan (PEMUPUKAN adalah SEDIKIT) dan (PENYIRAMAN adalah KURANG) maka (HASIL_TANAMAN adalah KERING) (1).
- (R2) Jika (JENIS_BIBIT adalah BIASA) dan (PEMUPUKAN adalah SEDIKIT) dan (PENYIRAMAN adalah SEDANG) maka (HASIL_TANAMAN adalah KERING) (1).
- (R3) Jika (JENIS_BIBIT adalah BIASA) dan (PEMUPUKAN adalah SEDIKIT) dan (PENYIRAMAN adalah PENUH) maka (HASIL_TANAMAN adalah STANDAR) (2).
- (R4) Jika (JENIS_BIBIT adalah BIASA) dan (PEMUPUKAN adalah SEDANG) dan (PENYIRAMAN adalah KURANG) maka (HASIL_TANAMAN adalah KERING) (1).
- (R5) Jika (JENIS_BIBIT adalah BIASA) dan (PEMUPUKAN adalah SEDANG) dan (PENYIRAMAN adalah SEDANG) maka (HASIL_TANAMAN adalah STANDAR) (2).
- (R6) Jika (JENIS_BIBIT adalah BIASA) dan (PEMUPUKAN adalah SEDANG) dan (PENYIRAMAN adalah PENUH) maka (HASIL_TANAMAN adalah BAGUS) (3).

- (R7) Jika (JENIS_BIBIT adalah BIASA) dan (PEMUPUKAN adalah BANYAK) dan (PENYIRAMAN adalah KURANG) maka (HASIL_TANAMAN adalah STANDAR) (2).
- (R8) Jika (JENIS_BIBIT adalah BIASA) dan (PEMUPUKAN adalah BANYAK) dan (PENYIRAMAN adalah SEDANG) maka (HASIL_TANAMAN adalah BAGUS) (3).
- (R9) Jika (JENIS_BIBIT adalah BIASA) dan (PEMUPUKAN adalah BANYAK) dan (PENYIRAMAN adalah PENUH) maka (HASIL_TANAMAN adalah BAGUS) (3).
- (R10) Jika (JENIS_BIBIT adalah UNGGUL) dan (PEMUPUKAN adalah SEDIKIT) dan (PENYIRAMAN adalah KURANG) maka (HASIL_TANAMAN adalah KERING) (1).
- (R11) Jika (JENIS_BIBIT adalah UNGGUL) dan (PEMUPUKAN adalah SEDIKIT) dan (PENYIRAMAN adalah SEDANG) maka (HASIL_TANAMAN adalah STANDAR) (2).
- (R12) Jika (JENIS_BIBIT adalah UNGGUL) dan (PEMUPUKAN adalah SEDIKIT) dan (PENYIRAMAN adalah PENUH) maka (HASIL_TANAMAN adalah BAGUS) (3).
- (R13) Jika (JENIS_BIBIT adalah UNGGUL) dan (PEMUPUKAN adalah SEDANG) dan (PENYIRAMAN adalah KURANG) maka (HASIL_TANAMAN adalah STANDAR) (2).

(R14) Jika (JENIS_BIBIT adalah UNGGUL) dan (PEMUPUKAN adalah SEDANG) dan (PENYIRAMAN adalah SEDANG) maka (HASIL_TANAMAN adalah BAGUS) (3).

(R15) Jika (JENIS_BIBIT adalah UNGGUL) dan (PEMUPUKAN adalah SEDANG) dan (PENYIRAMAN adalah PENUH) maka (HASIL_TANAMAN adalah BAGUS) (3).

(R16) Jika (JENIS_BIBIT adalah UNGGUL) dan (PEMUPUKAN adalah BANYAK) dan (PENYIRAMAN adalah KURANG) maka (HASIL_TANAMAN adalah BAGUS) (3).

(R17) Jika (JENIS_BIBIT adalah UNGGUL) dan (PEMUPUKAN adalah BANYAK) dan (PENYIRAMAN adalah SEDANG) maka (HASIL_TANAMAN adalah BAGUS) (3).

(R18) Jika (JENIS_BIBIT adalah UNGGUL) dan (PEMUPUKAN adalah BANYAK) dan (PENYIRAMAN adalah PENUH) maka (HASIL_TANAMAN adalah BAGUS) (3).

3.2.2.4 Implikasi

Fungsi implikasi yang digunakan dalam *fuzzy* Sugeno adalah fungsi implikasi *min* (minimum). Dimana fungsi implikasi *min* akan di terapkan pada setiap *rules* yang dibuat sebelumnya. Untuk *output* dari fungsi implikasi *min* diperoleh dengan memilih variabel *input* dengan derajat keanggotaan terendah. Formulasi berikut digunakan untuk menggambarkan fungsi ini:

$$\alpha - \text{predikat}_i = \min (\mu_{\text{JenisBibit}_i}[\text{x}]; \mu_{\text{Pemupukan}_i}[\text{x}]; \mu_{\text{Penyiraman}_i}[\text{x}]) \quad (3.9)$$

Hasil dari fungsi implikasi untuk setiap *rules* digunakan dalam proses *defuzzifikasi*.

3.2.2.5 Defuzzyfikasi

Dalam teknik Sugeno, *defuzzifikasi* dilakukan dengan merata-ratakan hasil dari fungsi implikasi pada komposisi aturan yang diberikan, dengan hasil *weighted average*. Prosedur defuzzifikasi secara umum digambarkan sebagai berikut:

$$Z = \frac{\sum_{i=1}^n \alpha\text{-predikat}_i z_i}{\sum_{i=1}^n \alpha\text{-predikat}_i} \quad (3.10)$$

Di mana:

z = Keluatan (*output*) akhir.

n = Jumlah aturan.

α – predikat_{*i*} = *Fire Strength* hasil fungsi implikasi aturan ke-*i*.

z_{*i*} = Keluaran (*output*) dari aturan ke-*i*.

Nilai **Z** yang dinyatakan sebagai nilai desimal akan dikonversi ke nilai satuan. Pedoman berikut akan digunakan untuk hasil pembulatan ke atas nilai **z** ini sebagai parameter untuk menghitung *output* hasil panen.

1. Jika hasil pembulatan dari nilai **z** sama dengan 1, maka hasil tanaman adalah Kering.
2. Jika hasil pembulatan dari nilai **z** sama dengan 2, maka hasil tanaman adalah Standar.
3. Jika hasil pembulatan dari nilai **z** sama dengan 3, maka hasil tanaman adalah Bagus.

3.2 Contoh Perhitungan *Fuzzy Sugeno*

3.2.1 Perhitungan Manual

Player menanam tanaman dengan jenis bibit yang memiliki nilai kualitas 5, lalu *player* memberi pupuk dengan nilai 20 dan penyiraman dengan nilai 10. Apa hasil tanaman yang akan diperoleh oleh *player*?

3.2.1.1 *Fuzzyfikasi*

Tahap pertama yaitu menentukan nilai *crisp* dari Jenis Bibit, Pemupukan, Penyiraman ke dalam himpunan *fuzzy* dan menentukan derajat keanggotaannya.

- a. Derajat keanggotaan variabel Jenis Bibit:

Pada perhitungan variabel Jenis Bibit dengan nilai $x = 5$ terhadap himpunan biasa dan himpunan unggul:

$$\mu_{\text{JenisBibitBiasa}}(x) = \frac{6 - x}{6 - 4}; 4 < x < 6$$

$$\mu_{\text{JenisBibitUnggul}}(x) = \frac{x - 4}{6 - 4}; 4 < x < 6$$

Hasil perhitungan menggunakan rumus linear turun, dan linear naik diperoleh nilai derajat keanggotaan Jenis Bibit biasa dan unggul adalah:

$$\mu_{\text{JenisBibitBiasa}}(5) = 0,5$$

$$\mu_{\text{JenisBibitUnggul}}(5) = 0,5$$

- b. Derajat keanggotaan variabel Pemupukan:

Pada perhitungan variabel Pemupukan dengan nilai $x = 20$ terhadap himpunan sedikit, himpunan sedang dan himpunan banyak:

$$\mu_{\text{PemupukanSedikit}}(x) = 0; x \geq 17.5$$

$$\mu_{\text{PemupukanSedang}}(x) = \frac{22.5 - x}{22.5 - 17.5}; 17.5 \leq x \leq 22.5$$

$$\mu_{\text{PemupukanBanyak}}(x) = \frac{x - 17.5}{22.5 - 17.5}; 17.5 < x < 22.5$$

Hasil perhitungan menggunakan rumus linear turun, kurva segitiga dan linear naik diperoleh nilai derajat keanggotaan Pemupukan sedikit, sedang dan banyak adalah:

$$\mu_{\text{PemupukanSedikit}}(20) = 0$$

$$\mu_{\text{PemupukanSedang}}(20) = 0,5$$

$$\mu_{\text{PemupukanBanyak}}(20) = 0,5$$

c. Derajat keanggotaan variabel Penyiraman:

Pada perhitungan variabel Penyiraman dengan nilai $x = 10$ terhadap himpunan kurang, himpunan sedang dan himpunan penuh:

$$\mu_{\text{PenyiramanKurang}}(x) = 1; x \leq 12.5$$

$$\mu_{\text{PenyiramanSedang}}(x) = 0; x < 17.5$$

$$\mu_{\text{PenyiramanPenuh}}(x) = 0; x \leq 17.5$$

Hasil perhitungan menggunakan rumus linear turun, kurva segitiga dan linear naik diperoleh nilai derajat keanggotaan Penyiramann kurang, sedang dan penuh adalah:

$$\mu_{\text{PenyiramanKurang}}(10) = 1$$

$$\mu_{\text{PenyiramanSedang}}(10) = 0$$

$$\mu_{\text{PenyiramanPenuh}}(10) = 0$$

3.2.1.2 Implikasi

Fungsi implikasi minimum kemudian diterapkan ke dalam *rules* yang telah ditentukan sehingga diperoleh nilai α – *predikat* dari setiap *rules*.

$$\begin{aligned}
 [\text{R1}] \alpha - \text{predikat}_1 &= \min(\mu_{\text{JenisBibitBiasa}}[5]; \mu_{\text{PemupukanSedikit}}[20]; \\
 &\quad \mu_{\text{PenyiramanKurang}}[10]) \\
 &= \min(0,5; 0; 1) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 [\text{R2}] \alpha - \text{predikat}_2 &= \min(\mu_{\text{JenisBibitBiasa}}[5]; \mu_{\text{PemupukanSedikit}}[20]; \\
 &\quad \mu_{\text{PenyiramanSedang}}[10]) \\
 &= \min(0,5; 0; 0) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 [\text{R3}] \alpha - \text{predikat}_3 &= \min(\mu_{\text{JenisBibitBiasa}}[5]; \mu_{\text{PemupukanSedikit}}[20]; \\
 &\quad \mu_{\text{PenyiramanPenuh}}[10]) \\
 &= \min(0,5; 0; 0) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 [\text{R4}] \alpha - \text{predikat}_4 &= \min(\mu_{\text{JenisBibitBiasa}}[5]; \mu_{\text{PemupukanSedang}}[20]; \\
 &\quad \mu_{\text{PenyiramanKurang}}[10]) \\
 &= \min(0,5; 0,5; 1) \\
 &= 0,5
 \end{aligned}$$

$$\begin{aligned}
 [\text{R5}] \alpha - \text{predikat}_5 &= \min(\mu_{\text{JenisBibitBiasa}}[5]; \mu_{\text{PemupukanSedang}}[20]; \\
 &\quad \mu_{\text{PenyiramanSedang}}[10]) \\
 &= \min(0,5; 0,5; 0) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
[\text{R6}] \alpha - \text{predikat}_6 &= \min(\mu_{\text{JenisBibitBiasa}}[5]; \mu_{\text{PemupukanSedang}}[20]; \\
&\quad \mu_{\text{PenyiramanPenuh}}[10]) \\
&= \min(0,5; 0,5; 0) \\
&= 0
\end{aligned}$$

$$\begin{aligned}
[\text{R7}] \alpha - \text{predikat}_7 &= \min(\mu_{\text{JenisBibitBiasa}}[5]; \mu_{\text{PemupukanBanyak}}[20]; \\
&\quad \mu_{\text{PenyiramanKurang}}[10]) \\
&= \min(0,5; 0,5; 1) \\
&= 0,5
\end{aligned}$$

$$\begin{aligned}
[\text{R8}] \alpha - \text{predikat}_8 &= \min(\mu_{\text{JenisBibitBiasa}}[5]; \mu_{\text{PemupukanBanyak}}[20]; \\
&\quad \mu_{\text{PenyiramanSedang}}[10]) \\
&= \min(0,5; 0,5; 0) \\
&= 0
\end{aligned}$$

$$\begin{aligned}
[\text{R9}] \alpha - \text{predikat}_9 &= \min(\mu_{\text{JenisBibitBiasa}}[5]; \mu_{\text{PemupukanBanyak}}[20]; \\
&\quad \mu_{\text{PenyiramanPenuh}}[10]) \\
&= \min(0,5; 0,5; 0) \\
&= 0
\end{aligned}$$

$$\begin{aligned}
[\text{R10}] \alpha - \text{predikat}_{10} &= \min(\mu_{\text{JenisBibitUnggul}}[5]; \mu_{\text{PemupukanSedikit}}[20]; \\
&\quad \mu_{\text{PenyiramanKurang}}[10]) \\
&= \min(0,5; 0; 1) \\
&= 0
\end{aligned}$$

$$\begin{aligned}
[\text{R11}] \alpha - \text{predikat}_{11} &= \min(\mu_{\text{JenisBibitUnggul}}[5]; \mu_{\text{PemupukanSedikit}}[20]; \\
&\quad \mu_{\text{PenyiramanSedang}}[10]) \\
&= \min(0,5; 0; 0)
\end{aligned}$$

