

**PENGEMBANGAN *GAME PUZZLE* SEBAGAI
PEMBELAJARAN HURUF JEPANG *HIRAGANA* DAN
KATAKANA MENGGUNAKAN METODE *FISHER-YATES*
SHUFFLE DAN *FUZZY TSUKAMOTO* BERBASIS ANDROID**

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV
Politeknik Negeri Malang

Oleh:

DWI SRI WAHYU

NIM. 1641727005



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

AGUSTUS 2017

**PENGEMBANGAN *GAME PUZZLE* SEBAGAI
PEMBELAJARAN HURUF JEPANG *HIRAGANA* DAN
KATAKANA MENGGUNAKAN METODE *FISHER-YATES*
SHUFFLE DAN *FUZZY TSUKAMOTO* BERBASIS ANDROID**

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV
Politeknik Negeri Malang

Oleh:

DWI SRI WAHYU

NIM. 1641727005



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

AGUSTUS 2017

HALAMAN PENGESAHAN

PENGEMBANGAN *GAME PUZZLE* SEBAGAI PEMBELAJARAN HURUF JEPANG *HIRAGANA* DAN *KATAKANA* MENGGUNAKAN METODE *FISHER-YATES* *SHUFFLE* DAN *FUZZY TSUKAMOTO* BERBASIS ANDROID

Disusun oleh:

Dwi Sri Wahyu NIM. 1641727005

Skripsi ini telah diuji pada tanggal 30 Agustus 2017

Disetujui oleh:

- | | | | |
|------------------|---|--|-------|
| 1. Penguji I | : | <u>Erfan Rohadi, ST., M.Eng., Ph.D</u> | |
| | | NIP. 19720123 200801 1 006 | |
| 2. Penguji II | : | <u>Mungki Astiningrum, ST.,M.Kom</u> | |
| | | NIP. 19771030 200501 2 001 | |
| 3. Pembimbing I | : | <u>Putra Prima Arhandi, S.T.,M.Kom</u> | |
| | | NIP. 19861103 201404 1 001 | |
| 4. Pembimbing II | : | <u>Kadek Sarjuna B., S.Kom., M.T</u> | |

Mengetahui,

Ketua Jurusan
Teknologi Informasi

Ketua Program Studi
Teknik Informatika

Rudy Ariyanto, S.T., M.Cs.
NIP. 19711110 199903 1 002

Ir. Deddy Kusbianto P., M.MKom.
NIP. 19621128 198811 1 001

PERNYATAAN

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

Malang, Agustus 2017

Dwi Sri Wahyu

ABSTRAK

Wahyu, Dwi Sri. “Pengembangan *Game Puzzle* Sebagai Pembelajaran Huruf Jepang *Hiragana* Dan *Katakana* Menggunakan Metode *Fisher-Yates Shuffle* Dan *Fuzzy Tsukamoto* Berbasis Android “. **Pembimbing: (1) Putra Prima Arhandi, S.T., M.Kom, (2) Kadek Sarjuna Batubulan, S.Kom., M.T.**

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

Perkembangan teknologi membuat banyak *game* yang dapat dimainkan oleh masyarakat sebagai hiburan. Banyak *game* yang dibuat hanya mengedepankan sisi hiburan semata, maka diperlukan sebuah *game* yang dapat dijadikan media pembelajaran agar masyarakat memperoleh rangsangan hiburan yang edukatif, salah satu *game* edukatif yaitu *game puzzle*. *Game puzzle* adalah permainan yang menarik dan menyenangkan serta memiliki fungsi khusus untuk memicu imajinasi, menyederhanakan informasi dan memperjelas sajian agar mudah dipahami dan diingat. Aplikasi yang dibangun berbentuk *game puzzle* sebagai media pembelajaran mengenai huruf Jepang *Hiragana* dan *Katakana*. Selain itu terdapat menu pembelajaran untuk menulis huruf Jepang *Hiragana* dan *Katakana*. Pada aplikasi ini diimplementasikan algoritma *Fisher-Yates Shuffle* pada proses pengacakan potongan gambar huruf Jepang dan *Fuzzy Tsukamoto* untuk menghasilkan nilai skor untuk menentukan tingkat kemampuan pengguna. Dari hasil pengujian aplikasi telah berjalan dengan baik dan dari hasil kuesioner yang telah disebarkan, diperoleh bahwa aplikasi sudah baik untuk diterapkan dengan prosentase 64,4 %. Dengan demikian, hasil dari aplikasi ini dapat dijadikan media pembelajaran bagi pengguna yang akan mempelajari huruf Jepang *Hiragana* dan *Katakana* sehingga diharapkan dapat menambah pengetahuan untuk proses pembelajaran huruf Jepang.

Kata kunci : huruf Jepang *Hiragana* dan *Katakana*, *Fisher-Yates Shuffle*, *Fuzzy Tsukamoto*

ABSTRACT

Wahyu, Dwi Sri. *“Developing Puzzle Game As A Learning Japanese Hiragana And Katakana Using Fisher-Yates Shuffle Method And Fuzzy Tsukamoto Based Android”*. Advisors: (1) **Putra Prima Arhandi, S.T., M.Kom**, (2) **Kadek Sarjuna Batubulan, S.Kom., M.T.**

Thesis, Informatics Engineering Programe, Department of Information Technology, State Polytechnic of Malang. 2017.

Development in technology makes games can be played by the people for entertainment. But most of games are made only to promote the entertain side, so that game can be used as a media of learning to stimulate that people to get educational stimulus, and one of them is a puzzle game. A puzzle game is an exciting and fun game and has a special function to trigger imagination, simplify information and clarify the content to make it easy to understand and remember. Applications built in the form of puzzle games as a media of learning on Japanese letters Hiragana and Katakana. Beside that there is a menu to write Japanese letters Hiragana and Katakana on the application. This application implemented Fisher-Yates Shuffle algorithm on the process of scrambling pieces of Japanese fonts and Fuzzy Tsukamoto to produce a score score to determine the level of user ability. From the results of testing the application has been running well and from the results of questionnaires that have been disseminated, obtained that the application is good to be applied with the percentage of 64.4%. The results of this application can be used as a media of learning for users who are interest to study Japanese letters Hiragana and Katakana so it is expected to increase knowledge for the learning process of Japanese letters.

Keywords : Japanese letters Hiragana and Katakana, Fisher-Yates Shuffle, Fuzzy Tsukamoto

KATA PENGANTAR

Puji Syukur kami panjatkan kehadirat Allah SWT atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan laporan akhir dengan judul “Pengembangan *Game Puzzle* Sebagai Pembelajaran Huruf Jepang *Hiragana* Dan *Katakana* Menggunakan Metode Fisher-Yates Shuffle Dan Fuzzy Tsukamoto Berbasis Android”. Laporan akhir ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Bapak Rudy Ariyanto, ST., M.CS., selaku ketua jurusan Teknologi Informasi.
2. Bapak Ir. Deddy Kusbiantoro P.A., M.MKOM selaku ketua program studi Teknik Informatika.
3. Bapak Putra Prima Arhandi, S.T., M.Kom., selaku pembimbing I yang telah memberikan bimbingan dan dukungan hingga penyusunan laporan ini selesai.
4. Bapak Kadek Sarjuna Batubulan, S.Kom., M.T., selaku pembimbing II yang telah memberikan bimbingan dan dukungan hingga penyusunan laporan ini selesai.
5. Dosen-dosen pengajar Program Studi Teknik Informatika yang telah memberikan bimbingan dan ilmunya.
6. Kedua orang tua kami dan segenap keluarga yang selalu memberikan doa dan memberikan dukungan moral maupun material.
7. Teman-teman angkatan dan alih jenjang Program Studi Teknik Informatika khususnya TI-4F yang telah membantu menyelesaikan skripsi ini dan selalu memberikan semangat.
8. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan skripsi dari awal hingga akhir yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari dalam penyusunan laporan ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, Agustus 2017

Penulis

DAFTAR ISI

	Halaman
COVER	i
HALAMAN JUDUL	ii
HALAMAN PENGESAHAN	iii
PERNYATAAN	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Sistematika Penulisan	2
BAB II. LANDASAN TEORI	4
2.1 Android	4
2.2 Unity Game Engine	4
2.3 Puzzle	5
2.4 Adobe Illustrator	5
2.5 Adobe Photoshop	5
2.6 Adobe Flash	6
2.7 Huruf <i>Hiragana</i>	6
2.8 Huruf <i>Katakana</i>	8
2.9 Algoritma <i>Fisher-Yates Shuffle</i>	11
2.10 Fungsi Keanggotaan	12
2.12 Metode Fuzzy Tsukamoto	15
BAB III. METODOLOGI PENELITIAN	17
3.1 Studi Literatur	17
3.2 Pengumpulan Data	17
3.3 Perancangan Sistem	17
3.3.1 Konsep (<i>Concept</i>)	17

3.3.2	Desain (<i>Design</i>).....	18
3.3.3	Pengumpulan Bahan (<i>Material Collecting</i>)	18
3.3.4	Pembuatan (<i>Assembly</i>)	18
3.3.5	Pengujian (<i>Testing</i>)	18
3.3.6	Distribusi (<i>Distribution</i>).....	18
BAB IV. ANALIS DAN PERANCANGAN.....		19
4.1	Konsep (<i>Concept</i>).....	19
4.1.1	Gambaran Umum Sistem	19
4.1.2	Analisa Kebutuhan Non Fungsional	19
4.1.3	Analisis Kebutuhan Fungsional	20
4.1.3.1	Usecase Diagram	20
4.1.3.2	Work Breakdown Structure (WBS)	24
4.1.3.3	<i>Flowchart</i>	25
4.2.1	Perancangan Storyboard.....	27
4.2.2	Perancangan Materi.....	30
4.2.3	Perancangan <i>Puzzle</i>	30
4.2.4	Perancangan Metode <i>Fisher Yates Shuffle</i>	33
4.2.5	Penerapan <i>Fuzzy Tsukamoto</i>	37
4.3.4.1	Pembentukan Himpunan Fuzzy Tsukamoto	37
4.3.4.3	<i>Fuzzifikasi</i>	38
BAB V. IMPLEMENTASI.....		52
5.1	Pengumpulan Bahan (<i>Material Collecting</i>) Objek dan Karakter.....	52
5.2	Pembuatan (<i>Assembly</i>) Implementasi Antar Muka	61
5.2.1	Halaman Menu Utama	61
5.2.2	Tampilan Menu Pembelajaran	62
5.2.3	Tampilan Menu Materi Pembelajaran <i>Hiragana</i>	62
5.2.4	Tampilan Menu Materi Pembelajaran <i>Katakana</i>	63
5.2.5	Tampilan Menu Putar Video Pembelajaran	63
5.2.7	Tampilan Menu Permainan <i>Puzzle</i>	64
5.2.8	Tampilan Menu <i>Puzzle</i> yang telah Diselesaikan.....	65
5.2.9	Tampilan Menu Lanjut ke Menu Latihan	65
5.2.10	Tampilan Menu Latihan	66
5.2.11	Tampilan Latihan Menulis	66
5.2.13	Tampilan Menu Skor.....	67

5.3 Implementasi Audio	68
5.4 Implementasi Proses.....	68
5.4.1 Proses Pengacakan Pada Permainan <i>Puzzle</i>	68
5.4.2 Proses Nilai <i>Score</i>	69
BAB VI. PENGUJIAN DAN PEMBAHASAN	73
6.1 Pengujian Aplikasi (<i>Testing</i>).....	73
6.1.1 Hasil dan Kasus Pengujian Alpha.....	73
6.1.3 Pengujian Beta	74
BAB VII. KESIMPULAN DAN SARAN	79
7.1 Kesimpulan	79
7.2 Saran	79
Daftar Pustaka	80

DAFTAR GAMBAR

	Halaman
Gambar 3. 1 Metode Pengembangan Multimedia Development Life Cycle	17
Gambar 4. 1 Use Case Diagram	21
Gambar 4. 2 <i>Work Breakdown Structure</i>	24
Gambar 4. 3 <i>Flowchart Game Puzzle</i> dan Waktu	25
Gambar 4. 4 <i>Flowchart</i> Praktik Menulis	26
Gambar 4. 5 Curva Input Variabel Point Puzzle	38
Gambar 4. 6 Curva Input Variabel Point Menulis	40
Gambar 4. 7 Curva Input Variabel Waktu	41
Gambar 4. 8 Curva Input Variabel Skor	42
Gambar 5.1 Desain tombol dan <i>icon</i>	52
Gambar 5.2 Desain <i>background</i> aplikasi	53
Gambar 5.3 Desain pembuatan huruf Jepang <i>Hiragana</i>	53
Gambar 5.4 Desain pembuatan huruf Jepang <i>Katakana</i>	54
Gambar 5.5 Proses pemotongan gambar	54
Gambar 5.6 <i>Divide Slide</i>	55
Gambar 5.7 Proses setelah dilakukan <i>Divide Slice</i>	55
Gambar 5.8 Proses simpan potongan gambar	56
Gambar 5.9 Menyimpan potongan gambar	56
Gambar 5.10 Memasukkan objek ke dalam canvas	57
Gambar 5.11 Proses tambah layer pada <i>masking</i>	57
Gambar 5.12 Proses menghilangkan <i>Outlines</i>	58
Gambar 5.13 Hasil dari menghilangkan <i>outlines</i> pada objek huruf Jepang	58
Gambar 5.14 Proses memilih Mask	59
Gambar 5.15 Proses Mask	59
Gambar 5.16 Proses animasi <i>masking</i>	60
Gambar 5.17 Hasil proses animasi <i>Masking</i>	60
Gambar 5.18 Tampilan Menu Utama	61
Gambar 5.19 Tampilan Menu Pembelajaran	62
Gambar 5.20 Tampilan Menu Materi Pembelajaran <i>Hiragana</i>	62
Gambar 5.21 Tampilan Menu Pembelajaran <i>Katakana</i>	63
Gambar 5.22 Tampilan Menu Putar Video Pembelajaran	63
Gambar 5.23 Ilustrasi cara menulis huruf Jepang	64
Gambar 5.24 Tampilan permainan <i>puzzle</i>	64
Gambar 5.25 Tampilan <i>Puzzle</i> yang telah diselesaikan	65
Gambar 5.26 Tampilan Blok <i>Puzzle</i> Salah	65
Gambar 5.27 Tampilan Menu Latihan	66
Gambar 5.28 Tampilan Latihan Menulis	66
Gambar 5. 29 Tampilan Pengecekan Latihan Menulis	67
Gambar 5. 30 Tampilan Menu Skor	67
Gambar 5.31 Implementasi Audio	68
Gambar 6.1 Grafik Hasil Kuesioner	77
Gambar 6.2 Hasil <i>console</i> pengacakan potongan gambar huruf Jepang	68

DAFTAR TABEL

	Halaman
Tabel 2. 1 Tabel Huruf <i>Hiragana</i> Sei on	6
Tabel 2. 2 Tabel Huruf <i>Hiragana</i> Daku on	7
Tabel 2. 3 Tabel Huruf <i>Hiragana</i> Yo on	8
Tabel 2. 4 Tabel Huruf <i>Katakana</i>	9
Tabel 2. 5 Huruf <i>Katakana</i> Dakuon	9
Tabel 2. 6 Tabel Huruf <i>Hiragana</i> Yo on	10
Tabel 2. 7 Tabel hasil pengacakan Fisher-Yates Shuffle	12
Tabel 4. 1 Tabel penjelasan actor	20
Tabel 4. 2 Deskripsi usecase diagram.....	21
Tabel 4. 3 Spesifikasi skenario : Menu Mulai.....	22
Tabel 4. 4 Spesifikasi Skenario: Informasi Aplikasi	23
Tabel 4. 5 Tabel spesifikasi skenario dari Petunjuk Aplikasi.....	23
Tabel 4. 6 Storyboard game Puzzle Huruf <i>Hiragana</i> dan <i>Katakana</i>	27
Tabel 4. 7 Tabel Figure <i>Puzzle</i> Huruf <i>Hiragana</i> dan <i>Katakana</i>	30
Tabel 4.8 Perancangan <i>Practice</i>	32
Tabel 4.9 Penerapan metode <i>Fisher-Yates Shuffle</i>	33
Tabel 4. 10 Fuzzy Rules	43
Tabel 6. 1 Kasus dan pengujian alpha	71
Tabel 6. 2 Kuesioner Aplikasi	75
Tabel 6. 3 Hasil Kuesioner	76

DAFTAR LAMPIRAN

Lampiran 1. Biodata Penulis

Lampiran 2. Kuesioner

Lampiran 3. Lembar Bimbingan

Lampiran 4. Lembar Revisi

BAB I. PENDAHULUAN

1.1 Latar Belakang

Pada saat ini bahasa Jepang di Indonesia dipelajari di lembaga pendidikan formal ditingkat Sekolah Menengah Atas yang mulai memberikan pelajaran bahasa Jepang kepada peserta didiknya. Selain itu, mulai bermunculan lembaga pendidikan nonformal yang memberikan atau kursus bahasa Jepang.

