# **RANCANG BANGUN *GAME SMART CHICKEN* MENGGUNAKAN METODE FUZZY TSUKAMOTO**

**SKRIPSI**

DigunakanSebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

**Oleh:**

**RIKA NOMILASARI NIM. 1441183004**

****

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2017**

# **RANCANG BANGUN *GAME SMART CHICKEN* MENGGUNAKAN METODE FUZZY TSUKAMOTO**

**SKRIPSI**

DigunakanSebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

**Oleh:**

**RIKA NOMILASARI NIM. 1441183004**

****

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2017**

# **HALAMAN PENGESAHAN**

**RANCANG BANGUN *GAME SMART CHICKEN* MENGGUNAKAN METODE FUZZY TSUKAMOTO**

**Disusun Oleh :**

**RIKA NOMILASARI NIM. 1441183004**

**Skripsi ini telah diuji pada tanggal 29 Agustus 2017**

**Disetujui Oleh :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Penguji I | : | Mungki Astiningrum, ST., M.KOM.  NIP. 197710302005012001 | ……………… |
| 2 | Penguji II | : | Pramana Yoga Saputra, S.KOM., MMT.  NIP. 198805042015041001 | ……………… |
| 3 | Pembimbing I | : | Ariadi Retno Tri Hayati Ririd, S.KOM., M.KOM.  NIP. 198108102005012002 | ……………… |
| 4 | Pembimbing II | : | Indra Dharma Wijaya, ST., MMT.  NIP. 1973051020008011010 | ……………… |

Mengetahui,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Ketua Jurusan  Teknologi Informasi | Ketua Program Studi  Teknik Informatika | | Rudy Ariyanto, S.T., M.Cs. | Ir. Deddy Kusbianto Purwoko Aji, M.MKom | | NIP. 19711110 199903 1 002 | |  | | --- | | NIP. 19621128 198811 1 001 | | |  |

|  |  |
| --- | --- |
| **PERNYATAAN** |  |
| Dengan ini saya menyatakan bahwa laporan Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. |  |
|  |  |

Malang …….. 2017

Rika Nomilasari

# **ABSTRAK**

**Nomilasari, Rika.** “Rancang Bangun Game Smart Chicken Menggunakan Metode Fuzzy Tsukamoto”. **Pembimbing: (1) Ariadi Retno Tri Hayati Ririd, S.Kom.,M.Kom, (2) Indra Dharma Wijaya, ST., MMT.**

**Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2017.**

Matematika merupakan bidang studi yang amat berguna dan banyak memberi bantuan dalam mempelajari berbagai disiplin ilmu yang lain. Namun, dalam kehidupan sehari-hari banyak orang yang tidak menyukai bidang ini dikarenakan kerumitan dalam memahaminya. Cara mempelajari matematika pun tidak disertai dengan cara yang lebih mudah dan menyenangkan, sehingga hal ini yang melatar belakangi pembangunan game ‘*Smart Chicken*’.

Dalam penelitian ini dibuatlah game yang berisi tentang materi matematika dasar yang menggunakan metode *Fuzzy Tsukamoto*. Penerapan metode *Fuzzy Tsukamoto* ke dalam game bertujuan untuk menentukan skor yang didapat pemain saat mengambil poin berupa telur dan menjawab pertanyaan yang diberikan. Pemain akan mendapatkan skor dengan rentang tertentu berdasarkan 3 variabel *fuzzy* yaitu variabel poin, variabel nyawa, dan variabel waktu.

Hasil dari penelitian ini yaitu *game Smart Chicken* memiliki tampilan keseluruhan yang menarik dan berfungsi secara baik. HHHal ini dapat diketahui melalui prosentase jawaban dari setiap pengguna atau responden terhadap pertanyaan kuesioner sebesar 70%.

**Kata Kunci:** Game Edukasi, *Platformer* 2D, *Fuzzy Tsukamoto*

# ***ABSTRACT***

***Nomilasari, Rika.*** *“The Development Game Smart Chicken Using Fuzzy Tsukamoto Method”.* ***Advisors: (1) Ariadi Retno Tri Hayati Ririd, S.Kom.,M.Kom, (2) Indra Dharma Wijaya, ST., MMT.***

***Thesis, Informatics Engineering Study Programs, Department of Information Technology, State Polytechnic of Malang 2017.***

*Mathematics is a field of study which is very useful and offers assistance in learning a variety of other disciplines. However, in daily life a lot of people feel hard to understand this field due to its complexity. The way to learn Mathematics is not supported with easier methods. Therefore game ' Smart Chicken ' is developed.*

*In this study a single game contains material about basic Maths methods using Fuzzy Tsukamoto. The application of Fuzzy Tsukamoto methods in games is to determine the score earned by player while taking eggs and answer the questions provided. Players will get a score with a certain range based on 3 fuzzy variables namely points variable, life variable and time variable.*

*The results of this research, namely games Smart Chicken has an exciting overall interface and proper functions. It can be known through a percentage of response from any users or respondents to the questionnaire as much as 70%.*

***Keywords:*** *Education Game, Platformer 2D, Fuzzy Tsukamoto*

# **KATA PENGANTAR**

Puji Syukur kami panjatkan kehadirat Allah SWT atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan laporan akhir dengan judul “Rancang Bangun *Game Smart Chicken*  menggunakan metode *Fuzzy Tsukamoto* ”. Laporan akhir ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Dalam penulisan laporan ini penulis menyampaikan terimakasih kepada pihak-pihak yang telah membantu dalam menyelesaikan laporan skripsi ini, khususnya kepada:

1. Bapak Ir. Tundung Subali Patma, MT., Direktur Politeknik Negeri Malang.
2. Bapak Rudy Ariyanto, ST., M.Cs., selaku Ketua Jurusan Teknologi Informasi.
3. Bapak Ir. Deddy Kusbianto Purwoko Aji , M.MKom, selaku Ketua Program Study Teknik Informatika.
4. Ibu Ariadi Retno Trihayati Ririd, S.Kom., M.Kom, dan Bapak Indra Dharma Wijaya, ST., MMT, selaku Dosen Pembimbing yang telah memberikan bimbingan dan petunjuk kepada penulis dalam meyelesaikan Laporan Skripsi.
5. Orang tua dan keluarga yang telah memberikan bantuan doa dan dukungan, baik secara moril maupun materiil selama ini kepada penulis.
6. Teman dan sahabat yang telah memberikan bantuan dan dukungan dalam menyelesaikan Laporan Skripsi ini.
7. Seluruh dosen, staf, dan karyawan Politeknik Negeri Malang yang telah membantu penulis dalam menyelesaikan laporan skripsi ini.

Akhirnya penulis berharap karya ilmiah ini dapat bermanfaat bagi mahasiswa Politeknik Negeri Malang dan masyarakat pada umumnya.

Malang, …….. 2017

Rika Nomilasari

# **DAFTAR ISI**

Halaman

[SAMPUL DEPAN i](#_Toc490190503)

[HALAMAN JUDUL ii](#_Toc490190504)

[HALAMAN PENGESAHAN iii](#_Toc490190505)

[PERNYATAAN iv](#_Toc490190506)

[ABSTRAK v](#_Toc490190507)

[*ABSTRACT* vi](#_Toc490190508)

[KATA PENGANTAR vii](#_Toc490190509)

[DAFTAR ISI viii](#_Toc490190510)

[DAFTAR GAMBAR xi](#_Toc490190511)

[DAFTAR TABEL xii](#_Toc490190512)

[DAFTAR LAMPIRAN xiii](#_Toc490190513)

[BAB I. PENDAHULUAN 14](#_Toc490190514)

[1.1 Latar Belakang 14](#_Toc490190515)

[1.2 Rumusan Masalah 15](#_Toc490190516)

[1.3 Tujuan 15](#_Toc490190517)

[1.4 Batasan Masalah 15](#_Toc490190518)

[1.5 Sistematika Penulisan 15](#_Toc490190519)

[BAB II. LANDASAN TEORI 17](#_Toc490190520)

[2.1 Game Edukasi 17](#_Toc490190521)

[2.2 Logika Fuzzy Tsukamoto 17](#_Toc490190522)

[2.3 Unity *Game Engine* 19](#_Toc490190523)

[2.4 Android 19](#_Toc490190524)

[2.5 Smartphone 19](#_Toc490190525)

[2.6 Storyboard 20](#_Toc490190526)

[2.7 Multimedia Development Life Cycle (MDLC) 20](#_Toc490190527)

[BAB III METODOLOGI PENELITIAN 21](#_Toc490190528)

[3.1 Konsep 22](#_Toc490190529)

[3.2 Desain 22](#_Toc490190530)

[3.3 Pengumpulan Bahan 23](#_Toc490190531)

[3.4 Perakitan 23](#_Toc490190532)

[3.5 Pengujian 23](#_Toc490190533)

[3.6 Distribusi 24](#_Toc490190534)

[BAB IV ANALISIS DAN PERANCANGAN 25](#_Toc490190535)

[4.1 Analisa Sistem 25](#_Toc490190536)

[4.1.1 Gambaran Umum Sistem 25](#_Toc490190537)

[4.1.2 Analisis Kebutuhan 25](#_Toc490190538)

[4.2 Perancangan 26](#_Toc490190539)

[4.2.1 Perancangan Materi 26](#_Toc490190540)

[4.2.2 Perancangan Karakter 26](#_Toc490190541)

[4.2.3 Perancangan Gameplay 26](#_Toc490190542)

[4.2.4 Perancangan Algoritma Gameplay 28](#_Toc490190543)

[4.2.5 Perancangan Kecerdasan Buatan 29](#_Toc490190544)

[4.2.5.1 Proses Fuzzifikasi 31](#_Toc490190545)

[4.2.5.2 Proses Inferensi 35](#_Toc490190546)

[4.2.5.3 Proses Defuzzifikasi 40](#_Toc490190547)

[4.2.6 Perancangan Storyboard 41](#_Toc490190548)

[4.2.7 Use Case 42](#_Toc490190549)

[4.2.8 *Work Breakdown Structure* (WBS) 45](#_Toc490190550)

[BAB V. IMPLEMENTASI 46](#_Toc490190551)

[5.1 Implementasi pembuatan Asset Gambar 46](#_Toc490190552)

[5.1.1 Pembuatan Background 46](#_Toc490190553)

[5.1.2 Pembuatan Karakter pada Adobe Illustrator CS6 47](#_Toc490190554)

[5.2 Implementasi Audio Backsound 48](#_Toc490190555)

[5.3 Implementasi Scene 49](#_Toc490190556)

[5.3.1 Scene Main Menu 49](#_Toc490190557)

[5.3.2 Scene Peta 50](#_Toc490190558)

[5.3.3 Scene Tampilan Stage 1 50](#_Toc490190559)

[5.3.4 Scene Tampilan Stage 2 53](#_Toc490190560)

[5.3.5 Scene Tampilan Stage 3 54](#_Toc490190561)

[5.3.6 Scene Tampilan Stage 4 56](#_Toc490190562)

[5.3.7 Scene Tampilan Cara bermain 59](#_Toc490190563)

[5.3.8 Tampilan Hasil *Scoring* 59](#_Toc490190564)

[5.4 Implementasi Metode Fuzzy Tsukamoto 60](#_Toc490190565)

[BAB VI. PENGUJIAN DAN PEMBAHASAN 66](#_Toc490190566)

[6.1 Pengujian Aplikasi *Game* 66](#_Toc490190567)

[6.1.2 Pengujian Alpha 67](#_Toc490190568)

[6.1.3 Pengujian Beta 69](#_Toc490190569)

[6.1.4 Pengujian Metode Fuzzy Tsukamoto 70](#_Toc490190570)

[6.2 Pembahasan Aplikasi *Game* 73](#_Toc490190571)

[6.2.1 Hasil Pengujian Alpha 73](#_Toc490190572)

[6.2.2 Hasil Pengujian Beta 73](#_Toc490190573)

[BAB VII. PENUTUP 79](#_Toc490190574)

[7.1 Kesimpulan 79](#_Toc490190575)

[7.2 Saran 79](#_Toc490190576)

[DAFTAR PUSTAKA 80](#_Toc490190577)

# **DAFTAR GAMBAR**

[Gambar 2. 1 Tahapan proses dalam logika kabur](#_Toc489602563) 18

[Gambar 2. 2 Contoh storyboard 20](#_Toc489602564)

[Gambar 3. 1 Metode MDLC](#_Toc489602602) 20

[Gambar 4. 1 Flowchart Gameplay ……………………………………...………..............28](#_Toc489602601)

[Gambar 4. 2 Diagram Fuzzifikasi 32](#_Toc489602602)

[Gambar 4. 3 Diagram Fuzzifikasi 33](#_Toc489602603)

[Gambar 4. 4 Diagram Fuzzifikasi 34](#_Toc489602604)

[Gambar 4. 5 Diagram Fuzzifikasi 35](#_Toc489602605)

[Gambar 4. 6 Use case 43](#_Toc489602606)

[Gambar 4. 7 Work Breakdown Structure Game Smart Chicken 45](#_Toc489602607)

Gambar 5. 1 Sketch Background 46

Gambar 5. 2 Hasil Gambar Background 47

Gambar 5. 3 Hasil Gambar Background 47

Gambar 5. 4 Hasil gambar Player 48

Gambar 5. 5 Sprites Player 48

Gambar 5. 6 Hasil Gambar Background 49

Gambar 5. 7 Main Menu 50

Gambar 5. 8 Tampilan peta 50

Gambar 5. 9 Tampilan Stage 1 51

Gambar 5. 10 Tampilan pertanyaan pada Stage 1 52

Gambar 5. 11 Tampilan Pause Menu pada Stage 1 52

Gambar 5. 12 Tampilan Death Menu pada Stage 1 52

Gambar 5. 13 Tampilan Stage 2 53

Gambar 5. 14 Tampilan pertanyaan pada Stage 2 53

Gambar 5. 15 Tampilan pause menu pada Stage 2 54

Gambar 5. 16 Tampilan Death menu pada Stage 2 54

Gambar 5. 17 Tampilan Stage 3 55

Gambar 5. 18 Tampilan pertanyaan dalam Stage 3 55

Gambar 5. 19 Tampilan pause Menu dalam Stage 3 56

Gambar 5. 20 Tampilan death Menu dalam Stage 3 56

Gambar 5. 21 Tampilan Stage 4 57

Gambar 5. 22 Tampilan pertanyaan dalam Stage 4 57

Gambar 5. 23 Tampilan pause menu dalam Stage 4 58

Gambar 5. 24 Tampilan death menu dalam Stage 4 58

Gambar 5. 25 Tampilan cara bermain 59

Gambar 5. 26 Tampilan hasil scoring 59

# **DAFTAR TABEL**

[Tabel 4. 1 Tabel Karakter Player 26](#_Toc489603647)

[Tabel 4. 2 Variabel Input dan Output 29](#_Toc489603648)

[Tabel 4. 3 Rule Base 30](#_Toc489603649)

[Tabel 4. 4 Storyboard game Smart Chicken 41](#_Toc489603650)

[Tabel 4. 5 Use case Main 43](#_Toc489603651)

[Tabel 4. 6 Use case Peta 44](#_Toc489603652)

[Tabel 4. 7 Use case Pengaturan 44](#_Toc489603653)

[Tabel 4. 8 Use case Keluar 44](#_Toc489603654)

[Tabel 6. 1 Pengujian Black Box 66](#_Toc489605245)

[Tabel 6. 2 Pengujian Alpha 67](#_Toc489605246)

[Tabel 6. 3 Kuisioner pengujian beta 69](#_Toc489605247)

[Tabel 6. 4 Pengujian metode Fuzzy Tsukamoto 70](#_Toc489605248)

[Tabel 6. 5 Hasil kuesioner 73](#_Toc489605249)

[Tabel 6. 6 Hasil pengujian 74](#_Toc489605250)

[Tabel 6. 7 Hasil pengujian 74](#_Toc489605251)

[Tabel 6. 8 Hasil pengujian 75](#_Toc489605252)

[Tabel 6. 9 Hasil pengujian 75](#_Toc489605253)

[Tabel 6. 10 Hasil pengujian 75](#_Toc489605254)

[Tabel 6. 11 Hasil pengujian 76](#_Toc489605255)

[Tabel 6. 12 Hasil pengujian 76](#_Toc489605256)

[Tabel 6. 13 Hasil pengujian 76](#_Toc489605257)

[Tabel 6. 14 Hasil pengujian 77](#_Toc489605258)

# **DAFTAR LAMPIRAN**

Lampiran 1. Lembar Bimbingan Skripsi

Lampiran 2. Lembar Kuisioner Uji Coba

Lampiran 3. *Source Code*

# **BAB I. PENDAHULUAN**

## **Latar Belakang**

Pada zaman yang sudah modern ini, kemajuan teknologi sudah tidak dapat dihindari hampir dalam semua aspek kehidupan. Dimulai dari kegiatan sosial hingga kegiatan pribadi dapat dikaitkan dengan semua kemajuan teknologi. Kemajuan teknologi salah satunya yaitu *game*. Indonesia memiliki jumlah pemain game yang cukup besar dan jumlah tersebut diperkirakan akan bertambah 33% tiap tahunnya. *Game* merupakan salah satu media teknologi yang populer di kalangan masyarakat baik anak kecil maupun orang dewasa karena game adalah salah satu bentuk hiburan yang dapat dijadikan sebagai penyegar pikiran dari kepenatan akibat padatnya aktivitas sehari-hari.

*Game* dinilai mampu merangsang kemampuan berfikir sekaligus mengasah kemampuan kecerdasan si pemain itu sendiri. Karena di dalam game biasanya dibuat berbagai permasalahan yang harus diselesaikan oleh si pemainnya. Game dengan unsur edukasi juga mendapat perhatian dari kalangan developer Indonesia dikarenakan melihat pentingnya pembelajaran dalam suatu game. Tak jarang juga developer Indonesia seperti Bamboo cannon studio dan sekar media membuat game edukasi untuk membuat belajar lebih terasa menyenangkan.