$$= 0$$

$$\begin{aligned} \text{[R12]} \alpha - \text{predikat}_{12} &= \min(\mu_{\text{JenisBibitUnggul}}[5]; \mu_{\text{PemupukanSedikit}}[20]; \\ &\quad \mu_{\text{PenyiramanPenuh}}[10]) \\ &= \min(0,5; 0; 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} \text{[R13]} \alpha - \text{predikat}_{13} &= \min(\mu_{\text{JenisBibitUnggul}}[5]; \\ &\quad \mu_{\text{PemupukanSedang}}[20]; \mu_{\text{PenyiramanKurang}}[10]) \\ &= \min(0,5; 0,5; 1) \\ &= 0,5 \end{aligned}$$

$$\begin{aligned} \text{[R14]} \alpha - \text{predikat}_{14} &= \min(\mu_{\text{JenisBibitUnggul}}[5]; \mu_{\text{PemupukanSedang}}[20]; \\ &\quad \mu_{\text{PenyiramanSedang}}[10]) \\ &= \min(0,5; 0,5; 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} \text{[R15]} \alpha - \text{predikat}_{15} &= \min(\mu_{\text{JenisBibitUnggul}}[5]; \mu_{\text{PemupukanSedang}}[20]; \\ &\quad \mu_{\text{PenyiramanPenuh}}[10]) \\ &= \min(0,5; 0,5; 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} \text{[R16]} \alpha - \text{predikat}_{16} &= \min(\mu_{\text{JenisBibitUnggul}}[5]; \mu_{\text{PemupukanBanyak}}[20]; \\ &\quad \mu_{\text{PenyiramanKurang}}[10]) \\ &= \min(0,5; 0,5; 1) \\ &= 0,5 \end{aligned}$$

$$\begin{aligned} \text{[R17]} \alpha - \text{predikat}_{17} &= \min(\mu_{\text{JenisBibitUnggul}}[5]; \mu_{\text{PemupukanBanyak}}[20]; \\ &\quad \mu_{\text{PenyiramanSedang}}[10]) \end{aligned}$$

$$= \min(0,5; 0,5; 0)$$

$$= 0$$

$$[R18] \alpha - predikat_{18} = \min(\mu_{JenisBibitUnggul}[5]; \mu_{PemupukanBanyak}[20]; \\ \mu_{PenyiramanPenuh}[10])$$

$$= \min(0,5; 0,5; 0)$$

$$= 0$$

3.2.1.3 Defuzzyfikasi

Langkah selanjutnya yaitu menentukan hasil nilai variabel linguistik dari aturan-aturan yang sudah dibuat. Setelah nilai $\alpha - predikat$ dari setiap aturan diperoleh, selanjutnya masuk pada proses *defuzzyfikasi* untuk memperoleh nilai z .

$$z = \frac{\sum_{i=1}^n \alpha - predikat_i z_i}{\sum_{i=1}^n \alpha - predikat_i} = \frac{\sum_{i=1}^{18} \alpha - predikat_i z_i}{\sum_{i=1}^{18} \alpha - predikat_i}$$

$$\begin{aligned} & \alpha - predikat_1 * z_1 + \alpha - predikat_2 * z_2 + \alpha - predikat_3 * z_3 + \\ & \alpha - predikat_4 * z_4 + \alpha - predikat_5 * z_5 + \alpha - predikat_6 * z_6 + \\ & \alpha - predikat_7 * z_7 + \alpha - predikat_8 * z_8 + \alpha - predikat_9 * z_9 + \\ & \alpha - predikat_{10} * z_{10} + \alpha - predikat_{11} * z_{11} + \alpha - predikat_{12} * z_{12} + \\ & \alpha - predikat_{13} * z_{13} + \alpha - predikat_{14} * z_{14} + \alpha - predikat_{15} * z_{15} + \\ & \alpha - predikat_{16} * z_{16} + \alpha - predikat_{17} * z_{17} + \alpha - predikat_{18} * z_{18} \\ = & \frac{\alpha - predikat_1 + \alpha - predikat_2 + \alpha - predikat_3 + \alpha - predikat_4 + \\ & \alpha - predikat_5 + \alpha - predikat_6 + \alpha - predikat_7 + \alpha - predikat_8 + \\ & \alpha - predikat_9 + \alpha - predikat_{10} + \alpha - predikat_{11} + \alpha - predikat_{12} + \\ & \alpha - predikat_{13} + \alpha - predikat_{14} + \alpha - predikat_{15} + \alpha - predikat_{16} + \\ & \alpha - predikat_{17} + \alpha - predikat_{18}}{\end{aligned}$$

$$\begin{aligned} & 0 * 1 + 0 * 1 + 0 * 2 + 0,5 * 1 + 0 * 2 + 0 * 3 + 0,5 * 2 + 0 * 3 + 0 * 3 + 0 * 1 + \\ & 0 * 2 + 0 * 3 + 0,5 * 2 + 0 * 3 + 0 * 3 + 0,5 * 3 + 0 * 3 + 0 * 3 \\ = & \frac{0 + 0 + 0 + 0,5 + 0 + 0 + 0,5 + 0 + 0 + 0 + 0 + 0 + 0,5 + 0 + 0 + 0,5 + 0 + 0}{\end{aligned}$$

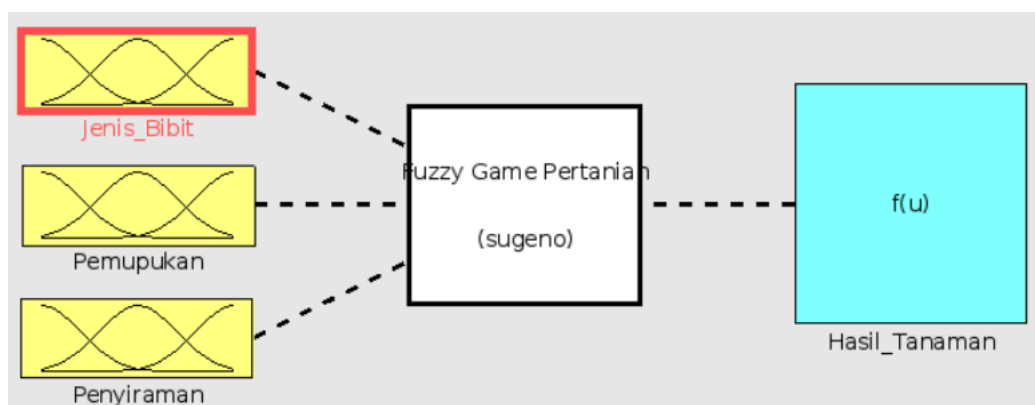
$$= \frac{4}{2} = 2$$

Sesuai dengan *rules* penentuan hasil tanaman yang sudah dibuat, dari perhitungan diperoleh hasil nilai z adalah 2, sehingga hasil tanaman yang ditanaman oleh *player* adalah standar.

3.2.2 Perhitungan Matlab

Untuk mendapatkan hasil yang valid maka juga dilakukan perhitungan menggunakan Matlab. Matlab merupakan bahasa pemrograman teknis yang sangat andal untuk melakukan proses simulasi dan visualisasi data. Matlab memiliki fitur *Fuzzy Logic Toolbox* yang dapat digunakan untuk simulasi metode *fuzzy* Sugeno ini. Untuk nilai *input* sama dengan nilai yang di-*input*-kan pada perhitungan manual. Dimana *player* memasukkan nilai *input* variabel Jenis Bibit yaitu 5, variabel Pemupukan yaitu 20 dan variabel Penyiraman yaitu 10.

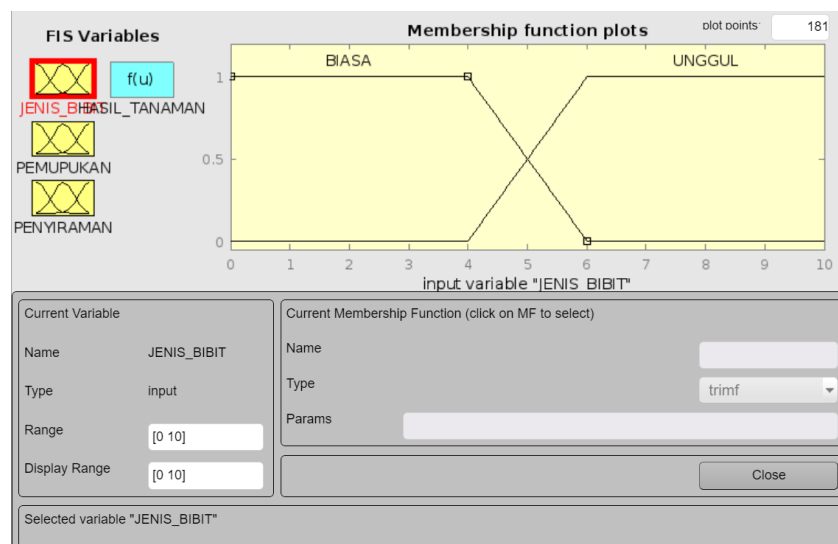
Langkah pertama yaitu memasukkan variabel *input* dan *output fuzzy*. Dimana pada penelitian ini terdapat 3 variabel *input* dan 1 variabel *output*.



Gambar 3.14 Matlab: Variabel *input* dan *output* pada proses *fuzzy*

Selanjutnya menentukan fungsi keanggotaan dari masing-masing variabel *input* dan *output fuzzy*.

Fungsi keanggotaan variabel Jenis Bibit memiliki dua himpunan yaitu himpunan Biasa dan himpunan Unggul. Semesta pembicara pada variabel Jenis Bibit yaitu $[0-10]$. Pada himpunan Biasa fungsi keanggotaan yang digunakan yaitu representasi linear turun. Dimana derajat keanggotaan 1 berada pada nilai 4 dan derajat keanggotaan 0 berada di nilai 6. Sebaliknya, pada himpunan Unggul fungsi keanggotaan yang digunakan yaitu representasi linear naik, dimana derajat keanggotaan 1 berada pada nilai 6 dan derajat keanggotaan 0 berada pada nilai 4.

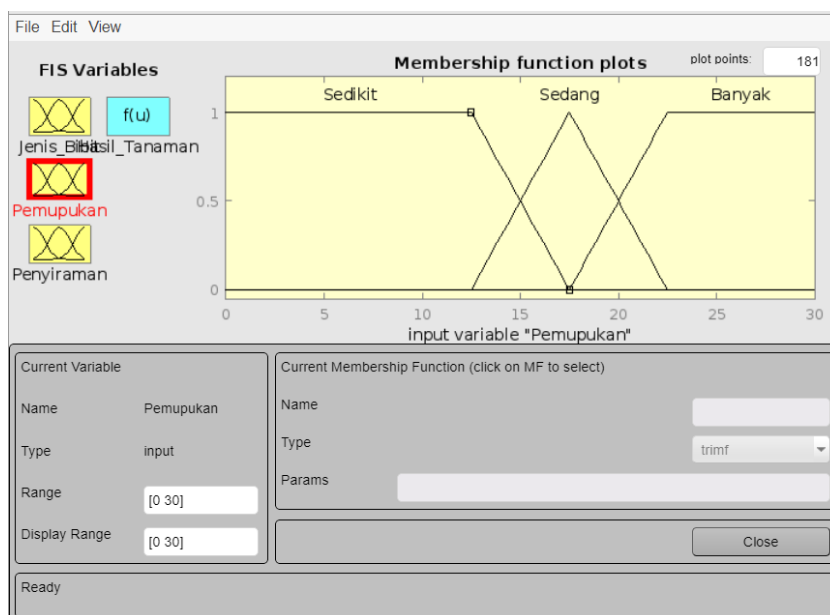


Gambar 3.15 Matlab: Fungsi keanggotaan variabel Jenis Bibit.

Fungsi keanggotaan variabel Pempukan memiliki tiga himpunan *fuzzy* yaitu himpunan Sedikit, himpunan Sedang dan himpunan Banyak. Semesta pembicara pada variabel Pemupukan yaitu $[0-30]$. Pada himpunan Sedikit fungsi keanggotaan yang digunakan yaitu representasi linear turun. Dimana derajat keanggotaan 1 berada pada nilai 12.5 dan derajat keanggotaan 0 berada di nilai 17.5.

Pada himpunan Sedang fungsi keanggotaan yang digunakan yaitu kurva segitiga, dimana derajat keanggotaan 1 berada pada nilai 22.5 dan derajat keanggotaan 0 berada pada nilai 17.5

Pada himpunan Banyak fungsi keanggotaan yang digunakan yaitu representasi linear naik, dimana derajat keanggotaan 1 berada pada nilai 17.5 dan derajat keanggotaan 0 berada pada nilai 12.5 dan 22.5.