Umumnya proses pembelajaran bahasa Jepang di lembaga pendidikan formal maupun nonformal dilakukan dengan menggunakan media pembelajaran buku, dan penggunaan multimedia dalam pembelajaran masih belum maksimal. Metode pembelajaran tersebut cenderung membuat proses pembelajaran menjadi membosankan dan menyebabkan pembelajar cepat jenuh. Sehingga diperlukan suatu inovasi untuk mengembangkan media pembelajaran baru berbasis multimedia dengan memanfaatkan perkembangan ilmu teknologi.

Perkembangan ilmu teknologi dimasa ini dapat dimanfaatkan untuk mengembangkan aplikasi pada perangkat *mobile* dalam membantu proses pembelajaran. Penggunaan perangkat *mobile* saat ini dapat mendukung penerapan *game* yang bernilai edukatif yang mudah diterima serta dapat digunakan dimanapun dan kapanpun. *Game* merupakan salah satu media yang digunakan untuk menyampaikan suatu pesan kepada orang lain dalam bentuk permainan yang menghibur. Dengan semakin majunya ilmu teknologi pada perangkat *mobile* membuat banyak *game* dapat dikembangkan, misalnya *game puzzle*.

Game puzzle adalah permainan yang menarik dan menyenangkan serta memiliki fungsi khusus untuk memicu imajinasi, menyederhanakan informasi dan memperjelas sajian agar mudah dipahami dan diingat. Selain itu *puzzle* memiliki kelebihan mengasah otak, melatih koordinasi mata, tangan, melatih nalar dan melatih kesabaran [1]. Sehingga penulis akan membangun sebuah *game puzzle* yang bernilai edukatif sebagai media pembelajaran mengenai huruf Jepang *Hiragana* dan *Katakana*.

Salah satu pengembangan metode yang dapat mendukung pembuatan *game puzzle* adalah algoritma *Fisher-Yates Shuffle*. *Fisher-Yates Shuffle* adalah sebuah algoritma untuk menghasilkan suatu permutasi acak dari suatu himpunan

terhingga, sehingga potongan gambar huruf yang telah muncul tidak akan muncul lagi di sesi yang sama [2]. Pada tahap akhir terdapat skor yang digunakan untuk menentukan skor akhir permainan. Pada tahap skor ini metode yang digunakan adalah metode *Fuzzy Tsukamoto* yang direpresentasikan dengan gambar bintang. Metode ini dipilih karena setiap konsekuensi pada aturan yang berbentuk IF-THEN direpresentasikan dengan himpunan fuzzy dengan fungsi keanggotaan. Sebagai hasilnya, output dari setiap aturan diberikan secara tegas berdasarkan α , kemudian diperoleh hasil akhir dengan menggunakan rata-rata terpusat [3].

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka perumusan masalah yang didapat yaitu:

Bagaimana menerapkan algoritma *Fisher-Yates Shuffle* untuk mengacak *puzzle* dan Fuzzy Tsukamoto untuk *scoring* pada *Game Puzzle* sebagai pembelajaran huruf Jepang *Hiragana* dan *Katakana*?

1.3 Tujuan

Tujuan dari pembuatan *game* ini yaitu :

Menerapkan algoritma *Fisher-Yates Shuffle* pada *game puzzle* dan metode Fuzzy Tsukamoto untuk *scoring*.

1.4 Batasan Masalah

Batasan masalah dari pembuatan *game* ini yaitu :

- Aplikasi ini berisi *puzzle* huruf Jepang *Hiragana* *Sei on* dan *Katakana* *Sei on* serta cara menulis untuk huruf Jepang *Hiragana* *Sei on* dan *Katakana* *Sei on*.
- Aplikasi ini bersifat *Single Player*.
- Aplikasi ini ditujukan pada pengguna usia 16 tahun.

1.5 Sistematika Penulisan

Uraian dalam laporan skripsi penulis menyusun dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisikan gambaran umum dari penulisan skripsi. Di dalamnya dijelaskan mengenai latar belakang permasalahan, rumusan masalah,

batasan masalah yang diambil, tujuan yang ingin dicapai, 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

Terdiri dari langkah-langkah yang akan membimbing penulis 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

Pada bab ini diuraikan dengan jelas sistem yang akan dibuat dan kebutuhan sistem yang meliputi kebutuhan fungsional dan kebutuhan non-fungsional. Rancangan sistem meliputi rancangan model, rancangan arsitektur sistem, rancangan proses, rancangan prosedural, rancangan data dan rancangan antarmuka pengguna (*user interface*).

BAB V IMPLEMENTASI

Pada bab ini, implementasi sistem dipaparkan secara detil sesuai rancangan dan komponen (*tools*) bahasa pemrograman yang dipakai. Implementasi rancangan proses dapat disertai dengan potongan kode pada proses yang dimaksud.

BAB VI PENGUJIAN DAN PEMBAHASAN

Pada bab ini Pengujian merupakan proses untuk menentukan apakah hasil dari tugas akhir sudah sesuai dengan kebutuhan sistem dan berjalan sesuai lingkungan yang diinginkan. Sedangkan Pembahasan merupakan argumentasi rasional dari penulis yang disusun secara sistematis berdasarkan fakta ilmiah yang diperoleh dari hasil pengujian.

BAB VII KESIMPULAN

Bab ini berisi uraian singkat dan jelas tentang hasil tugas akhir yang diperoleh sesuai dengan tujuan penelitian.

BAB II. LANDASAN TEORI

Pada bab ini akan membahas mengenai teori – teori yang digunakan dalam pembuatan aplikasi *Game Puzzle* Sebagai Pembelajaran Huruf Jepang *Hiragana* dan *Katakana* Berbasis Android.

2.1 Android

Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembang untuk melakukan pengembangan sesuai yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan di bawah GNU, *General Public License versi 2* (GPLv2), yang lebih dikenal dengan istilah copyleft, lisensi di mana setiap perbaikan pihak ketiga harus terus di bawah syarat (terms). Android didistribusikan di bawah *Apache Software Lisence* (ASL/Apache 2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen handset khususnya) dapat memilih untuk meningkatkan *platform* tanpa harus memberikan perbaikan mereka ke masyarakat *open source*. Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka di bawah lisensi apapun yang mereka inginkan. Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan. [4]

Android merupakan *platform* yang digunakan untuk mengimplementasikan aplikasi ini.

2.2 Unity Game Engine

Unity merupakan ekosistem pengembangan *game* mesin render yang kuat terintegrasi dengan satu set lengkap alat intuitif dan alur kerja yang cepat untuk membuat konten 3D interaktif, penerbitan multiplatform yang mudah, ribuan kualitas, asset siap pakai di Asset Store dan berbagai pengetahuan di komunitas. Unity 3D ini digunakan sebagai media pembuatan Aplikasi *Game Puzzle* Huruf *Hiragana* dan *Katakana* dari pembuatan setiap objeknya sampai penyelesaian pembuatan aplikasi. Kami menggunakan *software* Unity 3D ini karena unity memiliki banyak *tools* yang dapat digunakan untuk membuat latar berupa 3 dimensi dan pengoperasiannya. [5]

Unity merupakan *software maker* atau aplikasi yang digunakan untuk membangun aplikasi ini.

2.3 Puzzle

Menurut Kamus Besar Bahasa Indonesia *puzzle* adalah teka-teki. *Puzzle* merupakan permainan yang membutuhkan kesabaran dan ketekunan dalam proses merangkai. [6]

Game puzzle merupakan salah satu pilihan dari beberapa media pembelajaran yang menyenangkan dan mudah untuk dilakukan yakni sesuatu yang diwujudkan secara visual dalam bentuk dua dimensi sebagai curahan perasaan dan pikiran. *Game puzzle* merupakan bentuk permainan yang menantang daya kreatifitas dan ingatan siswa lebih mendalam dikarenakan munculnya motivasi untuk senantiasa mencoba memecahkan masalah, namun tetap menyenangkan sebab bisa di ulang-ulang. Tantangan dalam permainan ini akan selalu memberikan efek ketagihan untuk selalu mencoba, mencoba dan terus mencoba hingga berhasil. [7]

Definisi *puzzle* digunakan sebagai landasan untuk membangun *game puzzle* di aplikasi ini.

2.4 Adobe Illustrator

Adobe Illustrator merupakan salah satu software untuk membuat desain grafis. Software ini sangat populer dan sudah diakui kecanggihannya. Kelengkapan fasilitas dan kemampuannya yang luar biasa dalam mendesain grafis, menjadikan software ini mulai banyak dipakai oleh para desainer komputer, karena keberadaannya benar-benar mampu membantu dan memudahkan pemakai dalam menyelesaikan pekerjaan desain grafis. [10]

Adobe Illustrator merupakan *software* yang digunakan dalam mendesain objek yang digunakan pada aplikasi ini.

2.5 Adobe Photoshop

Adobe Photoshop adalah program pengolah gambar yang dikenal *smart-program* yang memungkinkan penggunanya untuk melakukan modifikasi dengan pengeditan foto digital. Adobe Photoshop hadir dengan beragam menu dan *tool* untuk mempermudah proses modifikasi dan pengeditan. [11]

Adobe Photoshop digunakan untuk memotong gambar huruf *Hiragana* dan *Katakana* menjadi potongan *puzzle* yang akan digunakan untuk objek *game puzzle*.

2.6 Adobe Flash

Adobe Flash merupakan perangkat lunak komputer yang digunakan untuk membuat animasi, video, gambar vector maupun bitmap dan multimedia interaktif. Flash didesain dengan kemampuan untuk membuat animasi 2 dimensi yang handal dan ringan sehingga flash banyak digunakan untuk membangun dan memberikan efek animasi pada website, CD interaktif dan yang lainnya.[12]

Adobe Flash digunakan untuk membuat animasi berupa video mengenai tata cara penulisan huruf *Hiragana* dan *Katakana* yang benar.

2.7 Huruf Hiragana

Huruf *hiragana* melambangkan suku kata tunggal, dan digunakan untuk menulis kata-kata yang berasal dari Jepang asli.

Huruf *Hiragana* terbentuk dari modifikasi dan penyederhanaan kanji, sehingga huruf *hiragana* coretannya melengkung dan tidak bersudut tajam, sehubungan dengan hal tersebut menyebutkan bahwa huruf *hiragana* digunakan mulai jaman Edo, yang pada awal mulanya digunakan oleh kaum wanita sehingga *hiragana* dikenal dengan huruf wanita, dalam bahasa Jepang disebut *onnade*.

Huruf *Hiragana* berjumlah empat puluh enam huruf di dalamnya mencakup lima huruf vokal yaitu a i u e o sisanya adalah suku kata dengan deretan ka, sa, ta, na, ha, ma, ya ra, wa dan satu konsonan yaitu n serta satu kata bantu o yang kadangkala diucapkan wo. [8]

Empat puluh enam huruf *hiragana* tersebut di atas disebut *Sei on*, selain itu dalam penggunaannya terdapat *Daku on*, *Handaku on* dan *Yoo on*.

Huruf *hiragana* yang termasuk *Sei on* adalah sebagai berikut :

Tabel 2. 1 Tabel Huruf *Hiragana* Sei on

あ a	い i	う u	え e	お o
か ka	き ki	く ku	け ke	こ ko
さ sa	し si	す su	せ se	そ so

た ta	ち chi	つ tsu	て te	と to
な na	に ni	ぬ nu	ね ne	の no
は ha	ひ hi	ふ fu	へ he	ほ ho
ま ma	み mi	む mu	め me	も mo
や ya	–	ゆ yu	–	よ yo
ら ra	り ri	る ru	れ re	ろ ro
わ wa	ん n	を wo	–	–

Contoh :

さくら / *Sakura* / „bunga sakura “
 わたし / *Watashi* / „saya “
 ほんや / *Honya* / „toko buku “

Daku on berjumlah dua puluh lima huruf *hiragana*, yang termasuk di dalamnya adalah sebagai berikut :

Tabel 2. 2 Tabel Huruf *Hiragana Daku on*

が ga	ぎ gi	ぐ gu	げ ge	ご go
ざ za	じ ji	ず zu	ぜ ze	ぞ zo
だ da	ぢ ji	づ zu	で de	ど do
ば ba	び bi	ぶ bu	べ be	ぼ bo
ぱ pa	ぴ pi	ぷ pu	ぺ pe	ぽ po

Contoh :

おどり / *odori* / „tarian “
 かぞく / *kazoku* / „keluarga “
 たばこ / *tabako* / „rokok “

Yoo on berjumlah tiga puluh tiga huruf *hiragana*, yang termasuk di dalamnya adalah sebagai berikut.

Tabel 2. 3 Tabel Huruf *Hiragana* Yo on

きゃ kya	きゅ kyu	きょ kyo
しゃ sha	しゅ shu	しょ sho
ちゃ cha	ちゅ chu	ちょ cho
にゃ nya	にゅ nyu	にょ nyo
ひゃ hya	ひゅ hyu	ひょ hyo
みゃ mya	みゅ myu	みょ myo
りゃ rya	りゅ ryu	りょ ryo
ぎゃ gya	ぎゅ gyu	ぎょ gyo
じゃ ja	じゅ ju	じょ jo
びゃ bya	びゅ byu	びょ byo
ぴゃ pya	ぴゅ pyu	ぴょ pyo

Definisi Huruf *Hiragana* digunakan sebagai landasan teori dalam pembangunan objek aplikasi ini.

2.8 Huruf *Katakana*

Huruf *Katakana* sama seperti huruf *Hiragana* yaitu melambangkan suku kata tunggal, tetapi mempunyai fungsi yang berbeda dengan huruf *hiragana*. Huruf *Katakana* selain digunakan untuk menulis kata-kata yang berasal dari bahasa asing, juga digunakan untuk penekanan suatu kata yang berasal dari Jepang asli.

Huruf *Katakana* juga terbentuk dari modifikasi kanji dengan cara mengambil salah satu bagian kanji, sehingga di dalam huruf *katakana* tidak akan menemukan coretannya yang melengkung seperti *hiragana*, *katakana* bentuk hurufnya terkesan kaku, karena setiap coretannya bersudut tajam sehingga *katakana* disebut sebagai huruf laki-laki.

Huruf *Katakana* berjumlah empat puluh enam huruf, karena dalam di dalam huruf *Katakana* tidak terdapat huruf o yang berfungsi sebagai kata bantu.

Susunan huruf *katakana* sama dengan huruf *Hiragana*, di dalamnya mencakup lima huruf vokal yaitu a i u e o dan sisanya adalah suku kata dengan deretan ka, sa, ta, na, ha, ma, ya ra, wa dan satu konsonan yaitu n. [8]

Berikut ini merupakan daftar huruf *Katakana*:

Tabel 2. 4 Tabel Huruf *Katakana* Sei on

ア a	イ i	ウ u	エ e	オ o
カ ka	キ ki	ク ku	ケ ke	コ ko
サ sa	シ shi	ス su	セ se	ソ so
タ ta	チ chi	ツ tsu	テ te	ト to
ナ na	ニ ni	ヌ nu	ネ ne	ノ no
ハ ha	ヒ hi	フ fu	ヘ he	ホ ho
マ ma	ミ mi	ム mu	メ me	モ mo
ヤ ya	-	ユ yu	-	ヨ yo
ラ ra	リ ri	ル ru	レ re	ロ ro
ワ wa	-	-	-	ヲ wo
ン n				

Daku on berjumlah dua puluh lima huruf *hiragana*, yang termasuk di dalamnya adalah sebagai berikut :

Tabel 2. 5 Huruf *Katakana* Daku on

ガ ga	ギ gi	グ gu	ゲ ge	ゴ go
ザ za	ジ ji	ズ zu	ゼ ze	ゾ zo
ダ da	ヂ ji	ヅ zu	デ de	ド do
バ ba	ビ bi	ブ bu	ベ be	ボ bo
パ pa	ピ pi	プ pu	ペ pe	ポ po

Yoo on berjumlah tiga puluh tiga huruf *hiragana*, yang termasuk di dalamnya adalah sebagai berikut :

Tabel 2. 6 Tabel Huruf *Hiragana* Yo on

キヤ kya	キュ kyu	キョ kyo
シャ sha	シュ shu	ショ sho
チャ cha	チュ chu	チョ cho
ニヤ nya	ニュ nyu	ニョ nyo
ヒヤ hya	ヒュ hyu	ヒョ hyo
ミヤ mya	ミユ myu	ミョ myo
リヤ rya	リュ ryu	リョ ryo
ギャ gya	ギュ gyu	ギョ gyo
ジャ ja	ジュ ju	ジョ jo
ビヤ bya	ビュ byu	ビョ byo
ピヤ pya	ピュ pyu	ピョ pyo

Huruf *katakana* berfungsi untuk menuliskan :

- a. Kata-kata yang berasal dari bahasa asing

Contoh : パイロット / pairotto /,, pilot “

- b. Nama orang dan tempat asing serta kata-kata benda asing

Contoh :

1. Kata-kata yang berasal dari bahasa asing

Contoh : パイロット / pairotto /,, pilot “

2. Nama orang dan tempat asing serta kata-kata benda asing

Contoh :

nama orang アミル (Amir) ジョン (john)

nama tempat インドネシア (Indonesia)

benda asing ボート (boat = perahu)

Definisi Huruf Katakan digunakan sebagai landasan teori dalam pembangunan objek aplikasi ini.