Matematika merupakan bidang studi yang amat berguna dan banyak memberi bantuan dalam mempelajari berbagai disiplin ilmu yang lain. Wawasan pendidikan matematika sangat penting  bagi peserta didik karena materi ini membawa peserta kepemahaman tentang karakteristik matematika yang memiliki objek abstrak, ber tumpu pada kesepakatan, pola pikir dedukatif, memiliki simbol yang kosong dari arti, memperhatikan semesta pembicaraan dan konsisten dalam sistemnya. Tapi kenyataan, matematika sering kali disalah artikan oleh sebagian kaum pelajar, terkadang mata pelajaran ini dianggap sebagai mata pelajaran yang tidak menyenangkan. Kenyataan ini tidak jarang berubah menjadi suatu kebencian terhadap apa saja yang berhubungan dengan matematika.

Berdasarkan latar belakang yang telah diuraikan, maka perlu dibangun sebuah *game* untuk mengenalkan kepada anak-anak mengenai *game* edukasi dengan mengambil tema matematika yang dapat dimainkan dengan mudah serta memiliki nilai edukasi di dalamnya. Dengan perancangan game yang membantu minat anak dalam bermain serta belajar matematika.

## **Rumusan Masalah**

Berdasarkan latar belakang diatas, maka permasalahan yang dapat penulis ambil dalam skripsi adalah:

1. Bagaimana merancang dan membangun *game 2D* pengenalan matematika dasar kelas 1 SD?
2. Bagaimana menerapkan metode *Fuzzy Tsukamoto* pada *game smart chicken*?

## **Tujuan**

Tujuan dari pembuatan *game* ini adalah :

1. Untuk membuat sebuah game sebagai media untuk memperkenalkan matematika dasar kelas 1 SD.
2. Dapat menerapkan metode Fuzzy *Tsukamoto* pada *game smart chicken.*

## **Batasan Masalah**

Batasan masalah dalam pembahasan skripsi adalah :

1. Permainan ini mengambil materi dari buku matematika.
2. Karakter yang tersedia pada *game* terdapat satu tokoh yaitu ayam.
3. *Game* ini dibangun menggunakan Engine *game* Unity.
4. *Game* dimainkan satu pemain (*Single Player*).
5. *Game* ini berformat 2D
6. *Game* ini dirancang untuk *smartphones* Android.
7. *Game* ini ditujukan untuk anak kelas 1 sd.

## **Sistematika Penulisan**

**BAB I PENDAHULUAN**

Pada bab ini dijelaskan mengenai hal-hal yang bersifat umum seperti latar belakang, rumusan masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

**BAB II LANDASAN TEORI**

Pada bab ini berisi teori-teori yang mendasari dan berkaitan dengan masalah perencanaan dan pembuatan aplikasi yang digunakan untuk memudahkan pemahaman dan pemecahan terhadap masalah yang ada.

**BAB III METODOLOGI PENELITIAN**

Pada bab ini menjelaskan langkah-langkah yang digunakan penulis untuk memilih metode, teknik, prosedur apa yang tepat, dan tools apa yang akan digunakan sehingga setiap tahap penelitian dapat dilakukan dengan tepat.

**BAB IV ANALISIS DAN PERANCANGAN**

Berisi pembahasan mengenai analisis perencanaan dan pembuatan terhadap sistem. Termasuk di dalamnya adalah Flowchart, Storyboard, Usecase Diagram, dan WBS (Work Breakdown Structure).

**BAB V IMPLEMENTASI**

Membahas tentang uraian mengenai proses implementasi dari sistem yang telah dirancang sebelumnya ke dalam bahasa pemograman. Disertai dengan gambaran interface aplikasi.

**BAB VI PENGUJIAN DAN PEMBAHASAN**

Membahas tentang hasil uji coba aplikasi, jalannya uji coba bertahap sistem yang dikembangkan. Dapat terdiri dari metode uji, tujuan uji, proses iji coba serta analisa hasil uji coba.

**BAB VII KESIMPULAN**

Bab ini berisi kesimpulan yang diperoleh dari hasil pengujian dan saran-saran yang dibutuhkan untuk kesempurnaan sistem sehingga sistem tersebut dapat disemurnakan dan dikembangkan di kemudian hari.

# **BAB II. LANDASAN TEORI**

Tinjauan pustaka merupakan bagian yang akan membahas tentang penyelesaian masalah yang akan memberikan jalan keluarnya. Dalam hal ini akan dijabarkan beberapa teori yang berkaitan dengan masalah.

## **Game Edukasi**

Dalam dunia edukasi, interaksi antara murid dengan gurunya merupakan suatuhal yang mutlak. Jika murid harus manusia, maka guru bisa berupa benda lain seperti buku, pengalaman, dan lain sebagainya. Salah satunya adalah game. Mengapa *game*? Karena game merupakan program yang bersifat interaktif, bisa merespon apa yang dilakukan pengguna. Sehingga proses belajar pun bisa terjadi di sini.

Penggunaan game sebagai sarana pendidikan sebetulnya bukan merupakan hal yang tabu ataupun salah. Karena  game bersifat entertain atau menghibur, yang disini adalah bermain. Psikologi manusia adalah lebih suka bermain daripada belajar serius. Dalam game, pendidikan diberikan lewat praktek atau pembelajaran dengan praktek (*learning by doing*). Dalam *game*, pemain seolah masuk ke dalam dunia baru tempat mereka bisa melakukan apa saja. *Game* secara tidak langsung mendidik manusia lewat apa yang mereka kerjakan dalam *game* tersebut. Apa yang mereka kerjakan dalam game tersebut mempengaruhi pola pikir dan perilaku mereka. Ini merupakan bagian dari edukasi, namun, karena banyak game pada saat ini lebih mengedepankan kekerasan dan kriminalitas dalam permainannya, maka pendidikan yang diberikan pun adalah pendidikan yang tidak baik. Dalam hal ini yang mendapat efek paling besar adalah anak-anak. Mengapa anak-anak? Karena anak-anak terkadang tidak dapat menyaring informasi yang dia dapatkan. Yang akhirnya membuat mereka merasa sah-sah saja melakukan kekerasan di dunia nyata sebagaimana yang mereka lakukan di dunia maya.

## **Logika *Fuzzy Tsukamoto***

Logika fuzzy pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika *fuzzy* adalah teori himpunan *fuzzy*. Pada teori himpunan fuzzy, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dari penalaran degan logika *fuzzy* tersebut [1].

Sistem *fuzzy* mempunyai beberapa keuntungan bila dibandingkan dengan sistem tradisional, misalkan pada jumlah aturan yang dipergunakan. Pemrosesan awal sejumlah besar nilai menjadi sebuah nilai derajat keanggotaan pada sistem *fuzzy* mengurangi jumlah nilai menjadi sebuah nilai derajat keanggotaan pada sistem *fuzzy* mengurangi jumlah nilai yang harus dipergunakan pengontrol untuk membuat suatu keputusan. Keuntungan lainnya adalah sistem *fuzzy* mempunyai kemampuan penalaran yang mirip dengan kemampuan penalaran manusia. Hal ini disebabkan karena sistem *fuzzy* mempunyai kemampuan untuk memberikan respon berdasarkan informasi yang bersifat kualitatif, tidak akurat, dan ambigu [2].

Penerapan logika *fuzzy* dapat meningkatkan kinerja sistem kendali dengan menekan munculnya fungsi-fungsi liar pada keluaran yang disebabkan oleh fluktuasi pada variabel masukannya. Pendekatan logika *fuzzy* secara garis besar diimplementasikan dalam tiga tahapan sebagai berikut :

1. Tahap pengaburan (*fuzzification*) yakni pemetaan dari masukan tegas ke himpunan kabur.
2. Tahap inferensi, yakni pembangkitan aturan kabur.
3. Tahap penegasan (*defuzzification*), yakni tranformasi keluaran dari nilai kabur ke nilai tegas.

Penegasan

Inferensi

Pengaburan

Masukan Keluaran

Gambar 2. 1 Tahapan proses dalam logika kabur (Kurniawan, 2004)

Ada beberapa metode untuk merepresentasikan hasil logika *fuzzy* yaitu metode Tsukamoto, Sugeno dan Mamdani. Pada metode Tsukamoto, setiap konsekuen direpresentasikan dengan himpunan *fuzzy* dengan fungsi keanggotaan monoton. Output hasil inferensi masing-masing aturan adalah z,  berupa himpunan biasa (crisp) yang ditetapkan berdasarkan -predikatnya. Hasil akhir diperoleh dengan menggunakan rata-rata terbobotnya. Adapun tahapan-tahapan yang digunakan dalam menghitung *Fuzzy Tsukamoto* yaitu: Fuzzifikasi, Mesin Inferensi, dan Defuzzifikasi.

## **Unity *Game Engine***

Unity merupakan suatu aplikasi yang digunakan untuk mengembangkan *game multi platform* yang didesain untuk mudah digunakan. Unity itu bagus dan penuh perpaduan dengan aplikasi yang profesional. Unity secara rinci dapat digunakan untuk membuat *video game* 3D, real time animasi 3D dan visualisasi arsitektur dan isi serupa yang interaktif lainnya. Editor Unity dapat menggunakan plugin untuk web player dan menghasilkan *game browser* yang didukung oleh Windows dan Mac. Dalam Unity disediakan berbagai pilihan bahasa pemograman untuk mengembangkan *game*, antara lain JavaScript, C#, dan BooScript serta menggunakan editor MonoDevelop yang sudah tersedia dalam Unity [3].

## **Android**

Android adalah suatu sistem operasi yang berjalan pada *smatphone* saat ini dan menyesuaikan spesifikasi di kelas *low-end* hingga *high-end*. Hampir semua vendor saat ini mengembangkan produknya dengan sistem operasi Android, karena peminatnya yang semakin meningkat tajam. Sekitar 1,4 miliar penduduk dunia menggunakan android, dan diperkirakan pada tahun 2019 semua sekitar 2,8 miliar penduduk akan menggunakan android. Versi terbaru android saat ini adalah Nougat [4].

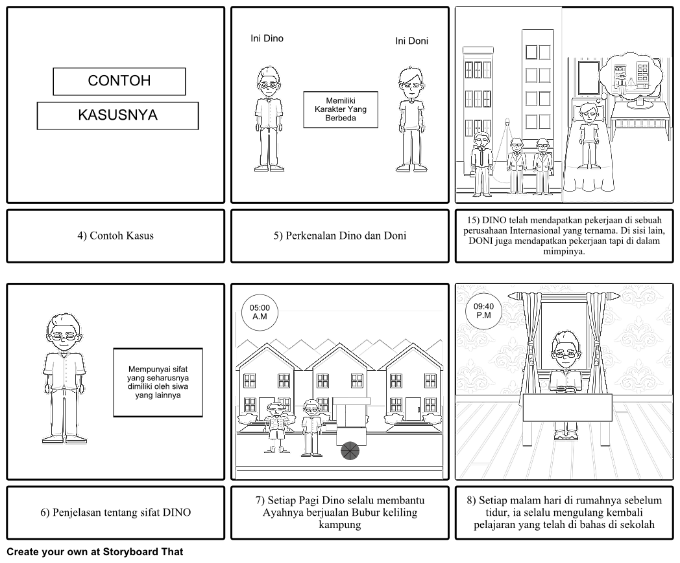
## **Smartphone**

Smartphone secara harfiah artinya telepon pintar, yakni telepon seluler yang memiliki kemampuan seperti PC walaupun terbatas. Selain itu, *smartphone* juga mendukung *email* dan *organizer*. Fitur lainnya adalah kemampuannya untuk ditambah aplikasi-aplikasi baru [5]. *Smartphone* merupakan perangkat ponsel yang tidak hanya bisa digunakan untuk berkomunikasi dasar (sms dan telepon), tetapi juga di dalamnya terdapat fungsi PDA (*Personal Digital Assistant*) atau disebut sebagai pembantu digital pribadi adalah sebuah alat elektronik yang berbasis komputer dan berbentuk kecil serta dapat dibawa kemana-mana.

Pada perkembangannya, *smartphone* juga mempunyai fungsi sebagai *media player portable*, *low end digital compact camera, pocket camera* dan GPS. *Smartphone* modern juga dilengkapi dengan layar *touchscreen* resolusi tinggi, *browser* yang mampu menampilkan *full web* seperti pada PC, serta akses data WiFi dan *internet broadband*.

## **Storyboard**

*Storyboard* merupakan visualisasi ide dari aplikasi yang dibangun, sehingga dapat memberikan gambaran dari aplikasi yang akan dihasilkan. *Storyboard* juga dapat dikatakan sebagai *visual script* yang akan dijadikan *outline* dari sebuah proyek, ditampilkan *shoot by shoot* yang biasa disebut dengan istilah *scene.* Dalam pembuatan *game* ini juga diperlukan *storyboard* agar lebih mempermudah dalam merancang adegan di setiap level.



Gambar 2. 2 Contoh storyboard

## **Multimedia Development Life Cycle (MDLC)**

Pengembangan multimedia agar dapat dimasukkan dalam pembelajaran harus melalui tahapan-tahapan yang terancang dengan baik dan runtut agar produk multimedia yang dihasilkan memiliki kualitas yang baik dan tepat digunakan dalam pembelajaran. Medode yang digunakan dalam pengembangan multimedia ini adalah *Multimedia Development Life Cycle*. Pengembangan metode multimedia ini dilakukan berdasarkan enam tahap, yaitu *concept* (pengonsepan), *design* (pendesignan), *material collecting* (pengumpulan materi), *assembly* (pembuatan), *testing* (pengujian), dan *distribution* (pendistribusian). Keenam tahap ini tidak harus berurutan dalam praktiknya, tahap-tahap tersebut dapat saling bertukar posisi. Meskipun begitu, tahap *concept* memang harus menjadi hal yang pertama kali dikerjakan[6].

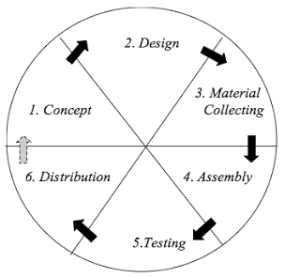
## **Relevansi**

Berdasarkan penelitian sebelumnya yang dilakukan oleh Inandiastya herinda putri yang berjudul “Pengembangan Game Edukasi Platformer Petualangan Rama Sinta berbasis Android” pada tahun 2016 di Politeknik Negeri Malang, pembuatan *game* menggunakan *tool* Unity engine. Dengan persamaan tool dan konsep edukasi yang ada, penulis akan membangun *game* dengan memberikan edukasi yang berbeda berupa materi matematika menggunakan Unity *Game Engine.*

Pada tahun 2016 ada penelitian yang menggunakan metode Fuzzy Tsukamoto oleh Tri setianto yang berjudul “Pembuatan Game Math Task dengan metode Fuzzy Tsukamoto*”* di Universitas Mulawarman. Penerapan metode dalam penelitian ini diletakkan pada hasil skor dengan menggunakan 2 variabel. Dengan menggunakan metode yang sama, penulis akan menerapkan pada hasil skor dan memiliki 3 variabel yang menjadi penentu hasil skor.

# **BAB III METODOLOGI PENELITIAN**

Metode pengembangan pada permainan edukasi pada Rancang Bangun *Game Smart Chicken* menggunakan metode Fuzzy Tsukamoto yaitu *Multimedia Development Live Cycle* (MDLC). Model MDLC memiliki 6 tahapan yang ditujukan pada gambar



Gambar 3. 1 Metode MDLC

Berikut ini merupakan penjelasan dari tahapan-tahapan sistematis dari metode *Multimedia Development Live Cycle* (MDLC) :

## **Konsep**

Konsep merupakan tahapan awal dalam memulai perancangan sebuah sistem. Konsep dimulai dengan menentukan target, mencari kebutuhan pengguna dan bagaimana merancang sistem. Jenis aplikasi ini merupakan aplikasi pembelajaran yang ditujukan pada anak sekolah dasar dengan rentang umur yang sudah ditentukan. Kebutuhan pengguna yang akan digunakan berupa soal-soal matematika dan soal tersebut akan diimplementasikan ke dalam sebuah *game.*

## **Desain**

Desain ataupun perancangan merupakan tahapan lanjut setelah melakukan pengkonsepan. Pada tahap ini, dibuat spesifikasi secara rinci dalam sebuah perancangan proyek multimedia dan menjabarkan secara rinci bagaimana proyek

akan dibuat. Sasaran yang ingin dicapai oleh sistem ini yaitu memberikan cara lain dengan media *game* menggunakan materi matematika dasar.

## **Pengumpulan Bahan**

Material Collecting adalah tahap dimana pengumpulan data yang disesuaikan kebutuhan adalah sebagai berikut :

* Pertanyaan : berisi pertanyaan-pertanyaan yang berkaitan dengan soal matematika
* Teks : *font* apa yang nantinya akan digunakan dalam *game smart chicken*
* Gambar : gambar pada proyek yang digunakan sebagai *background*, *object game, dan interface*. Gabar yang digunakan bertipe .jpg dan .png
* Animasi : objek animasi yang digunakan adalah berupa karakter pada cerita.
* Audio : audio bertipe .wav atau .mp3

## **Perakitan**

Tahap Assembly atau perakitan adalah tahap dimana semua objek atau bahan multimedia dibuat. Semua objek atau material dibuat dan digabungkan menjadi satu aplikasi yang utuh. Dalam tahapan ini material diolah dalam game engine Unity sesuai dengan konsep dan desain yang sudah direncanakan sebelumnya.

## **Pengujian**

Pengujian dilakukan dengan menggunakan metode *White Box* dan metode *Black Box*:

* Metode pengujian *White Box* menggunakan struktur kontrol desain prosedural (structural testing) untuk memperoleh *test case*. Pengujian ini mengasumsikan bahwa logik spesifik adalah penting dan harus diuji untuk menjamin sistem melakukan fungsi dengan benar.
* Metode pengujian *Black Box* merupakan pengujian *user interface* kepada pengguna apakah sistem dapat dioperasikan atau tidak. Pengujian ini dilakukan dengan memberikan *data sample* sebagai nilai masukan dan dibandingkan dengan informasi yang dihasilkan, disbut juga pengujian berbasis skenario.