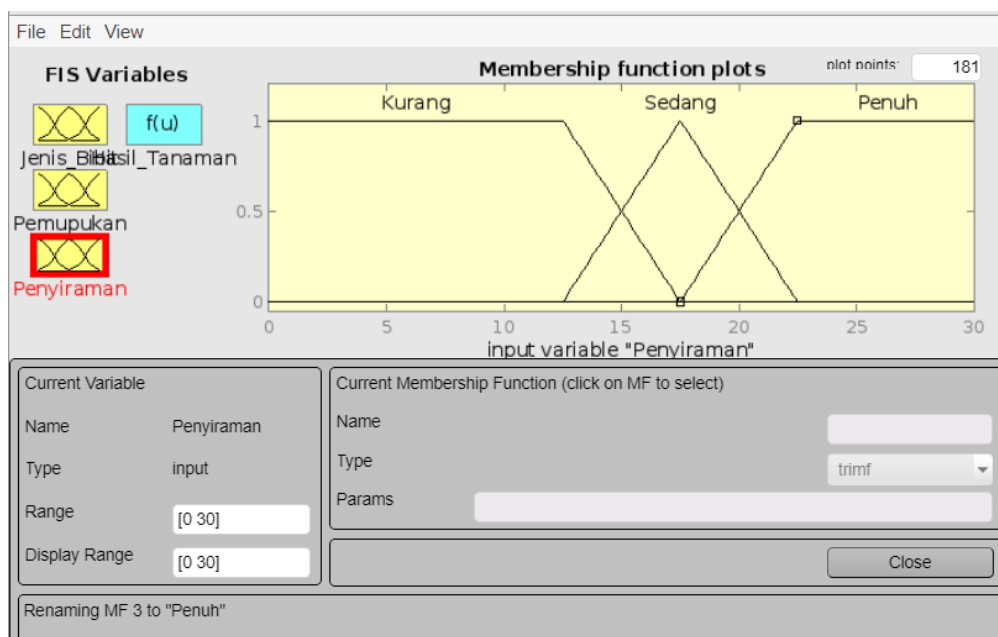


Gambar 3.16 Matlab: Fungsi keanggotaan variabel Pemupukan.

Fungsi keanggotaan variabel Penyiraman memiliki tiga himpunan yaitu himpunan Rendah, himpunan Sedang dan himpunan Tinggi. Semesta pembicara pada variabel Penyiraman yaitu [0-30]. Pada himpunan Kurang fungsi keanggotaan yang digunakan yaitu linear turun. Dimana derajat keanggotaan 1 berada pada nilai 12.5 dan derajat keanggotaan 0 berada di nilai 17.5.

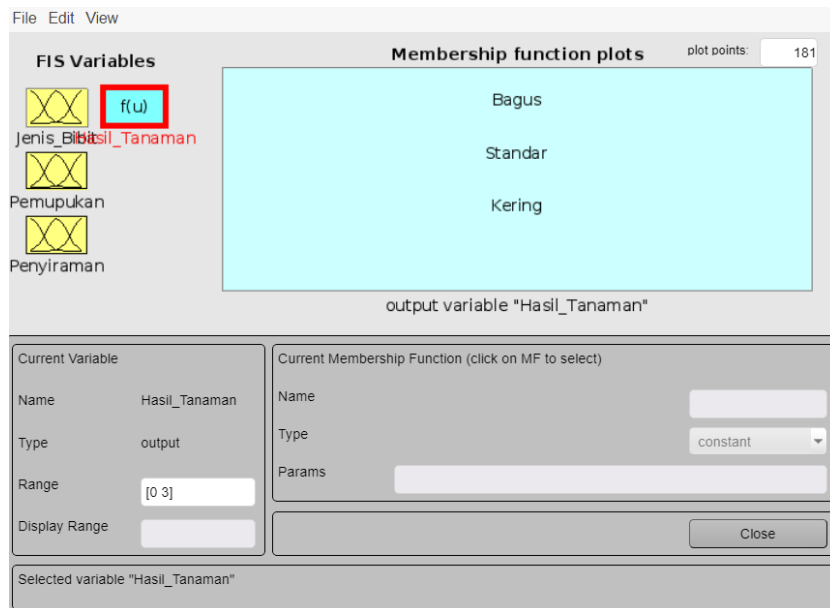
Pada himpunan Sedang fungsi keanggotaan yang digunakan yaitu segitiga, dimana derajat keanggotaan 1 berada pada nilai 22.5 dan derajat keanggotaan 0 berada pada nilai 17.5

Pada himpunan Penuh fungsi keanggotaan yang digunakan yaitu linear naik, dimana derajat keanggotaan 1 berada pada nilai 17.5 dan derajat keanggotaan 0 berada pada nilai 12.5 dan 22.5.



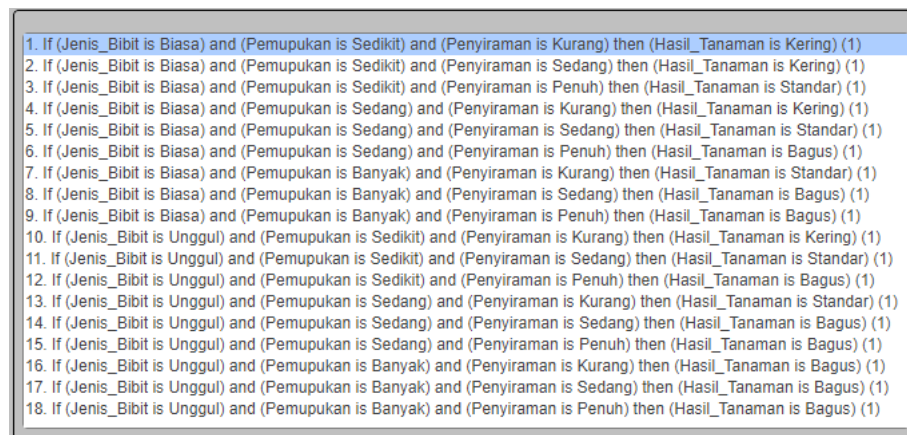
Gambar 3.17 Matlab: Fungsi keanggotaan variabel Penyiraman.

Fungsi keanggotaan Hasil Tanaman yaitu memiliki tiga himpunan yaitu Kering, Standar dan Bagus.



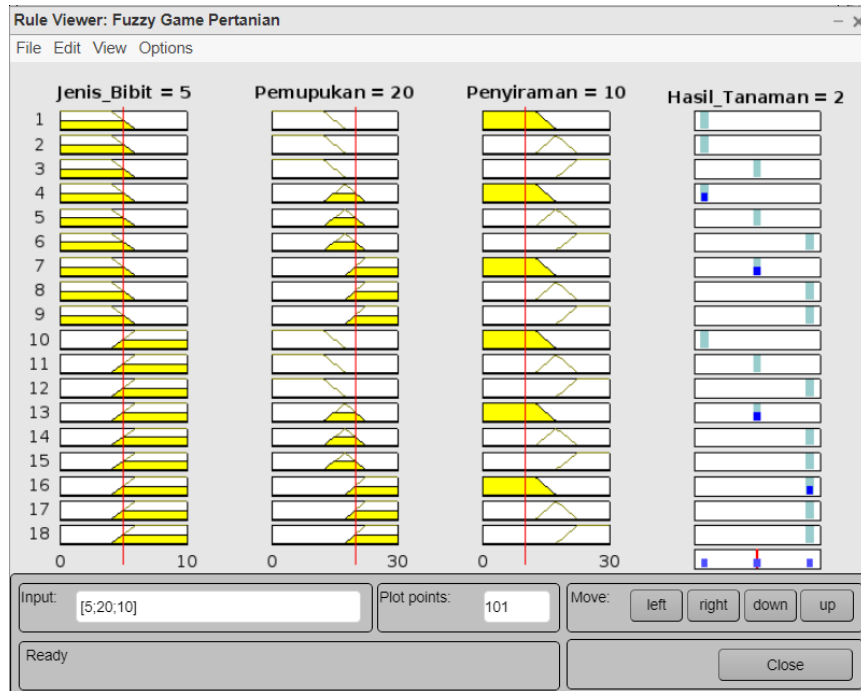
Gambar 3.18 Matlab: Fungsi keanggotaan variabel Hasil Tanaman.

Selanjutnya menambahkan *rules* yang akan diterapkan pada *fuzzy*. Pada *game* ini memiliki 18 *rules*.



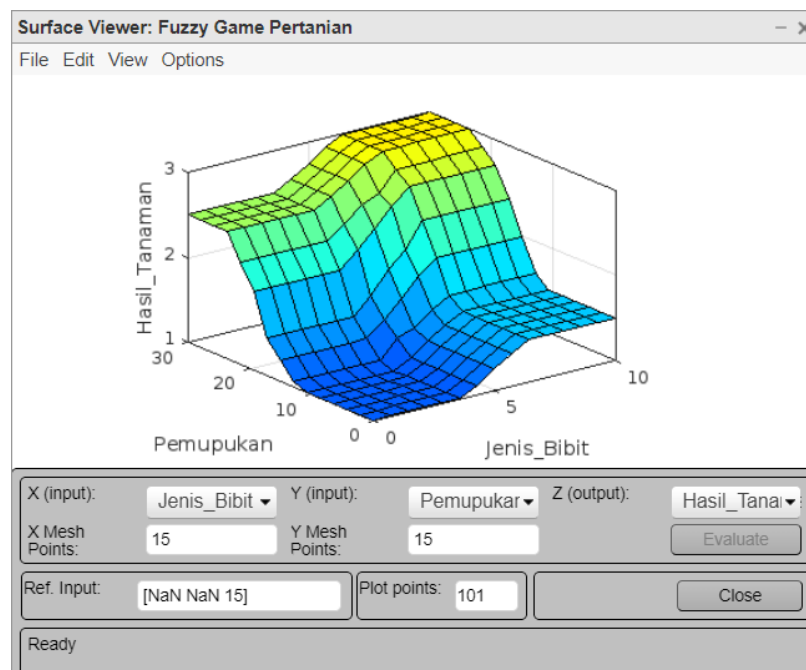
Gambar 3.19 Matlab: *Fuzzy rules*.

Selanjutnya untuk melihat hasil dari proses *fuzzy* yang sudah dibuat dapat dilihat di menu *View-Rules*, dimana nilai *input* disesuaikan dengan contoh kasus yaitu nilai Jenis Bibit 5, nilai Pemupukan 20 dan nilai Penyiraman 10.



Gambar 3.20 Matlab: Hasil perhitungan.

Untuk melihat grafik dari berbagai peluang nilai *input*, bisa menuju pada menu *View-Surface*.



Gambar 3.21 Matlab: *Surface Viewer*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Desain Sistem

Pada bab ini menjelaskan bagaimana metode HFSSM dan algoritma *fuzzy* Sugeno dipraktikkan setelah perencanaan yang matang. Pengujian dilakukan untuk menilai apakah program yang dikembangkan berjalan sesuai dengan tujuan penelitian. Pada pengembangan *game* ini penulis menggunakan seperangkat alat yaitu berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

4.1.1 Spesifikasi Perangkat Keras (*Hardware*)

Dalam pengembangan *game* ini perangkat keras digunakan untuk memberi dukungan sebagai fungsi utama dalam pengembangan *game* ini baik berupa *input*, pemrosesan, *output*, penyimpanan dll. Spesifikasi perangkat keras yang digunakan adalah sebagai berikut:

Tabel 4.1 Spesifikasi perangkat keras

| No. | Perangkat Keras | Spesifikasi |
|-----|-----------------------------|--|
| 1. | <i>Monitor</i> | 13 inch |
| 2. | <i>SSD</i> | 500 GB |
| 3. | <i>RAM</i> | 8 GB |
| 4. | <i>Processor</i> | <i>AMD Ryzen 3 3250U with Radeon Graphics 2.60 GHz</i> |
| 5. | <i>Mouse & Keyboard</i> | <i>Active</i> |
| 6. | <i>Speaker</i> | <i>Active</i> |

4.1.2 Spesifikasi Perangkat Lunak (*Software*)

Dalam pengembangan *game* ini perangkat lunak digunakan untuk melakukan perintah program untuk menjalankan kebutuhan pada pengembangan *game* ini. Spesifikasi perangkat lunak yang digunakan pada pengembangan *game* ini adalah sebagai berikut:

Tabel 4.2 Spesifikasi perangkat lunak

| No. | Perangkat Lunak | Spesifikasi |
|-----|----------------------|---------------------------|
| 1. | Sistem Operasi | <i>Windows 11 Home</i> |
| 2. | <i>Game Engine</i> | <i>Unity 2020 3.4</i> |
| 3. | Desain 3D | Make Human & Blender |
| 4. | Desain 2D | Photoshop CS6 |
| 5. | <i>Script Writer</i> | <i>Visual Studio 2019</i> |

4.2 Implementasi Permainan

Perancangan aplikasi permainan Saung Sawah ini diimplementasikan ke dalam perangkat *mobile* berbasis *Android*, dan sebagai hasilnya terdapat deskripsi setiap *scene* pada bab pembahasan ini. Hasil implementasi tersebut ditunjukkan berikut ini.




4.2.1 Tampilan Menu utama

Tampilan pada saat memulai permainan, *scene* pertama akan menampilkan menu utama seperti pada gambar 4.1. Pada tampilan menu utama terdapat pilihan tombol “*Start*”, “*Setting*” dan “*Exit*”. Untuk penjelasan dari masing-masing tombol pada tabel 4.3.



Gambar 4.1 Tampilan Menu utama

Tabel 4.3 Keterangan menu pada menu utama.

| No. | Menu | Keterangan |
|-----|---|--|
| 1. |  | Tombol “ <i>Start</i> ” untuk memulai permainan. |
| 2. |  | Tombol “ <i>Setting</i> ” akan mengarahkan ke menu pengaturan permainan. |
| 3 |  | Tombol “ <i>Exit</i> ” untuk keluar atau mengakhiri permainan |

Ketika pemain menekan tombol “*Start*” maka pemain akan diarahkan ke *scene 2* yaitu cerita awal dalam permainan.

4.2.2 Tampilan Cerita Awal Permainan

Setelah pemain memilih opsi "Start" dari menu utama, mereka akan diarahkan ke tampilan cerita awal dalam *game*. Dimana pemain akan disuguhkan adegan singkat sebelum menuju ke *gameplay*.



Gambar 4.2 Tampilan cerita awal permainan.