2.9 Algoritma *Fisher-Yates Shuffle*

Versi asli dari algoritma ini pertama kali diterbitkan pada tahun 1938. *Fisher-Yates Shuffle* (dinamai berdasarkan penemunya, Ronald Fisher dan Frank Yates) digunakan untuk mengubah urutan masukan yang diberikan secara acak. Permutasi yang dihasilkan oleh algoritma ini muncul dengan probabilitas yang sama.

Algoritma ini dinyatakan bias karena permutasi yang dihasilkan oleh algoritma ini muncul dengan probabilitas yang sama, hal ini dibuktikan dengan percobaan mengacak suatu set kartu yang dilakukan berulang.

Metode *Fisher-Yates* secara umum adalah :

1. Ketika masih ada elemen tersisa untuk diacak.
2. Ambil elemen secara acak dari elemen yang tersisa.
3. Kemudian tukar dengan elemen saat ini

Terdapat metode modern pada *fisher-yates shuffle*, metode ini dibuat untuk menyempurnakan metode sebelumnya, dengan menuliskan (N, M-K) :

1. N adalah elemen yang keluar secara random di setiap iterasi.
2. M adalah jumlah elemen yang di iterasi.
3. K adalah pengurang yang digunakan agar iterasi terus terjadi sampai pada elemen terakhir yang posisinya belum tergantikan.

Pada metode modern angka yang terpilih tidak dicoret, tetapi posisinya ditukar dengan angka terakhir dari angka yang belum terpilih. Berikut ini adalah contoh pengerjaan dari versi modern.

Range adalah jumlah angka yang belum terpilih, *roll* adalah angka acak yang terpilih, *scratch* adalah daftar angka yang belum terpilih, *result* adalah hasil permutasi yang akan didapatkan. [2]

Tabel 2. 7 Tabel hasil pengacakan Fisher-Yates Shuffle

Range	Roll	Scratch	Result
		12345678	
1-8	5	1234 8 67	5
1-7	3	12748 6	3 5
1-6	4	127 6 8	4 3 5
1-5	5	1276 8	8 4 3 5
1-4	2	1 6 7	2 8 4 3 5
1-3	3	1 6	7 2 8 4 3 5
1-2	1	6	1 7 2 8 4 3 5
Hasil Pengacakan			6 1 7 2 8 4 3 5

Definisi algoritma Fisher Yates-Shuffle digunakan sebagai landasan pengimplementasian metode aplikasi yang diletakkan pada saat *game puzzle* untuk mengacak potongan gambar huruf Jepang *Hiragana* dan *Katakana*.

2.10 Fungsi Keanggotaan

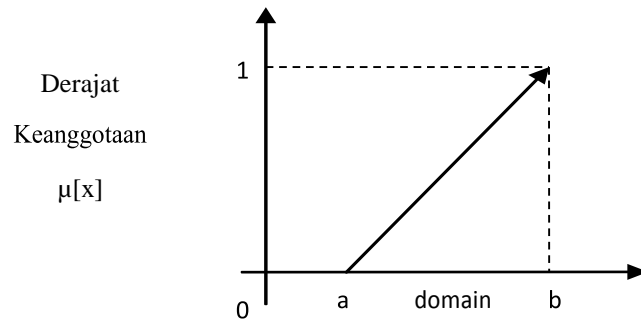
Fungsi keanggotaan adalah grafik yang mewakili besar dari derajat keanggotaan masing- masing variabel *input* yang berada dalam *interval* antara 0 sampai 1. Derajat keanggotaan sebuah variabel “x” dilambangkan dengan *symbol* $\mu[x]$. *Rule-rule* nilai menggunakan nilai keanggotaan sebagai faktor bobot untuk menentukan pengaruhnya pada saat melakukan *inferensi* untuk menarik kesimpulan. [13]

Ada beberapa fungsi yang bisa digunakan:

1. Representasi Linear

Representasi Linear adalah pemetaan input ke derajat keanggotannya digambarkan sebagai suatu garis lurus. Pada representasi linear terdapat 2 kemungkinan, yaitu:

- a. Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke arah kanan menuju nilai domain yang memiliki derajat keanggotaan lebih tinggi.

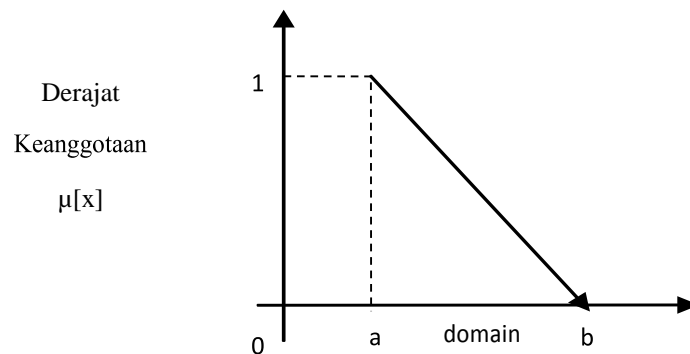


Gambar 2.2 Representasi Kurva *Linear Naik*

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} 0 & ; x \leq a \\ \left(\frac{x-a}{b-a}\right) & ; a \leq x \leq b \\ 1 & ; x \geq b \end{cases} \dots\dots\dots(2.1)$$

- b. Penurunan himpunan dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.



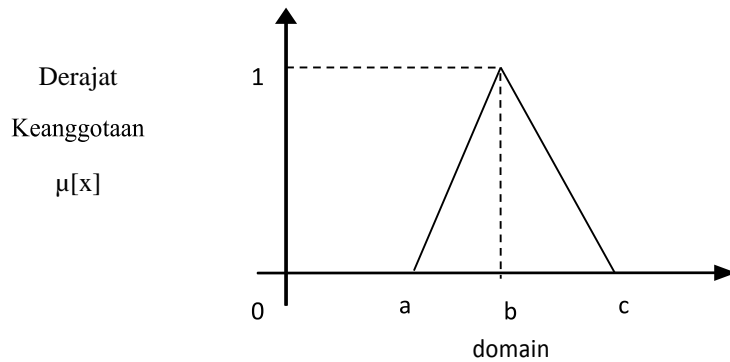
Gambar 2.3 Representasi Kurva *Linear Turun*

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} 1 & ; x \leq a \\ \left(\frac{b-x}{b-a}\right) & ; a \leq x \leq b \\ 0 & ; x \geq b \end{cases} \dots\dots\dots(2.2)$$

2. Representasi Kurva Segetiga

Kurva segitiga pada dasarnya terbentuk dari gabungan antara 2 garis (*linear*).



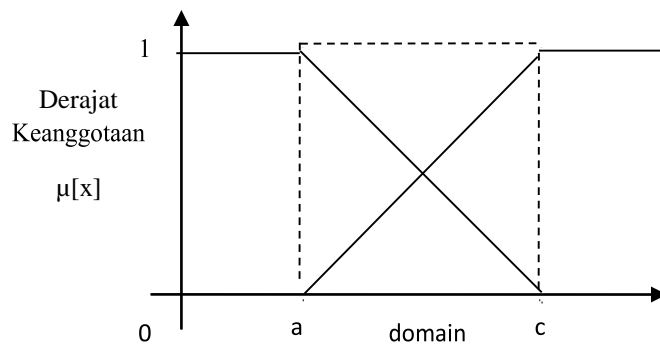
Gambar 2.4 Representasi Kurva Segitiga

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} 0 & ; x \leq a \text{ atau } x \geq c \\ \left(\frac{x-a}{b-a}\right); & a \leq x \leq b \\ \left(\frac{c-x}{c-b}\right); & b \leq x \leq c \end{cases} \dots\dots\dots(2.3)$$

3. Representasi Kurva Bahu

Daerah yang terbentuk di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk kurva segitiga, pada sisi kanan dan kirinya akan naik turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Himpunan *fuzzy* “bahu”, digunakan untuk mengakhiri variabel suatu daerah *fuzzy*.



Gambar 2.5 Representasi Kurva Bahu

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} 0 & ; x \geq d \text{ atau } x \leq a \\ \left(\frac{x-a}{b-a}\right) & ; a \leq x \leq b \\ \left(\frac{d-x}{d-c}\right) & ; c \leq x \leq d \\ 1 & ; b \leq x \leq c \end{cases} \dots\dots\dots(2.4)$$

2.12 Metode Fuzzy Tsukamoto

Pada metode Tsukamoto, setiap konsekuensi pada aturan yang berbentuk If-Then harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan. Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan secara tegas (crisp) berdasarkan α -predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot. [13]

a. Metode Tsukamoto

Secara umum bentuk model Fuzzy Tsukamoto adalah If (X IS A) and (Y IS B) Then (Z IS C) Dimana A, B, C adalah himpunan fuzzy. Dalam inferensinya, metode Tsukamoto menggunakan tahapan berikut:

1. Fuzzyfikasi
2. Pembentukan basis pengetahuan fuzzy (*Rule* dalam bentuk IF...THEN)
3. Mesin inferensi

Menggunakan fungsi implikasi MIN untuk mendapatkan nilai α -predikat tiap-tiap rule ($\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$). Kemudian masing-masing nilai α -predikat ini digunakan untuk menghitung keluaran hasil inferensi secara tegas (crisp) masing-masing rule ($z_1, z_2, z_3, \dots, z_n$).

4. Defuzzyfikasi

Menggunakan metode rata-rata (*average*).

$$Z^* = \sum \frac{\mu(z).z}{\mu(z)} \dots\dots\dots(2.5)$$

Pada metode Tsukamoto, setiap aturan direpresentasikan menggunakan himpunan-himpunan fuzzy, dengan fungsi keanggotaan. Untuk menentukan nilai output crisp/hasil yang tegas (Z) dicari dengan cara mengubah *input* (berupa himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy) menjadi suatu bilangan pada domain himpunan fuzzy tersebut. Cara ini disebut dengan

metode defuzzifikasi (penegasan). Metode defuzzifikasi yang digunakan dalam metode Tsukamoto adalah metode defuzzifikasi rata-rata terpusat (*Center Average Defuzzifier*). [9]

Definisi Fuzzy Tsukamoto digunakan sebagai landasan pengimplementasian metode aplikasi yang diletakkan pada saat menampilkan *score* di akhir permainan.

BAB III. METODOLOGI PENELITIAN

3.1 Studi Literatur

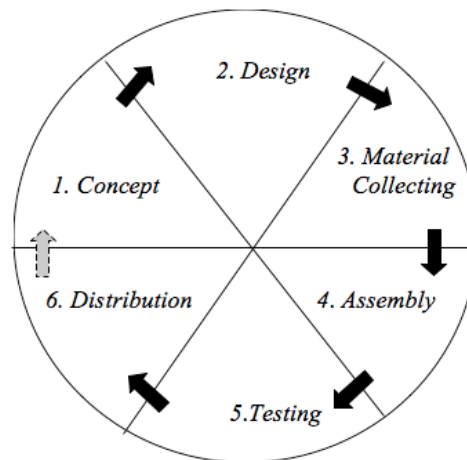
Studi literatur ini sebagai dasar teori yang akan melandasi pengerjaan aplikasi ini dengan mencari literatur-literatur yang berkaitan pada media elektronik maupun media cetak sebagai bahan untuk pembuatan aplikasi ini.

3.2 Pengumpulan Data

Tahapan pengumpulan data ini dilakukan untuk memperoleh adalah pengumpulan data penunjang yang dilakukan dengan pengumpulan materi dari buku mengenai huruf *Hiragana* dan *Katakana* yang digunakan sebagai landasan teori dalam penulisan laporan skripsi ini.

3.3 Perancangan Sistem

Dalam pengembangan aplikasi *game puzzle* huruf jepang *hiragana katakana* menggunakan metode *Multimedia Development Life Cycle* (MDLC). Dimana proses metode MDLC terdiri dari beberapa fase yaitu :



Gambar 3. 1 Metode Pengembangan Multimedia Development Life Cycle

3.3.1 Konsep (*Concept*)

Pada tahapan ini merancang aplikasi *game puzzle* pengenalan huruf *Hiragana* dan *Katakana* ini dimulai dengan menentukan target pengguna dari aplikasi ini.

3.3.2 Desain (*Design*)

Pada tahap ini menjelaskan mengenai arsitektur pengguna *storyboard* untuk menggambarkan deskripsi tiap *scene* dengan mencantumkan semua objek multimedia yang telah diberi penjelasan mengenai alur jalannya aplikasi.

3.3.3 Pengumpulan Bahan (*Material Collecting*)

Material Collecting adalah tahap dimana pengumpulan bahan yang sesuai dengan kebutuhan dilakukan. Adapun bahan – bahan yang dibutuhkan antara lain:

- a. Gambar : semua gambar yang akan digunakan, seperti gambar tombol, gambar puzzle huruf jepang *hiragana* dan *katakana*, icon *game*, dll. Gambar tersebut berformat .png lalu dijadikan objek pada aplikasi ini.
- b. Audio : file audio yang digunakan dalam aplikasi ini bertipe .mp3.
- c. *Interface* : Tampilan yang dibuat dan digunakan sebagai fitur aplikasi.

3.3.4 Pembuatan (*Assembly*)

Dari *material collecting* yang telah tersedia kemudian diolah dengan memanfaatkan *game engine* Unity sesuai dengan konsep dan desain yang sudah di rencanakan sebelumnya. Pembuatan aplikasi *game puzzle Hiragana* dan *Katakana* dan pengimplementasian dilakukan pada tahap ini.

3.3.5 Pengujian (*Testing*)

Pengujian dilakukan setelah di-*build* pada *smartphone* kemudian diuji langsung kepada *user*, apakah aplikasi ini dapat dioperasikan dengan baik dan fitur – fitur yang ada dapat berjalan.

3.3.6 Distribusi (*Distribution*)

Pada tahap ini, project yang telah selesai diuji akan disimpan dalam bentuk file .apk dengan tujuan dapat didistribusikan dan digunakan oleh pengguna dengan target usia 16 tahun.

BAB IV. ANALISIS DAN PERANCANGAN

4.1 Konsep (*Concept*)

Pada tahapan konsep merancang aplikasi *game puzzle* pengenalan huruf *Hiragana* dan *Katakana* ini dimulai dengan gambaran umum sistem hingga menentukan target pengguna dari aplikasi ini.

4.1.1 Gambaran Umum Sistem

Gambaran umum dari aplikasi *Game Puzzle* Sebagai Pembelajaran Huruf Jepang *Hiragana* dan *Katakana* antara lain,

- a. Aplikasi ini dimainkan oleh satu orang saja atau *Single Player*.
- b. Aplikasi ini berbasis Android atau dijalankan pada perangkat *Mobile* dengan sistem operasi Android (*Smartphone*).
- c. Aplikasi ini merupakan aplikasi pembelajaran mengenai bentuk dan cara penulisan huruf Jepang *Hiragana* dan *Katakana* yang diawali dengan pengenalan huruf Jepang *Hiragana* dan *Katakana* sebagai pendahuluan berisi materi pengantar sebelum melakukan pembelajaran. Pada saat proses pembelajaran, aplikasi akan memberikan gambaran huruf *Hiragana* dan *Katakana* berupa gambar dan video animasi secara langsung kepada pengguna dan permainan *puzzle* serta latihan untuk cara penulisan huruf Jepang *Hiragana* dan *Katakana*.
- d. Aplikasi ini digunakan untuk pengguna berusia 16 tahun.

4.1.2 Analisa Kebutuhan Non Fungsional

Analisa kebutuhan non fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah aplikasi dan menjelaskan analisis kebutuhan perangkat lunak, analisa kebutuhan perangkat keras, serta analisis kebutuhan pengguna. Beberapa perangkat lunak (software) yang dibutuhkan dalam membangun aplikasi yaitu, :

- a. Sistem Operasi Windows 7
- b. Unity versi 5.6.3

Digunakan untuk membangun aplikasi *game puzzle* sebagai media pembelajaran huruf Jepang *Hiragana* dan *Katakana*

c. Android SDK dan Java Development Kit (JDK)

Digunakan sebagai penunjang pada saat *build* aplikasi ke dalam bentuk .apk

d. Adobe Illustrator CC 2014

Digunakan untuk membuat asset yang dibutuhkan dan juga untuk membuat objek huruf Jepang *Hiragana* dan *Katakana*

e. Adobe Flash CC 2014

Digunakan untuk membuat animasi cara penulisan huruf Jepang *Hiragana* dan *Katakana*

4.1.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang diterapkan dalam sebuah aplikasi dan menjelaskan kebutuhan yang diperlukan aplikasi agar aplikasi dapat berjalan serta sesuai dengan kebutuhan. Analisis yang dilakukan dimodelkan dengan menggunakan *usecase diagram*, *Work Breakdown Structure* (WBS), dan *flowchart*.

4.1.3.1 Usecase Diagram

Usecase diagram ini digunakan untuk menggambarkan hubungan sejumlah *external actor* dengan *usecase* yang terdapat dalam sistem. *Usecase diagram* ini hanya menggambarkan keadaan lingkungan sistem yang dapat dilihat dari luar oleh *actor*. Pada *usecase* ini terdapat satu *actor*, yaitu *user*. Penjelasan mengenai *actor* dapat pada tabel 4.1 berikut.

Tabel 4. 1 Tabel penjelasan actor

No	Nama Actor	Definisi
1.	<i>User</i>	Orang yang berinteraksi dengan system, yaitu pengguna yang mengakses semua fungsi yang disediakan sistem.

a. Skenario *Usecase Diagram*

Usecase pada aplikasi berhubungan dengan interaksi *user* terhadap sistem.