## **Distribusi**

Dalam tahapan ini, aplikasi yang telah selesai di uji dan dinyatakan baik sesuai dengan tujuan pembuatan, akan didistribusikan dengan cara mengunggah ke sebuah toko online Android. Dengan tujuan agar masyarakat dapat mengunduh dan menikmati game lebih mudah.

# **BAB IV ANALISIS DAN PERANCANGAN**

Pada bab ini akan dibahas secara rinci mengenai analisis dan perancangan game edukasi pembelajaran matematika berbasis android yang akan diimplementasikan dengan kegunaan dan tujuan yang telah ditentukan pada bab sebelumnya.

## **Analisa Sistem**

Analisa sistem merupakan penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

### **Gambaran Umum Sistem**

Game edukasi pembelajaran matematika berbasis Android ini berjenis *platformer* dengan misi menyelesaikan pertanyaan pada setiap kategori yang sudah disediakan. Matematika yang diterapkan dalam game platformer ini berupa logika, numerik dan matematika realistis. Sebelum memulai permainan akan tampil sebuah menu dengan pilihan yang disediakan. Kemudian pemain akan memasuki sebuah peta dengan berbagai pilihan kategori. Setelah memasuki bagian peta kemudian, pemain dapat memainkan permainan dan mendapatkan nilai yang akan dijadikan ranking. Sasaran pengguna game ditujukan untuk anak kelas 1 SD. Sistem *game smart chicken* ini menggunakan *Fuzzy Tsukamoto*. Metode ini menggunakan kecerdasan buatan untuk memberikan hasil akhir dari sebuah permainan berdasarkan poin, nyawa, dan waktu.

### **Analisis Kebutuhan**

Dalam pembuatan aplikasi, maka dibutuhkan spesifikasi software yang mendukung. Kebutuhan software:

* Sistem operasi Windows 10, min Windows 7
* Unity *game engine* versi 5.5.1
* Adobe Illustrator CS6
* Android SDK and Java Development Kit (JDK)
* Bfxr Sound Effect

Kebutuhan Hardware yang digunakan dalam membangun aplikasi adalah :

* Processor Intel Core i5, 2.20 GHz
* 4 GB RAM
* Smartphone Android

## **Perancangan**

Berikut ini merupakan perancangan materi yang akan digunakan dalam pembuatan aplikasi “Rancang Bangun Game Smart Chicken menggunakan Fuzzy Tsukamoto”.

### **Perancangan Materi**

Materi yang disampaikan dalam *game smart chicken* yaitu mengenalkan ragam soal matematika yang diselipkan dalam sebuah permainan anak-anak, karena dengan bermain game smart chicken maka pemain akan mendapatkan beberapa manfaat untuk belajar matematika.

### **Perancangan Karakter**

Karakter dalam game hanya ada satu. Karakter akan berada pada tempat yang berbeda. Karakter *player* dimainkan oleh pemain dengan tombol *controller*. Rancangan aset karakter yang akan ditampilkan pada *game* tersebut dijelaskan pada tabel :

Tabel 4. 1 Tabel Karakter Player

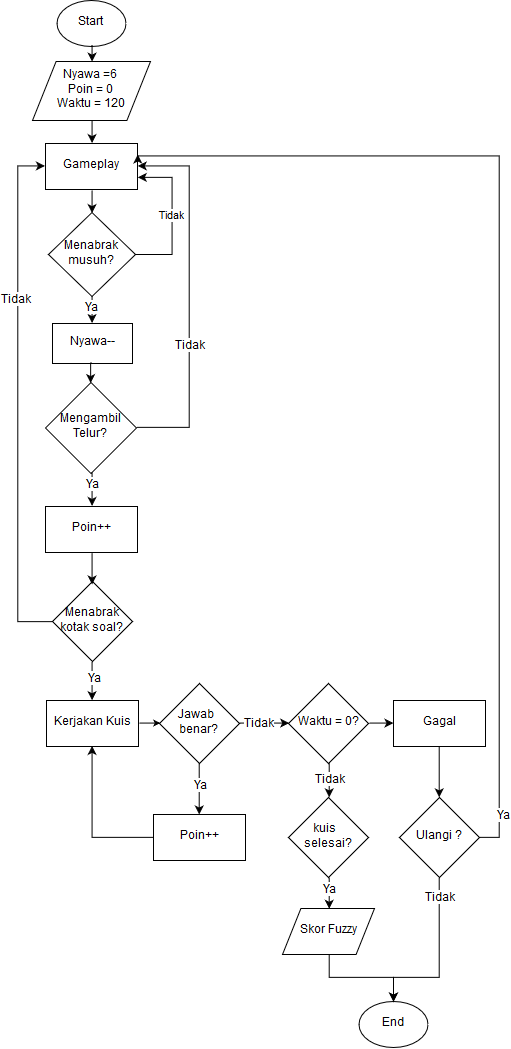
|  |  |  |
| --- | --- | --- |
| No | Nama Karakter | Gambar |
| 1 | Master Mata Merah |  |

### **Perancangan Gameplay**

Dalam permainan “*Smart Chicken*” secara keseluruhan alur adalah pemain harus mengambil semua telur dan menyelesaikan soal matematika yang disediakan dengan bobot yang berbeda dalam setiap *stage* nya. Terdapat 4 *stage* dengan latar belakang yang berbeda yaitu pada *stage* 1 terdapat latar belakang hutan, *stage* 2 terdapat latar belakang malam hari, *stage* 3 terdapat latar belakang musim salju, dan *stage* 4 terdapat latar belakang pantai. Dalam setiap *stage* terdapat misi untuk mencari telur dan menjawab soal matematika. Ketika pemain membuka menu awal akan ada pilihan tombol *play* kemudian pemain akan memilih lokasi yang tersedia pada peta. Lokasi yang tersedia pada permainan yaitu latarbelakang siang hari, malam hari, salju dan pantai. Pemain akan memasuki *stage* yang dipilih maka akan memasuki tempat yang berlatar belakang sesuai dengan pilihan yang ada pada peta, di dalam arena permainan tersebut player akan mengumpulkan poin berupa telur yang sudah tersebar. Ketika mengumpulkan telur, maka pemain juga dihadapkan pada sebuah rintangan berupa batu, ular, jamur beracun. Jika pemain terkena rintangan, maka nyawa pemain akan berkurang satu poin. Ketika pemain terkena rintangan secara terus-menerus hingga nyawa pemain habis maka pemain akan kalah. Setelah mendapatkan seluruh telur, kemudian pemain akan dihadapkan pada sebuah kotak yang berisi sebuah pertanyaan. Di dalam kotak pertanyaan tersebut disediakan pertanyaan seputar matematika dasar dengan tingkatan kelas 1 SD. Setelah menyentuh kotak, pemain diharuskan menjawab pertanyaan dengan benar dan tepat agar mendapatkan poin. Lalu, pemain akan mendapatkan skor berupa bintang 1, bintang 2, dan bintang 3 setelah menyelesaikan semua soal. Untuk menentukan perolehan bintang terdapat 3 variabel untuk mendapatkan skor bintang yaitu berdasarkan variabel waktu, nyawa, dan poin. Sistem *scoring* yang ada pada game adalah sebagai berikut :

* Awal bermain pemain memiliki poin = 0.
* Waktu yang disediakan adalah 120 detik atau 2 menit.
* Nyawa yang disediakan sebanyak 6.
* Setiap telur yang diambil memiliki nilai 5 poin.
* Dalam satu kali permainan terdapat 7 soal.
* Setiap pertanyaan dijawab benar maka bernilai 20.
* Hasil akhir kalkulasi nilai dimasukkan dalam metode Fuzzy Tsukamoto.

### **Perancangan Algoritma *Gameplay***

Flowchart menjelaskan bagaimana jalannya permainan pada sebuah *game.* Berikut adalah flowchart gameplay “*Smart Chicken*”, ditunjukkan pada gambar 4.1 :

Gambar 4. 1 Flowchart Gameplay

### **Perancangan Kecerdasan Buatan**

Dalam *game* ini algoritma fuzzy tsukamoto diterapkan pada hasil *scoring*, yang akan menghasilkan bintang sesuai *score* yang diperoleh. *Score* yang di dapatkan oleh pemain meliputi *score* waktu pada saat bermain game dan *point* dalam menjawab pertanyaan, mengambil item dan menyelesaikan misinya. Maka dari itu terdapat 3 variabel, yaitu nyawa, waktu dan *point* sebagai input dan *score* sebagai output. Berikut adalah contoh perhitungan manual yang akan dilakukan :

* Pemain mendapatkan poin sebesar 250, nyawa sebanyak 6, dan waktu sebanyak 101 detik. Maka akan dicari perangkingan berdasarkan 3 variabel tersebut. Berapakah bintang yang didapatkan dari pemain?
* Pembentukan Himpunan *Fuzzy Logic*

Tabel 4. 2 Variabel Input dan Output

|  |  |  |
| --- | --- | --- |
| Variabel | Himpunan Input Fuzzy | Range |
| Nyawa | Sedikit | 0 – 2 |
| Sedang | 1 – 5 |
| Banyak | 4 – 6 |
| Waktu | Lambat | 0 – 40 |
| Normal | 20 – 100 |
| Cepat | 80 – 120 |
| Poin | Rendah | 0 – 120 |
| Sedang | 90 – 210 |
| Tinggi | 180 – 300 |
| Scoring | Kurang | 0 – 400 |
| Cukup | 350 – 850 |
| Baik | 800 – 1200 |

Data diatas sudah ditentukan sebagai batasan nilai dari masing-masing atribut untuk memudahkan dalam proses fuzzifikasi. Ada 4 variabel fuzzy yang dimodelkan, yaitu :

1. Nyawa, terdiri dari 3 himpunan fuzzy, yaitu Sedikit, Sedang, dan Banyak.
2. Waktu, terdiri dari 3 himpunan fuzzy, yaitu Lambat, Normal, dan Cepat.
3. Poin, terdiri dari 3 himpunan fuzzy, yaitu Rendah, Sedang, dan Tinggi.
4. Score, terdiri dari 3 himpunan fuzzy, yaitu Kurang, Cukup, dan Baik.

Terdapat 27 rule base yang dihasilkan, rule base ini nantinya akan menghasilkan score akhir kurang, cukup, dan baik untuk mendapatkan perankingan berupa bintang 1, bintang 2, dan bintang 3. Tabel di bawah ini merupakan pembetukan rule base :

Tabel 4. 3 Rule Base

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IF | NYAWA (H) | WAKTU (C) | POINT (P) | SCORE (H) |
| R1 | BANYAK | LAMBAT | RENDAH | KURANG |
| R2 | BANYAK | LAMBAT | SEDANG | CUKUP |
| R3 | BANYAK | LAMBAT | TINGGI | BAIK |
| R4 | BANYAK | NORMAL | RENDAH | KURANG |
| R5 | BANYAK | NORMAL | SEDANG | CUKUP |
| R6 | BANYAK | NORMAL | TINGGI | BAIK |
| R7 | BANYAK | CEPAT | RENDAH | KURANG |
| R8 | BANYAK | CEPAT | SEDANG | CUKUP |
| R9 | BANYAK | CEPAT | TINGGI | BAIK |
| R10 | SEDANG | LAMBAT | RENDAH | KURANG |
| R11 | SEDANG | LAMBAT | SEDANG | CUKUP |
| R12 | SEDANG | LAMBAT | TINGGI | BAIK |
| R13 | SEDANG | NORMAL | RENDAH | KURANG |
| R14 | SEDANG | NORMAL | SEDANG | CUKUP |
| R15 | SEDANG | NORMAL | TINGGI | BAIK |
| R16 | SEDANG | CEPAT | RENDAH | KURANG |
| R17 | SEDANG | CEPAT | SEDANG | CUKUP |
| R18 | SEDANG | CEPAT | TINGGI | BAIK |
| R19 | SEDIKIT | LAMBAT | RENDAH | KURANG |
| R20 | SEDIKIT | LAMBAT | SEDANG | CUKUP |
| R21 | SEDIKIT | LAMBAT | TINGGI | BAIK |
| R22 | SEDIKIT | NORMAL | RENDAH | KURANG |
| R23 | SEDIKIT | NORMAL | SEDANG | CUKUP |
| R24 | SEDIKIT | NORMAL | TINGGI | BAIK |
| R25 | SEDIKIT | CEPAT | RENDAH | KURANG |
| R26 | SEDIKIT | CEPAT | SEDANG | CUKUP |
| R27 | SEDIKIT | CEPAT | TINGGI | BAIK |

#### **Proses Fuzzifikasi**

Fuzzifikasi adalah proses pengubahan nilai tegas/real yang ada dalam fungsi keanggotaan. Dalam sistem ini dibuat 4 fungsi fuzzifikasi yaitu Fuzzifikasi Nyawa, Fuzzifikasi Waktu, Fuzzifikasi Poin dan Fuzzifikasi Score.

1. Fungsi Keanggotaan Nyawa (H)

Atribut Nyawa dibagi menjadi 3 kelompok yaitu sedikit (nyawa kurang dari 2), sedang (nyawa diantara 1 dan 5), dan banyak (nyawa tidak lebih dari 6). Dari pembagian tersebut dapat ditentukan nilai derajat keanggotaan dari himpunan fuzzy sedikit, sedang, dan banyak.

µ sedikit (H) =

µ sedang (H) =

µ banyak (H) =

μNyawaSedikit [6] = (2-6) / 2 = 0

μNyawaSedang [6] = (5-6) / 2 = 0

μNyawaBanyak [6] = (6-4) / 2 = 1

Gambar 4. 2 Diagram Fuzzifikasi

1. Fungsi Keanggotaan Waktu (C)

Atribut Waktu dibagi menjadi 3 kelompok yaitu lambat (waktu kurang dari 40), normal (waktu diantara 20 dan 100), dan cepat (waktu tidak lebih dari 120). Dari pembagian tersebut dapat ditentukan nilai derajat keanggotaan dari himpunan fuzzy lambat, normal, dan cepat.

µ cepat (C) =

µ normal (C) =

µ lambat (C) =

μWaktuCepat [101] = (101-80) / 40 = 0,35

μWaktuNormal [101] = (100-101) / 40 = 0

μWaktuLambat [101] = (40-101) / 40 = 0

Gambar 4. 3 Diagram Fuzzifikasi

1. Fungsi Keanggotaan Poin (P)

Atribut Poin dibagi menjadi 3 kelompok yaitu rendah (poin kurang dari 120), sedang (poin diantara 90 dan 210), dan tinggi (poin tidak lebih dari 300). Dari pembagian tersebut dapat ditentukan nilai derajat keanggotaan dari himpunan fuzzy rendah, sedang, dan tinggi.

µ rendah (P) =

µ sedang (P) =

µtinggi (P) =

μPoinTinggi [250] = (250-180) / 120 = 0,58

μPoinSedang [250] = (210-250) / 60 = 0

μPoinRendah [250] = (120-250) / 120 = 0

Gambar 4. 4 Diagram Fuzzifikasi

1. Fungsi Keanggotaan Score (S)

Atribut Score dibagi menjadi 3 kelompok yaitu kurang (score kurang dari 400), cukup (score diantara 350 dan 850), dan baik (score tidak lebih dari 1200). Dari pembagian tersebut dapat ditentukan nilai derajat keanggotaan dari himpunan fuzzy kurang, cukup, dan baik.