Pada *scene* ini pemain mendapatkan surat dari sang kakek berupa penawaran untuk bersedia meneruskan usaha pertanian kakeknya di desa. Setelah pemain menerima tawaran dari sang kakek, pemain akan diarahkan ke *gameplay* dari permainan ini.

4.2.3 Tampilan *Gameplay*

Tampilan *gameplay* merupakan tampilan inti dalam permainan, dimana pemain akan memulai permainannya.

4.2.3.1 Tampilan *Scene Dialog*

Pada awal *scene* akan ditampilkan dialog antara karakter utama dan NPC kakek Oland.



Gambar 4.3 Tampilan *scene* dialog NPC Kakek Oland dan karakter utama

4.2.3.2 Tampilan *Scene* Misi

Setelah itu pemain akan mendapatkan misi dari NPC Kakek Oland, dan diharapkan pemain segera menyelesaikan misi yang telah diberikan dan pemain bisa mendapatkan hadiah dari NPC Kakek Oland.



Gambar 4.4 Tampilan misi pemain.

4.2.3.3 Tampilan *Scene* Pemilihan Lahan dan Benih Tanaman

Di dalam *gameplay* pemain dapat melakukan aktifitas seperti bercocok tanam, berternak dan aktifitas lainnya. Pada saat bercocok tanaman pemain terlebih

dahulu memilih lahan yang akan dia tanami, setelah itu melakukan pemilihan benih yang tersedia pada panel.

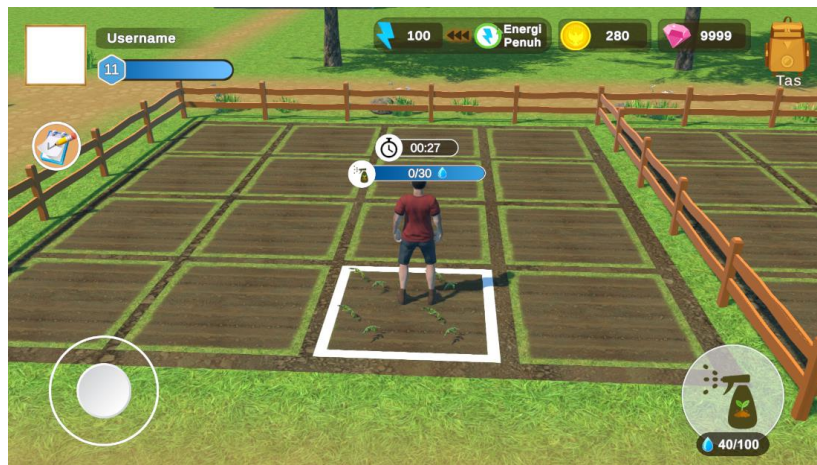


Gambar 4.5 Tampilan *scene* pemilihan lahan dan benih.

Setelah melakukan pemilihan lahan dan benih, selanjutnya pemain dapat melakukan aksi penanaman benih dengan mengklik tombol tanaman, sehingga akan muncul bibit tanaman pada lahan.

4.2.3.4 Tampilan *Scene* Fase Penyemprotan Pupuk

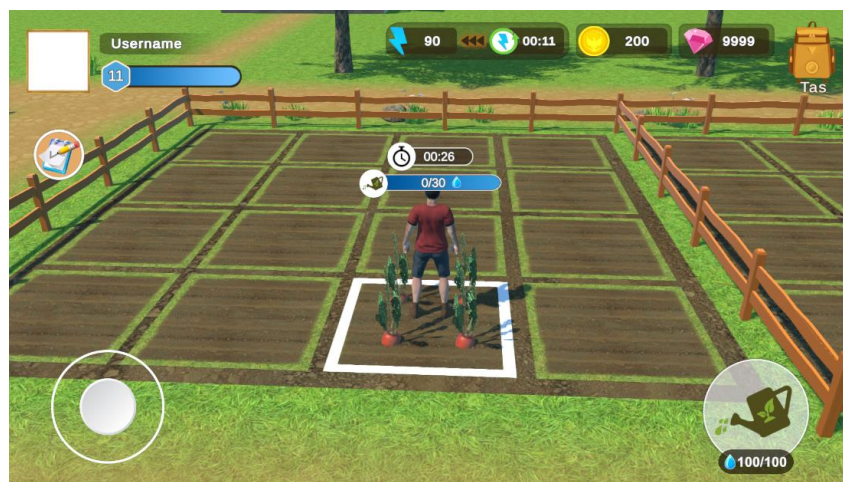
Setelah itu pemain akan memasuki fase penyemprotan pupuk pada tanaman, dan akan muncul bar waktu dan juga kebutuhan pupuk yang diperlukan agar tanaman bisa tumbuh dengan baik.



Gambar 4.6 Tampilan *scene* fase penyemprotan pupuk.

4.2.3.5 Tampilan *Scene* Fase Penyiraman

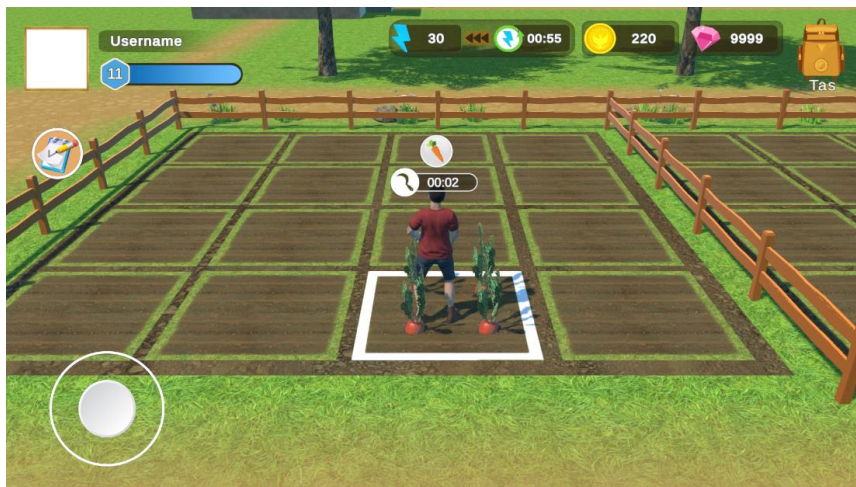
Setelah waktu pada fase penyemprotan pupuk telah selesai, maka selanjutnya pemain akan memasuki fase penyiraman tanaman, di mana pada fase penyiraman tanaman juga akan menampilkan waktu dan juga kebutuhan air.



Gambar 4.7 Tampilan *scene* fase penyiraman.

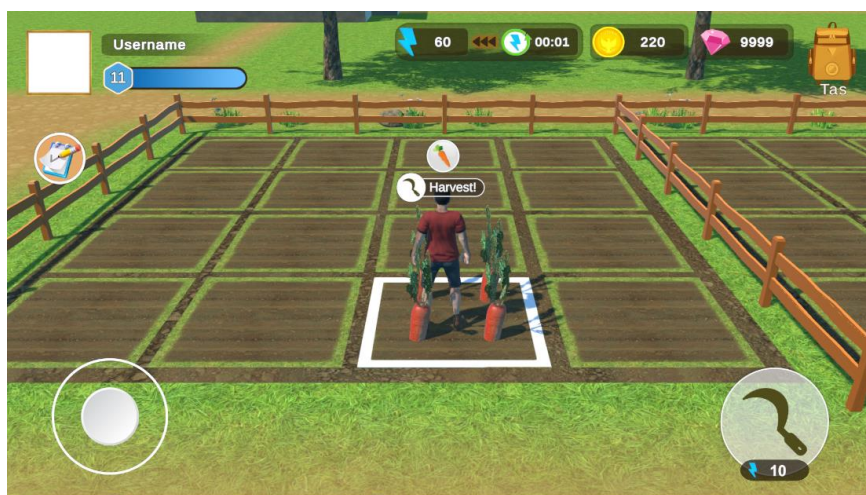
4.2.3.6 Tampilan *Scene* Fase *Pra-Panen* dan Panen

Setelah melewati fase penyiraman, maka pemain akan memasuki fase pematangan tanaman sebelum tanaman siap untuk dipanen.



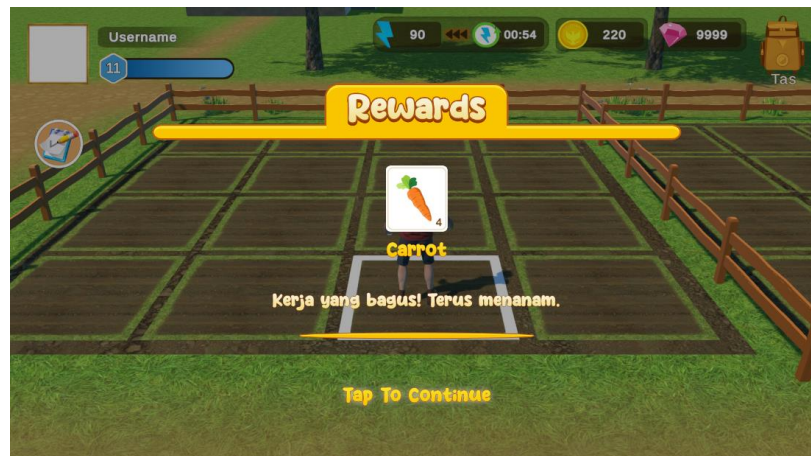
Gambar 4.8 Tampilan *scene* menunggu tanaman siap panen.

Ketika tanaman sudah siap untuk dipanen maka akan muncul panel “*harvest*” di atas tanaman tersebut.



Gambar 4.9 Tampilan *scene* tanaman siap panen.

Ketika pemain klik *button Harvest* maka akan muncul panel *rewards* tanaman.



Gambar 4.10 Tampilan panel *rewards* tanaman.

4.2.3.7 Tampilan *Scene* Lahan Peternakan

Selain bercocok tanam, pemain juga bisa beternak ayam, di mana pemain bisa memanfaatkan hasil telurnya dan menjualnya.



Gambar 4.11 Tampilan lahan peternakan.

Jika ayam memasuki waktu bertelur, maka dia akan menuju ke kandangnya untuk mengerami telurnya. Dan pemain bisa mengambil telurnya ketika telur sudah siap untuk diambil.



Gambar 4.12 Tampilan ayam sudah menghasilkan telur.

Ketika pemain mengklik *button* Telur maka akan muncul panel *Rewards Eggs*.



Gambar 4.13 Tampilan panel mendapatkan telur.

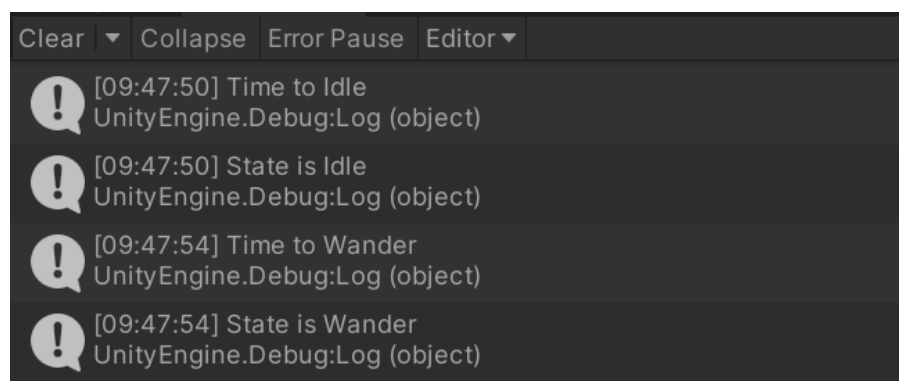
4.3 Uji Coba

Pada tahap uji coba dilakukan untuk mengetahui seberapa valid metode yang diusulkan dapat diimplementasikan pada *game Saung Sawah*.

4.3.1 Uji Coba FSM Pada NPC

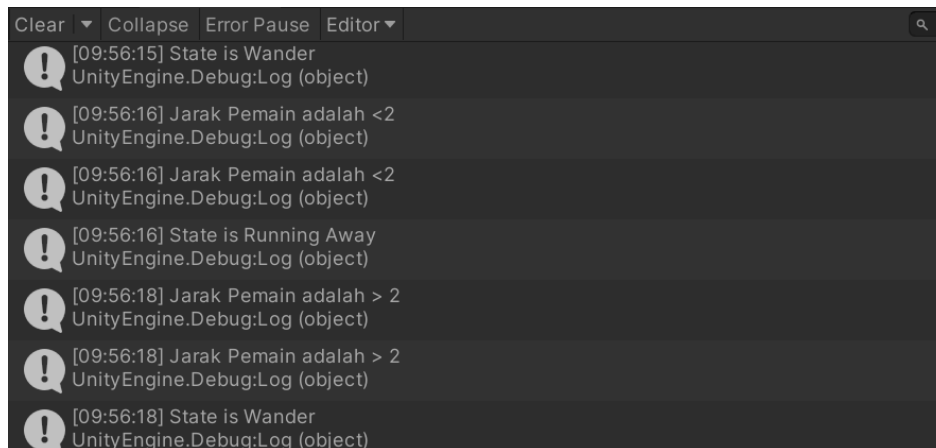
4.3.1.1 FSM Pada NPC Ayam

Pada pengujian FSM terhadap NPC Ayam ini dilakukan untuk menentukan desain perilaku NPC Ayam di dalam *game*. Dalam FSM NPC Ayam, pada saat sistem dimulai NPC Ayam berada pada *state idle* ketika memasuki *time to wander* maka NPC Ayam akan bertransisi ke *state wander* ketika memasuki *time to idle*, NPC Ayam akan kembali ke *state idle*. Pada saat jarak NPC Ayam dan *player* kurang dari 2, maka NPC Ayam akan bertransisi ke *state running away* dan menjauh dari *player*. Ketika jarak NPC Ayam dan *player* lebih dari 2 maka NPC Ayam akan kembali ke *state idle* atau *state wander* tergantung *state* yang berjalan sebelum transisi ke *state running away*. Ketika memasuki fase bertelur maka NPC Ayam akan menuju kandang dan bertransisi ke *state laying egg*.