Diagram *usecase* pada gambar 4.1 dilakukan oleh *actor* sebagai *User* yang menggambarkan bagaimana *actor* dapat menjalankan fungsi-fungsi yang terdapat pada aplikasi.



Gambar 4. 1 Use Case Diagram

Pada diagram *usecase* yang ditunjukkan pada gambar 4.1 menjelaskan masing-masing deskripsi dari *usecase* diagram yang dilakukan oleh *user* dalam penggunaan aplikasi. Pada saat *user* memulai aplikasi menampilkan menu utama yaitu menu *Start*, *user* memilih menu *Hiragana* atau *Katakana* untuk mengakses materi huruf Jepang *Hiragana* atau *Katakana* berupa video animasi. Kemudian *user* akan menyusun *puzzle* dengan cara melakukan *drag and drop* hingga membentuk gambar huruf *Hiragana* atau *Katakana*. Selain itu terdapat juga tombol *practice* untuk melakukan latihan cara penulisan huruf *Hiragana* dan *Katakana* yang benar. Terdapat juga tombol *about* yang menuju halaman informasi pembuat aplikasi, serta tombol latihan yang menuju ke halaman petunjuk penggunaan aplikasi, serta tombol keluar untuk keluar dari aplikasi.

Berikut merupakan deskripsi *Usecase Diagram* yang dijabarkan dalam bentuk tabel 4.2 sebagai berikut.

Tabel 4. 2 Deskripsi usecase diagram

No.	Kode	Nama <i>Usecase</i>	Keterangan
1.	UC-P01	Menu Mulai	Sistem akan menuju ke tampilan halaman menu pembelajaran huruf <i>Hiragana</i> atau huruf <i>Katakana</i> .
2.	UC-P02	Informasi	Sistem akan menampilkan data diri dari pembuat aplikasi.
3.	UC-P03	Petunjuk	Sistem akan menampilkan petunjuk pengoperasian aplikasi.

Berikut adalah tabel spesifikasi skenario dari Menu Mulai. Untuk penjelasan selanjutnya terdapat pada tabel 4.3

Tabel 4. 3 Spesifikasi skenario : Menu Mulai

Nama Usecase	Menu Mulai	
Nomor	UC-P01	
Aktor	<i>User</i>	
Kondisi Awal	<i>User</i> memilih tombol Mulai	
	Reaksi Aktor	Reaksi Sistem
	1. <i>User</i> memilih tombol mulai 3. <i>User</i> memilih tombol <i>Hiragana</i> 5. <i>User</i> memilih tombol <i>Katakana</i> 7. <i>User</i> memilih pilihan huruf 9. <i>User</i> memilih tombol <i>puzzle</i> 11. <i>User</i> memainkan puzzle dengan menyusun potongan gambar membentuk gambar huruf yang dimaksud 13. <i>User</i> memilih tombol <i>practice</i> 15. <i>User</i> melakukan praktik menulis huruf yang dimaksud 17. <i>User</i> memilih tombol huruf selanjutnya	2. Sistem akan menampilkan <i>scene</i> pilihan materi pembelajaran 4. Sistem akan menampilkan <i>scene</i> materi <i>Hiragana</i> 6. Sistem akan menampilkan <i>scene</i> materi <i>Katakana</i> 8. Sistem akan menampilkan <i>scene</i> video ilustrasi penulisan huruf pertama berdasarkan pilihan huruf 10. Sistem akan menampilkan <i>scene</i> <i>Puzzle</i> 12. Sistem menanggapi reaksi aktor dengan menyelesaikan permainan atau mengulang permainan 14. Sistem akan menampilkan <i>scene</i> <i>Practice</i> yakni praktik menulis huruf yang dimaksud 16. Sistem memberikan predikat bintang berdasarkan kegiatan yang telah dilakukan 18. Sistem menampilkan <i>scene</i> huruf selanjutnya
Kondisi Akhir	Mengakses huruf hingga menyelesaikan <i>practice</i> menulis huruf	
Skenario Alternatif	<i>Button</i> tidak berfungsi	

Berikut adalah tabel spesifikasi skenario dari Informasi Aplikasi. Untuk penjelasan selanjutnya terdapat pada tabel 4.4.

Tabel 4. 4 Spesifikasi Skenario: Informasi Aplikasi

Nama Usecase	Informasi Aplikasi	
Nomor	UC-P02	
Aktor	<i>User</i>	
Kondisi Awal	<i>User</i> memilih tombol Informasi Aplikasi	
Reaksi Aktor		Reaksi Sistem
1. <i>User</i> mengakses halaman Informasi 3. <i>User</i> memilih tombol Kembali		2. Sistem akan menampilkan <i>Scene</i> Informasi Aplikasi 4. Sistem menampilkan <i>Scene</i> Menu Utama
Kondisi Akhir		Menampilkan halaman Informasi aplikasi
Skenario Alternatif		<i>Button</i> tidak berfungsi

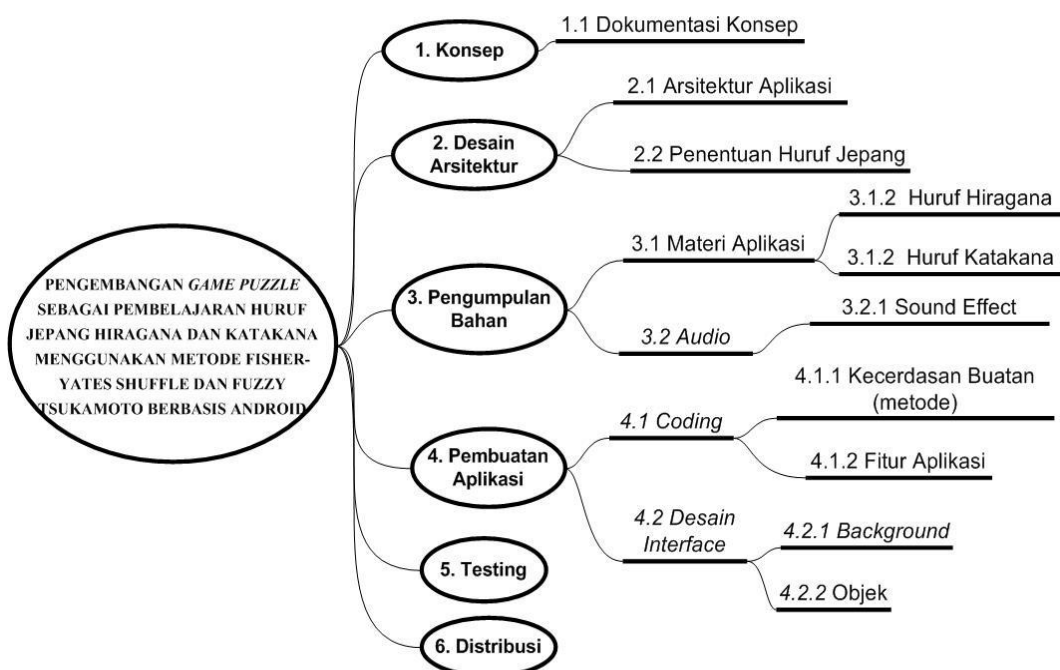
Berikut adalah tabel spesifikasi skenario dari Petunjuk Aplikasi. Untuk penjelasan selanjutnya terdapat pada tabel 4.5

Tabel 4. 5 tabel spesifikasi skenario dari Petunjuk Aplikasi

Nama Usecase	Petunjuk Aplikasi	
Nomor	UC-P03	
Aktor	<i>User</i>	
Kondisi Awal	<i>User</i> memilih tombol Petunjuk Aplikasi	
Reaksi Aktor		Reaksi Sistem
1. <i>User</i> mengakses halaman Petunjuk 3. <i>User</i> memilih tombol Lanjut 5. <i>User</i> memilih tombol Kembali		2. Sistem akan menampilkan <i>Scene</i> Petunjuk Aplikasi 4. Sistem menampilkan halaman selanjutnya 6. Sistem menampilkan halaman sebelumnya
Kondisi Akhir		Menampilkan halaman petunjuk aplikasi
Skenario Alternatif		<i>Button</i> tidak berfungsi

4.1.3.2 Work Breakdown Structure (WBS)

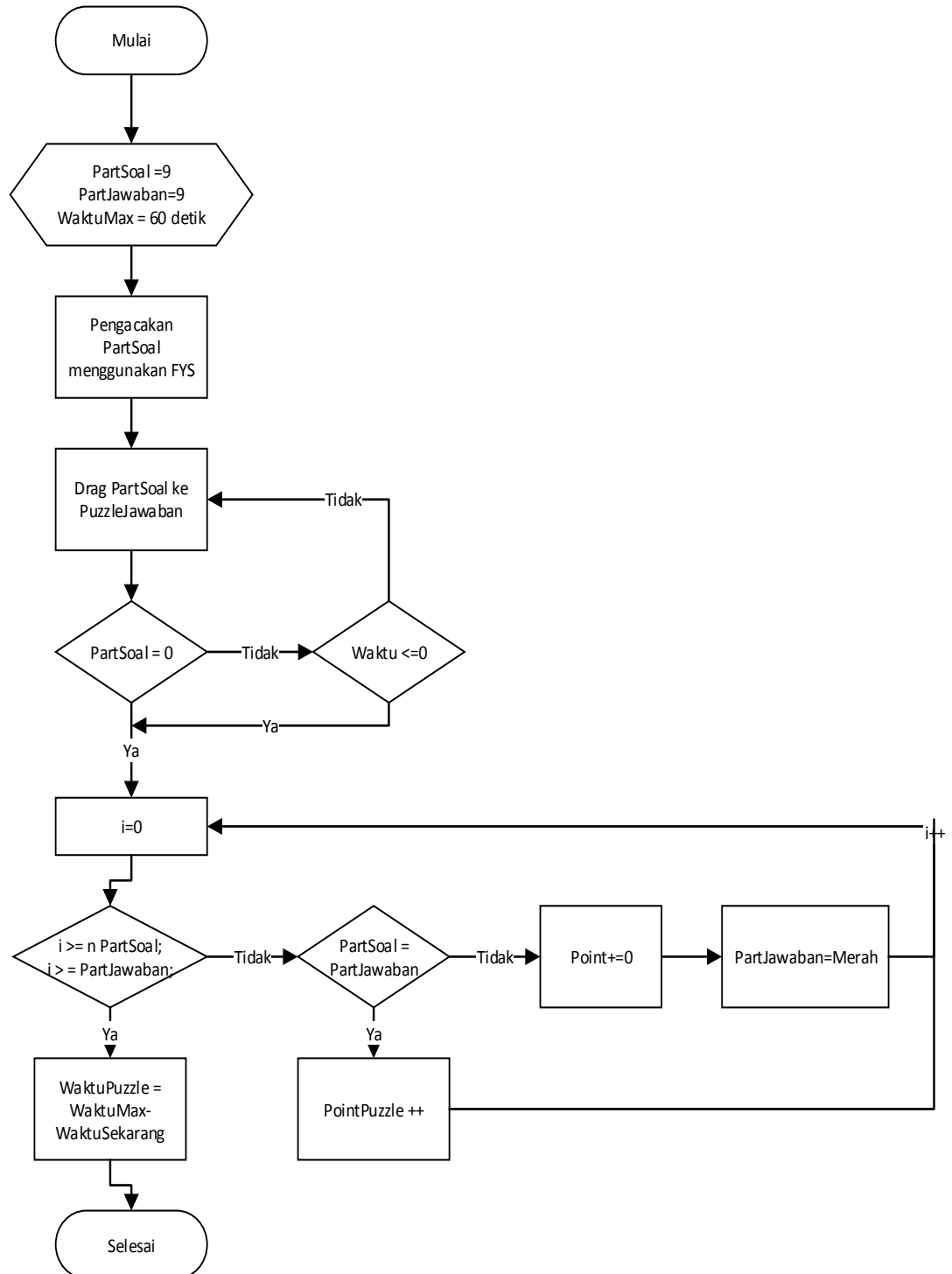
Pada Pembuatan ini Struktur kerja yang akan diterapkan pada aplikasi ini menggunakan konsep dan alur dari *Multimedia Development Life Cycle* dan dijabarkan sesuai dengan pekerjaan yang harus dikerjakan saat pembuatan aplikasi ini. Berikut WBS yang di gambarkan pada gambar 4.2. Pada *Work Breakdown Structure* terdapat 6 tahapan dalam penyusunan aplikasi ini, diantara Konsep yang memuat dokumentasi konsep merancang aplikasi *game puzzle* pengenalan huruf *Hiragana* dan *Katakana* ini dimulai dengan menentukan target pengguna dari aplikasi ini. Desain Arsitektur berisi mengenai arsitektur pengguna *storyboard* untuk menggambarkan deskripsi tiap *scene* dengan mencantumkan semua objek multimedia yang telah diberi penjelasan mengenai alur jalannya aplikasi. Pengumpulan bahan adalah tahap dimana pengumpulan bahan yang sesuai dengan kebutuhan dilakukan. Pada tahapan pembuatan aplikasi dilakukan implementasi metode desain *interface*. Pengujian dilakukan setelah di-*build* pada *smartphone* kemudian diuji langsung kepada *user*, apakah aplikasi ini dapat dioperasikan dengan baik dan fitur – fitur yang ada dapat berjalan. Pada tahap ini, project yang telah selesai diuji akan disimpan dalam bentuk file .apk dengan tujuan dapat didistribusikan dan digunakan oleh pengguna dengan target usia 16 tahun.



Gambar 4. 2 Work Breakdown Structure

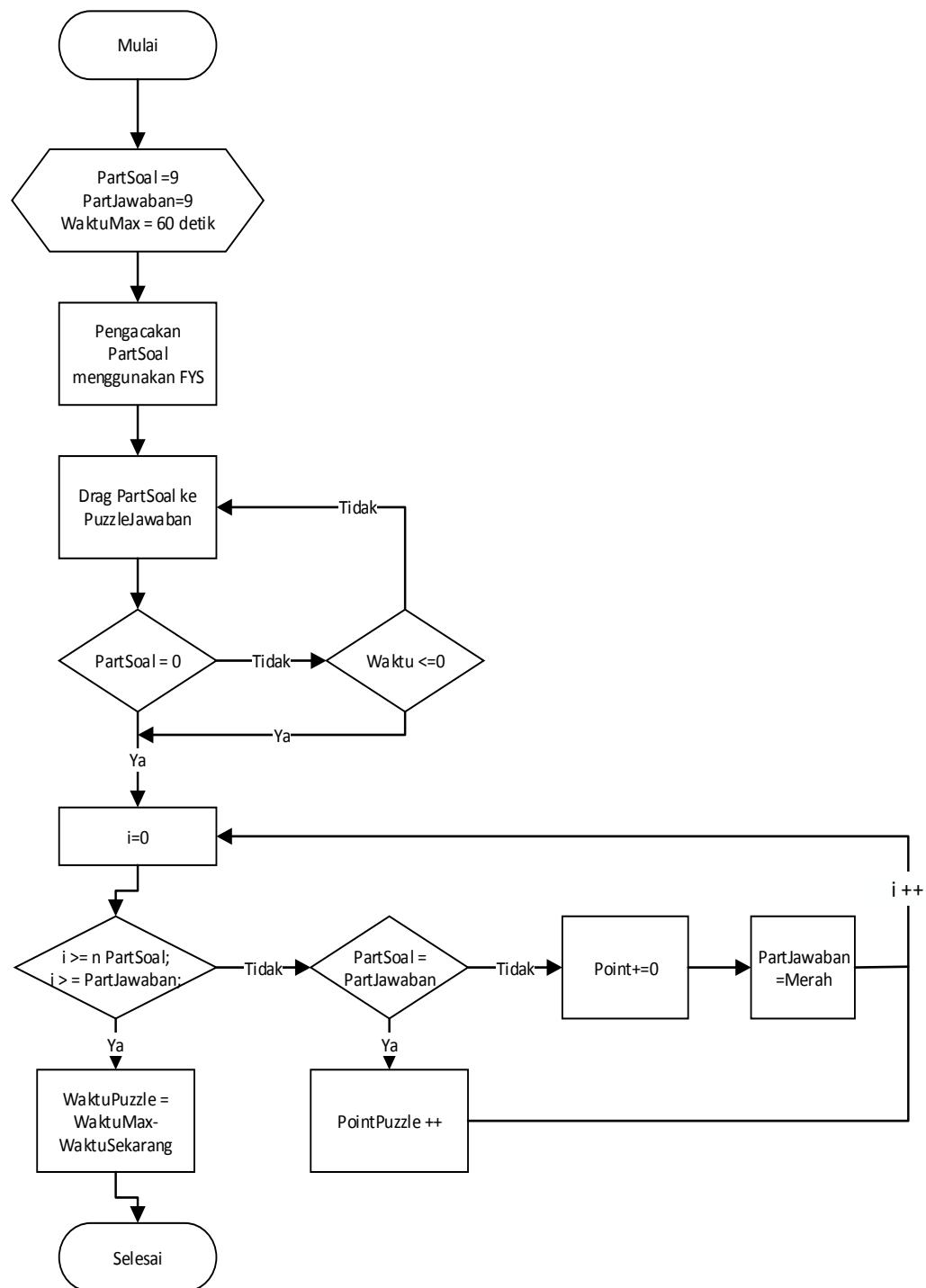
4.1.3.3 Flowchart

Dibawah ini merupakan gambar *flowchart game puzzle* dan waktu dari aplikasi pembelajaran Huruf Jepang *Hiragana* dan *Katakana*.