µ kurang (Z) =

µ cukup (Z) =

µ baik (Z) =

Gambar 4. 5 Diagram Fuzzifikasi

#### **Proses Inferensi**

Berdasarkan 27 aturan *fuzzy*, akan ditentukan nilai α dan z  
untuk masing-masing aturan. α adalah nilai keanggotaan anteseden dari setiap  
aturan, sedangkan z adalah nilai perkiraan skor dari setiap  
aturan. Berikut ini adalah langkah-langkah untuk mengkonversi 27 aturan  
*fuzzy* tersebut sehingga diperolah nilai α dan z dari setiap aturan. Nilai keanggotaan anteseden untuk setiap aturan *fuzzy* yang dinotasikan dengan α  
diperoleh dengan rumus sebagai berikut:

[R1] IF poin RENDAH Nyawa BANYAK Waktu LAMBAT Score KURANG  
α1 = Min (μPoinRendah μNyawaBanyak μWaktuLambat)

= (0 , 1 , 0) = 0

z1 = 400 – (0 \* 400) = 400

[R2] IF poin RENDAH Nyawa BANYAK Waktu NORMAL Score CUKUP

α2 = Min (μPoinRendah μNyawaBanyak μWaktuNormal)

= (0 , 1 , 0) = 0

z2 = 850 – (0 \* 250) = 850

[R3] IF poin RENDAH Nyawa BANYAK Waktu CEPAT Score CUKUP

α3 = Min (μPoinRendah μNyawaBanyak μWaktuCepat)

= (0 , 1 , 0,35) = 0

z3 = 850 – (0 \* 250) = 850

[R4] IF poin RENDAH Nyawa SEDANG Waktu LAMBAT Score KURANG

α4 = Min (μPoinRendah μNyawaSedang μWaktuLambat)

= (0 , 0 , 0) = 0

z4 = 400 – (0 \* 400) = 400

[R5] IF poin RENDAH Nyawa SEDANG Waktu NORMAL Score KURANG

α5 = Min (μPoinRendah μNyawaSedang μWaktuNormal)

= (0 , 0 , 0) = 0

z5 = 400 – (0 \* 400) = 400

[R6] IF poin RENDAH Nyawa SEDANG Waktu CEPAT Score CUKUP

α6 = Min (μPoinRendah μNyawaSedang μWaktuCepat)

= (0 , 0 , 0,35) = 0

z6 = 850 – (0 \* 250) = 850

[R7] IF poin RENDAH Nyawa SEDIKIT Waktu LAMBAT Score KURANG

α7 = Min (μPoinRendah μNyawaSedikit μWaktuLambat)

= (0 , 0 , 0) = 0

z7 = 400 – (0 \* 400) = 400

[R8] IF poin RENDAH Nyawa SEDIKIT Waktu NORMAL Score CUKUP

α8 = Min (μPoinRendah μNyawaSedikit μWaktuNormal)

= (0 , 0 , 0) = 0

z8 = 850 – (0 \* 250) = 850

[R9] IF poin RENDAH Nyawa SEDIKIT Waktu CEPAT Score KURANG

α9 = Min (μPoinRendah μNyawaSedikit μWaktuCepat)

= (0 , 0 , 0,35) = 0

z9 = 400 – (0 \* 400) = 400

[R10] IF poin SEDANG Nyawa BANYAK Waktu LAMBAT Score KURANG

α10 = Min (μPoinSedang μNyawaBanyak μWaktuLambat)

= (0 , 1 , 0) = 0

z10 = 400 – (0 \* 400) = 400

[R11] IF poin SEDANG Nyawa BANYAK Waktu NORMAL Score CUKUP

α11 = Min (μPoinSedang μNyawaBanyak μWaktuNormal)

= (0 , 1 , 0) = 0

z11 = 850 – (0 \* 250) = 850

[R12] IF poin SEDANG Nyawa BANYAK Waktu CEPAT Score BAIK

α12 = Min (μPoinSedang μNyawaBanyak μWaktuCepat)

= (0 , 1 , 0,35) = 0

z12 = 800 + (0 \* 400) = 800

[R13] IF poin SEDANG Nyawa SEDANG Waktu LAMBAT Score KURANG

α13 = Min (μPoinSedang μNyawaSedang μWaktuLambat)

= (0 , 0 , 0) = 0

z13 = 400 – (0 \* 400) = 400

[R14] IF poin SEDANG Nyawa SEDANG Waktu NORMAL Score CUKUP

α14 = Min (μPoinSedang μNyawaSedang μWaktuNormal)

= (0 , 0 , 0) = 0

z14 = 850 – (0 \* 250) = 850

[R15] IF poin SEDANG Nyawa SEDANG Waktu CEPAT Score BAIK

α15 = Min (μPoinSedang μNyawaSedang μWaktuCepat)

= (0 , 0 , 0,35) = 0

z15 = 800 + (0 \* 400) = 800

[R16] IF poin SEDANG Nyawa SEDIKIT Waktu LAMBAT Score KURANG

α16 = Min (μPoinSedang μNyawaSedikit μWaktuLambat)

= (0 , 0 , 0,35) = 0

z16 = 800 + (0 \* 400) = 800

[R17] IF poin SEDANG Nyawa SEDIKIT Waktu NORMAL Score CUKUP

α17 = Min (μPoinSedang μNyawaSedikit μWaktuNormal)

= (0 , 0 , 0) = 0

z17 = 850 – (0 \* 250) = 850

[R18] IF poin SEDANG Nyawa SEDIKIT Waktu CEPAT Score CUKUP

α18 = Min (μPoinSedang μNyawaSedikit μWaktuCepat)

= (0 , 0 , 0,35) = 0

z18 = 850 – (0 \* 250) = 850

[R19] IF poin TINGGI Nyawa BANYAK Waktu LAMBAT Score KURANG

α19 = Min (μPoinTinggi μNyawaBanyak μWaktuLambat)

= (0,5833 , 1 , 0) = 0

z19 = 800 + (0 \* 400) = 800

[R20] IF poin TINGGI Nyawa BANYAK Waktu NORMAL Score BAIK

α20 = Min (μPoinTinggi μNyawaBanyak μWaktuNormal)

= (0,5833 , 1 , 0) = 0

z20 = 800 + (0 \* 400) = 800

[R21] IF poin TINGGI Nyawa BANYAK Waktu CEPAT Score BAIK

α21 = Min (μPoinTinggi μNyawaBanyak μWaktuCepat)

= (0,5833 , 1 , 0,35) = 1

z21 = 800 + (1 \* 400) = 940

[R22] IF poin TINGGI Nyawa SEDANG Waktu LAMBAT Score KURANG

α22 = Min (μPoinTinggi μNyawaSedang μWaktuLambat)

= (0,5833 , 0 , 0) = 0

z22 = 400 – (0 \* 400) = 400

[R23] IF poin TINGGI Nyawa SEDANG Waktu NORMAL Score BAIK

α23 = Min (μPoinTinggi μNyawaSedang μWaktuNormal)

= (0,5833 , 0 , 0) = 0

z23 = 800 + (0 \* 400) = 800

[R24] IF poin TINGGI Nyawa SEDANG Waktu CEPAT Score BAIK

α24 = Min (μPoinTinggi μNyawaSedang μWaktuCepat)

= (0,5833 , 0 , 0,35) = 0

z24 = 800 + (0 \* 400) = 800

[R25] IF poin TINGGI Nyawa SEDIKIT Waktu LAMBAT Score CUKUP

α25 = Min (μPoinTinggi μNyawaSedikit μWaktuLambat)

= (0,5833 , 0 , 0) = 0

z25 = 850 – (0 \* 250) = 850

[R26] IF poin TINGGI Nyawa SEDIKIT Waktu NORMAL Score BAIK

α26 = Min (μPoinTinggi μNyawaSedikit μWaktuNormal)

= (0,5833 , 0 , 0) = 0

z26 = 800 + (0 \* 400) = 800

[R27] IF poin TINGGI Nyawa SEDIKIT Waktu CEPAT Score BAIK

α27 = Min (μPoinTinggi μNyawaSedikit μWaktuCepat)

= (0,5833 , 0 , 0,35) = 0

z27 = 800 + (0 \* 400) = 800

#### **Proses Defuzzifikasi**

Setelah melakukan proses inferensi maka dilakukan proses defuzzifikasi secara terpusat dengan menggunakan rumus :

Dengan tahap tahap perhitungan diatas maka dapat diketahui jika Poin memiliki nilai 250, Nyawa memiliki nilai 6, dan Waktu memiliki nilai 101 maka hasil perhitungan Z mendapatkan nilai 1010, yang berarti pemain tersebut mendapatkan skor bintang 3.

### **Perancangan Storyboard**

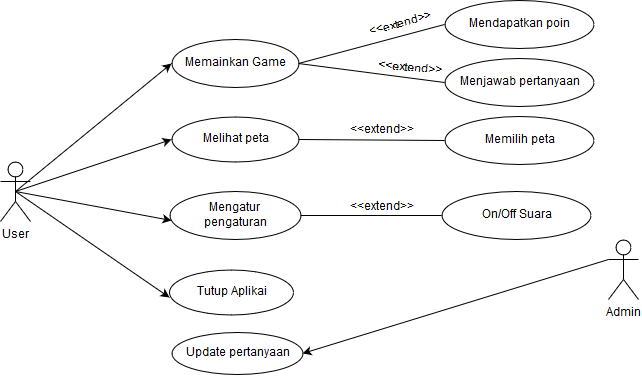
Tabel 4. 4 Storyboard game Smart Chicken

|  |  |  |
| --- | --- | --- |
| No. | Storyboard | Deskripsi |
| 1. |  | **Tampilan Menu Utama**  Pada tampilan menu utama akan muncul beberapa sub menu seperti gambar disamping yaitu Main, Peta, Pengaturan, Keluar. Pada menu main, pemain dapat langsung menuju menu peta lalu memilih stage dan dapat langsung memainkan. Pada menu peta terdapat peta stage untuk memilih stage. Pada menu pengaturan terdapat pengaturan suara. Menu keluar untuk mengakhiri aplikasi. |
| 2. |  | **Tampilan Pilihan Stage**  Pada tampilan pilihan stage, pemain dapat memilih stage sesuai dengan minat/keinginan. |
| 3. |  | **Tampilan Stage Game**  Pada tampilan stage game ini pemain akan mengambil telur yang sudah disebar, menghindari rintangan. |
| 4. |  | Kemudian pada saat sudah sampai akhir pemain akan menemukan kotak perrtanyaan yang harus disentuh oleh pemain. |
| 5. |  | Setelah pemain menyentuh kotak tersebut maka akan keluar panel yang mengeluarkan soal beserta jawabannya. |
| 6. |  | **Tampilan Pengaturan**  Pada tampilan menu pengaturan pemain dapat menghilangkan suara yang ada dalam permainan. |

### **Use Case**

Kegiatan yang dilakukan pemain digambarkan menggunakan use case. Hak akses yang dapat dilakukan oleh pemain dan admin yaitu :

1. Player memainkan game dengan menu Main di main menu.
2. Player dapat memilih stage dalam menu peta dengan memilih menu peta di main menu.
3. Player dapat melakukan pengaturan pada game, yaitu dengan mengatur volume dengan mode aktif atau nonaktif.
4. Player dapat keluar dari aplikasi dengan memilih menu tutup aplikasi.
5. Admin/pembuat dapat melakukan update pertanyaan secara berkala dengan mengupdate aplikasi.



Gambar 4. 6 Use case

* + - 1. Deskripsi Use Case

Tabel 4. 5 Use case Main

|  |  |  |
| --- | --- | --- |
| *Use Case Name:*  Memainkan Game | *ID* : UC.01  Siklus 1 | *Importance Level* :  *High* |
| *Primary Actor* :  *User* | *Use Case Type* : | |
| *Trigger* :  *Pemain klik tombol main* | | |
| *Pre-Condition :*  *Pemain memulai permainan* | | |
| *Post-Condition :*  *Masuk tampilan menu peta* | | |
| *Brief Description* :  *Use case* ini berfungsi untuk memulai permainan | | |
| *Normal flow events* :   1. Pemain klik tombol main 2. Masuk tampilan menu peta | | |
| *Alternative flow* : | | |

Tabel 4. 6 Use case Peta

|  |  |  |
| --- | --- | --- |
| *Use Case Name:*  Melihat Peta | *ID* : UC.02  Siklus 2 | *Importance Level* :  *High* |
| *Primary Actor* :  *User* | *Use Case Type* : | |
| *Trigger* :  *Pemain klik tombol menu peta* | | |
| *Pre-Condition :*  *Pemain memilih stage dalam peta* | | |
| *Post-Condition :*  *Masuk lihat peta* | | |
| *Brief Description* :  *Use case* ini berfungsi untuk memilih stage dalam peta | | |
| *Normal flow events* :   1. Pemain klik tombol menu peta 2. Masuk tampilan permainan dari stage yang dipilih | | |
| *Alternative flow* : | | |

Tabel 4. 7 Use case Pengaturan

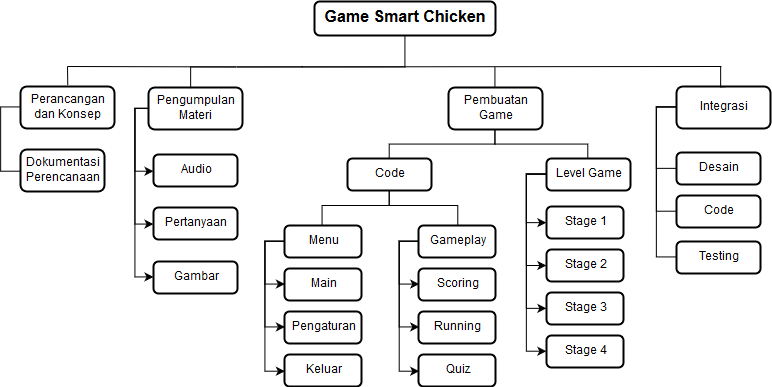
|  |  |  |
| --- | --- | --- |
| *Use Case Name:*  Mengatur Pengaturan | *ID* : UC.03  Siklus 3 | *Importance Level* :  *High* |
| *Primary Actor* :  *User* | *Use Case Type* : | |
| *Trigger* :  *Pemain klik tombol menu pengaturan* | | |
| *Pre-Condition :*  *Pemain memilih stage dalam pengaturan* | | |
| *Post-Condition :*  *Masuk lihat menu pengaturan* | | |
| *Brief Description* :  *Use case* ini berfungsi untuk memilih mode on/off pada suara | | |
| *Normal flow events* :   1. Pemain klik tombol menu pengaturan 2. Masuk tampilan pengaturan | | |
| *Alternative flow* : | | |

Tabel 4. 8 Use case Keluar

|  |  |  |
| --- | --- | --- |
| *Use Case Name:*  Keluar | *ID* : UC.04  Siklus 4 | *Importance Level* :  *High* |
| *Primary Actor* :  *User* | *Use Case Type* : | |
| *Trigger* :  *Pemain klik tombol exit/keluar* | | |
| *Pre-Condition :*  *Pemain keluar dari permainan* | | |
| *Post-Condition :*  *Keluar dari aplikasi* | | |
| *Brief Description* :  *Use case* ini berfungsi untuk keluar dari aplikasi. | | |
| *Normal flow events* :   1. Pemain klik tombol keluar 2. Aplikasi tertutup | | |
| *Alternative flow* : | | |

### ***Work Breakdown Structure* (WBS)**

Pada *Work Breakdown Structure* akan dijelaskan bagian pembuatan aplikasi ini.



Gambar 4. 7 Work Breakdown Structure Game Smart Chicken

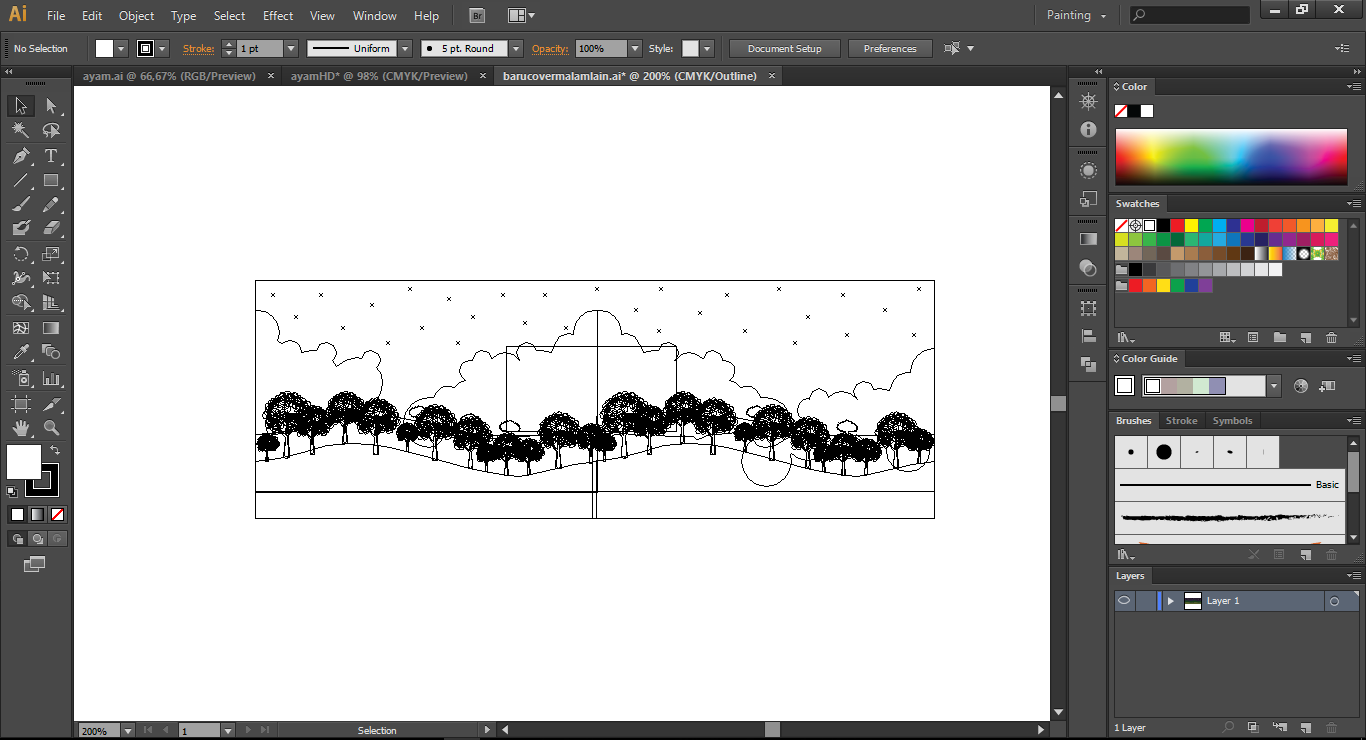
# **BAB V. IMPLEMENTASI**

Bab V berisi implementasi merupakan tahapan dimana aplikasi yang sudah dibuat dan dirancang lalu di uji kelayakannya. Pada tahap ini akan dijelaskan mengenai aplikasi yang dirancang meliputi implementasi pembuatan asset, implementasi scene, dan implementasi algoritma *fuzzy tsukamoto*.