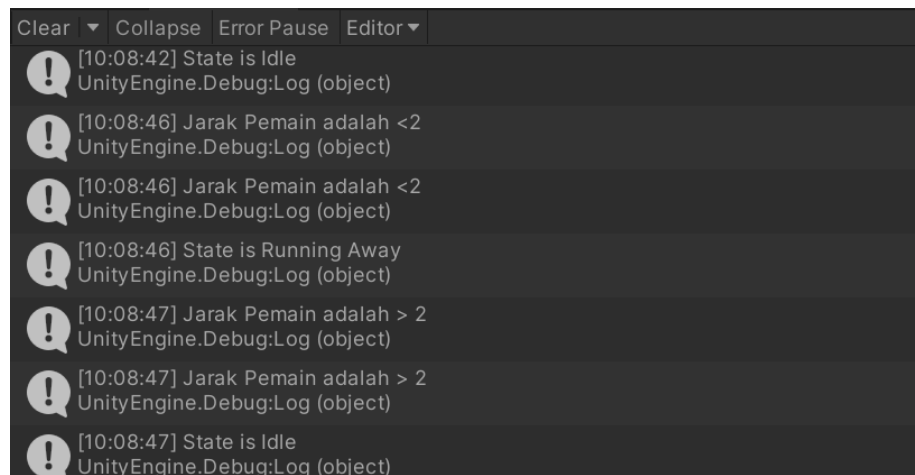


Gambar 4.14 Unity: Console perubahan *idle state* ke *wander state*.

Pada Gambar 4.14 adalah hasil ketika *idle state* bertransisi ke *wander state*. Ketika jarak *player* kurang dari 2 dari NPC Ayam, maka NPC Ayam akan melakukan *running away state*. Ketika jarak pemain dan NPC Ayam lebih dari 2, maka akan kembali ke *state awal*, sebelum *running away state*.

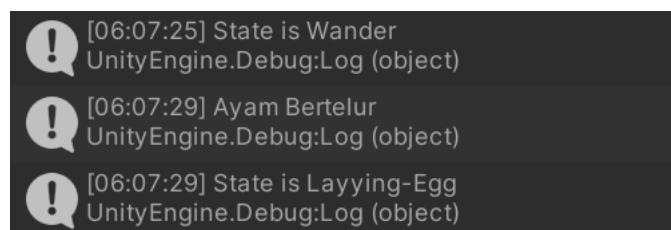


Gambar 4.15 Unity: Console perubahan wander state ke running away state.



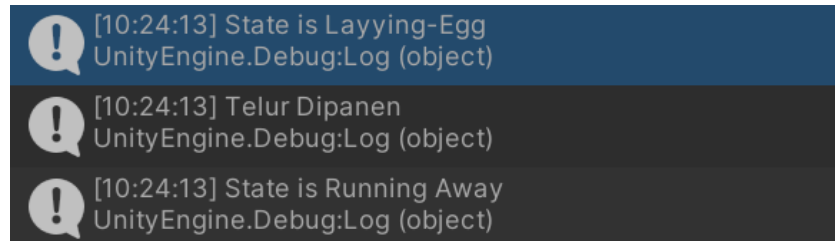
Gambar 4.16 Unity: Console perubahan Idle State ke RunningAway State.

Pada kondisi ketika ayam bertelur, dia akan memasuki *state Laying-Egg*.



Gambar 4.17 Unity: Console perubahan wander state ke laying-egg state.

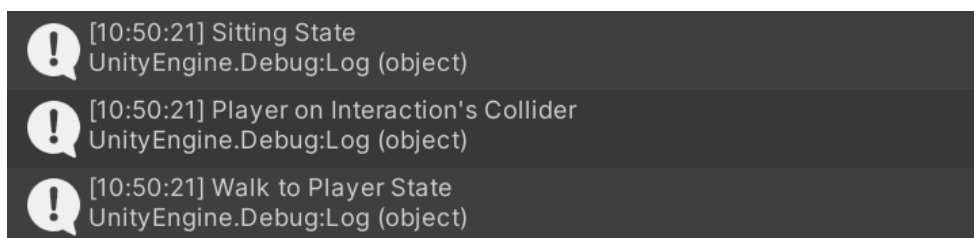
Dan ketika telur telah diambil oleh pemain, maka NPC Ayam akan kembali pada *state Wander State*.



Gambar 4.18 Unity: Console perubahan *state* ketika telur dipanen.

4.3.1.2 FSM pada NPC Kakek Oland

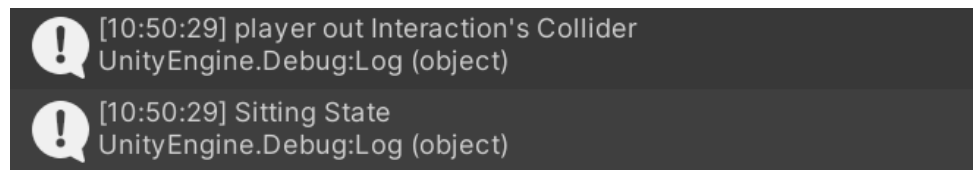
Dalam FSM NPC Kakek Oland, *state* awal yaitu *sitting state*, ketika *player* berada di *interaction's collider* maka NPC Kakek Oland akan melakukan transisi ke *walk to player state*, ketika NPC berada di *player's collider* maka NPC Kakek Oland akan melakukan transisi ke *interaction state* dan akan mengaktifkan *button mission*. Dimana ketika *player* mengklik *button mission* maka akan menampilkan panel misi. Ketika *player* tidak berada pada *interaction's collider* maka NPC Kakek Oland akan kembali ke *sitting state*.



Gambar 4.19 Unity: Console perubahan *state* ketika *player on interaction's collider*.



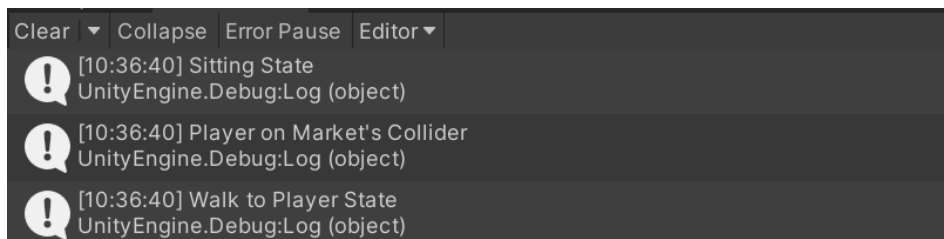
Gambar 4.20 Unity: Console perubahan state ketika NPC on player's collider.



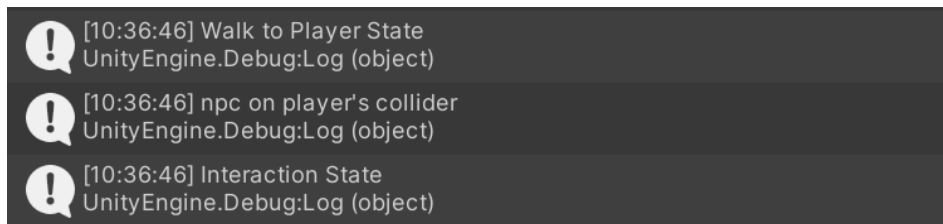
Gambar 4.21 Unity: Console perubahan state ketika player out interaction's collider.

4.3.1.3 FSM Pada NPC Om Giant

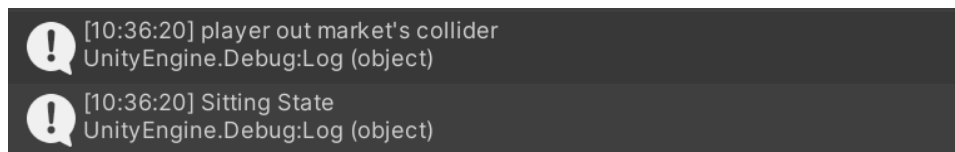
Dalam FSM NPC Om Giant, *state* awal yaitu *sitting state*, ketika *player* berada di *market's collider* maka NPC Om Giant akan melakukan transisi ke *walk to player state*, ketika NPC berada di *player's collider* maka NPC Om Giant akan melakukan transisi ke *interaction state* dan akan mengaktifkan *button market*. Dimana ketika *player* mengklik *button market* maka akan menampilkan panel *market*. Ketika *player* tidak berada pada *market's collider* maka NPC Om Giant akan kembali ke *sitting state*.



Gambar 4.22 Unity: Console perubahan state ketika player on market's collider.



Gambar 4.23 Unity: Console perubahan *state* ketika NPC on player's collider.



Gambar 4.24 Unity: Console perubahan *state* ketika player out market's collider.

Hasil uji coba dari penelitian metode FSM terhadap setiap NPC dengan berbagai nilai masukan variabel *input* dirangkum di bawah ini, beserta uraian hasil yang tercantum dalam Tabel 4.4 berikut:

Tabel 4.4 Hasil uji coba FSM pada NPC

| Uji Coba Ke- | NPC | Input | Output | Keterangan |
|--------------|------|---|---|------------|
| 1 | Ayam | Jika <i>time to idle</i> | Maka <i>Idle State</i> | Valid |
| 2 | Ayam | Jik <i>time to wander</i> | Maka <i>Wander State</i> | Valid |
| 3 | Ayam | Jika jarak <i>player</i> dan NPC ayam < 2 | Maka <i>Running Away State</i> | Valid |
| 4 | Ayam | Jika jarak <i>player</i> dan NPC ayam > 2 | Maka <i>Wander State</i> | Valid |
| 5 | Ayam | Jika fase bertelur | Maka NPC ayam menuju ke kandang dan <i>Laying-Egg State</i> | Valid |

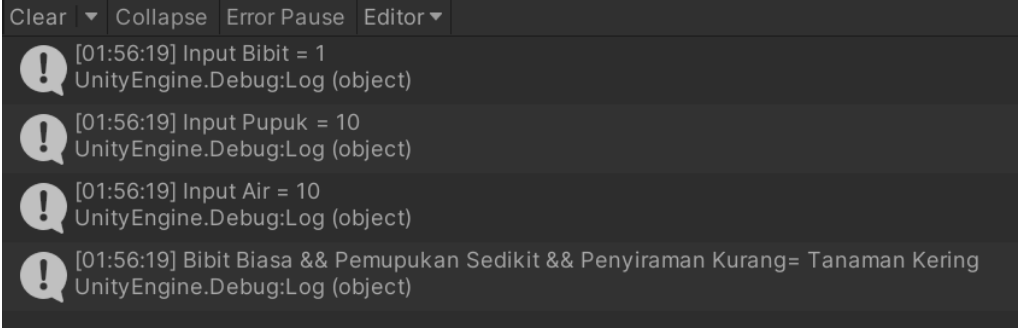
| | | | | |
|----|-------------|---|----------------------------------|-------|
| 6 | Ayam | Jika telur dipanen | Maka <i>Wander State</i> | Valid |
| 7 | Kakek Oland | Jika <i>player on interaction's collider</i> | Maka <i>Walk to Player State</i> | Valid |
| 8 | Kakek Oland | Jika NPC on <i>player's collider</i> | Maka <i>Interaction's State</i> | Valid |
| 9 | Kakek Oland | Jika <i>player out interaction's collider</i> | Maka <i>Sitting State</i> | Valid |
| 10 | Om Giant | Jika <i>player on market's collider</i> | Maka <i>Walk to Player State</i> | Valid |
| 11 | Om Giant | Jika NPC on <i>player's collider</i> | Maka <i>Interaction's State</i> | Valid |
| 12 | Om Giant | Jika <i>player out market's collider</i> | Maka <i>Sitting State</i> | Valid |

Dari Tabel 4.4 dapat kita lihat bahwa hasil validasi adalah 100%. Hasil ini dinyatakan sangat baik karena percobaan ini sesuai dengan metodologi FSM yang dirancang sehingga tidak mempengaruhi hasil validasi FSM dari setiap NPC pada *game Saung sawah* ini.

4.3.2 Uji Coba *Fuzzy Sugeno* Pada Sistem Tanaman

Uji coba algoritma *fuzzy Sugeno* dilakukan untuk menentukan hasil tanaman yang ditanam oleh pemain berdasarkan aturan yang telah dibuat sebelumnya. Beberapa pengujian dilakukan dengan menggunakan berbagai nilai variabel input. Nilai ini digunakan untuk menentukan hasil panen yang akan dicapai oleh pemain.

Pengujian ke-1 dilakukan dengan nilai *input* variabel Jenis Bibit 1, variabel Pemupukan 10 dan nilai *input* variabel Penyiraman 10. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Kering.