Gambar 4. 3 Flowchart Game Puzzle dan Waktu

Dibawah ini merupakan gambar *flowchart* praktik menulis dari aplikasi pembelajaran Huruf Jepang *Hiragana* dan *Katakana*.



Gambar 4. 4 *Flowchart* Praktik Menulis

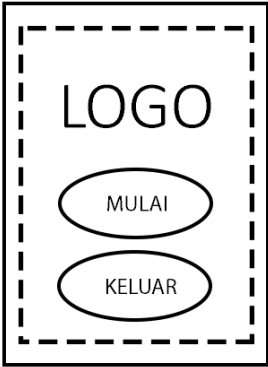
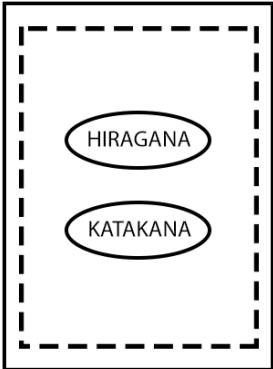

4.2 Desain (*Design*)


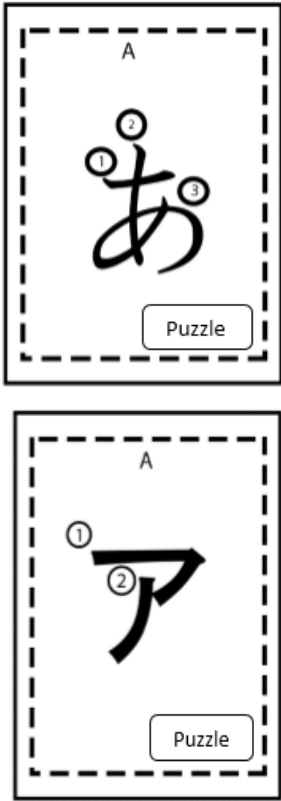
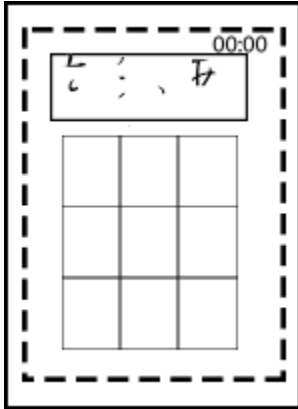
Pada tahap ini menjelaskan mengenai *storyboard* untuk menggambarkan deskripsi dengan mencantumkan semua objek multimedia yang telah diberi penjelasan mengenai alur jalannya aplikasi.

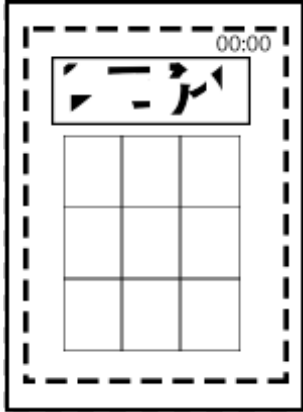
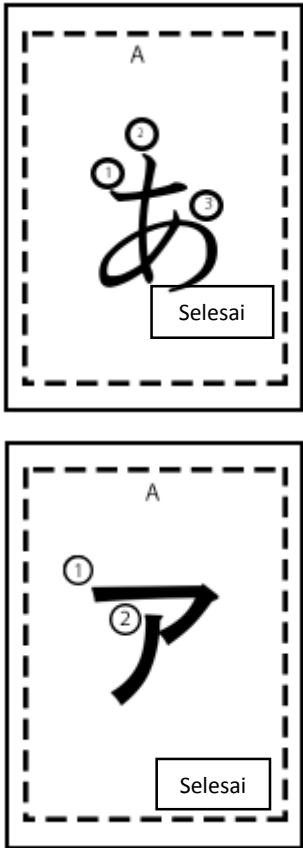
4.2.1 Perancangan Storyboard

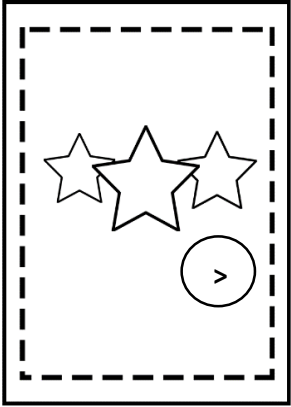
Berikut ini alur aplikasi pembelajaran huruf Jepang *Hiragana* dan *Katakana* dalam bentuk *storyboard*.

Tabel 4. 6 Storyboard game Puzzle Huruf *Hiragana* dan *Katakana*

No	Storyboard	Deskripsi/Penjelasan
1.		<p>Tampilan Menu</p> <ul style="list-style-type: none"> - Pada menu awal terdapat tombol Mulai dan Keluar.
2.		<p>Tampilan Menu</p> <ul style="list-style-type: none"> - Terdapat tampilan pilihan menu huruf <i>Hiragana</i> dan <i>Katakana</i>.
3.		<p>Tampilan Menu Huruf <i>Hiragana</i> dan <i>Katakana</i></p> <ul style="list-style-type: none"> - Pada menu <i>Hiragana</i> dan <i>Katakana</i> berjumlah 46 huruf, berisi huruf a i u e o suku kata deretan ka, sa, ta, na, ha, ma, ya ra, wa, dan satu konsonan yaitu n serta satu kata bantu o

		yang biasanya diucapkan wo.
4.		<p>Tampilan Animasi Pola Huruf <i>Hiragana/Katakana</i></p> <ul style="list-style-type: none"> - Pada menu animasi pengguna dapat melihat pola huruf <i>hiragana/katakana</i> yang dimunculkan. - Terdapat tombol <i>Puzzle</i> untuk menuju halaman menu <i>Game Puzzle</i> - Pada menu animasi pengguna dapat melihat pola huruf <i>hiragana/katakana</i> yang dimunculkan. - Terdapat tombol <i>Puzzle</i> untuk menuju halaman menu <i>Game Puzzle</i>
6.		<p>Tampilan <i>Game Puzzle</i></p> <ul style="list-style-type: none"> - Pada menu ini berisi <i>puzzle</i> huruf <i>Hiragana/Katakana</i>. - Pada menu ini metode <i>Fisher-Yates Shuffle</i> diimplementasikan

		
7.		<p>Tampilan menu mengikuti pola huruf <i>hiragana/katakana</i></p> <ul style="list-style-type: none"> - Pada menu mengikuti pola, pengguna mengikuti pola huruf <i>hiragana/katakana</i> untuk mendapatkan <i>score</i>. - Pada menu ini metode Fuzzy Tsukamoto diimplementasikan. - Pada menu ini terdapat tombol Selesai untuk pengecekan penulisan apakah pengguna telah mengikuti pola menulis dengan benar.

8.		<p>Akhir permainan</p> <ul style="list-style-type: none"> - Diakhir permainan user akan mendapatkan nilai <i>score</i> berupa bintang 1, bintang 2, atau bintang 3.
----	---	---

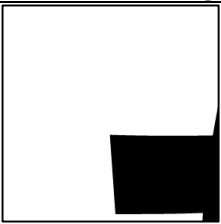

4.2.2 Perancangan Materi

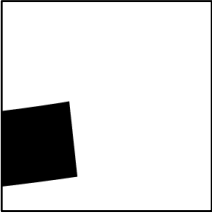
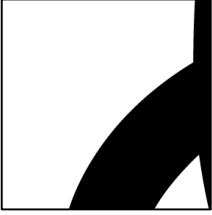
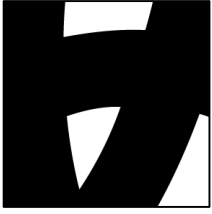


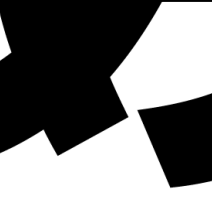
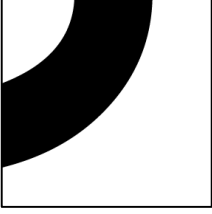
Materi yang ingin disampaikan dalam *game puzzle* huruf *Hiragana* dan *Katakana* yaitu mengenalkan *puzzle* huruf *Hiragana* dan *Katakana*. Selain itu terdapat materi untuk mengenalkan cara penulisan huruf *Hiragana* dan *Katakana* dengan benar.

4.2.3 Perancangan *Puzzle*

Jenis *puzzle* terdiri dari potongan huruf *Hiragana* dan *Katakana* dengan jumlah potongan gambar huruf sebanyak 9 buah. Berikut adalah tabel perancangan figure *puzzle* yang akan digunakan pada *game puzzle* huruf *Hiragana* dan *Katakana*. Potongan gambar huruf tersebut akan di acak posisinya menggunakan algoritma *Fisher Yates-Shuffle*.

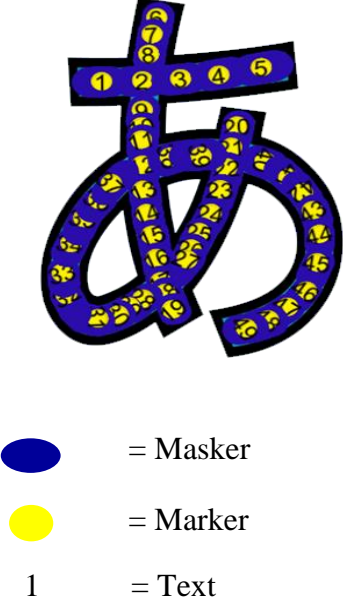




Tabel 4. 7 Tabel Figure *Puzzle* Huruf *Hiragana* dan *Katakana*

No	<i>Puzzle Huruf Hiragana</i>
1	
2	

3	
4	
5	
6	
7	
8	
9	

Berikut adalah tabel perancangan praktik yang akan digunakan pada aplikasi ini.

Tabel 4.8 Perancangan Menu Praktik Menulis

No	Practice Huruf	Keterangan
1.	 <p>  = Masker  = Marker 1 = Text </p>	<p>Pada menu <i>practice</i>, setiap huruf dirancang dengan beberapa <i>node</i> yang didalamnya terdiri dari Masker, Marker, dan Text. Pada Masker berfungsi untuk pewarnaan ketika pengguna menulis. Pada Marker berfungsi sebagai <i>background</i> dari Text. Pada Text berfungsi untuk membantu pembuatan langkah – langkah penulisan. Setiap huruf memiliki jumlah <i>node</i> berbeda-beda dikarenakan setiap huruf memiliki bentuk yang berbeda-beda dan berdasarkan langkah-langkah menulisnya juga berbeda.</p> <p>Gambar No. 1 merupakan perancangan <i>practice</i> untuk huruf Hiragana “A” terdapat 49 <i>node</i>.</p>
2.		<p>Gambar No. 2 merupakan perancangan <i>practice</i> untuk huruf Hiragana “U” terdapat 18 <i>node</i>.</p>
3.		<p>Gambar No. 3 merupakan perancangan <i>practice</i> untuk huruf Katakana “O” terdapat 29 <i>node</i>.</p>

4.2.4 Perancangan Metode *Fisher Yates Shuffle*

Pada aplikasi ini algoritma *Fisher-Yates Shuffle* diterapkan pada pengacakan *Puzzle* yang ditampilkan. Pengacakan *puzzle* yang diacak dengan algoritma *Fisher-Yates Shuffle* memiliki urutan yang berbeda dengan sebelumnya, sehingga pengguna tidak jenuh ketika mendapatkan potongan gambar yang sama. *Fisher-Yates Shuffle* terdapat beberapa tahap dalam proses pengacakannya. Berikut ini adalah tahapan dari *Fisher-Yates Shuffle*.

- a. Menentukan nilai n
- b. Memilih sebuah angka acak (y) diantara 1 sampai dengan n
- c. Menukar posisi angka acak dengan angka terakhir dengan range $1 - n$
- d. Memindahkan angka acak ke list array
- e. Mengatur nilai n , dimana $n = n - 1$
- f. Jika n masih memenuhi $n > 0$ maka melakukan kembali proses pilih angka acak dimana angka acak kurang dari sama dengan n , angka acak lebih dari sama dengan 1 ($1 \leq y \leq n$)
- g. Jika $n = 0$ maka proses pengacakan telah selesai.

Dari tahapan proses pengacakan *Fisher-Yates Shuffle*, berikut ini penerapannya.

Tabel 4.9 Penerapan metode *Fisher-Yates Shuffle*

Langkah – langkah <i>Fisher-Yates Shuffle</i>	Hasil Pengacakan									
Menentukan nilai <i>n</i> . Nilai <i>n</i> pada penerapan pengacakan <i>puzzle</i> di aplikasi ini yaitu, <i>n</i> = {0, 1, 2, 3, 4, 5, 6, 7, 8}	<table><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	4	5	6	7	8
0	1	2								
3	4	5								
6	7	8								

Memilih sebuah angka acak diantara 1 sampai dengan n ($1 - n$). Misal angka acak yang terpilih adalah 2.	<table><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	4	5	6	7	8
0	1	2								
3	4	5								
6	7	8								
Menukar posisi angka acak dengan angka terakhir dengan <i>range</i> $1 - n$. Apabila sebelumnya $n = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ maka saat ini $n = \{0, 1, 8, 3, 4, 5, 6, 7\}$	<table><tr><td>0</td><td>1</td><td>8</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td></td></tr></table>	0	1	8	3	4	5	6	7	
0	1	8								
3	4	5								
6	7									
Memindahkan angka acak ke list array. Misalkan r (list array) = $\{2\}$	<table><tr><td>0</td><td>1</td><td>8</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td></td></tr></table>	0	1	8	3	4	5	6	7	
0	1	8								
3	4	5								
6	7									
Mengatur nilai n , dimana $n = n-1$, $(9-1) = 8$. Saat ini $n = \{0, 1, 8, 3, 4, 5, 6, 7\}$	<table><tr><td>0</td><td>1</td><td>8</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td></td></tr></table>	0	1	8	3	4	5	6	7	
0	1	8								
3	4	5								
6	7									

Jika n masih memenuhi $n > 0$ maka melakukan kembali proses pilih angka acak dimana angka acak kurang dari sama dengan n , angka acak lebih dari sama dengan 1 ($1 \leq y \leq n$). Kemudian menukar posisi angka acak dengan angka terakhir dengan *range* 1 – n .

Nilai n saat ini yaitu {0, 1, 8, 3, 4, 5, 6, 7}, misal angka acak yang terpilih yaitu 0. Dimana $n-1$, $(8-1) = 7$,

Maka saat ini nilai $n = \{7, 1, 8, 3, 4, 5, 6\}$

Sehingga $r = \{2, 0\}$

7	1	8
3	4	5
6		

Nilai n sebelumnya yaitu {7, 1, 8, 3, 4, 5, 6}, misal angka acak yang terpilih yaitu, 5. Dimana $n-1$,

$(7-1) = 6$

Maka saat ini nilai $n = \{7, 1, 8, 3, 4, 6\}$

Sehingga $r = \{2, 0, 5\}$

7	1	8
3	4	6

Nilai n sebelumnya yaitu {7, 1, 8, 3, 4, 6}, misal angka acak yang terpilih yaitu, 1. Dimana $n-1$,

$(6-1) = 5$

Maka saat ini nilai $n = \{7, 6, 8, 3, 4\}$

Sehingga $r = \{2, 0, 5, 1\}$

7	6	8
3	4	

Nilai n sebelumnya yaitu $\{7, 6, 8, 3, 4\}$, misal angka acak yang terpilih yaitu, 4. Dimana $n-1, (5-1) = 4$
Maka saat ini nilai $n = \{7, 6, 8, 3\}$
Sehingga $r = \{2, 0, 5, 1, 4\}$

7	6	8
3		

Nilai n sebelumnya yaitu $\{7, 6, 8, 3\}$, misal angka acak yang terpilih yaitu, 8. Diman $n-1, (5-1) = 4$
Maka saat ini nilai $n = \{7, 6, 3\}$
Sehingga $r = \{2, 0, 5, 1, 4, 8\}$

7	6	3

Nilai n sebelumnya yaitu $\{7, 6, 3\}$, misal angka acak yang terpilih yaitu, 3. Dimana $n-1, (3-1) = 2$
Maka saat ini nilai $n = \{7, 6\}$
Sehingga $r = \{2, 0, 5, 1, 4, 8, 3\}$

7	6	

Nilai n sebelumnya yaitu $\{7, 6\}$, misal angka acak yang terpilih yaitu, 6. Dimana $n-1, (2-1) = 1$
Maka saat ini nilai $n = \{7\}$
Sehingga $r = \{2, 0, 5, 1, 4, 8, 3, 6\}$

7		

Nilai n sebelumnya yaitu $\{7, 6\}$ dimana $n-1, (1-1) = 0$, angka acak yang tersisa yaitu, 7. Jika $n = 0$, maka pengacakan telah selesai.

Sehingga $r =$
 $\{2, 0, 5, 1, 4, 8, 3, 6, 7\}$

Setelah semua angka diacak, hasil angka yang ditampung pada r (list array) dimunculkan sehingga hasil pengacakan *Fisher-Yates Shuffle* pada *game puzzle* yaitu, $\{2, 0, 5, 1, 4, 8, 3, 6, 7\}$.