## **Implementasi pembuatan Asset Gambar**

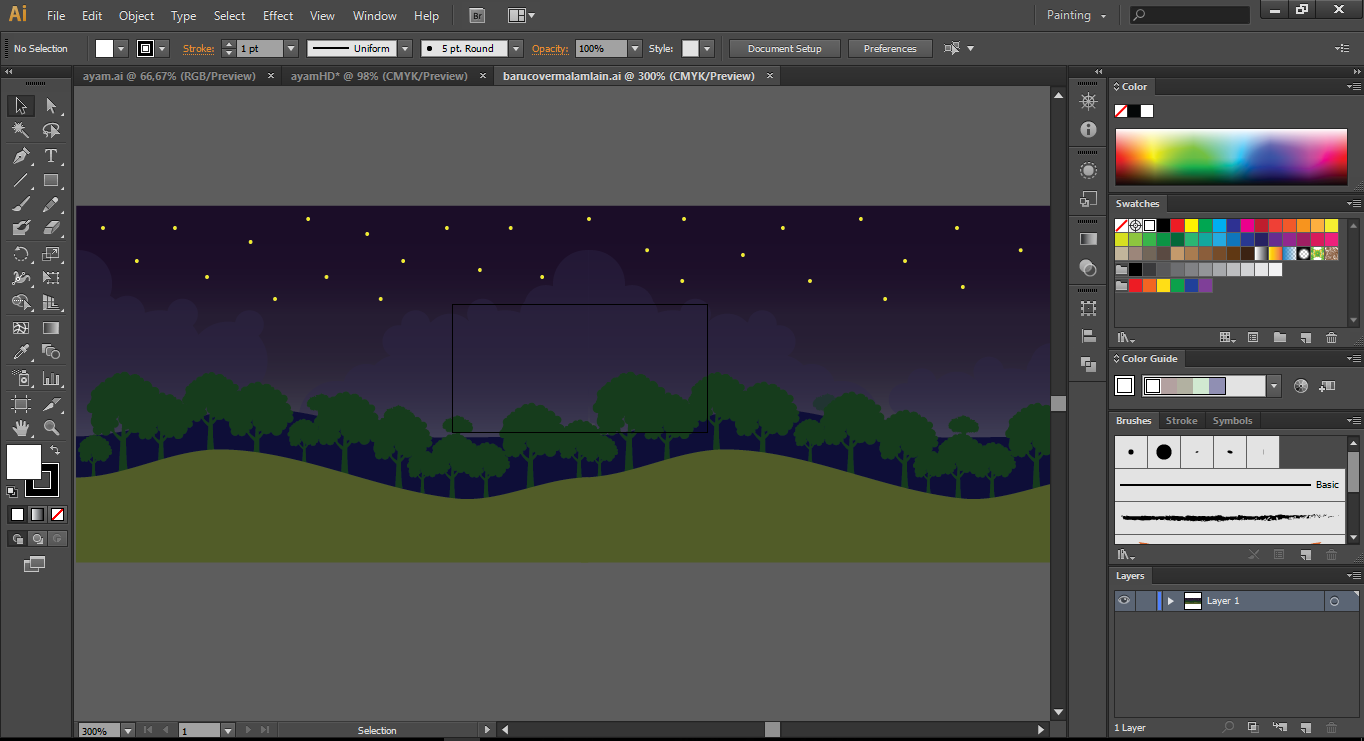
Berikut ini merupakan langkah-langkah pembuatan asset yang terdapat dalam game.

### Pembuatan Background

Dalam pembuatan background, software yang digunakan adalah Adobe Illustrator CS6. Berikut sket awal pembuatan background *game.* 

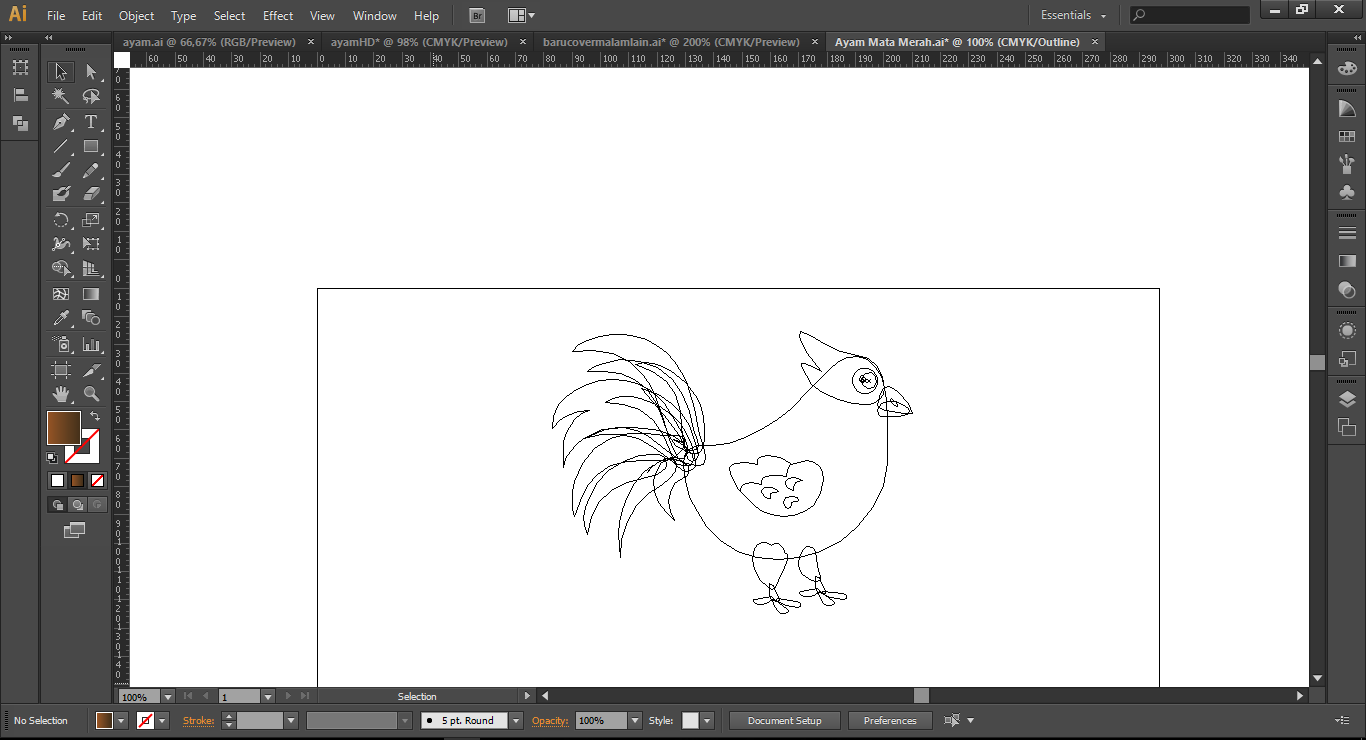
Gambar 5. 1 Sketch Background

Setelah sket di atas proses selanjutnya adalah pembuatan secara digital, adapun tools yang digunakan dari software Adobe Illustrator CS6 adalah Pen Tool, Direct Selection Tool, Rectangle Tool, Polygon Tool, dan Fitur lainnya. Setelah membentuk background sesuai dengan sket dilakukan pewarnaan, untuk mendapatkan gradasi warna diinginkan, dapat menggunakan Transparancy Tool atau Gradient tool.



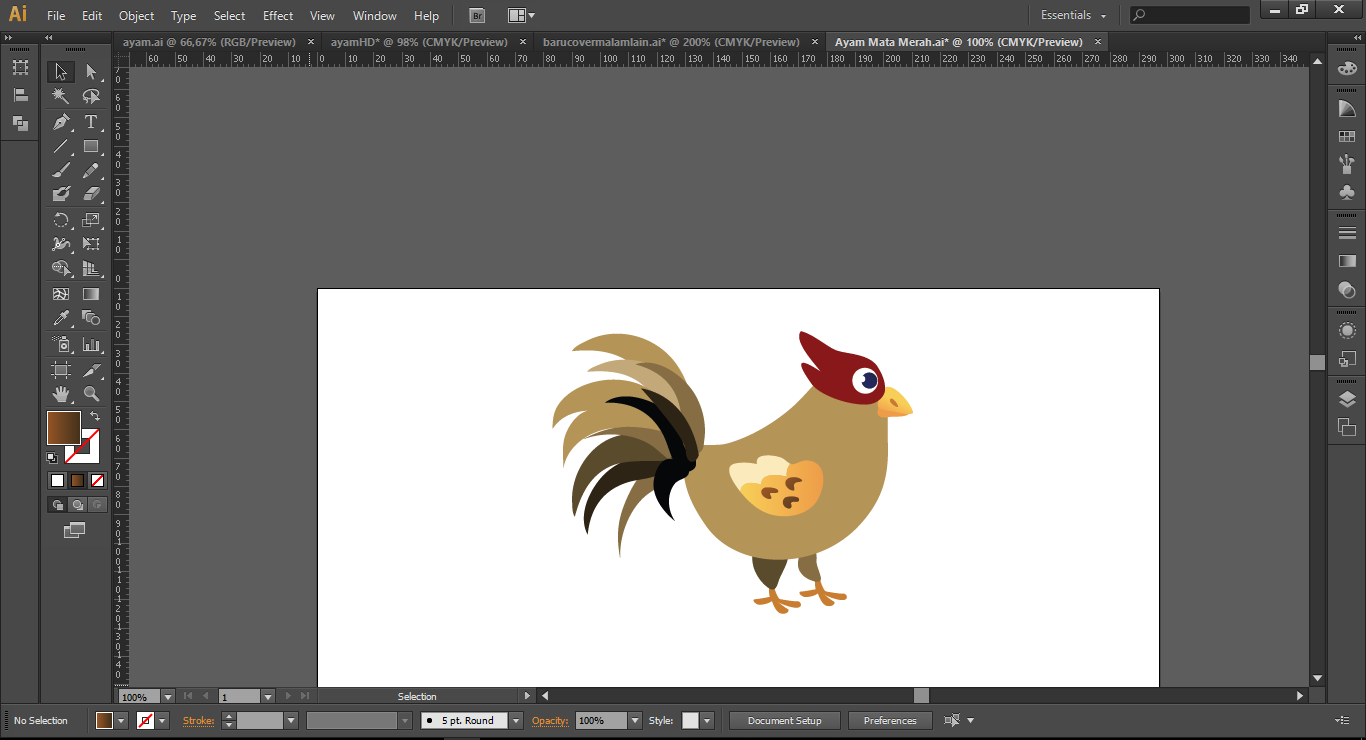
Gambar 5. 2 Hasil Gambar Background

### Pembuatan Karakter pada Adobe Illustrator CS6

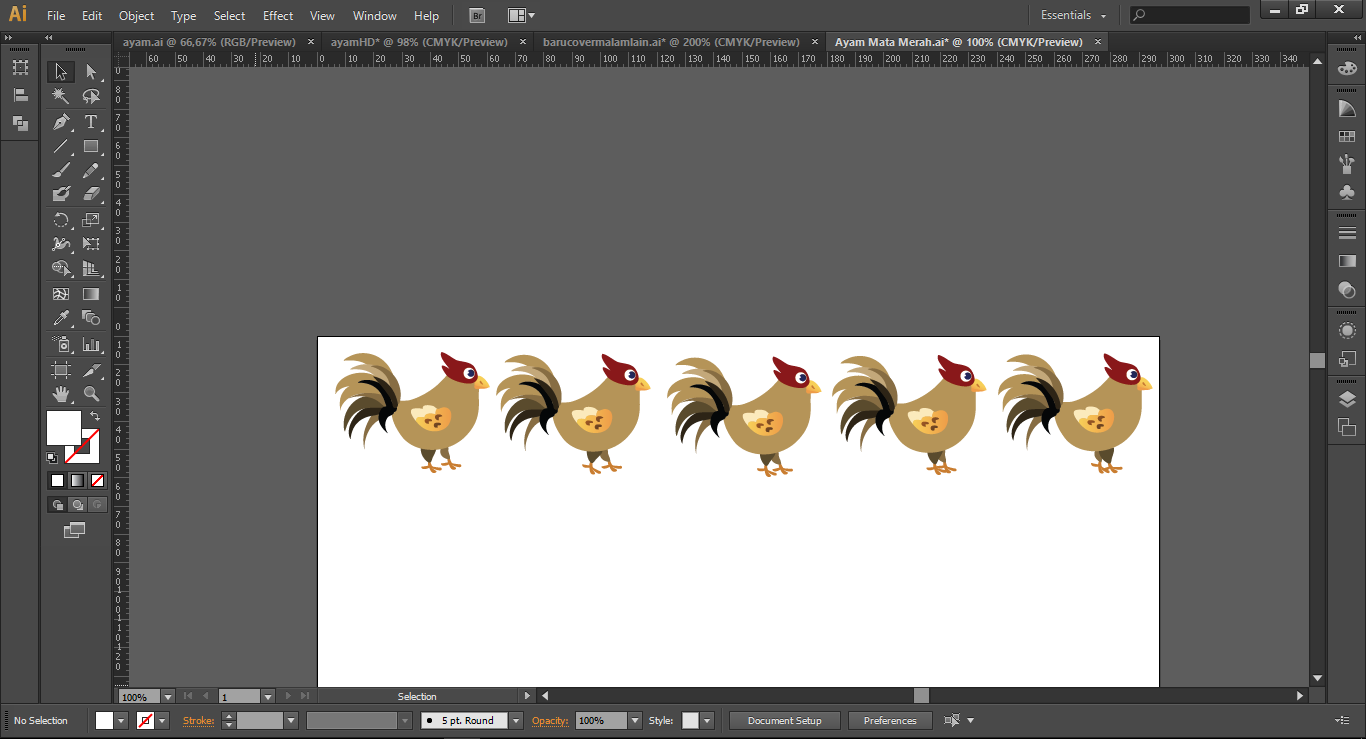


Gambar 5. 3 Hasil Gambar Background

Setelah sket di atas proses selanjutnya adalah pembuatan secara digital, adapun tools yang digunakan dari software Adobe Illustrator CS6 adalah Pen Tool, Direct Selection Tool, Rectangle Tool, Polygon Tool, dan Fitur lainnya. Setelah membentuk background sesuai dengan sket dilakukan pewarnaan, untuk mendapatkan gradasi warna diinginkan, dapat menggunakan Transparancy Tool atau Gradient tool.



Gambar 5. 4 Hasil gambar Player



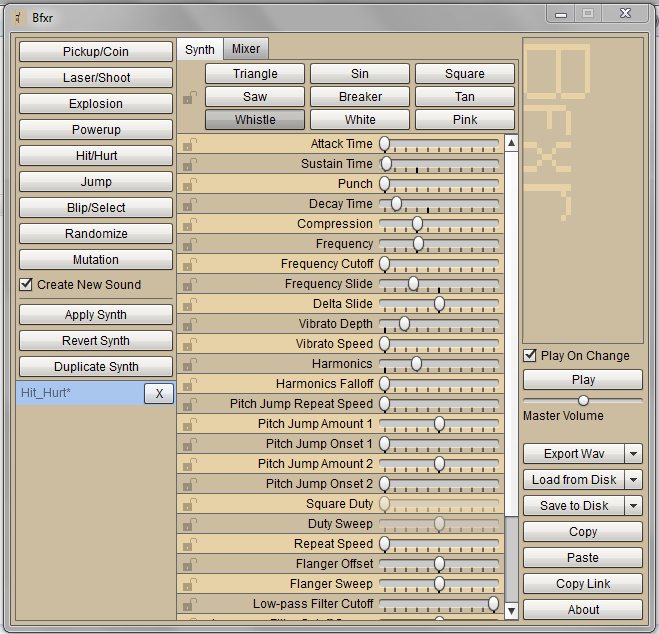
Gambar 5. 5 Sprites Player

Pembuatan karakter *player* dilakukan dengan menggunakan *software* Adobe Illustrator CS6. *Tool*  yang digunakan adalah *Pen Tool, Direct Selection Tool, Rectangle Tool, Polygon Tool*, dan Fitur lainnya.

## **Implementasi Audio Backsound**

Backsound pada permainan menggunakan lagu yang didapat dari internet. Untuk backsound pada main menu menggunakan music Jambalaya Loop.

Sound effect drag drop dibuat dengan menggunakan aplikasi Bfxr. Cukup mudah menggunakan aplikasi ini. Untuk membuat sound drop pertama pilih suara Hit\_Hurt pada panel sebelah kiri kemudian pilik effect pada menu Synth, pilih efek Whistle kemudian ubah pengaturan di bafak panel efek, untuk mencoba sound yang telah diubah klik tombol Play, atau klik Play On Change tombol tersebut akan membuat sound yang sedang di ubah dimainkan ketika terjadi perubahan. Setelah suara sesuai tekan tombol Export Wav untuk menyimpan.



Gambar 5. 6 Hasil Gambar Background

## **Implementasi Scene**

### **Scene Main Menu**

Menu utama atau main menu berisi tombol utama yaitu tombol Main, Pengaturan, Credit, Cara bermain, dan Keluar. Pada bagian tengah atas terdapat logo *game smart chicken* yang juga merupakan judul dari permainan. Pada scene ini backsound lagu yang digunakan adalah Jambalaya Loop.



Gambar 5. 7 Main Menu

### **Scene Peta**

Tampilan Petaberlatar belakangsungai dan rawa. Terdapat 5 tombol pada tampilan Peta yaitu tombol untuk Stage 1, 2, 3, 4 dan tombol home untuk kembali ke menu awal. Tombol setiap stage akan menuju ke halaman permainan sesuai level yang dipilih.



Gambar 5. 8 Tampilan peta

### **Scene Tampilan Stage 1**

Tampilan *stage 1* berlatar belakang hutan dan pegunungan pada siang hari. Terdapat 4 tombol pada tampilan *Stage 1* yaitu tombol navigasi kanan, navigasi kiri, navigasi atas dan pause. Navigasi kanan untuk menjalankan player berjalan ke arah kanan, navigasi kiri untuk menjalankan player berjalan ke arah kiri, navigasi atas untuk menjalankan player melompat dan tombol pause untuk menghentikan permainan.



Gambar 5. 9 Tampilan Stage 1

Tampilan pertanyaan ini merupakan tampilan saat pemain menyentuh kotak pertanyaan. Saat menyentuh kotak pertanyaan, maka muncul panel yang berisi soal. Terdapat 3 tombol pada tampilan panel pertanyaan yaitu tombol pilihan jawaban 1, pilihan jawaban 2, dan pilihan jawaban 3. Dari ketiga jawaban tersebut pemain harus memilih salah satunya, jika jawaban yang diklik benar maka poin akan menambah sebanyak 20 poin dan jika jawaban yang dipilih salah maka poin tidak bertambah. Pertanyaan dalam setiap stage nya akan disediakan sebanyak 7 soal. Berikut tampilan pertanyaan yang akan muncul.



Gambar 5. 10 Tampilan pertanyaan pada Stage 1

Berikut merupakan tampilan ketika pemain menekan tombol Jeda yang terletak di sisi pojok kanan atas. Pemain akan mendapatkan 2 tombol pilihan yaitu Lanjut, dan Home. Selama tampilan Jeda muncul timer pada permainan berhenti akan dilanjutkan apabila pemain menekan tombol Lanjut.