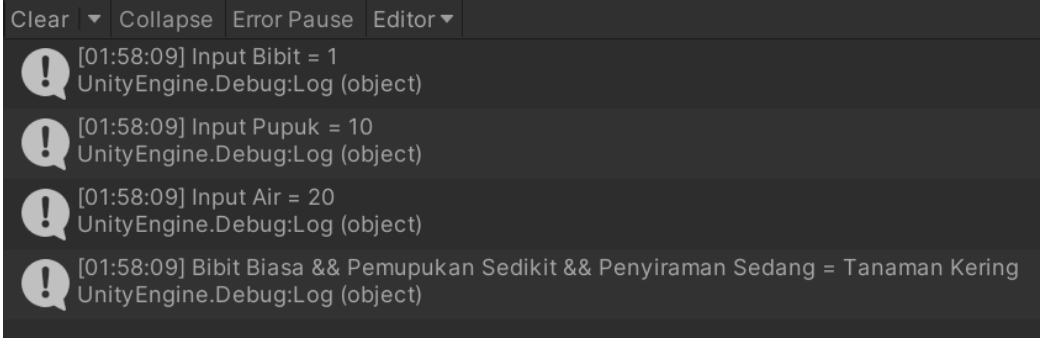
```

Clear ▼ Collapse Error Pause Editor ▼
[01:56:19] Input Bibit = 1
UnityEngine.Debug:Log (object)
[01:56:19] Input Pupuk = 10
UnityEngine.Debug:Log (object)
[01:56:19] Input Air = 10
UnityEngine.Debug:Log (object)
[01:56:19] Bibit Biasa & Pemupukan Sedikit & Penyiraman Kurang = Tanaman Kering
UnityEngine.Debug:Log (object)

```

Gambar 4.25 Unity: Console output hasil tanaman percobaan ke-1.

Pengujian ke-2 dilakukan dengan nilai *input* variabel Jenis Bibit 1, variabel Pemupukan 10 dan nilai *input* variabel Penyiraman 20. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Kering.



```

Clear ▼ Collapse Error Pause Editor ▼
[01:58:09] Input Bibit = 1
UnityEngine.Debug:Log (object)
[01:58:09] Input Pupuk = 10
UnityEngine.Debug:Log (object)
[01:58:09] Input Air = 20
UnityEngine.Debug:Log (object)
[01:58:09] Bibit Biasa & Pemupukan Sedikit & Penyiraman Sedang = Tanaman Kering
UnityEngine.Debug:Log (object)

```

Gambar 4.26 Unity: Console output hasil tanaman percobaan ke-2.

Pengujian ke-3 dilakukan dengan nilai *input* variabel Jenis Bibit 1, variabel Pemupukan 10 dan nilai *input* variabel Penyiraman 30. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Standar.

```

Clear ▼ Collapse Error Pause Editor ▼
[01:59:53] Input Bibit = 1
UnityEngine.Debug:Log (object)
[01:59:53] Input Pupuk = 10
UnityEngine.Debug:Log (object)
[01:59:53] Input Air = 30
UnityEngine.Debug:Log (object)
[01:59:53] Bibit Biasa && Pemupukan Sedikit && Penyiraman Penuh = Tanaman Standar
UnityEngine.Debug:Log (object)

```

Gambar 4.27 Unity: Console output hasil tanaman percobaan ke-3.

Pengujian ke-4 dilakukan dengan nilai *input* variabel Jenis Bibit 1, variabel Pemupukan 20 dan nilai *input* variabel Penyiraman 10. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Kering.

```

Clear ▼ Collapse Error Pause Editor ▼
[02:01:30] Input Bibit = 1
UnityEngine.Debug:Log (object)
[02:01:30] Input Pupuk = 20
UnityEngine.Debug:Log (object)
[02:01:30] Input Air = 10
UnityEngine.Debug:Log (object)
[02:01:30] Bibit Biasa && Pemupukan Sedang && Penyiraman Kurang = Tanaman Kering
UnityEngine.Debug:Log (object)

```

Gambar 4.28 Unity: Console output hasil tanaman percobaan ke-4.

Pengujian ke-5 dilakukan dengan nilai *input* variabel Jenis Bibit 1, variabel Pemupukan 20 dan nilai *input* variabel Penyiraman 20. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Standar.

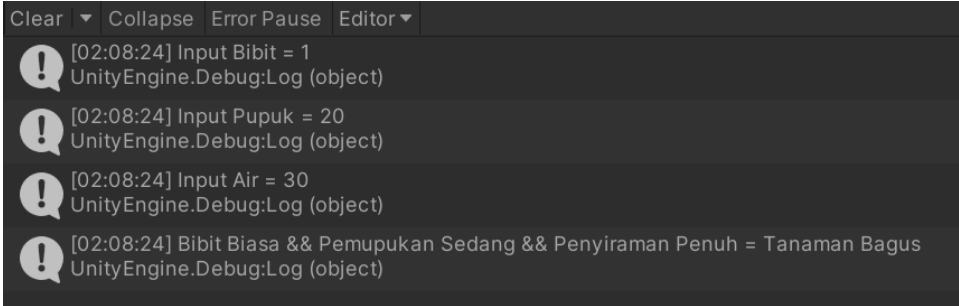
```

Clear ▼ Collapse Error Pause Editor ▼
[02:03:17] Input Bibit = 1
UnityEngine.Debug:Log (object)
[02:03:17] Input Pupuk = 20
UnityEngine.Debug:Log (object)
[02:03:17] Input Air = 20
UnityEngine.Debug:Log (object)
[02:03:17] Bibit Biasa && Pemupukan Sedang && Penyiraman Sedang = Tanaman Standar
UnityEngine.Debug:Log (object)

```

Gambar 4.29 Unity: *Console output* hasil tanaman percobaan ke-5

Pengujian ke-6 dilakukan dengan nilai *input* variabel Jenis Bibit 1, variabel Pemupukan 20 dan nilai *input* variabel Penyiraman 30. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Bagus.

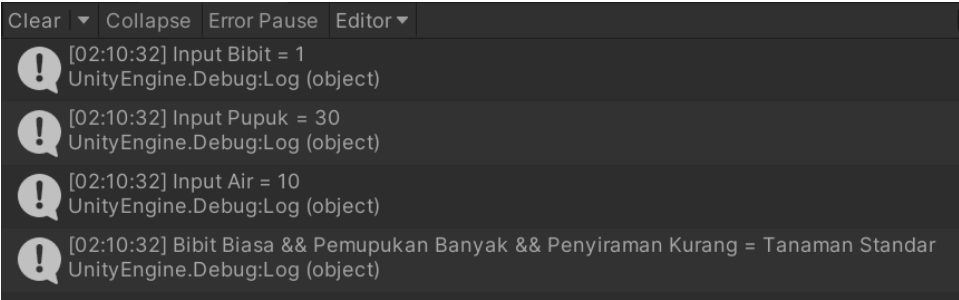


```

Clear ▼ Collapse Error Pause Editor ▼
[02:08:24] Input Bibit = 1
UnityEngine.Debug:Log (object)
[02:08:24] Input Pupuk = 20
UnityEngine.Debug:Log (object)
[02:08:24] Input Air = 30
UnityEngine.Debug:Log (object)
[02:08:24] Bibit Biasa && Pemupukan Sedang && Penyiraman Penuh = Tanaman Bagus
UnityEngine.Debug:Log (object)
  
```

Gambar 4.30 Unity: *Console output* hasil tanaman percobaan ke-6

Pengujian ke-7 dilakukan dengan nilai *input* variabel Jenis Bibit 1, variabel Pemupukan 30 dan nilai *input* variabel Penyiraman 10. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Standar.

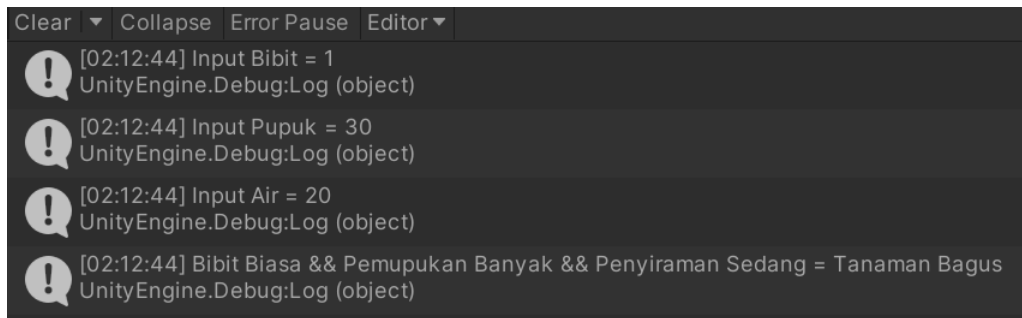


```

Clear ▼ Collapse Error Pause Editor ▼
[02:10:32] Input Bibit = 1
UnityEngine.Debug:Log (object)
[02:10:32] Input Pupuk = 30
UnityEngine.Debug:Log (object)
[02:10:32] Input Air = 10
UnityEngine.Debug:Log (object)
[02:10:32] Bibit Biasa && Pemupukan Banyak && Penyiraman Kurang = Tanaman Standar
UnityEngine.Debug:Log (object)
  
```

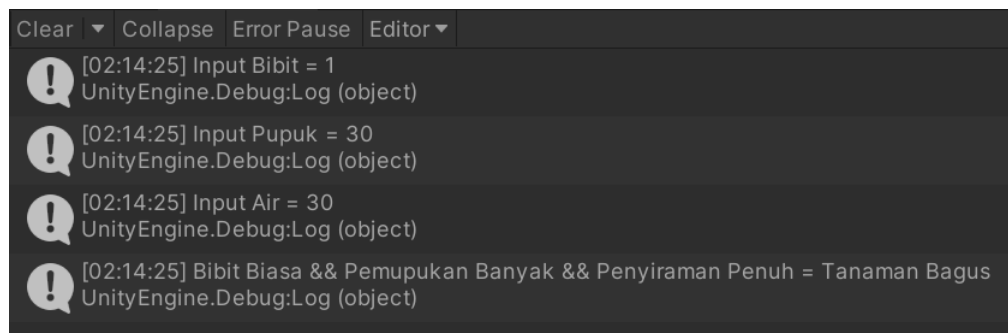
Gambar 4.31 Unity: *Console output* hasil tanaman percobaan ke-7

Pengujian ke-8 dilakukan dengan nilai *input* variabel Jenis Bibit 1, variabel Pemupukan 30 dan nilai *input* variabel Penyiraman 20. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Bagus.



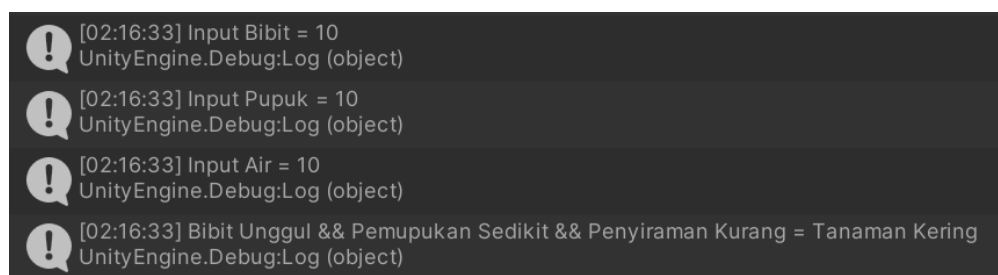
Gambar 4.32 Unity: Console output hasil tanaman percobaan ke-8

Pengujian ke-9 dilakukan dengan nilai *input* variabel Jenis Bibit 1, variabel Pemupukan 30 dan nilai *input* variabel Penyiraman 30. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Bagus.



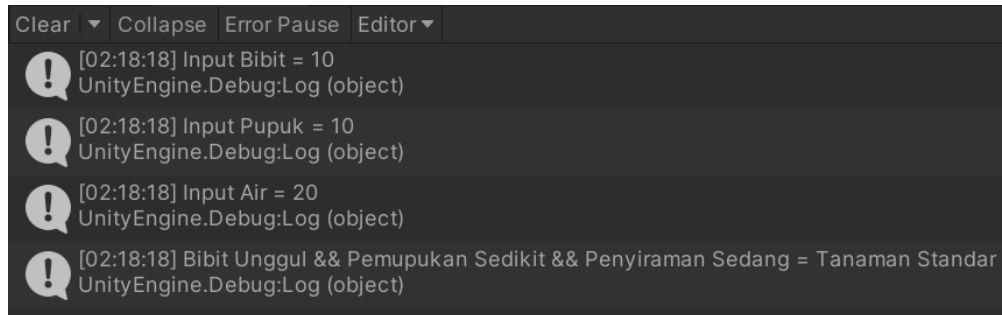
Gambar 4.33 Unity: Console output hasil tanaman percobaan ke-9

Pengujian ke-10 dilakukan dengan nilai *input* variabel Jenis Bibit 10, variabel Pemupukan 10 dan nilai *input* variabel Penyiraman 10. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Kering.



Gambar 4.34 Unity: Console output hasil tanaman percobaan ke-10.

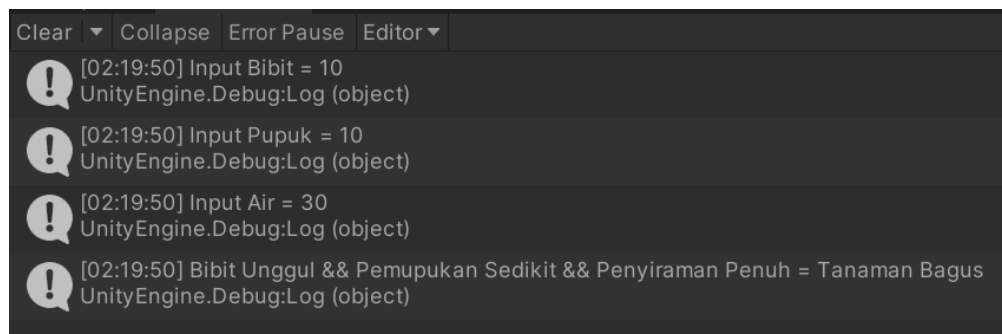
Pengujian ke-11 dilakukan dengan nilai *input* variabel Jenis Bibit 10, variabel Pemupukan 10 dan nilai *input* variabel Penyiraman 20. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Standar.



```
Clear ▼ Collapse Error Pause Editor ▼
[02:18:18] Input Bibit = 10
UnityEngine.Debug:Log (object)
[02:18:18] Input Pupuk = 10
UnityEngine.Debug:Log (object)
[02:18:18] Input Air = 20
UnityEngine.Debug:Log (object)
[02:18:18] Bibit Unggul && Pemupukan Sedikit && Penyiraman Sedang = Tanaman Standar
UnityEngine.Debug:Log (object)
```

Gambar 4.35 Unity: Console output hasil tanaman percobaan ke-11.