2	0	5
1	4	8
3	6	7

4.2.5 Penerapan *Fuzzy Tsukamoto*

Dalam aplikasi ini *Fuzzy Tsukamoto* diterapkan pada hasil *score*, yang nantinya akan menghasilkan gambar bintang sesuai *score* yang diperoleh *user*. Berikut ini adalah tahap-tahap analisa dari *Fuzzy Tsukamoto* :

4.3.4.1 Pembentukan Himpunan Fuzzy Tsukamoto

Dalam aplikasi ini digunakan 3 variabel *input* dan 1 variabel *output*, variabel tersebut yaitu :

1. Variabel Point *Puzzle* sebagai variabel *input*.
2. Variabel Point Menulis sebagai variabel *input*.
3. Variabel Waktu sebagai variabel *input*.
4. Variabel Skor sebagai variabel *output*.

4.3.4.2 Nilai Linguistik

Dari tiga variabel yang digunakan, maka tiap-tiap variabel linguistik memiliki himpunan sebagai berikut :

1. Variabel Ketepatan *Puzzle* terbagi menjadi 3 *input* yaitu :
 - a. Sedikit
 - b. Sedang
 - c. Banyak
2. Variabel Ketepatan Menulis terbagi menjadi 3 *input* yaitu :
 - a. Rendah
 - b. Sedang
 - c. Tinggi

3. Variabel Waktu terbagi menjadi 3 *input* yaitu :
 - a. Lambat
 - b. Sedang
 - c. Cepat
4. Variabel Skor terbagi menjadi 3 *output* yaitu :
 - a. Kurang
 - b. Cukup
 - c. Baik

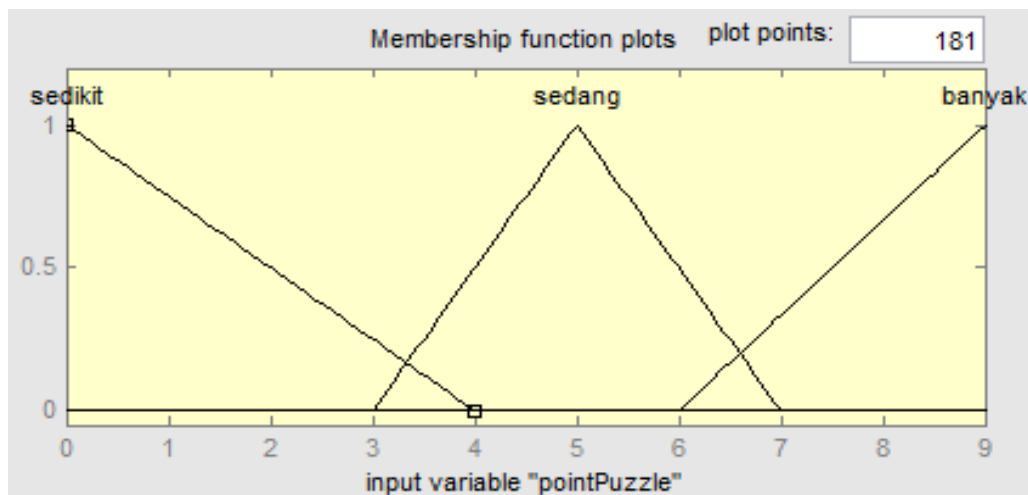
4.3.4.3 Fuzzifikasi

Merupakan proses memetakan nilai *crisps* (numerik) ke dalam himpunan *fuzzy* dan menentukan derajat keanggotaannya. Secara garis besar pemetaan nilai *crisps* ke dalam himpunan *fuzzy* dijelaskan sebagai berikut :

1. Fungsi Keanggotaan Point Puzzle

Proses menentukan *range* himpunan sedikit, sedang, banyak. Nilai maksimal dari fungsi keanggotaan ketepatan *puzzle* yaitu 9. Nilai *range* untuk variabel ketepatan *puzzle* sebagai berikut :

- a. Sedikit = 0 – 4
- b. Sedang = 3 – 7
- c. Banyak = 6 – 9



Gambar 4. 5 Curva Input Variabel Point Puzzle

Himpunan *fuzzy* rendah memiliki domain $[0, 4]$, dengan derajat keanggotaan sedikit ($=1$) terletak pada nilai 0 – 4. Fungsi keanggotaan untuk himpunan rendah

direpresentasikan ke dalam bentuk fungsi linear turun seperti dalam persamaan 4.1 dibawah ini.

$$\mu \text{ rendah } (x) = \begin{cases} 1 & ; x \leq 0 \\ \left(\frac{4-x}{4-0}\right) & ; 0 \leq x \leq 4 \\ 0 & ; x \geq 4 \end{cases} \dots\dots\dots(4.1)$$

Himpunan *fuzzy* sedang memiliki domain [3, 7], dengan derajat keanggotaan sedang tertinggi [=1] terletak pada nilai 3 – 7. Fungsi keanggotaan untuk himpunan sedang direpresentasikan ke dalam bentuk fungsi linear segitiga seperti dalam persamaan 4.2 dibawah ini.

$$\mu \text{ sedang } (x) = \begin{cases} 0 & ; x \leq 3 \text{ atau } x \geq 7 \\ \left(\frac{x-3}{5-3}\right) & ; 3 \leq x \leq 5 \\ \left(\frac{7-x}{7-5}\right) & ; 5 \leq x \leq 7 \end{cases} \dots\dots\dots(4.2)$$

Himpunan *fuzzy* banyak memiliki domain [6, 9], dengan derajat keanggotaan banyak tertinggi [=1] terletak pada nilai 6 – 9. Fungsi keanggotaan untuk himpunan sedang direpresentasikan ke dalam bentuk fungsi linear naik seperti dalam persamaan 4.3 dibawah ini.

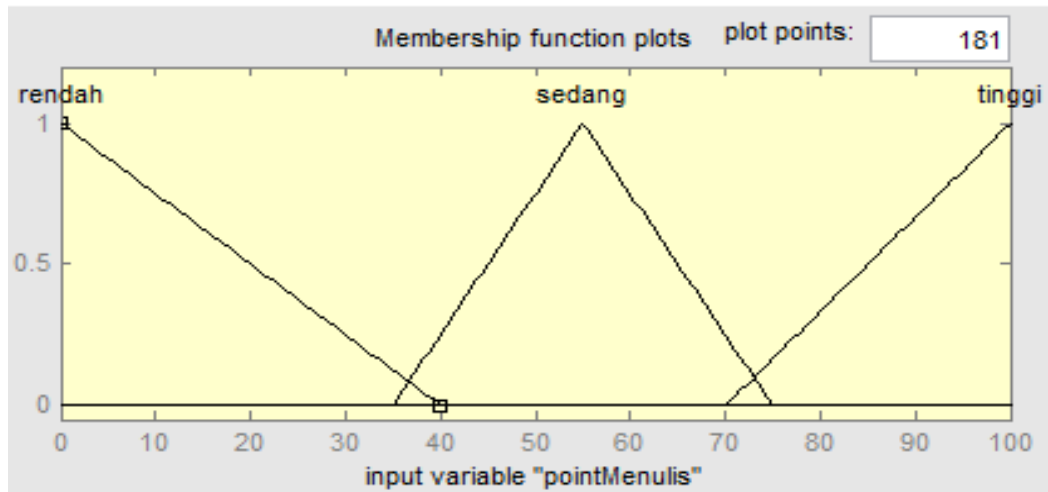
$$\mu \text{ banyak } (x) = \begin{cases} 0 & ; x \leq 6 \\ \left(\frac{x-6}{9-6}\right) & ; 6 \leq x \leq 9 \\ 1 & ; x \geq 9 \end{cases} \dots\dots\dots(4.3)$$

2. Fungsi Keanggotaan Point Menulis

Proses menentukan *range* himpunan rendah, sedang, tinggi. Nilai maksimal dari fungsi keanggotaan ketepatan menulis yaitu 100. Pada point menulis ini dilakukan nilai normalisasi sehingga dapat menghasilkan nilai 100. Untuk tata cara menulis yang sesuai dengan urutan *node* mendapatkan nilai satu point dan yang salah bernilai nol point. Perolehan jumlah skor pada point menulis didapatkan dari nilai normalisasi berdasarkan total urutan *node* yang benar dibagi dengan jumlah *node* dikali seratus. Nilai *range* untuk variabel ketepatan menulis sebagai berikut :

- a. Rendah = 0 – 40
- b. Sedang = 35 – 75

c. Tinggi = 70 – 100



Gambar 4. 6 Curva Input Variabel Point Menulis

Himpunan *fuzzy* rendah memiliki domain $[0, 40]$, dengan derajat keanggotaan sedikit ($=1$) terletak pada nilai $0 - 40$. Fungsi keanggotaan untuk himpunan rendah direpresentasikan ke dalam bentuk fungsi linear turun seperti dalam persamaan 4.4 dibawah ini.

$$\mu_{\text{rendah}}(x) = \begin{cases} 1 & ; x \leq 0 \\ \left(\frac{40-x}{40-0}\right) & ; 0 \leq x \leq 40 \\ 0 & ; x \geq 40 \end{cases} \dots\dots\dots(4.4)$$

Himpunan *fuzzy* sedang memiliki domain $[35, 75]$, dengan derajat keanggotaan sedang tertinggi [$=1$] terletak pada nilai $35 - 75$. Fungsi keanggotaan untuk himpunan sedang direpresentasikan ke dalam bentuk fungsi linear segitiga seperti dalam persamaan 4.5 dibawah ini.

$$\mu_{\text{sedang}}(x) = \begin{cases} 0 & ; x \leq 35 \text{ atau } x \geq 75 \\ \left(\frac{x-35}{55-35}\right) & ; 35 \leq x \leq 55 \\ \left(\frac{75-x}{75-55}\right) & ; 55 \leq x \leq 75 \end{cases} \dots\dots\dots(4.5)$$

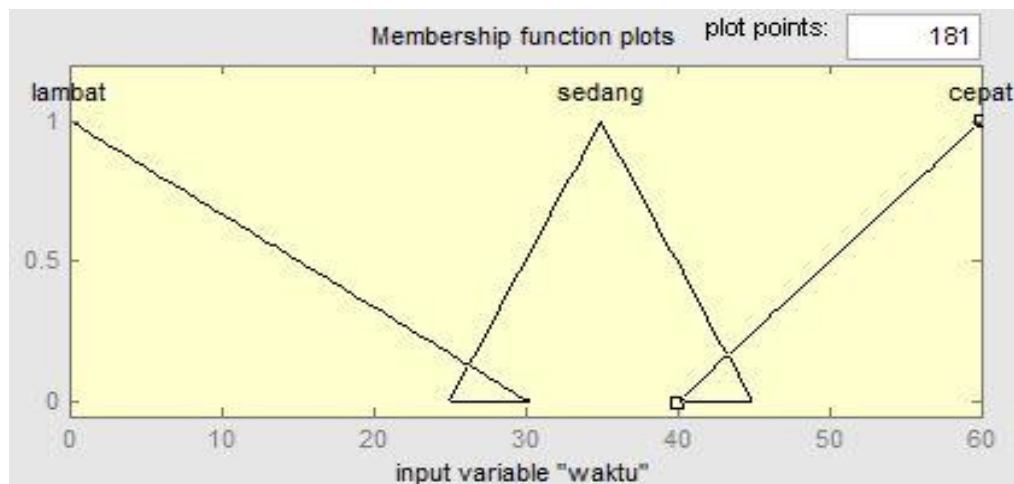
Himpunan *fuzzy* banyak memiliki domain $[70, 100]$, dengan derajat keanggotaan banyak tertinggi [$=1$] terletak pada nilai $70 - 100$. Fungsi keanggotaan untuk himpunan sedang direpresentasikan ke dalam bentuk fungsi linear naik seperti dalam persamaan 4.6 dibawah ini.

$$\mu \text{ banyak } (x) = \begin{cases} 0 & ; x \leq 70 \\ \left(\frac{x-70}{100-30} \right) & ; 70 \leq x \leq 100 \\ 1 & ; x \leq 100 \end{cases} \dots\dots\dots(4.6)$$

3. Fungsi Keanggotaan Waktu

Proses menentukan *range* himpunan sedikit, sedang, banyak. Nilai maksimal dari fungsi keanggotaan waktu yaitu 60. Nilai *range* untuk variabel waktu sebagai berikut :

- Sedikit = 0 – 30
- Sedang = 25 – 45
- Banyak = 40 – 60



Gambar 4. 7 Curva Input Variabel Waktu

Himpunan *fuzzy* rendah memiliki domain [0, 30], dengan derajat keanggotaan sedikit (=1) terletak pada nilai 0 – 30. Fungsi keanggotaan untuk himpunan sedikit direpresentasikan ke dalam bentuk fungsi linear turun seperti dalam persamaan 4.7 dibawah ini.

$$\mu \text{ lambat } (x) = \begin{cases} 1 & ; x \leq 0 \\ \left(\frac{30-x}{30-0} \right) & ; 0 \leq x \leq 30 \\ 0 & ; x \leq 30 \end{cases} \dots\dots\dots(4.7)$$

Himpunan *fuzzy* sedang memiliki domain [25, 45], dengan derajat keanggotaan sedang tertinggi [=1] terletak pada nilai 25 – 45. Fungsi keanggotaan untuk himpunan sedang direpresentsikan ke dalam bentuk fungsi linear segitiga seperti dalam persamaan 4.8 dibawah ini.

$$\mu_{\text{sedang}}(x) = \begin{cases} 0 & ; x \leq 25 \text{ atau } x \geq 45 \\ \left(\frac{x-25}{35-25}\right) & ; 25 \leq x \leq 35 \\ \left(\frac{45-x}{45-35}\right) & ; 35 \leq x \leq 45 \end{cases} \dots\dots\dots(4.8)$$

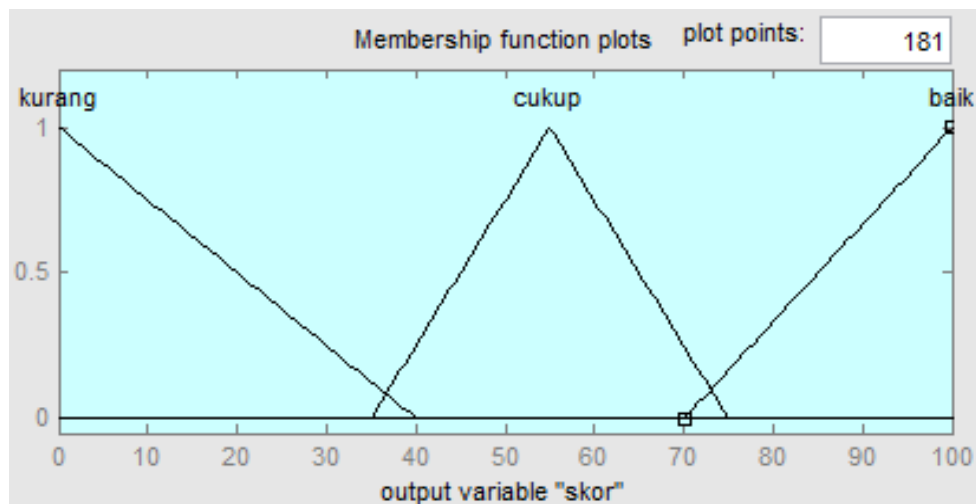
Himpunan *fuzzy* banyak memiliki domain [40, 60], dengan derajat keanggotaan banyak tertinggi [=1] terletak pada nilai 40 – 60. Fungsi keanggotaan untuk himpunan cepat direpresentasikan ke dalam bentuk fungsi linear naik seperti dalam persamaan 4.9 dibawah ini.

$$\mu_{\text{cepat}}(x) = \begin{cases} 0 & ; x \leq 40 \\ \left(\frac{x-40}{60-40}\right) & ; 40 \leq x \leq 60 \\ 1 & ; x \geq 60 \end{cases} \dots\dots\dots(4.9)$$

4. Fungsi Keanggotaan Skor

Setelah proses pembentukan fungsi keanggotaan, dilakukan pembuatan *rule base system*. Sebelum membuat *rule base system* ditentukan nilai diagram *fuzzy output*. Proses menentukan *range* himpunan kurang, cukup, baik. Nilai maksimal dari fungsi keanggotaan skor yaitu 100. Nilai *range* untuk variabel skor adalah sebagai berikut :

- a. Kurang = 0 – 40
- b. Cukup = 35 – 75
- c. Baik = 70 – 100



Gambar 4. 8 Curva Input Variabel Skor

Himpunan *fuzzy* kurang memiliki domain [0, 40], dengan derajat keanggotaan sedikit (=1) terletak pada nilai 0 – 40. Fungsi keanggotaan untuk

himpunan kurang direpresentasikan ke dalam bentuk fungsi linear turun seperti dalam persamaan 4.10 dibawah ini.

$$\mu \text{ kurang } (x) = \begin{cases} 1 & ; x \leq 0 \\ \left(\frac{40-x}{40-0} \right) & ; 0 \leq x \leq 40 \\ 0 & ; x \geq 40 \end{cases} \dots\dots\dots(4.10)$$