Gambar 5. 11 Tampilan Pause Menu pada Stage 1

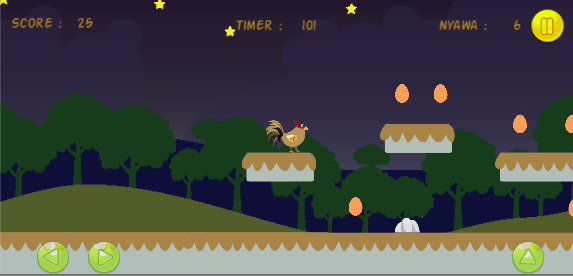
Apabila waktu yang diberikan habis maka pemain mendapatkan tampilan kamu kalah, pemain dapat melakukan dua pilihan kembali ke Menu Level atau Ulang permainan.



Gambar 5. 12 Tampilan Death Menu pada Stage 1

### **Scene Tampilan Stage 2**

Tampilan *stage 2* berlatar belakang hutan dan pemandangan pada malam hari. Terdapat 4 tombol pada tampilan *Stage 2* yaitu tombol navigasi kanan, navigasi kiri, navigasi atas dan pause. Navigasi kanan untuk menjalankan player berjalan ke arah kanan, navigasi kiri untuk menjalankan player berjalan ke arah kiri, navigasi atas untuk menjalankan player dengan melompat dan tombol pause untuk menghentikan permainan sejenak.



Gambar 5. 13 Tampilan Stage 2

Pada tampilan pertanyaan pada *stage 2* berbeda dari tampilan pertanyaan dari stage sebelumnya, karena yang membedakan setiap stage nya adalah tingkat kesulitan pada arena permainan disetiap *stage* nya.



Gambar 5. 14 Tampilan pertanyaan pada Stage 2

Berikut merupakan tampilan ketika pemain menekan tombol Jeda yang terletak di sisi pojok kanan atas. Pemain akan mendapatkan 2 tombol pilihan yaitu Lanjut, dan *Home*. Selama tampilan Jeda muncul timer pada permainan berhenti akan dilanjutkan apabila pemain menekan tombol Lanjut.



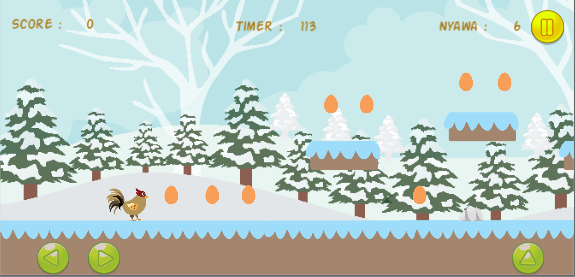
Gambar 5. 15 Tampilan pause menu pada Stage 2



Gambar 5. 16 Tampilan Death menu pada Stage 2

### **Scene Tampilan Stage 3**

Tampilan *stage 3* berlatar belakang salju. Terdapat 4 tombol pada tampilan *Stage 3* yaitu tombol navigasi kanan, navigasi kiri, navigasi atas dan pause. Navigasi kanan untuk menjalankan player berjalan ke arah kanan, navigasi kiri untuk menjalankan player berjalan ke arah kiri, navigasi atas untuk menjalankan player dengan melompat dan tombol pause untuk menghentikan permainan.



Gambar 5. 17 Tampilan Stage 3

Pada tampilan pertanyaan pada *stage 3* berbeda dari tampilan pertanyaan dari stage sebelumnya, karena yang membedakan setiap stage nya adalah tingkat kesulitan disetiap stage nya.



Gambar 5. 18 Tampilan pertanyaan dalam Stage 3

Berikut merupakan tampilan ketika pemain menekan tombol Jeda yang terletak di sisi pojok kanan atas. Pemain akan mendapatkan 2 tombol pilihan yaitu Lanjut, dan Home. Selama tampilan Jeda muncul timer pada permainan berhenti akan dilanjutkan apabila pemain menekan tombol Lanjut.



Gambar 5. 19 Tampilan pause Menu dalam Stage 3



Gambar 5. 20 Tampilan death Menu dalam Stage 3

### **Scene Tampilan Stage 4**

Tampilan *stage 4* berlatar belakang pantai. Terdapat 4 tombol pada tampilan *Stage 4* yaitu tombol navigasi kanan, navigasi kiri, navigasi atas dan pause. Navigasi kanan untuk menjalankan player berjalan ke arah kanan, navigasi kiri untuk menjalankan player berjalan ke arah kiri, navigasi atas untuk menjalankan player dengan melompat dan tombol pause untuk menghentikan permainan.



Gambar 5. 21 Tampilan Stage 4

Pada tampilan pertanyaan pada *stage 3* berbeda dari tampilan pertanyaan dari stage sebelumnya, karena yang membedakan setiap stage nya adalah tingkat kesulitan disetiap stage nya.



Gambar 5. 22 Tampilan pertanyaan dalam Stage 4

Berikut merupakan tampilan ketika pemain menekan tombol Jeda yang terletak di sisi pojok kanan atas. Pemain akan mendapatkan 2 tombol pilihan yaitu Lanjut, dan *Home*. Selama tampilan Jeda muncul timer pada permainan berhenti akan dilanjutkan apabila pemain menekan tombol Lanjut.



Gambar 5. 23 Tampilan pause menu dalam Stage 4



Gambar 5. 24 Tampilan death menu dalam Stage 4

### **Scene Tampilan Cara bermain**



Gambar 5. 25 Tampilan cara bermain

Tampilan cara bermain ini merupakan tampilan saat pemain memilih menu berinisialkan tanda tanya pada bangian pojok kanan atas. Saat memilih tombol tersebut, maka akan muncul scene seperti gambar diatas yang berisikan tata cara bermain. Terdapat 3 tombol dalam scene cara bermain yaitu tombol navigasi kanan, tombol navigasi kiri, dan tombol home. Tombol navigasi kanan untuk melihat langkah selanjutnya, tombol naigasi kiri untuk melihat cara sebelumnya dan tombol home untuk kembali ke halaman home.

### **Tampilan Hasil *Scoring***



Gambar 5. 26 Tampilan hasil scoring

Pada tampilan hasil scoring merupakan hasil akhir dari setiap *stage* nya. Scoring yang dihasilkan berupa Tampilan pertanyaan ini merupakan tampilan saat pemain menyentuh kotak pertanyaan. Saat menyentuh kotak pertanyaan, maka muncul panel yang berisi soal. Terdapat 3 tombol pada tampilan panel pertanyaan yaitu tombol pilihan jawaban 1, pilihan jawaban 2, dan pilihan jawaban 3. Dari ketiga jawaban tersebut pemain harus memilih salah satunya, jika jawaban yang diklik benar maka poin akan menambah sebanyak 20 poin dan jika jawaban yang dipilih salah maka poin tidak bertambah. Pertanyaan dalam setiap stage nya akan disediakan sebanyak 7 soal.

## **Implementasi Metode Fuzzy Tsukamoto**

Implementasi dilakukan ketika pemain telah mengambil telur dan menyelesaikan seluruh pertanyaan sehingga menghasilkan hasil *scoring* berupa bintang. Algoritma *Fuzzy Tsukamoto* dipilih karena algoritma ini memiliki perhitungan secara terpusat, maka dalam aplikasi nya ke dalam sebuah scoring game dinilai cocok.

Pada *game* ini algoritma Fuzzy Tsukamoto dibuat pada sebuah class tersendiri yang dinamakan Fuzzy.cs class ini yang nantinya akan dijadikan acuan apabila ingin menggunakan algoritma Fuzzy Tsukamoto. Berilut *source code* dari class Fuzzy.cs.

using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
public class Fuzzy : MonoBehaviour {  
    private static Fuzzy dataManagerObj;  
    public static Fuzzy getFuzzyObj{  
        get{  
            if(dataManagerObj == null){  
                dataManagerObj = GameObject.FindObjectOfType<Fuzzy>();  
                if(dataManagerObj == null){  
                    GameObject dataManagerTempObj = new GameObject("Fuzzy");  
                    DontDestroyOnLoad(dataManagerTempObj);  
                    dataManagerObj = dataManagerTempObj.AddComponent<Fuzzy>();  
                }  
            }  
            return dataManagerObj;  
        }  
    }  
    public float nyawaSedikit; // 0 - 2  
    public float nyawaSedang;  // 1 - 5  
    public float nyawaBanyak;  // 4 - 6  
  
    public float poinRendah;    // 0 - 120  
    public float poinSedang;    // 90 - 210  
    public float poinTinggi;    // 180 - 300  
  
    public float waktuLambat;   // 0 - 40  
    public float waktuNormal;    // 20 - 100  
    public float waktuCepat;    // 80 - 120  
  
    float[] minimum = new float[27];  
    float[] nilaiZ = new float[27];  
  
    float nyawa;  
    float poin;  
    float waktu;

    // Use this for initialization  
    void Start () {

}

  public int hitungFuzzy(float nyawa, float poin, float waktu) {  
        nyawaSedikit = getNilaiTurunNyawa(nyawa, 0, 2);  
        nyawaSedang = getNilaiSedangNyawa(nyawa, 1, 3, 5);  
        nyawaBanyak = getNilaiNaikNyawa(nyawa, 4, 6);  
        Debug.Log("Nyawa nya adalah : " + nyawaSedikit + " sedikit," + nyawaSedang + " sedang, " + nyawaBanyak + " banyak.");  
  
        poinRendah = getNilaiTurunPoin(poin, 0, 120);  
        poinSedang = getNilaiSedangPoin(poin, 90, 150, 210);  
        poinTinggi = getNilaiNaikPoin(poin, 180, 300);  
        Debug.Log("Poin nya adalah :" + poinRendah + " rendah," + poinSedang + " sedang, " + poinTinggi + " tinggi.");  
  
        waktuLambat = getNilaiTurunWaktu(waktu, 0, 40);  
        waktuNormal = getNilaiSedangWaktu(waktu, 20, 60, 100);  
        waktuCepat = getNilaiNaikWaktu(waktu, 80, 120);  
        Debug.Log("Waktu nya adalah :" + waktuLambat + " lambat," + waktuNormal + " normal, " + waktuCepat + " cepat.");  
  
        cariMin();  
        for (int i = 0; i < minimum.Length; i++)  
        {  
            Debug.Log("Nilai minimum " + i + " : " + minimum[i]);  
        }  
        CariNilaiZ();  
        for (int i = 0; i < nilaiZ.Length; i++)  
        {  
            Debug.Log("Nilai z " + i + " : " + nilaiZ[i]);  
        }  
        float averageZ = defuzzifikasi();  
        int star = getStar(averageZ);  
        Debug.Log("Average Z " + averageZ + ", Star Get : " + star);  
        return star;  
    }  
    public void cariMin() {  
        minimum[0] = Mathf.Min(nyawaSedikit, poinRendah, waktuLambat);  
        minimum[1] = Mathf.Min(nyawaSedikit, poinRendah, waktuNormal);  
        minimum[2] = Mathf.Min(nyawaSedikit, poinRendah, waktuCepat);  
        minimum[3] = Mathf.Min(nyawaSedikit, poinSedang, waktuLambat);  
        minimum[4] = Mathf.Min(nyawaSedikit, poinSedang, waktuNormal);  
        minimum[5] = Mathf.Min(nyawaSedikit, poinSedang, waktuCepat);  
        minimum[6] = Mathf.Min(nyawaSedikit, poinTinggi, waktuLambat);  
        minimum[7] = Mathf.Min(nyawaSedikit, poinTinggi, waktuNormal);  
        minimum[8] = Mathf.Min(nyawaSedikit, poinTinggi, waktuCepat);  
  
        minimum[9] = Mathf.Min(nyawaSedang, poinRendah, waktuLambat);  
        minimum[10] = Mathf.Min(nyawaSedang, poinRendah, waktuNormal);  
        minimum[11] = Mathf.Min(nyawaSedang, poinRendah, waktuCepat);  
        minimum[12] = Mathf.Min(nyawaSedang, poinSedang, waktuLambat);  
        minimum[13] = Mathf.Min(nyawaSedang, poinSedang, waktuNormal);  
        minimum[14] = Mathf.Min(nyawaSedang, poinSedang, waktuCepat);  
        minimum[15] = Mathf.Min(nyawaSedang, poinTinggi, waktuLambat);  
        minimum[16] = Mathf.Min(nyawaSedang, poinTinggi, waktuNormal);  
        minimum[17] = Mathf.Min(nyawaSedang, poinTinggi, waktuCepat);  
  
        minimum[18] = Mathf.Min(nyawaBanyak, poinRendah, waktuLambat);  
        minimum[19] = Mathf.Min(nyawaBanyak, poinRendah, waktuNormal);  
        minimum[20] = Mathf.Min(nyawaBanyak, poinRendah, waktuCepat);  
        minimum[21] = Mathf.Min(nyawaBanyak, poinSedang, waktuLambat);  
        minimum[22] = Mathf.Min(nyawaBanyak, poinSedang, waktuNormal);  
        minimum[23] = Mathf.Min(nyawaBanyak, poinSedang, waktuCepat);  
        minimum[24] = Mathf.Min(nyawaBanyak, poinTinggi, waktuLambat);  
        minimum[25] = Mathf.Min(nyawaBanyak, poinTinggi, waktuNormal);  
        minimum[26] = Mathf.Min(nyawaBanyak, poinTinggi, waktuCepat);  
    }  
    public void CariNilaiZ() {  
        nilaiZ[0] = scoreKurang(minimum[0], 0, 400f);  
        nilaiZ[1] = scoreCukup(minimum[1], 200f, 1000f                                    

    public void CariNilaiZ() {  
        nilaiZ[0] = scoreKurang(minimum[0], 0, 400f);  
        nilaiZ[1] = scoreCukup(minimum[1], 200f, 1000f);  
        nilaiZ[2] = scoreCukup(minimum[2], 200f, 1000f);  
        nilaiZ[3] = scoreKurang(minimum[3], 0, 400f);  
        nilaiZ[4] = scoreKurang(minimum[4], 0, 400f);   
      nilaiZ[5] = scoreCukup(minimum[5], 200f, 1000f);  
        nilaiZ[6] = scoreKurang(minimum[6], 0, 400f);  
        nilaiZ[7] = scoreCukup(minimum[7], 200f, 1000f);  
        nilaiZ[8] = scoreKurang(minimum[8], 0, 400f);  
        nilaiZ[9] = scoreKurang(minimum[9], 0, 400f);  
        nilaiZ[10] = scoreCukup(minimum[10], 200f, 1000f);  
        nilaiZ[11] = scoreBaik(minimum[11], 800f, 1200f);  
        nilaiZ[12] = scoreKurang(minimum[12], 0, 400f);  
        nilaiZ[13] = scoreCukup(minimum[13], 200f, 1000f);  
        nilaiZ[14] = scoreBaik(minimum[14], 800f, 1200f);  
        nilaiZ[15] = scoreKurang(minimum[15], 0, 400f);  
        nilaiZ[16] = scoreCukup(minimum[16], 200f, 1000f);  
        nilaiZ[17] = scoreCukup(minimum[17], 200f, 1000f);  
        nilaiZ[18] = scoreKurang(minimum[18], 0, 400f);  
        nilaiZ[19] = scoreBaik(minimum[19], 800f, 1200f);  
        nilaiZ[20] = scoreBaik(minimum[20], 800f, 1200f);  
        nilaiZ[21] = scoreKurang(minimum[21], 0, 400f);  
        nilaiZ[22] = scoreBaik(minimum[22], 800f, 1200f);  
        nilaiZ[23] = scoreBaik(minimum[23], 800f, 1200f);  
        nilaiZ[24] = scoreCukup(minimum[24], 200f, 1000f);  
        nilaiZ[25] = scoreBaik(minimum[25], 800f, 1200f);  
        nilaiZ[26] = scoreBaik(minimum[26], 800f, 1200f);  
    }  
    // Update is called once per frame  
    void Update () {  
    }  
    #region variable nyawa  
    float getNilaiTurunNyawa(float input, float \_titikRange1, float \_titikRange2){  
        if (input >= \_titikRange2){  
            return 0;  
        } else if (input <= \_titikRange1){  
            return 1;  
        } else if (input > \_titikRange1 && input < \_titikRange2){  
            return (\_titikRange2 - input) / (\_titikRange2 - \_titikRange1);  
        }  
        return 0;  
    }  
  
    float getNilaiSedangNyawa(float input, float \_titikRange1, float \_titikRange2, float \_titikRange3){  
        if (input <= \_titikRange1 || input >= \_titikRange3){  
            return 0;  
        } else if (input > \_titikRange1 && input < \_titikRange2){  
            return (input - \_titikRange1) / (\_titikRange2 - \_titikRange1);

} else if (input > \_titikRange2 && input < \_titikRange3){  
            return (\_titikRange3 - input) / (\_titikRange3 - \_titikRange2);  
        }  
        return 0;  
    }  
  
    float getNilaiNaikNyawa(float input, float \_titikRange1, float \_titikRange2){  
        if (input <= \_titikRange1){  
            return 0;  
        } else if (input >= \_titikRange2){  
            return 1;  
      } else if (input > \_titikRange1 && input < \_titikRange2){  
            return ( input - \_titikRange1) / (\_titikRange2 - \_titikRange1);  
        }  
        return 0;  
    }  
    #endregion variable nyawa

   #region variable poin  
    float getNilaiTurunPoin(float input, float \_titikRange1, float \_titikRange2){  
        if (input >= \_titikRange2){  
            return 0;  
        }  
        else if (input <= \_titikRange1){  
            return 1;  
        }  
        else if (input > \_titikRange1 && input < \_titikRange2){  
            return (\_titikRange2 - input) / (\_titikRange2 - \_titikRange1);  
        }  
        return 0;  
    }  
  
    float getNilaiSedangPoin(float input, float \_titikRange1, float \_titikRange2, float \_titikRange3){  
        if (input <= \_titikRange1 || input >= \_titikRange3){  
            return 0;  
        }  
        else if (input > \_titikRange1 && input < \_titikRange2){

return (input - \_titikRange1) / (\_titikRange2 - \_titikRange1);  
        }  
else if (input > \_titikRange2 && input < \_titikRange3){ return (\_titikRange3 - input) / (\_titikRange3 - \_titikRange2);  
        }

        return 0;  
    }

float getNilaiNaikPoin(float input, float \_titikRange1, float \_titikRange2){  
        if (input <= \_titikRange1){  
            return 0;  
        }  
        else if (input >= \_titikRange2){  
            return 1;  
        }  
        else if (input > \_titikRange1 && input < \_titikRange2){  
            return (input - \_titikRange1) / (\_titikRange2 - \_titikRange1);  
        }  
        return 0;  
    }  
    #endregion variable poin