Pengujian ke-12 dilakukan dengan nilai *input* variabel Jenis Bibit 10, variabel Pemupukan 10 dan nilai *input* variabel Penyiraman 30. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Bagus.



```
Clear ▼ Collapse Error Pause Editor ▼
[02:19:50] Input Bibit = 10
UnityEngine.Debug:Log (object)
[02:19:50] Input Pupuk = 10
UnityEngine.Debug:Log (object)
[02:19:50] Input Air = 30
UnityEngine.Debug:Log (object)
[02:19:50] Bibit Unggul && Pemupukan Sedikit && Penyiraman Penuh = Tanaman Bagus
UnityEngine.Debug:Log (object)
```

Gambar 4.36 Unity: Console output hasil tanaman percobaan ke-12.

Pengujian ke-13 dilakukan dengan nilai *input* variabel Jenis Bibit 10, variabel Pemupukan 20 dan nilai *input* variabel Penyiraman 10. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Standar.

```

Clear ▼ Collapse Error Pause Editor ▼
[02:21:51] Input Bibit = 10
UnityEngine.Debug:Log (object)
[02:21:51] Input Pupuk = 20
UnityEngine.Debug:Log (object)
[02:21:51] Input Air = 10
UnityEngine.Debug:Log (object)
[02:21:51] Bibit Unggul && Pemupukan Sedang && Penyiraman Kurang= Tanaman Standar
UnityEngine.Debug:Log (object)

```

Gambar 4.37 Unity: Console output hasil tanaman percobaan ke-13.

Pengujian ke-14 dilakukan dengan nilai *input* variabel Jenis Bibit 10, variabel Pemupukan 20 dan nilai *input* variabel Penyiraman 20. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Bagus.

```

Clear ▼ Collapse Error Pause Editor ▼
[02:23:39] Input Bibit = 10
UnityEngine.Debug:Log (object)
[02:23:39] Input Pupuk = 20
UnityEngine.Debug:Log (object)
[02:23:39] Input Air = 20
UnityEngine.Debug:Log (object)
[02:23:39] Bibit Unggul && Pemupukan Sedang && Penyiraman Sedang = Tanaman Bagus
UnityEngine.Debug:Log (object)

```

Gambar 4.38 Unity: Console output hasil tanaman percobaan ke-14

. Pengujian ke-15 dilakukan dengan nilai *input* variabel Jenis Bibit 10, variabel Pemupukan 20 dan nilai *input* variabel Penyiraman 30. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Bagus.

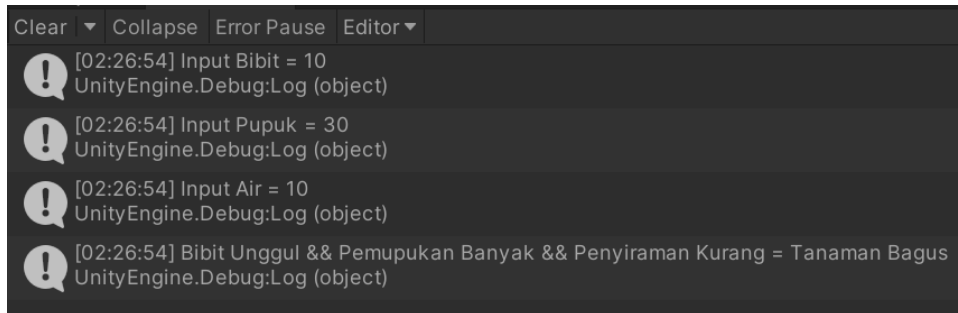
```

Clear ▼ Collapse Error Pause Editor ▼
[02:25:19] Input Bibit = 10
UnityEngine.Debug:Log (object)
[02:25:19] Input Pupuk = 20
UnityEngine.Debug:Log (object)
[02:25:19] Input Air = 30
UnityEngine.Debug:Log (object)
[02:25:19] Bibit Unggul && Pemupukan Sedang && Penyiraman Penuh = Tanaman Bagus
UnityEngine.Debug:Log (object)

```

Gambar 4.39 Unity: Console output hasil tanaman percobaan ke-15

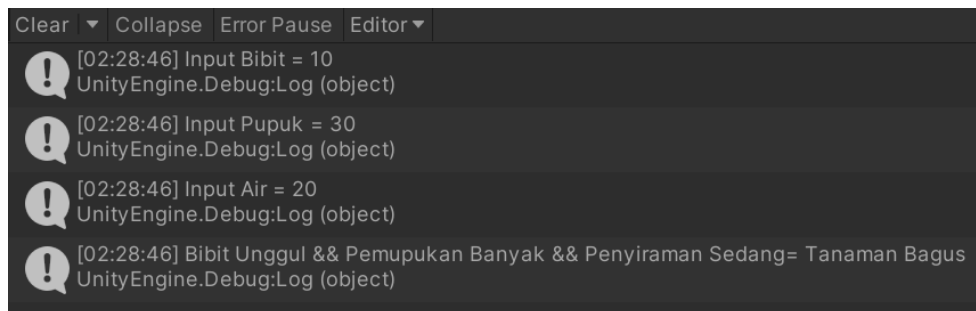
Pengujian ke-16 dilakukan dengan nilai *input* variabel Jenis Bibit 10, variabel Pemupukan 30 dan nilai *input* variabel Penyiraman 10. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Bagus.

A screenshot of the Unity console showing four log messages. The messages are: [02:26:54] Input Bibit = 10, [02:26:54] Input Pupuk = 30, [02:26:54] Input Air = 10, and [02:26:54] Bibit Unggul && Pemupukan Banyak && Penyiraman Kurang = Tanaman Bagus. Each message is preceded by a warning icon and the text 'UnityEngine.Debug:Log (object)'.

```
Clear ▼ Collapse Error Pause Editor ▼
[02:26:54] Input Bibit = 10
UnityEngine.Debug:Log (object)
[02:26:54] Input Pupuk = 30
UnityEngine.Debug:Log (object)
[02:26:54] Input Air = 10
UnityEngine.Debug:Log (object)
[02:26:54] Bibit Unggul && Pemupukan Banyak && Penyiraman Kurang = Tanaman Bagus
UnityEngine.Debug:Log (object)
```

Gambar 4.40 Unity: Console Output Hasil Tanaman Percobaan ke-16

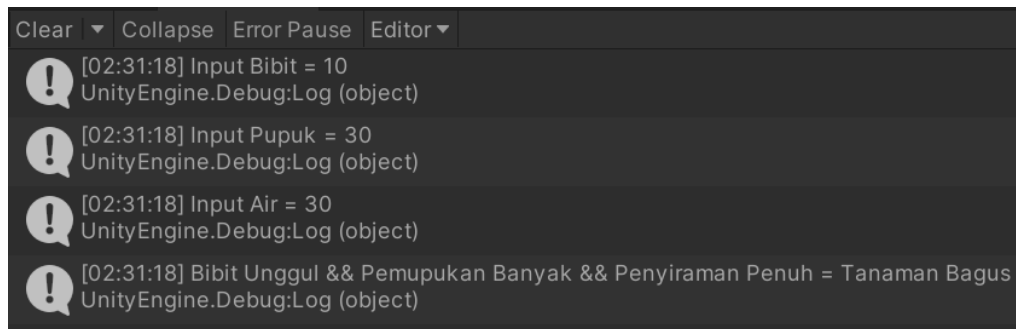
Pengujian ke-17 dilakukan dengan nilai *input* variabel Jenis Bibit 10, variabel Pemupukan 30 dan nilai *input* variabel Penyiraman 20. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Bagus.

A screenshot of the Unity console showing four log messages. The messages are: [02:28:46] Input Bibit = 10, [02:28:46] Input Pupuk = 30, [02:28:46] Input Air = 20, and [02:28:46] Bibit Unggul && Pemupukan Banyak && Penyiraman Sedang = Tanaman Bagus. Each message is preceded by a warning icon and the text 'UnityEngine.Debug:Log (object)'.

```
Clear ▼ Collapse Error Pause Editor ▼
[02:28:46] Input Bibit = 10
UnityEngine.Debug:Log (object)
[02:28:46] Input Pupuk = 30
UnityEngine.Debug:Log (object)
[02:28:46] Input Air = 20
UnityEngine.Debug:Log (object)
[02:28:46] Bibit Unggul && Pemupukan Banyak && Penyiraman Sedang = Tanaman Bagus
UnityEngine.Debug:Log (object)
```

Gambar 4.41 Unity: Console output hasil tanaman percobaan ke-17

Pengujian ke-18 dilakukan dengan nilai *input* variabel Jenis Bibit 10, variabel Pemupukan 30 dan nilai *input* variabel Penyiraman 30. Dan *output* dari nilai yang telah dimasukkan yaitu Tanaman Bagus.



Gambar 4.42 Unity: Console output hasil tanaman percobaan ke-18

Hasil uji coba dari penelitian terhadap permainan *Saung Sawah* dengan berbagai nilai masukan pada variabel *input* dirangkum pada Tabel 4.5 di bawah ini:

Tabel 4.5 Hasil uji coba algoritma *fuzzy* Sugeno.

| Percobaan ke- | Input | | | Output | Hasil Uji Coba | Keterangan |
|---------------|-------------|-----------|------------|---------------|----------------|------------|
| | Jenis Bibit | Pemupukan | Penyiraman | Hasil Tanaman | | |
| 1 | 1 | 10 | 10 | Kering | Kering | Valid |
| 2 | 1 | 10 | 20 | Kering | Kering | Valid |
| 3 | 1 | 10 | 30 | Standar | Standar | Valid |
| 4 | 1 | 20 | 10 | Kering | Kering | Valid |
| 5 | 1 | 20 | 20 | Standar | Standar | Valid |
| 6 | 1 | 20 | 30 | Bagus | Bagus | Valid |
| 7 | 1 | 30 | 10 | Standar | Standar | Valid |
| 8 | 1 | 30 | 20 | Bagus | Bagus | Valid |
| 9 | 1 | 30 | 30 | Bagus | Bagus | Valid |
| 10 | 10 | 10 | 10 | Kering | Kering | Valid |
| 11 | 10 | 10 | 20 | Standar | Standar | Valid |

| | | | | | | |
|----|----|----|----|---------|---------|-------|
| 12 | 10 | 10 | 30 | Bagus | Bagus | Valid |
| 13 | 10 | 20 | 10 | Standar | Standar | Valid |
| 14 | 10 | 20 | 20 | Bagus | Bagus | Valid |
| 15 | 10 | 20 | 30 | Bagus | Bagus | Valid |
| 16 | 10 | 30 | 10 | Bagus | Bagus | Valid |
| 17 | 10 | 30 | 20 | Bagus | Bagus | Valid |
| 18 | 10 | 30 | 30 | Bagus | Bagus | Valid |

Seperti terlihat pada Tabel 4.5, hasil validasi mencapai 100%. Karena percobaan sesuai dengan aturan *fuzzy* yang dibuat. Hasil validasi pada hasil tanaman dalam permainan *Saung Sawah*, hasil ini dianggap sangat baik.

4.4 Integrasi Islam

Islam sangat memuliakan profesi petani. Selain sebagai ladang penghasilan, bertani juga merupakan sebuah ibadah. Hal ini sejalan dengan sabda nabi yang diriwayatkan oleh Imam Bukhari. Rasulullah saw bersabda;

مَا مِنْ مُسْلِمٍ يَغْرِسُ غَرْسًا، أَوْ يَزْرَعُ زَرْعًا، فَيَأْكُلُ مِنْهُ طَيْرٌ أَوْ إِنْسَانٌ أَوْ بَهِيمَةٌ، إِلَّا كَانَ لَهُ بِهِ صَدَقَةٌ

“Tidaklah seorang muslim yang menanam tanaman atau bertani kemudian burung, manusia atau pun binatang ternak memakan hasilnya, kecuali semua itu merupakan sedekah baginya.” (H.R. Bukhari:2320).

Dalam Al-Quran, Allah subhanahu wa ta’ala telah menganugerahkan bumi yang di dalamnya terdapat berbagai macam tumbuhan untuk bisa dinikmati. Sebagaimana dalam Quran surat Ar-Rahman ayat 10-13, Allah subhanahu wa ta’ala berfirman;

وَالْأَرْضَ وَضَعَهَا لِلْأَنَامِ (١٠) فِيهَا فَاكِهَةٌ وَالنَّخْلُ ذَاتُ الْأَكْمَامِ (١١) وَالْحَبُّ ذُو الْعَصْفِ وَالرَّيْحَانُ (١٢)
فَبِأَيِّ آلَاءِ رَبِّكُمَا تُكَذِّبَانِ (١٣)

“Dan bumi telah dibentangkan-Nya untuk makhluk(-Nya). Di dalamnya ada buah-buahan dan pohon kurma yang mempunyai kelopak mayang. Dan biji-bijian yang berkulit dan bunga-bunga yang harum baunya. Maka nikmat Tuhanmu yang manakah yang kamu dustakan?” (QS. Ar-Rahman 10-13).