Himpunan *fuzzy* cukup memiliki domain [35, 75], dengan derajat keanggotaan sedang tertinggi [=1] terletak pada nilai 35 – 75. Fungsi keanggotaan untuk himpunan cukup direpresentasikan ke dalam bentuk fungsi linear segitiga seperti dalam persamaan 4.11 dibawah ini.

$$\mu \text{ cukup } (x) = \begin{cases} 0 & ; x \leq 35 \text{ atau } x \geq 75 \\ \left(\frac{x-35}{55-35} \right) & ; 35 \leq x \leq 55 \\ \left(\frac{75-x}{75-55} \right) & ; 55 \leq x \leq 75 \end{cases} \dots\dots\dots(4.11)$$

Himpunan *fuzzy* baik memiliki domain [70, 100], dengan derajat keanggotaan baik tertinggi [=1] terletak pada nilai 70 – 100. Fungsi keanggotaan untuk himpunan sedang direpresentasikan ke dalam bentuk fungsi linear naik seperti dalam persamaan 4.12 dibawah ini.

$$\mu \text{ baik } (x) = \begin{cases} 0 & ; x \leq 70 \\ \left(\frac{x-70}{100-70} \right) & ; 70 \leq x \leq 100 \\ 1 & ; x \geq 100 \end{cases} \dots\dots\dots(4.12)$$

4.3.4.4 Fuzzy Rules

Fuzzy Rules atau aturan-aturan yang diterapkan dalam penentuan *score* pada aplikasi ini berjumlah sebanyak 27 *rules* yaitu:

Tabel 4. 10 Fuzzy Rules

IF	KETEPATAN PUZZLE (A)	KETEPATAN MENULIS (H)	WAKTU (X)	SCORE (Z)
R1	SEDIKIT	RENDAH	LAMBAT	KURANG
R2	SEDIKIT	RENDAH	SEDANG	KURANG
R3	SEDIKIT	RENDAH	CEPAT	KURANG
R4	SEDIKIT	SEDANG	LAMBAT	KURANG
R5	SEDIKIT	SEDANG	SEDANG	CUKUP
R6	SEDIKIT	SEDANG	CEPAT	CUKUP
R7	SEDIKIT	TINGGI	LAMBAT	BAIK

R8	SEDIKIT	TINGGI	SEDANG	BAIK
R9	SEDIKIT	TINGGI	CEPAT	BAIK
R10	SEDANG	RENDAH	LAMBAT	KURANG
R11	SEDANG	RENDAH	SEDANG	CUKUP
R12	SEDANG	RENDAH	CEPAT	CUKUP
R13	SEDANG	SEDANG	LAMBAT	CUKUP
R14	SEDANG	SEDANG	SEDANG	BAIK
R15	SEDANG	SEDANG	CEPAT	BAIK
R16	SEDANG	TINGGI	LAMBAT	CUKUP
R17	SEDANG	TINGGI	SEDANG	BAIK
R18	SEDANG	TINGGI	CEPAT	BAIK
R19	BANYAK	RENDAH	LAMBAT	KURANG
R20	BANYAK	RENDAH	SEDANG	CUKUP
R21	BANYAK	RENDAH	CEPAT	CUKUP
R22	BANYAK	SEDANG	LAMBAT	CUKUP
R23	BANYAK	SEDANG	SEDANG	BAIK
R24	BANYAK	SEDANG	CEPAT	BAIK
R25	BANYAK	TINGGI	LAMBAT	BAIK
R26	BANYAK	TINGGI	SEDANG	BAIK
R27	BANYAK	TINGGI	CEPAT	BAIK

4.3.4.5 Mesin Inferensi dan Defuzzyfikasi

Fungsi *implikasi* yang digunakan adalah fungsi *implikasi* MIN untuk mendapatkan nilai α -predikat tiap–tiap rule ($\alpha_1, \alpha_2, \alpha_3, \dots \alpha_n$). Kemudian masing–masing nilai α -predikat ini digunakan untuk menghitung keluaran hasil inferensi secara tegas (*crisp*) masing–masing rule ($z_1, z_2, z_3, \dots z_n$). Dan proses *defuzzyfikasi* yang dilakukan dengan menggunakan metode Rata – Rata (*Average*).

$$z^* = \sum \frac{\alpha_i z_i}{\alpha_i} \dots\dots\dots(4.13)$$

4.3.4.6 Perhitungan perolehan point *Puzzle*

Pada *puzzle* terdiri dari sembilan potongan gambar huruf Jepang yang ditempatkan pada sembilan letak dari suatu kolom. Nilai yang didapatkan berdasarkan pada peletakan posisi potongan gambar yang sesuai dengan *puzzle* jawaban.

Puzzle jawaban merupakan kotak yang disediakan untuk menyusun potongan gambar yang benar yang memiliki indeks 0 hingga 8 yang ditunjukkan dibawah ini.

0	1	2
3	4	5
6	7	8

Puzzle jawaban diisi dengan potongan gambar yang didapat dari *puzzle* soal. *Puzzle* soal merupakan sekumpulan potongan gambar yang disajikan secara acak menggunakan metode *Fisher-Yates Shuffle* yang ditunjukkan dibawah ini.

3	2	1
5	6	4
9	8	7

Perolehan nilai untuk peletakan gambar yang sesuai dari potongan gambar yang berada pada *puzzle* soal ke *puzzle* jawaban yaitu bernilai satu poin dan yang salah bernilai nol poin. Perolehan jumlah skor pada point *puzzle* didapatkan dari total peletakan potongan gambar yang benar penyusunannya.

4.3.4.7 Perhitungan point menulis

Pada praktik menulis didapatkan nilai perhitungan berdasarkan tata cara menulis seperti video ilustrasi yang ditampilkan sebelumnya. Cara penulisan yang benar didapatkan dengan mengikuti urutan *node* sesuai dengan rancangan. Untuk tata cara menulis yang sesuai dengan urutan *node* mendapatkan nilai satu point dan yang salah bernilai nol point. Perolehan jumlah skor pada point menulis didapatkan dari nilai normalisasi berdasarkan total urutan *node* yang benar dibagi dengan jumlah *node* dikali seratus.

4.3.4.8 Perhitungan *Fuzzy*

Perhitungan dengan point *puzzle* atau jumlah benar ketika pengguna menyusun *puzzle* sebanyak 5. Untuk point menulis atau nilai pada saat pengguna menulis huruf yang dimunculkan sebanyak 98. sedangkan waktu 25 detik. Sebelum dilakukan inferensi perlu dicari terlebih dahulu derajat keanggotaan nilai tiap variabel dalam setiap himpunan.

- Point *puzzle* (5)

Point *puzzle* atau jumlah benar ketika menyusun *puzzle* sebanyak 5 berada pada area sedang, maka dapat dihitung menggunakan persamaan 4.2 dan 4.3.

Banyak : 0

Sedang : $(5 - 3) / 2 = 1$

Sedikit : 0

- Point menulis (98)

Nilai dari point menulis ini didapatkan dari nilai normalisasi jumlah *input* atau berapa banyak *node* yang diselesaikan oleh pengguna. Pada point menulis ini menggunakan nilai normalisasi yang akan menjadi nilai inputan pada perhitungan *Fuzzy*. Nilai normalisasi = (jumlah *input*) / (panjang *node* huruf) * 100.

Misal huruf A *Hiragana*, terdapat 49 *node*. Pengguna telah menyelesaikan 48 *node*. Sehingga nilai normalisasi = $(48) / (49) * 100 = 98$. Point 98 berada pada area tinggi, maka diitung menggunakan persamaan 4.6.

Tinggi : $(98 - 70) / 30 = 0,93$

Sedang : 0

Rendah : 0

- Waktu (25)

Nilai dari waktu didapatkan dari sisa waktu ketika pengguna menyelesaikan permainan menyusun *puzzle*. Waktu 25 detik berada pada area lambat, maka dihitung menggunakan persamaan 4.7 dan 4.8.

Cepat : 0

Sedang : $(25-25) / (25) = 0$

Lambat : $(30-25) / 30 = 0,16$

Setelah derajat keanggotaan diketahui, kemudian masuk kedalam *fuzzy rule*. *Fuzzy rule* diproses dan dicari yang cocok sesuai dengan derajat keanggotaan. *Fuzzy rule* secara lengkap disajikan dalam Tabel 4.8. Selanjutnya, nilai tiap variabel diambil nilai minimumnya dari setiap pencocokan *rule*.

R1. IF Point *Puzzle* SEDIKIT AND Point Menulis RENDAH AND Waktu LAMBAT THEN KURANG

$\alpha_{predikat_1} = \text{MIN} (0; 0 ; 0,16)$

= 0

R4. IF Point *Puzzle* SEDIKIT AND Point Menulis SEDANG AND Waktu LAMBAT THEN KURANG

$$\alpha_{\text{predikat}_2} = \text{MIN} (0; 0 ; 0,16)$$

$$= 0$$

R7. IF Point *Puzzle* SEDIKIT AND Point Menulis TINGGI AND Waktu LAMBAT THEN BAIK

$$\alpha_{\text{predikat}_3} = \text{MIN} (0; 0,93 ; 0,16)$$

$$= 0$$

R8. IF Point *Puzzle* SEDIKIT AND Point Menulis TINGGI AND Waktu SEDANG THEN BAIK

$$\alpha_{\text{predikat}_4} = \text{MIN} (0; 0,93 ; 0)$$

$$= 0$$

R9. IF Point *Puzzle* SEDIKIT AND Point Menulis TINGGI AND Waktu CEPAT THEN BAIK

$$\alpha_{\text{predikat}_5} = \text{MIN} (0; 0,93 ; 0)$$

$$= 0$$

R10. IF Point *Puzzle* SEDANG AND Point Menulis RENDAH AND Waktu LAMBAT THEN KURANG

$$\alpha_{\text{predikat}_6} = \text{MIN} (1; 0 ; 0,16)$$

$$= 0$$

R11. IF Point *Puzzle* SEDANG AND Point Menulis RENDAH AND Waktu SEDANG THEN CUKUP

$$\alpha_{\text{predikat}_7} = \text{MIN} (1; 0 ; 0)$$

$$= 0$$

R12. IF Point *Puzzle* SEDANG AND Point Menulis RENDAH AND Waktu CEPAT THEN CUKUP

$$\alpha_{\text{predikat}_8} = \text{MIN} (1; 0 ; 0)$$

$$= 0$$

R13. IF Point *Puzzle* SEDANG AND Point Menulis SEDANG AND Waktu LAMBAT THEN CUKUP

$$\alpha_{\text{predikat}_9} = \text{MIN} (1; 0 ; 0,16)$$

$$= 0$$

R14. IF Point *Puzzle* SEDANG AND Point Menulis SEDANG AND Waktu SEDANG THEN BAIK

$$\alpha_{\text{predikat}_{10}} = \text{MIN} (1; 0 ; 0)$$

$$= 0$$

R15. IF Point *Puzzle* SEDANG AND Point Menulis SEDANG AND Waktu CEPAT THEN BAIK

$$\alpha_{\text{predikat}_{11}} = \text{MIN} (1; 0 ; 0)$$

$$= 0$$

R16. IF Point *Puzzle* SEDANG AND Point Menulis TINGGI AND Waktu LAMBAT THEN CUKUP

$$\alpha_{\text{predikat}_{12}} = \text{MIN} (1; 0,93 ; 0,16)$$

$$= 0,16$$

R17. IF Point *Puzzle* SEDANG AND Point Menulis TINGGI AND Waktu SEDANG THEN BAIK

$$\alpha_{\text{predikat}_{13}} = \text{MIN} (1; 0,93 ; 0)$$

$$= 0$$

R18. IF Point *Puzzle* SEDANG AND Point Menulis TINGGI AND Waktu CEPAT THEN BAIK

$$\alpha_{\text{predikat}_{14}} = \text{MIN} (1; 0,93 ; 0)$$

$$= 0$$

R19. IF Point *Puzzle* BANYAK AND Point Menulis RENDAH AND Waktu LAMBAT THEN KURANG

$$\alpha_{\text{predikat}_{15}} = \text{MIN} (0; 0 ; 0,16)$$

$$= 0$$

R22. IF Point *Puzzle* BANYAK AND Point Menulis SEDANG AND Waktu LAMBAT THEN CUKUP

$$\alpha_{\text{predikat}_{16}} = \text{MIN} (0; 0 ; 0,16)$$

$$= 0$$

R25. IF Point *Puzzle* BANYAK AND Point Menulis TINGGI AND Waktu LAMBAT THEN BAIK

$$\alpha_{\text{predikat}_{17}} = \text{MIN} (0; 0,93 ; 0,16)$$

$$= 0$$

R26. IF Point *Puzzle* BANYAK AND Point Menulis TINGGI AND Waktu SEDANG THEN BAIK

$$\alpha_{\text{predikat}_{18}} = \text{MIN} (0; 0,93 ; 0)$$

$$= 0$$

R27. IF Point *Puzzle* BANYAK AND Point Menulis TINGGI AND Waktu CEPAT THEN BAIK

$$\alpha_{\text{predikat}_{19}} = \text{MIN} (0; 0,93 ; 0)$$

$$= 0$$

Kemudian mencocokkan hasil dari nilai minimal tiap variabel dengan komposisi aturan output yang telah dibuat.

Aturan pertama (R1)

$$\mu_{\text{kurang}}(Z) = \left(\frac{40-Z}{40} \right) = 0; Z = 40 - (40 \times 0) = 40$$

Aturan kedua (R4)

$$\mu_{\text{kurang}}(Z) = \left(\frac{40-Z}{40} \right) = 0; Z = 40 - (40 \times 0) = 40$$

Aturan ketiga (R7)

$$\mu_{\text{baik}}(Z) = \left(\frac{Z-70}{30} \right) = 0; Z = 70 + (30 \times 0) = 70$$

Aturan keempat (R8)

$$\mu_{\text{baik}}(Z) = \left(\frac{Z-70}{30} \right) = 0; Z = 70 + (30 \times 0) = 70$$

Aturan kelima (R9)

$$\mu_{\text{baik}}(Z) = \left(\frac{Z-70}{30} \right) = 0; Z = 70 + (30 \times 0) = 70$$

Aturan keenam (R10)

$$\mu_{\text{kurang}}(Z) = \left(\frac{40-Z}{40} \right) = 0; Z = 40 - (40 \times 0) = 40$$

Aturan ketujuh (R11)

$$\mu_{\text{cukup}}(Z) = \left(\frac{75-Z}{20} \right) = 0; Z = 75 - (20 \times 0) = 75$$

Aturan kedelapan (R12)

$$\mu_{\text{cukup}}(Z) = \left(\frac{75-Z}{20} \right) = 0; Z = 75 - (20 \times 0) = 75$$

Aturan kesembilan (R13)

$$\mu \text{ cukup } (Z) = \left(\frac{75-Z}{20} \right) = 0; Z = 75 - (20 \times 0) = 75$$

Aturan sepuluh (R14)

$$\mu \text{ baik } (Z) = \left(\frac{Z-70}{30} \right) = 0; Z = 70 + (30 \times 0) = 70$$

Aturan sebelas (R15)

$$\mu \text{ baik } (Z) = \left(\frac{Z-70}{30} \right) = 0; Z = 70 + (30 \times 0) = 70$$

Aturan duabelas (R16)

$$\mu \text{ cukup } (Z) = \left(\frac{75-Z}{20} \right) = 0,16; Z = 75 - (20 \times 0,16) = 71,8$$

Aturan tigabelas (R17)

$$\mu \text{ baik } (Z) = \left(\frac{Z-70}{30} \right) = 0; Z = 70 + (30 \times 0) = 70$$

Aturan empatbelas (R18)

$$\mu \text{ baik } (Z) = \left(\frac{Z-70}{30} \right) = 0; Z = 70 + (30 \times 0) = 70$$

Aturan limabelas (R19)

$$\mu \text{ kurang } (Z) = \left(\frac{40-Z}{40} \right) = 0; Z = 40 - (40 \times 0) = 40$$

Aturan enambelas (R22)

$$\mu \text{ cukup } (Z) = \left(\frac{75-Z}{20} \right) = 0,16; Z = 75 - (20 \times 0,16) = 71,8$$

Aturan tujuhbelas (R25)

$$\mu \text{ baik } (Z) = \left(\frac{Z-70}{30} \right) = 0; Z = 70 + (30 \times 0) = 70$$

Aturan delapanbelas (R26)

$$\mu \text{ baik } (Z) = \left(\frac{Z-70}{30} \right) = 0; Z = 70 + (30 \times 0) = 70$$

Aturan sembilanbelas (R27)

$$\mu \text{ baik } (Z) = \left(\frac{Z-70}{30} \right) = 0; Z = 70 + (30 \times 0) = 70$$

Setelah itu proses pengambilan keputusan *fuzzy* menggunakan perhitungan rata-rata (*average*).

$$Z = \sum \frac{\mu(z).z}{\mu(z)}$$

$$\begin{aligned} Z &= (0 \times 40) + (0 \times 40) + (0 \times 70) + (0 \times 70) + (0 \times 70) + (0 \times 40) + (0 \times 75) + (0 \times \\ &75) + (0 \times 75) + (0 \times 70) + (0 \times 70) + (0,16 \times 71,8) + (0 \times 70) + (0 \times 70) + (0 \times 40) \\ &+ (0 \times 75) + (0 \times 70) + (0 \times 70) + (0 \times 70) / (0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\ &+ 0 + 0,16 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 71,8 \approx 72 \text{ (Cukup)} \end{aligned}$$

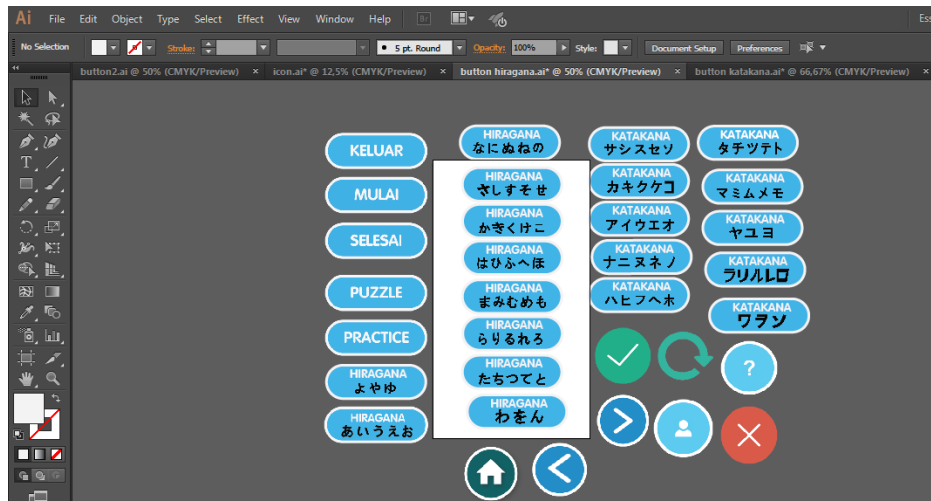
BAB V. IMPLEMENTASI

Setelah melakukan tahapan analisis dan perancangan aplikasi, maka pada bab ini akan dibahas tentang proses implementasi yang merupakan fase pengumpulan bahan (*material collecting*) dan pembuatan (*assembly*) pada metode perancangan *Multimedia Development Life Cycle*. Bagian pada bab ini meliputi implementasi aplikasi pembelajaran huruf Jepang *Hiragana* dan *Katakana*. Dalam aplikasi pembelajaran huruf Jepang *Hiragana* dan *Katakana* ini terdapat beberapa fitur proses pembelajaran, diantaranya materi pembelajaran sebagai informasi pengantar dari proses pembelajaran. Kemudian terdapat permainan menyusun *puzzle* dan latihan cara menulis huruf *Hiragana* dan *Katakana*.