#region variable waktu  
    float getNilaiTurunWaktu(float input, float \_titikRange1, float \_titikRange2){  
        if (input >= \_titikRange2){  
            return 0;  
        }  
        else if (input <= \_titikRange1){  
            return 1;  
        }  
        else if (input > \_titikRange1 && input < \_titikRange2){  
            return (\_titikRange2 - input) / (\_titikRange2 - \_titikRange1);  
        }  
        return 0;  
    }  
  
    float getNilaiSedangWaktu(float input, float \_titikRange1, float \_titikRange2, float \_titikRange3){  
        if (input <= \_titikRange1 || input >= \_titikRange3){  
            return 0;  
        }  
        else if (input > \_titikRange1 && input < \_titikRange2){  
            return (input - \_titikRange1) / (\_titikRange2 - \_titikRange1);  
        }  
        else if (input > \_titikRange2 && input < \_titikRange3){  
            return (\_titikRange3 - input) / (\_titikRange3 - \_titikRange2);  
        }  
        return 0;  
    }

float getNilaiNaikWaktu(float input, float \_titikRange1, float \_titikRange2){  
        if (input <= \_titikRange1){  
            return 0;  
        }

        }  
        else if (input >= \_titikRange2){  
            return 1;  
        }  
        else if (input > \_titikRange1 && input < \_titikRange2){  
            return (input - \_titikRange1) / (\_titikRange2 - \_titikRange1);  
        }  
        return 0;  
    }  
    #endregion variable Waktu  
  
    public float scoreKurang(float \_input, float \_titikRange1, float \_titikRange2) {   
        return (\_titikRange2-(\_input\*(\_titikRange2 - \_titikRange1)));  
    }  
    public float scoreCukup(float \_input, float \_titikRange2, float \_titikRange3) {   
        return (\_titikRange3-(\_input \* (\_titikRange3 - \_titikRange2)));  
    }  
    public float scoreBaik(float \_input, float \_titikRange1, float \_titikRange2) {   
        return (\_titikRange1+(\_input\*(\_titikRange2 - \_titikRange1)));  
    }  
  
    public float defuzzifikasi() {  
        float atas = 0, bawah = 0;  
        for (int i = 0; i < nilaiZ.Length; i++){  
            atas += nilaiZ[i] \* minimum[i];  
            bawah += minimum[i];  
        }  
        Debug.Log("Atas : " + atas + ", Bawah " + bawah);  
        return atas / bawah;  
    }  
  
    public int getStar(float \_nilaiAkhirFuzzy) {  
  
        if(\_nilaiAkhirFuzzy >= 0 && \_nilaiAkhirFuzzy <= 400f){  
            return 1;  
        }else if(\_nilaiAkhirFuzzy >= 400f && \_nilaiAkhirFuzzy <= 800f){  
            return 2;  
        } else if (\_nilaiAkhirFuzzy >= 800f && \_nilaiAkhirFuzzy <= 1200f){  
            return 3;  
        }  
        return 0;  
  
    }  
}

Dalam implementasi *script* Fuzzy.cs terdapat 3 variabel Poin, Nyawa, Waktu yang dipanggil dalam script TimeManager.cs.

Setelah variabel input terkakulasi, maka dalam script KalkulasiScore.cs memanggil fungsi bintang yang akan menampilkan hasil score akhir.

    public int getStar(){  
        int star = Fuzzy.getFuzzyObj.hitungFuzzy (nyawa, poin, waktu);  
        ff = GameObject.Find ("Fuzzy").GetComponent<Fuzzy> ();  
        Debug.Log (ff.name);  
        if (star == 1) {  
            b1 [0].SetActive (true);  
            b1 [1].SetActive (false);  
            b1 [2].SetActive (false);  
        }

        else if (star == 2) {  
            b1 [1].SetActive (true);  
            b1 [0].SetActive (false);  
            b1 [2].SetActive (false);  
        }  
        else if (star == 3) {  
            b1 [2].SetActive (true);  
            b1 [1].SetActive (false);  
            b1 [0].SetActive (false);  
        }  
        Debug.Log ("kalkulasi score star : " + star);  
        return star;  
    }

# **BAB VI. PENGUJIAN DAN PEMBAHASAN**

Bab VI berisi tentang pengujian yang dilakukan pada aplikasi permainan dari sisi pembuat maupun pengguna. Tahapan ini dilakukan setelah proses analisa, perancangan dan implementasi telah selesai dilakukan. Pengujian yang dulakukan meliputi dua tahapan yaitu pengujian *Alpha* dan pengujian *Betha.*

## **Pengujian Aplikasi *Game***

Pengujian aplikasi merupakan tahap selanjutnya setelah aplikasi selesai dalam pembuatannya dan dimaksudkan untuk menguji semua elemen-elemen perangkat lunak yang dibuat apakah sudah sesuai dengan perancangan.

* + 1. Rencana Pengujian

Adapun rancangan pengujian menu *game* yang akan diuji dengan teknik pengujian *Black Box*. Pengujian *Black Box* terfokus pada persyaratan fungsional perangkat lunak. Daftar rencana pengujian dapat dilihat pada tabel 6.1.

Tabel 6. 1 Pengujian Black Box

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Komponen yang di Uji** | **Skenario Butir Uji** | **Jenis Uji** |
|  | Tampilan awal | Terdapat tampilan menu utama | *Black Box* |
|  | Menu Main | Memilih tombol main untuk memulai permainan | *Black Box* |
|  | Menu Pengaturan | Memilih tombol pengaturan untuk mengatur suara dan musik | *Black Box* |
|  | Menu Credit | Memilih tombol kredit untuk melihat profil pembuat | *Black Box* |
|  | Tombol Suara | Memilih tombol suara untuk mematikan/menghidupkan suara | *Black Box* |
|  | Tombol Musik | Memilih tombol musik untuk mematikan/menghidupkan musik | *Black Box* |
|  | Tampilan Menu stage | Terdapat tampilan menu level dengan pilihan 3 stage permainan. | *Black Box* |
|  | Pilih Stage | Memilih salah satu Stage | *Black Box* |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Permainan di Stage 1 | Memulai permainan di Stage 1 | *Black Box* |
|  | Permainan di Stage 2 | Memulai permainan di Stage 2 | *Black Box* |
|  | Permainan di Stage 3 | Memulai permainan di Stage 3 | *Black Box* |
|  | Permainan di Stage 4 | Memulai permainan di Stage 4 | *Black Box* |
|  | Tombol Jeda | Memilih tombol Jeda untuk menghentikan permainan sementara. | *Black Box* |
|  | Tombol Keluar | Memilih tombol keluar untuk keluar dari aplikasi | *Black Box* |
|  | Tombol Resume | Memilih tombol resume untuk main lagi setelah tombol pause | *Black Box* |
|  | Tombol Home | Memlih tombol Home untuk kembali ke halaman utama | *Black Box* |
|  | Tombol Lanjut | Memlih tombol lanjut untuk melihat cara bermain | *Black Box* |
|  | Tombol Kembali | Memlih tombol kembali untuk melihat cara bermain | *Black Box* |
|  | Perolehan Skor Akhir | Memperoleh skor akhir dari total skor yang didapat dari setiap scene pada setiap stage. | *Black Box* |

### Pengujian Alpha

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian alpha pada *game smart chicken*. Berikut hasil pengujian dari game *smart chicken.*

Tabel 6. 2 Pengujian Alpha

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Kasus/diuji | Skenario Uji | Hasil Yang Diharapkan | Hasil Pengujian |
| 1. | Tampilan awal | Terdapat tampilan menu utama | Ketika memilih tombol masuk, tampilan Main Menu *game* akan tampil | Berhasil |
| 2. | Menu Main | Memilih tombol main untuk memulai permainan | Ketika memilih tombol Mulai tampilan *Gameplay* akan tampil | Berhasil |
| 3. | Menu Pengaturan | Memilih tombol pengaturan untuk mengatur suara dan musik | Ketika memilih tombol Pengaturan, panel sound akan tampil. | Berhasil |
| 4. | Menu Credit | Memilih tombol kredit untuk melihat profil pembuat | Ketika memilih tombol Credit akan tampil tampilan Credit. | Berhasil |
| 5. | Tombol Suara | Memilih tombol suara untuk mematikan/menghidupkan suara | Ketika memilih tombol Suara maka akan on/off suara. | Berhasil |
| 6. | Tampilan Menu stage | Terdapat tampilan menu level dengan pilihan 3 stage permainan. | Ketika memilih tombol keluar maka aplikasi akan tertutup. | Berhasil |
| 7. | Pilih Stage | Memilih salah satu Stage yang terdapat pada peta | Ketika memilih tombol Stage maka akan tampil Peta yang berisi macam stage. | Berhasil |
| 8. | Permainan di Stage 1 | Memulai permainan di Stage 1 | Ketika memilih tombol Stage 1 akan tampil Scene Gameplay | Berhasil |
| 9. | Permainan di Stage 2 | Memulai permainan di Stage 2 | Ketika memilih tombol Stage 2 akan tampil Scene Gameplay | Berhasil |
| 10. | Permainan di Stage 3 | Memulai permainan di Stage 3 | Ketika memilih tombol Stage 3 akan tampil Scene Gameplay | Berhasil |
| 11. | Permainan di Stage 4 | Memulai permainan di Stage 4 | Ketika memilih tombol Stage 4 akan tampil Scene Gameplay | Berhasil |
| 12. | Tombol Jeda | Memilih tombol Jeda untuk menghentikan permainan sementara. | Ketika memilih tombol jeda maka permainan berhenti. | Berhasil |
| 13. | Tampilan Berhasil | Tampilan berhasil | Ketika berhasil menyusun *puzzle* pada Level 1,2 dan 3, akan tampil panel skor dan bonus yang didapat. | Berhasil |
| 14. | Tampilan Gagal | Kehabisan waktu | Ketika waktu pada setiap level habis makan panel gagal akan muncul. | Berhasil |

### Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secar objektif, dimana pengujian langsung melibatkan para penggna aplikasi *game* secara langsung dengan menggunakan kuesioner mengenai kepuasan para pengguna terhadap *game* yang dibangun. Uji coba *game* *Smart Chicken* dilakukan pada anak sekolah dasar. Dengan jumlah responden sebanyak 10 orang. Teknik pengambilan sampel yang digunakan adalah simple random sampling. Berikut adalah kuesinoer yang diisi oleh pemain setelah mencoba *game Smart Chicken*.

Tabel 6. 3 Kuisioner pengujian beta

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Pertanyaan | Skala Jawaban | | | | |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Apakah tampilan halaman utama menarik? |  |  |  |  |  |
| 2. | Apakah tampilan *gameplay* secara umum menarik? |  |  |  |  |  |
| 3. | Apakah alur permainan mudah dipahami? |  |  |  |  |  |
| 4. | Apakah tombol arah / *controller* pada *gameplay* dapat dimainkan dengan mudah? |  |  |  |  |  |
| 5. | Apakah tingkat kesulitan dalam bermain sudah sesuai dengan tingkatan level? |  |  |  |  |  |
| 6. | Apakah dengan *game* ini membuat proses pembelajaran matematika lebih menyenangkan? |  |  |  |  |  |
| 7. | Apakah setelah bermain *game* ini membuat anda lebih mudah memahami matematika? |  |  |  |  |  |
| 8. | Apakah semua tombol-tombol berfungsi dengan baik? |  |  |  |  |  |
| 9. | Apakah karakter menarik untuk dimainkan? |  |  |  |  |  |

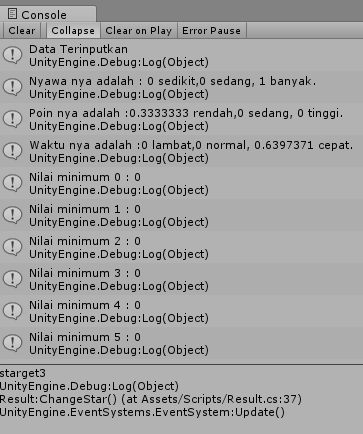
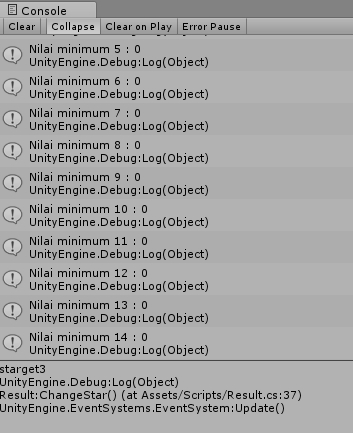
### Pengujian Metode Fuzzy Tsukamoto

Pengujian metode Fuzzy Tsukamotoditerapkan pada hasil akhir scoring yang memiliki 3 kriteria yaitu Poin, Nyawa, dan Waktu. Hasil pengujian akan dijelaskan pada tabel.

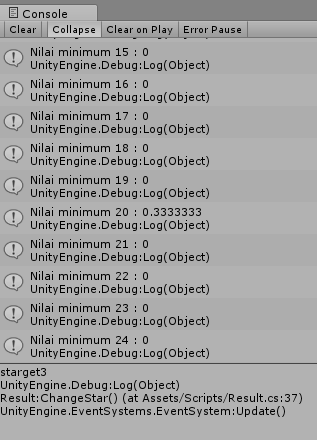
Tabel 6. 4 Pengujian metode Fuzzy Tsukamoto

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Poin** | **Nyawa** | **Waktu** | **Score** | **Min (α)** | **z** | **α \* z** |
| 0 | 1 | 0 | 1 | 0 | 400 | 0 |
| 0 | 1 | 0 | 2 | 0 | 850 | 0 |
| 0 | 1 | 0,525 | 2 | 0 | 850 | 0 |
| 0 | 0 | 0 | 1 | 0 | 400 | 0 |
| 0 | 0 | 0 | 1 | 0 | 400 | 0 |
| 0 | 0 | 0,525 | 2 | 0 | 850 | 0 |
| 0 | 0 | 0 | 1 | 0 | 400 | 0 |
| 0 | 0 | 0 | 2 | 0 | 850 | 0 |
| 0 | 0 | 0,525 | 1 | 0 | 400 | 0 |
| 0 | 1 | 0 | 1 | 0 | 400 | 0 |
| 0 | 1 | 0 | 2 | 0 | 850 | 0 |
| 0 | 1 | 0,525 | 3 | 0 | 800 | 0 |
| 0 | 0 | 0 | 1 | 0 | 400 | 0 |
| 0 | 0 | 0 | 2 | 0 | 850 | 0 |
| 0 | 0 | 0,525 | 3 | 0 | 800 | 0 |
| 0 | 0 | 0 | 1 | 0 | 400 | 0 |
| 0 | 0 | 0 | 2 | 0 | 850 | 0 |
| 0 | 0 | 0,525 | 2 | 0 | 850 | 0 |
| 0,583333333 | 1 | 0 | 1 | 0 | 400 | 0 |
| 0,583333333 | 1 | 0 | 3 | 0 | 800 | 0 |
| 0,583333333 | 1 | 0,525 | 3 | 0,525 | 1010 | 530,25 |
| 0,583333333 | 0 | 0 | 1 | 0 | 400 | 0 |
| 0,583333333 | 0 | 0 | 3 | 0 | 800 | 0 |
| 0,583333333 | 0 | 0,525 | 3 | 0 | 800 | 0 |
| 0,583333333 | 0 | 0 | 2 | 0 | 850 | 0 |
| 0,583333333 | 0 | 0 | 3 | 0 | 800 | 0 |
| 0,583333333 | 0 | 0,525 | 3 | 0 | 800 | 0 |
|  |  |  | **JUMLAH α** | 0,525 |  | 530,25 |
|  |  |  |  |  | **∑z / ∑α** | 1010 |
|  |  |  |  |  | Hasil Ranking | 3 |

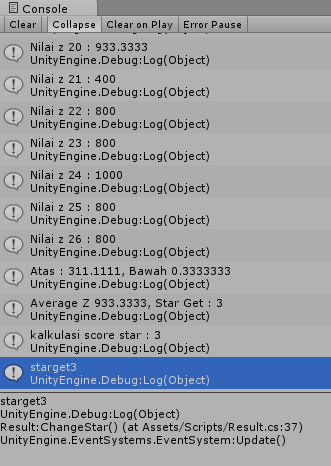
Pada *inspector* Unity diinisiasikan dengan *starget*. Dimana starget1 menunjukkan *score* dengan bintang 1, starget2 menunjukkan *score* dengan bintang 2*, dan* starget3 menunjukkan *score* dengan bintang *3.*

Gambar 6. 1 Tampilan Console Hasil perhitungan Fuzzy

Gambar 6. 2 Tampilan Console Hasil perhitungan Fuzzy

Gambar 6. 3 Tampilan Console Hasil perhitungan Fuzzy



Gambar 6. 4 Tampilan Console Hasil perhitungan Fuzzy

## **Pembahasan Aplikasi *Game***

### Hasil Pengujian Alpha

Berdasarkan hasil pengujian *Alpha* di tabel 6.2, diperoleh bahwa *Game Smart Chicken* ini secara fungsional mengeluarkan hasil sesuai dengan yang diharapkan.

### Hasil Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana pengujian langsung melibatkan para penggna aplikasi *game* secara langsung dengan menggunakan kuesioner mengenai kepuasan para pengguna terhadap *game* yang dibangun. Berikut adalah hasil kuesioner pengujian beta.