Pada ayat 11 Ibnu Katsir menjelaskan bahwa kata “*Faa Kihatun*” yakni memiliki makna buah-buahan dengan berbagai macam rasa, bentuk, bau dan warnanya. Sedangkan lafazh “*an-nakhl*” Karena manfaat yang ditawarkannya, baik buahnya basah atau kering, disebutkan secara terpisah. Dan menurut Ibnu Juraij yang mengutip Ibnu Abbas, “*al-akmam*” mengacu pada kelopak mayang. Banyak analis menyatakan pendapat yang sama, menyatakan bahwa “*akmam*” mengacu pada kelopak mayang yang merekah dan mengeluarkan sipulut dan kurma yang semula dikenal sebagai “*busr*” lalu “*rutab*” sebelum menjadi matang dan matang dengan sempurna.

Selanjutnya pada ayat 12, Ali bin AbuTalhah meriwayatkan dari Ibnu Abbas, bahwa makna dari kata: “*Dan biji-bijian yang berkulit.*” (Ar-Rahman: 12) ialah biji-bijian yang memiliki daun.

Al-Aufi telah meriwayatkan dari Ibnu Abbas, bahwa “*al-'asf*” berarti daun hijau dari tanaman yang dipotong bagian atasnya. Ketika dikeringkan, itu disebut “*asf*”. Hal yang sama dikatakan oleh Qatadah. Sedangkan Ad-Dahhak, dan Abu Malik, bahwa yang dimaksud dengan “*asf*” yaitu daun yang telah kering.

Ibnu Abbas, Mujahid, dan lain-lainnya mengatakan bahwa yang dimaksud dengan “*raihan*” ialah dedaunannya.

Lain halnya dengan Al-Hasan, ia mengatakan bahwa yang dimaksud dengan “*raihan*” ialah aroma yang harum yang dipakai seseorang. Sedangkan menurut Ali ibnu AbuTalhah yang telah meriwayatkan dari Ibnu Abbas, bahwa makna “*raihan*” yaitu tanaman yang hijau.

Makna yang dimaksud dengan biji-bijian yaitu tanaman yang mempunyai bulir dan daun-daunan yang melingkar pada batangnya. Seperti gandum, jewawut, dan lain sebagainya.

Menurut pendapat lain, “*al-'asf*” berarti daun-daun muda seperti sayuran. Sementara “*Raihan*” mengacu pada daun yang terjalin dengan biji-bijian tua.

Selanjutnya, ayat 13 berarti: "Wahai kedua jenis makhluk-Nya, jin dan manusia, maka nikmat Tuhan yang manakah yang kamu dustakan?" Ini adalah pendapat Mujahid dan para ulama lainnya, dan diperjelas dengan makna yang terkandung dalam konteks berikut. Menjadi. Dengan kata lain, anugerah Tuhanmu begitu nyata bagimu sehingga kamu begitu terpesona olehnya sehingga kamu tidak dapat mengingkari atau mengakuinya. Dan kita hanya bisa mengatakan apa yang dikatakan oleh jin yang beriman kepada Tuhan. "Ya Allah, tidak ada satupun dari karunia-Mu yang kami dustakan. Oleh karena itu, segala puji bagi-Mu."

Disebutkan bahwa Ibnu Abbas selalu menjawabnya dengan ucapan berikut, "Tidak, lalu yang manakah, wahai Tuhanku?" Dengan kata lain. dapat disebutkan bahwa kami tidak mendustakan sesuatu pun darinya.

Imam Ahmad mengatakan, telah meriwayatkan kepada kami Yahya ibnu Ishaq, telah meriwayatkan kepada kami Ibnu Lahi'ah, dari Abul Aswad, dari Urwah, dari Asma binti Abu Bakar yang telah meriwayatkan bahwa dia mendengar

Rasulullah Saw. pernah membaca satu rukun Al-Qur'an dalam solatnya sebelum diperintahkan untuk menyerukan dakwahnya secara terang-terangan, sedangkan orang-orang kafir mendengarkan-nya, yaitu firman Allah Subhanahu wa ta'ala.: Maka nikmat Tuhan kamu yang manakah yang kamu dustakan? (Ar-Rahman: 13).

Di ayat lainnya Allah subhanahu wa ta'ala juga memberikan nikmat untuk memudahkan manusia dalam bercocok tanam, sebagaimana diterangkan dalam surat Al-An'am ayat 99,

وَهُوَ الَّذِي أَنْزَلَ مِنَ السَّمَاءِ مَاءً فَأَخْرَجْنَا بِهِ نَبَاتَ كُلِّ شَيْءٍ فَأَخْرَجْنَا مِنْهُ خَضِرًا نُخْرِجُ مِنْهُ حَبًّا مُتَرَاكِبًا

“Dan Dialah yang menurunkan air dan langit, lalu Kami tumbuhkan dengan air itu segala macam tumbuh-tumbuhan, maka Kami keluarkan dari tumbuh-tumbuhan itu tanaman yang menghijau, Kami keluarkan dari tanaman yang menghijau itu butir yang banyak.” (QS. Al-An'am 99)

Wahaba az-Zuhairi menjelaskan bahwa ayat ini menunjukkan bahwa Allah sabhanahu wa ta'ala menurunkan air hujan yang menyebabkan tumbuhan tumbuh dan berkembang. Saat hujan turun, Allah menumbuhkan tanaman dalam berbagai bentuk, varietas, kualitas dan buah. Menurutnya, poin terpenting dalam ayat ini adalah renungan bagi manusia. Ketika berbuah, bagaimanakah keadaan buah yang tumbuh dari pohon itu, bagaimanakah keadaan pertumbuhannya, dan bagaimanakah keadaan buah ketika sudah matang dan lengkap, itu berubah dari keadaan kering ke keadaan yang mengandung air, dan buah-buahan serta bunga mekar.

Menurut ayat dan hadits yang disebutkan di atas, itu adalah berkah dan pengingat tentang manfaat bercocok tanam yang telah Allah sediakan akan kebutuhannya. Para pemikir Islam telah menyadari pentingnya peran petani.

Pentingnya pertanian dan status kaum tani telah dibahas dalam tulisan K.H. Hasyim Asy'ari. Tulisannya yang diterbitkan di Majalah Soeara Moeslimin, No. 2, pada tahun 14 Januari 1944 atau 19 Muharom 1363. Dalam artikel ini, ia secara singkat membahas masalah pertanian. Menurut K.H Hashim Asy'ari, pertanian bukan merupakan industri yang mandiri. Berbagai ayat yang ia kutip dari Alquran, al-Hadits, dan teks-teks ilmiah menunjukkan betapa besar perhatian yang diberikan terhadap bidang pertanian dalam ajaran Islam.

Karena itu, kita sebagai Muslim yang merupakan penduduk dengan mayoritas komunitas petani tidak merendahkan status petani. Mualaf harus selalu aktif dan berkeinginan untuk berkecimpung di dunia pertanian. Sentimen yang tumbuh dan menguat di dunia petanian perlu ditekankan. Dampaknya, hasil pertanian bisa lebih maksimal dengan biaya produksi yang lebih terjangkau. Oleh karena itu, hal ini akan berdampak positif pada lingkungan pertanian yang saat ini mengalami kemunduran.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil uji coba implementasi FSM dan algoritma *fuzzy* Sugeno pada *game Saung Sawah*, dapat disimpulkan bahwa penggunaan metode FSM dalam mendesain perilaku NPC dan metode logika *fuzzy* untuk menentukan hasil tanaman dapat diterapkan dengan sangat baik dan sesuai dengan validasi sebanyak 100% di mana prosentase validasi dari *output* metode FSM dalam mendesain perilaku NPC yaitu 100% dan untuk metode *fuzzy sugeno* dalam klasifikasi hasil tanaman yaitu 100%. Hasil pengujian telah dilakukan pada penelitian ini dengan hasil yang dirangkum pada Tabel 4.4 dan Tabel 4.5. Pada metode *fuzzy sugeno* ini menggunakan variabel *input* sebanyak tiga variabel *input*, yakni variabel jenis bibit, variabel pemupukan dan variabel penyiraman. Serta *output* berupa hasil tanaman kering, standar, bagus.

5.2 Saran

Dalam pembuatan *game Saung Sawah* ini masih belum mencapai sempurna, sehingga saran untuk penelitian berikutnya, yaitu:

1. Menambahkan *state* baru pada setiap NPC sehingga perilaku NPC lebih bervariasi dan tidak membosankan.
2. Menambahkan NPC baru sehingga *game* lebih banyak interaksi dan menyenangkan.
3. Menambahkan objek tanaman baru sehingga permainan lebih menarik.

DAFTAR PUSTAKA

- Aarsten, Van. (1953). Pengertian Pertanian Menurut Para Ahli, Sejarah dan Contohnya. Di akses 14 September 2022 jam 10.00 di halaman <https://dosenpertanian.com/pengertian-pertanian/>
- Anggraeni, Tri. (2021). Desain *Game* APD COVID-19 Simulator. *Journal of Animation & Games Studies*. 7(1), 43-62.
- Ardi, B. K., & Sari, Y. S. (2020). Aplikasi *Game Far From Reality Endless Runner* Berbasis *Android*. *JUKOMIKA (Jurnal Ilmu Komputer Dan Informatika)*. 3(1), 288-303.
- Arif, Yunifa M, M. Hardi, dan Supeno Mardi (2012). "Integrasi *Hierarchy Finite State Machine* Dan Logika *Fuzzy* Untuk Desain Strategi *Npc Game*". *Matics Journal*. 100-109.
- Auditya, G. Perdana. (2019). "Pengembangan *Game Dice* Sebagai Sarana Pembelajaran Pengucapan Bahasa Inggris Untuk Siswa SD". Skripsi. Yogyakarta: Universitas Muhammadiyah Yogyakarta.
- Aulianti, W. D., Karim, S. A., & Riska, M. (2021). Pengembangan *Game* Pendidikan Anti Korupsi Berbasis *Android*. *Jurnal Media TIK : Jurnal Media Pendidikan Teknik Informatika dan Komputer*. 4(2), 27-32.
- Badan Pusat Statistik. (2018). Hasil Survey Pertanian Antar Sensus (SUTAS) 2018.
- Caesar, Rio. (2015). Kajian Pustaka Perkembangan *Genre Games* Dari Masa Ke Masa. *Journal of Animation and Games Studies*. 1(2), 113-134.
- Costikyan, G. (2013). *Uncertainty in Games (Playful Thinking)*. MIT Press.
- Della, N. R., Wijaya, I. D., & Harijanto, B. (2017). Pengembangan *Game* Simulasi Penanaman Padi Situ Bagendit (Studi Kasus Pada Uptd Pertanian Kecamatan Ngantang). *Jurnal Informatika Polinema*. 3(4), 29-35.
- Deloitte, (2012). "*eTransform Africa: Agriculture Sector Study: Sector Assessment and Opportunities for ICT*". *Deloitte Project Report*.
- Jacky D. S., Virginia T., & Alwin M. S. (2019). Perancangan *Game* Simulasi Kewirausahaan. *Jurnal Teknik Informatika*. 14(1), 79-88.
Jakarta: Badan Pusat Statistik.
- Jenni Novak. (2012). *Game Development Essentials*. USA New York: Delma Cengage Learning. h.58.

- Julius r latumaresa. (2015). *Perekonomian Indonesia Dan Dinamika Ekonomi Global*. Jakarta: Mitra Wacana Media.
- Lia Widya, L. (2020). "Pengembangan Potensi Lokal Pertanian Dalam Peningkatan Ekonomi Masyarakat (Studi Pada *Home* Industri Merek Jajan *Japri* Di Pekon Pringsewu Provinsi Lampung)". Skripsi. Lampung: Institut Agama Islam Negeri (IAIN) Metro.
- Nurdin, A. (2013). *Pembangunan Game First Person Shooter 3D The Tracker*. Skripsi. Bandung: Universitas Komputer Indonesia.
- Nurshodiq, & Rahmadhani, A. Y. (2021). *Game Edukasi Flora dan Fauna Di Indonesia Untuk Meningkatkan Minat Belajar Siswa Berbasis Android Dengan Pendekatan Multyplayer*. *Ilmudata*. 1(1), 1-11.
- Rahadian, M. F., Suyatno, A., & Maharani, S. (2016). Penerapan Metode *Finite State Machine* Pada Game "The Relationship". *Jurnal Informatika Mulawarman*, 11(1), 14-22.
- Rianingtias, O. (2019). Pengembangan *Game* Edukasi Berbasis Android Sebagai Media Pembelajaran Biologi Bernuansa Motivasi Siswa Kelas Xi Di SMA/MA. Skripsi. Lampung: Universitas Islam Negeri Raden Intan.
- Rifandi, E. (2017). Perancangan Aplikasi *Game* Simulasi Lalu Lintas Berbasis Android Dengan Metode *Quad-Tree*. *Pelita Informatika Budi Darma*. 16(1), 28-33.
- Septian R. H. (2018). "Penerapan Metode *Finite State Machine* Pada Game "The Mahasiswa" Guna Membangun Perilaku *Non Playable Character*". Skripsi. Yogyakarta: Universitas Islam Indonesia.
- Siti Mahmudah. (2017). "Pengembangan *Game* Edukasi 3d "Finding Treasure" Sebagai Media Pembelajaran Perakitan Komputer Untuk Siswa Kelas X TKJ SMK Negeri 1 Ngawen". Skripsi. Yogyakarta: Universitas Negeri Yogyakarta.
- Suratiyah, K. (2006). *Ilmu Usaha Tani*. Jakarta: Penebar Swadaya.
- Triwibowo, B. A. (2016). "Pengembangan *Game* Edukasi Menggunakan *Unity 3d* Berdisiplin Bersepeda Di Jalan Raya Pada Anak Usia 8-11 Tahun". Skripsi. Yogyakarta: Universitas Negeri Yogyakarta.
- V. Machado Da Silva. (2018). "Technologies To Support the Technical Debt Management in Software Projects: a Qualitative Research."