Tabel 6. 5 Hasil kuesioner

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nama | JK | Umur | Skala Jawaban | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Safa | P | 11 | 3 | 4 | 5 | 5 | 5 | 4 | 4 | 4 | 3 |
| Gita | P | 10 | 4 | 5 | 4 | 5 | 4 | 3 | 4 | 4 | 3 |
| Soraya | P | 8 | 3 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 3 |
| Agnia | P | 9 | 4 | 4 | 5 | 5 | 4 | 4 | 4 | 4 | 3 |
| Rania | P | 10 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 4 | 4 |
| Diny | P | 8 | 4 | 4 | 5 | 4 | 4 | 4 | 5 | 5 | 3 |
| Ardi | L | 7 | 3 | 3 | 5 | 5 | 5 | 4 | 4 | 4 | 3 |
| Rehan | L | 11 | 4 | 4 | 4 | 4 | 4 | 3 | 4 | 5 | 4 |
| Nadiha | P | 10 | 3 | 3 | 5 | 5 | 5 | 4 | 5 | 4 | 3 |
| Rizky | L | 11 | 4 | 4 | 5 | 5 | 4 | 3 | 4 | 4 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sangat baik** | **Baik** | **Cukup** | **Kurang** | **Sangat Kurang** |
| 5 | 4 | 3 | 2 | 1 |

Berdasarkan hasil kuesioner tersebut, dicari prosentase masing-masing jawaban dengan menggunakan rumus : H = J/U\*100%

Keterangan :

H = Hasil perhitungan

J = Banyaknya jawaban (skala jawaban) oleh Responden

U = Jumlah *User*

Berikut ini hasil perhitungan prosentase dari jawaban hasil kuesioner yang terlah dilakukan terhadap 10 *user* yang dijadikan contoh penelitian *Game Smart Chicken.*

* + - * 1. Pertanyaan pertama “Apakah Tampilan Halaman Utama menarik?”

Tabel 6. 6 Hasil pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Skala jawaban | 1 | 2 | 3 | 4 | 5 |
| Responden | - | - | 4 | 6 | - |
| Hasil | 0% | 0% | 40% | 60% | 0% |

Berdasarkan hasil tabel, dapat dilihat bahwa 10 responden sebanyak 4 responden menjawab cukup dengan prosentase 40%, 6 responden menjawab baik dengan prosentase 60%. Dengan hasil tersebut dapat disimpulkan bahwa tampilan halaman utama pada game ini menarik.

* + - * 1. Pertanyaan pertama “Apakah Tampilan *gameplay* secara umum menarik?”

Tabel 6. 7 Hasil pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Skala jawaban | 1 | 2 | 3 | 4 | 5 |
| Responden | - | - | 2 | 6 | 2 |
| Hasil | 0% | 0% | 20% | 60% | 20% |

Berdasarkan hasil tabel, dapat dilihat bahwa 20 responden sebanyak 2 responden menjawab cukup dengan prosentase 20%, 6 responden menjawab baik dengan prosentase 60%, dan 2 responden menjawab sangat baik dengan prosentase 20%. Dengan hasil tersebut dapat disimpulkan bahwa tampilan *gameplay* secara umum pada game ini baik.

* + - * 1. Pertanyaan pertama “Apakah alur permainan mudah dipahami?”

Tabel 6. 8 Hasil pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Skala jawaban | 1 | 2 | 3 | 4 | 5 |
| Responden | - | - | - | 4 | 6 |
| Hasil | 0% | 0% | 0% | 40% | 60% |

Berdasarkan hasil tabel, dapat dilihat bahwa 10 responden sebanyak 4 responden menjawab baik dengan prosentase 40%, 6 responden menjawab sangat baik dengan prosentase 60%. Dengan hasil tersebut dapat disimpulkan bahwa alur permainan pada game ini mudah dipahami.

* + - * 1. Pertanyaan pertama “Apakah tombol arah / *controller* pada gameplay dapat dimainkan dengan mudah?”

Tabel 6. 9 Hasil pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Skala jawaban | 1 | 2 | 3 | 4 | 5 |
| Responden | - | - | - | 2 | 8 |
| Hasil | 0% | 0% | 0% | 20% | 80% |

Berdasarkan hasil tabel, dapat dilihat bahwa 10 responden sebanyak 2 responden menjawab baik dengan prosentase 20%, 8 responden menjawab sangat baik dengan prosentase 80%. Dengan hasil tersebut dapat disimpulkan bahwa tombol/ *controller* pada game ini dapat dimainkan dengan mudah.

* + - * 1. Pertanyaan pertama “Apakah Tingkat kesulitan dalam bermain sudah sesuai dengan tingkatan level?”

Tabel 6. 10 Hasil pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Skala jawaban | 1 | 2 | 3 | 4 | 5 |
| Responden | - | - | - | 7 | 3 |
| Hasil | 0% | 0% | 0% | 70% | 30% |

Berdasarkan hasil tabel, dapat dilihat bahwa 10 responden sebanyak 7 responden menjawab baik dengan prosentase 70%, dan 3 responden menjawab sangat baik dengan prosentase 30%. Dengan hasil tersebut dapat disimpulkan bahwa tingkat kesulitan di setiap level pada game ini baik.

* + - * 1. Pertanyaan pertama “Apakah dengan *game* ini membuat proses pembelajaran matematika lebih menyenangkan?”

Tabel 6. 11 Hasil pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Skala jawaban | 1 | 2 | 3 | 4 | 5 |
| Responden | - | - | 3 | 6 | 1 |
| Hasil | 0% | 0% | 30% | 60% | 10% |

Berdasarkan hasil tabel, dapat dilihat bahwa 10 responden sebanyak 3 responden menjawab cukup dengan prosentase 30%, 6 responden menjawab baik dengan prosentase 60%, dan 1 responden menjawab sangat baik dengan prosentase 10% . Dengan hasil tersebut dapat disimpulkan bahwa game ini membuat proses pembelajaran matematika lebih menyenangkan.

* + - * 1. Pertanyaan pertama “Apakah setelah bermain game ini membuat anda lebih mudah memahami matematika?”

Tabel 6. 12 Hasil pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Skala jawaban | 1 | 2 | 3 | 4 | 5 |
| Responden | - | - | - | 8 | 2 |
| Hasil | 0% | 0% | 0% | 80% | 20% |

Berdasarkan hasil tabel, dapat dilihat bahwa 10 responden sebanyak 8 responden menjawab baik dengan prosentase 80%, dan 2 responden menjawab sangat baik dengan prosentase 20%. Dengan hasil tersebut dapat disimpulkan bahwa setelah bermain game ini membuat anda lebih mudah memahami matematika dengan baik.

* + - * 1. Pertanyaan kedelapan “Apakah semua tombol berfungsi dengan baik?”

Tabel 6. 13 Hasil pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Skala jawaban | 1 | 2 | 3 | 4 | 5 |
| Responden | - | - | - | 8 | 2 |
| Hasil | 0% | 0% | 0% | 80% | 20% |

Berdasarkan hasil tabel, dapat dilihat bahwa 10 responden sebanyak 8 responden menjawab baik dengan prosentase 80%, dan 2 responden menjawab sangat baik dengan prosentase 20%. Dengan hasil tersebut dapat disimpulkan bahwa semua tombol berfungsi dengan baik.

* + - * 1. Pertanyaan kesembilan “Apakah karakter menarik untuk dimainkan?”

Tabel 6. 14 Hasil pengujian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Skala jawaban | 1 | 2 | 3 | 4 | 5 |
| Responden | - | - | 8 | 2 | - |
| Hasil | 0% | 0% | 80% | 20% | 0% |

Berdasarkan hasil tabel, dapat dilihat bahwa 10 responden sebanyak 8 responden menjawab cukup dengan prosentase 80%, 2 responden menjawab baik dengan prosentase 20%. Dengan hasil tersebut dapat disimpulkan bahwa karakter pada game ini menarik untuk dimainkan.

* + - * 1. Persentase Hasil Kuisioner

Persentase hasil kuisioner dari 10 responden berdasarkan 9 pertanyaan, memiliki hasil dengan rata-rata 70% hasil uji coba *user*. Berikut pada tabel 6.15 adalah tabel hasil kuisioner uji coba *user :*

Tabel 6. 15 Hasil kuisioner uji coba user.

|  |  |  |
| --- | --- | --- |
| **No** | **Hasil Skala** | **Persentase** |
| 1 | Baik | 60% |
| 2 | Baik | 60% |
| 3 | Sangat Baik | 60% |
| 4 | Sangat Baik | 80% |
| 5 | Baik | 70% |
| 6 | Baik | 60% |
| 7 | Baik | 80% |
| 8 | Baik | 80% |
| 9 | Cukup | 80% |
|  | **Rata - rata** | 70% |

Kesimpulan dari tabel 6.15 yaitu: Secara umum aplikasi ini mendapatkan nilai lebih dari 50%. Mengenai tampilan game, fungsi dari tombol, alur permainan, bentuk karakter, tingkat kesulitan, pengacakan pertanyaan, dan proses bermain menurut user sudah baik.

# **BAB VII. PENUTUP**

Bab VII menjelaskan tentang kesimpulan yang didapat pada saat pengerjaan skripsi melalui uji coba yang dilakukan dan analisa yang dilakukan dalam penelitiannya. Bab ini juga berisi saran yang bisa dilakukan untuk penelitian di masa yang akan datang.

## **Kesimpulan**

Kesimpulan yang didapatkan dari pembuatan skripsi “Rancang Bangun *Game Smart Chicken* menggunakan metode *Fuzzy Tsukamoto*” adalah sebagai berikut :

1. Dengan adanya *Game Smart Chicken* ini, pemain dapat bermain sambil belajar mengenai dasar-dasar matematika dengan tampilan yang menarik dan fungsi yang berjalan dengan baik berdasarkan hasil kuisioner terhadap responden yang mendapatkan prosentase rata-rata sebesar 70%.
2. Metode *fuzzy tsukamoto* yang diterapkan pada hasil scoring dapat berjalan dengan baik dan semestinya.

## **Saran**

*Game* ini bermanfaat sebagai media pengetahuan, namun masih ditemukan kelemahan dalam aplikasi ini, yaitu penambahan fitur *multiplayer* untuk *game* dan latarbelakang cerita. Diharapkan nantinya akan ada pengembangan fitur *multiplayer* agar permainan dapat dimainkan lebih dari satu anak, serta penambahan latar belakang agar pengetahuan yang di dapat mengenai matematika lebih dapat dikembangkan dan lebih baik lagi.

# **DAFTAR PUSTAKA**

[1] Kusumadewi, S., Purnomo, H. 2010. Aplikasi Logika Fuzzy untuk Pendukung Keputusan Edisi 2. Graha Ilmu. Yogyakarta.

[2] Sutojo T. S.Si.,M.Kom., Mulyanto Edy S.Si.,M.Kom., Suhartono Dr. Vincent. 2011. “*Kecerdasan Buatan*.” Yogyakarta. Andi Yogyakarta.

[3] Aisy Muhammad Rozsidy. 2016. “*Rancang Bangun Game FIghting Pewayangan Bahuraksa Arena.”* Malang. Politeknik Negeri Malang.

[4] Safaat H, Nazruddin. 2014. *Android Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*.Bandung. Informatika.

[5] Zaki, Ali. 2010. MemanfaatkanBeragam PerangkatTeknologi Digital.Yogyakarta: Salemba Infotek.

[6] Luther, A. C., 1994, *Authoring Interactive Multimedia, Academic Press, Inc*., Massachusettes.

Sumber Lain :

(17 Agustus 2017) Tersedia di : <http://www.viva.co.id/digital/digilife/162371-pemain-game-indonesia-naik-33-per-tahun>

**LAMPIRAN**

**PROFIL PENULIS**

****

Nama : Rika Nomilasari

Tempat, Tanggal Lahir : Malang, 25 November 1995

Alamat : Perum. Bumi Tunggulwulung Indah R-14

Malang

Usia : 22 tahun

Agama : Islam

Kewarganegaraan : Indonesia

Nomor Telepon : 081333229649

E-mail : [rikanomila@gmail.com](mailto:rikanomila@gmail.com)



**RIWAYAT PENDIDIKAN**

Tahun Lulus

2007 : SDN Sidokumpul Sidoarjo

2010 : SMP Muhammadiyah 02 Tayu, Pati

2013 : SMA Laboratorium UM, Malang

2014 : Program Studi D1 Teknologi Informasi Jurusan Teknik Elektronika Politeknik Negeri Malang

Scripts

using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using System.Linq;  
using UnityEngine.UI;  
using System;  
  
public class Pengacakan : MonoBehaviour {  
  
    TimeManager TM;  
    GameObject iconBenar;  
    GameObject Panel2;  
    public int pointsToAdd;  
    private List<int> arrayAngka;  
    private List<int> arrayAngkaShuffle;  
    //Image BtnBenar;  
    //public Text textTimer;  
    //Getinputfuzzy gif;  
  
    void Start()  
    {  
        TM = GameObject.Find ("txtTimer").GetComponent<TimeManager>();  
        //BtnBenar = GetComponent<Image> ();  
        iconBenar = GameObject.FindGameObjectWithTag ("TagBenar");  
        //Panel2 = GameObject.Find ("Result");  
        arrayCompleteQuiz = new List<int>();  
        //PushImageToList();  
        //gif = GameObject.Find("Getinputfuzzy").GetComponent<Getinputfuzzy>();  
  
        //{//Fungsi Random Quiz 2 -8  
        if (Application.loadedLevelName == "Level 1")  
            arrayAngka = new List<int>(){  
            0,1,2,3,4,5,6  
        };  
        arrayAngkaShuffle = ShuffleFYS(arrayAngka);  
        RandomPanelQuiz();  
        for (int i = 0; i < arrayCompleteQuiz.Count; i++)  
        {  
            print(arrayCompleteQuiz[i]);  
            Debug.Log ("pertanyaan"+ i);  
        }  
  
    }  
  
    void Update () {  
  
    }  
  
    public int nilai = 0;  
  
    #region QUIZ   
    [Header("Pertanyaan")]  
    public GameObject panelResult;  
    public GameObject panelpertanyaan;  
  
  
    int indexPanelSebelumnya;  
    int quizKe = 0;  
    List<int> arrayCompleteQuiz;  
    private void RandomPanelQuiz()  
    {  
        for (int i = 0; i < quizContainer.Length; i++)  
        {  
            quizContainer[i].SetActive(false);  
            //Debug.Log ("pertanyaan ke= "+ i);  
  
        }  
        quizContainer[arrayAngkaShuffle[quizKe]].SetActive(true);  
        arrayCompleteQuiz.Add(arrayAngkaShuffle[quizKe]);  
  
    }  
  
    public void scoreQuiz2\_8()  
    {  
        ScoreManager.AddPoints (pointsToAdd);  
        //nilai = nilai + 10;  
        //Debug.Log("hasilnya" + nilai);  
        if (arrayCompleteQuiz.Count < 7)  
        {  
            quizKe++;  
            RandomPanelQuiz();  
            Debug.Log ("pertanyaan ke= "+ quizKe);  
        }  
        else if (arrayCompleteQuiz.Count == 7)  
        {  
            panelResult.SetActive(true);  
            //Panel2.gameObject.SetActive (true);  
            //panelpertanyaan.SetActive (false);  
            //textTimer.gameObject.SetActive(false);  
        }  
    }  
  
    public void jawab\_benarQuiz()  
    {  
        StartCoroutine(    PendingJawaban ());  
        ScoreManager.AddPoints (pointsToAdd);  
        //nilai = nilai + 10;  
        //Debug.Log("hasilnya " + nilai);  
        if (arrayCompleteQuiz.Count < 7)  
        {  
            quizKe++;  
            //Debug.Log ("ambil"+ gif.getStar());  
            Debug.Log ("pertanyaan ke= "+ quizKe);  
            //RandomPanelQuiz();  
        }  
        else if (arrayCompleteQuiz.Count == 7)  
        {  
            TM.isFinish = true;  
            panelResult.SetActive(true);  
            //Debug.Log ("ambil"+ gif.getStar());  
            //Panel2.gameObject.SetActive (true);  
            //panelpertanyaan.SetActive (false);  
  
            //textTimer.gameObject.SetActive(false);  
        }  
  
    }  
  
    IEnumerator PendingJawaban(){          
        yield return new WaitForSeconds (1f);  
        RandomPanelQuiz();  
        //Debug.Log ("pengacakan quiz"+ arrayAngkaShuffle);  
    }   
  
  
  
    public void jawab\_salahQuiz()  
    {  
        StartCoroutine (PendingJawaban());  
        //ScoreManager.AddPoints (pointsToAdd);  
        //nilai = nilai + 0;  
        //Debug.Log("hasilnya " + nilai);  
        if (arrayCompleteQuiz.Count < 7)  
        {  
            quizKe++;  
            //Debug.Log ("ambil"+ gif.getStar());  
            Debug.Log ("pertanyaan ke= "+ quizKe);  
            //RandomPanelQuiz();  
        }  
        else if (arrayCompleteQuiz.Count == 7)  
        {  
            TM.isFinish = true;  
            panelResult.SetActive(true);  
            //Debug.Log ("ambil"+ gif.getStar());  
            //Panel2.gameObject.SetActive (true);  
            //panelpertanyaan.SetActive (false);  
  
            //textTimer.gameObject.SetActive(false);  
        }  
  
    }  
    #endregion QUIZ   
  
  
    List<int> ShuffleFYS(List<int> a)  
    {  
        for (int i = a.Count - 1; i > 0; i--)  
        {  
            int rnd = UnityEngine.Random.Range(0, i);  
            int temp = a[i];  
            a[i] = a[rnd];  
            a[rnd] = temp;  
        }  
        return a;  
        //Debug.Log ("pengacakan quiz"+ a);  
    }  
  
  
  
    //public int UniqueRandomInt(List<int> arrayParam, int min, int max)  
    //{  
    //    int val = UnityEngine.Random.Range(min, max);  
    //    if (arrayParam != null)  
    //    {  
    //        while (arrayParam.Contains(val))  
    //        {  
    //            val = UnityEngine.Random.Range(min, max);  
    //        }  
    //    }  
    //    return val;  
    //}  
  
    public GameObject[] quizContainer;  
  
}