5.1 Pengumpulan Bahan (*Material Collecting*) Objek dan Karakter

Implementasi objek dan karakter terdiri atas pembuatan objek dan karakter. Objek dan karakter pada aplikasi ini dibuat dengan menggunakan *software* Adobe Illustrator CC 2014. Adapun pembuatan *asset* yang digunakan untuk melengkapi fitur aplikasi dan karakter yang digunakan yaitu sebagai berikut.

Proses pembuatan *asset* desain tombol dan *icon* untuk melengkapi aplikasi digambarkan pada gambar sebagai berikut.



Gambar 5.1 Desain tombol dan *icon*

Proses pembuatan desain *background* aplikasi digambarkan pada gambar sebagai berikut.



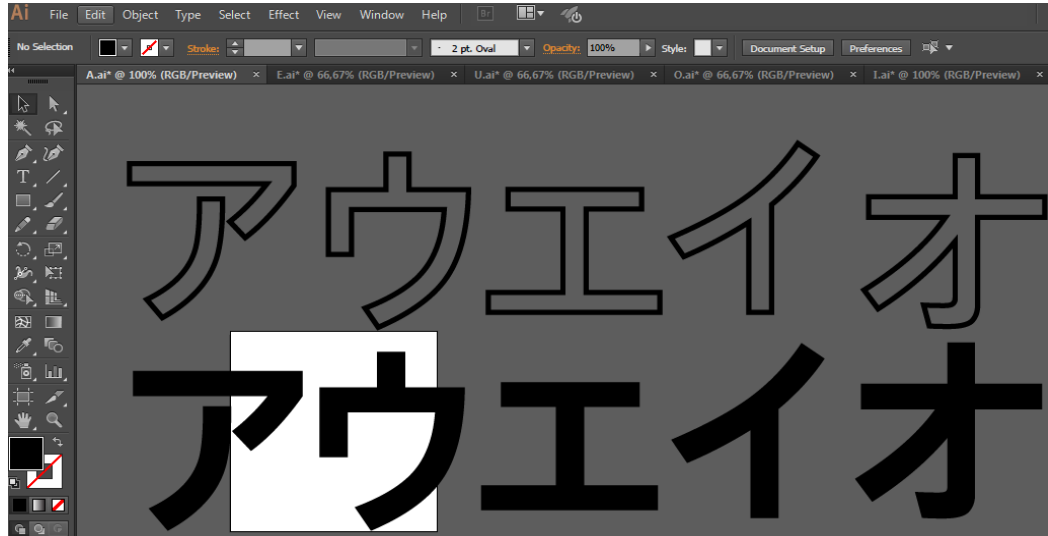
Gambar 5.2 Desain *background* aplikasi

Proses pembuatan objek huruf Jepang *Hiragana* digambarkan pada gambar sebagai berikut. Proses pembuatan huruf Jepang *Hiragana* menggunakan Adobe Illustrator dengan memakai *pen tool*.



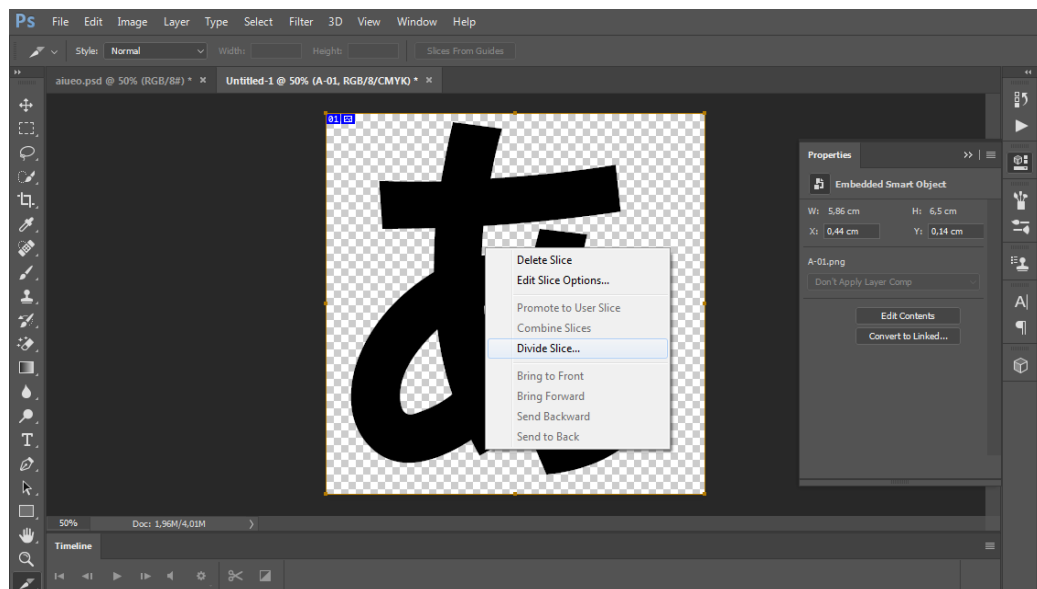
Gambar 5.3 Desain pembuatan huruf Jepang *Hiragana*

Proses pembuatan objek huruf Jepang *Hiragana* digambarkan pada gambar sebagai berikut. Proses pembuatan huruf Jepang *Hiragana* menggunakan Adobe Illustrator dengan memakai *pen tool*.



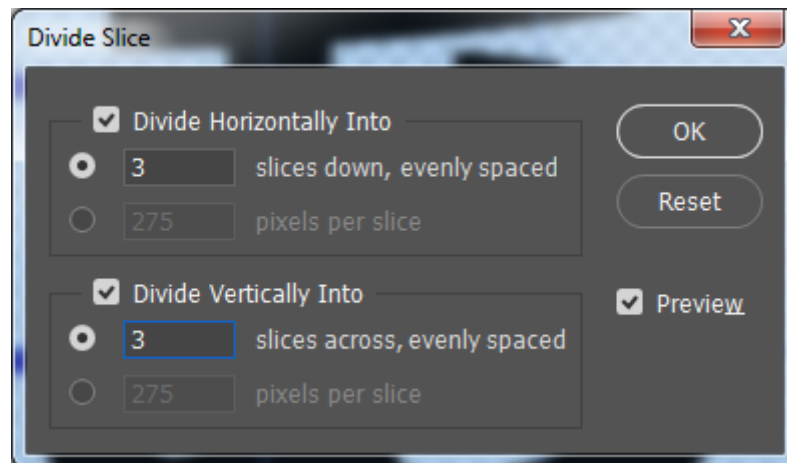
Gambar 5.4 Desain pembuatan huruf Jepang *Katakana*

Proses pemotongan gambar menggunakan Adobe Photoshop dengan *slice tool*. Kemudian memilih *Divide Slice* untuk memotong gambar menjadi 9 gambar dengan mengklik kanan pada objek.



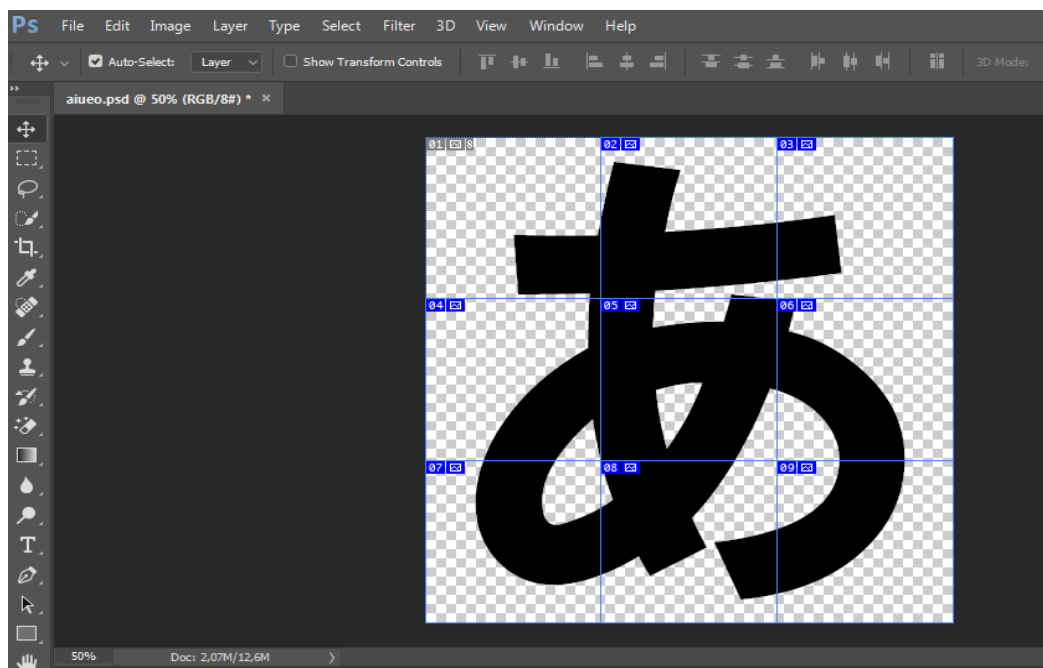
Gambar 5.5 Proses pemotongan gambar

Kemudian akan muncul kotak dialog dibawah ini. *Divide Horizontally Into* untuk memotong atau membagi gambar secara mendatar. Untuk memotong atau membagi gambar menjadi 9 dengan mengisi nilai angka 3 untuk *Divide Horizontally Into* untuk memotong gambar ke bawah, mengisi nilai angka 3 untuk *Divide Vertically Into* untuk memotong gambar ke samping.



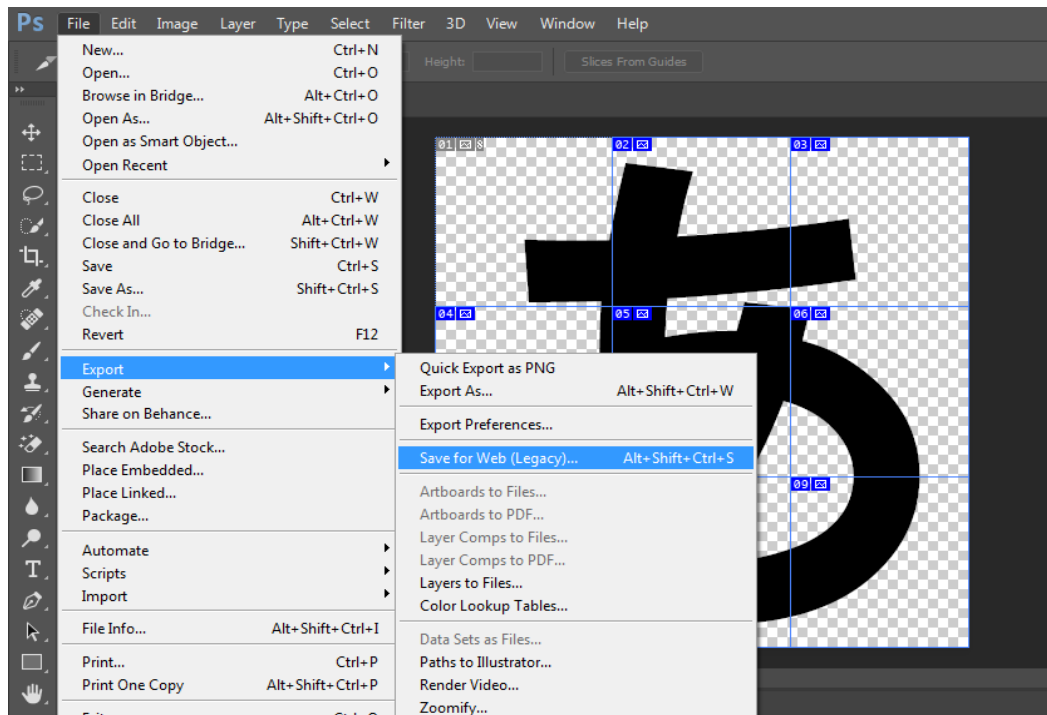
Gambar 5.6 *Divide Slide*

Proses pemotongan gambar menjadi 9 potongan gambar yang akan digunakan untuk menyusun *puzzle* digambarkan pada gambar sebagai berikut.



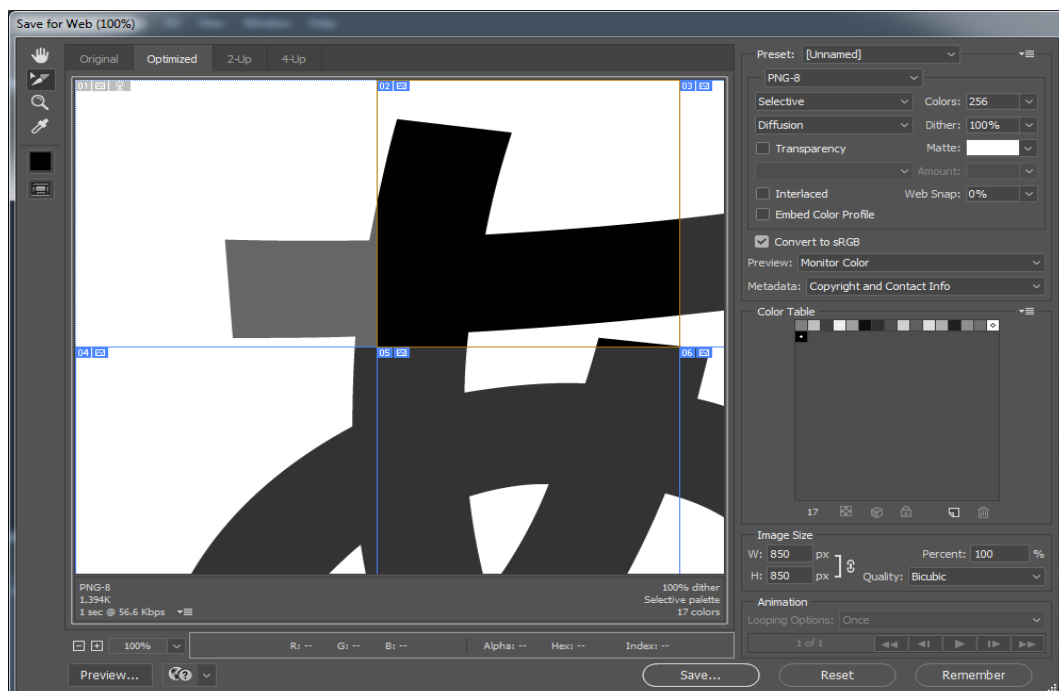
Gambar 5.7 Proses setelah dilakukan *Divide Slice*

Kemudian simpan potongan gambar dengan cara memilih File -> Export -> Save for Web



Gambar 5.8 Proses simpan potongan gambar

Setelah itu akan muncul tampilan seperti gambar dibawah ini. Menekan tombol Save untuk menyimpan potongan gambar.



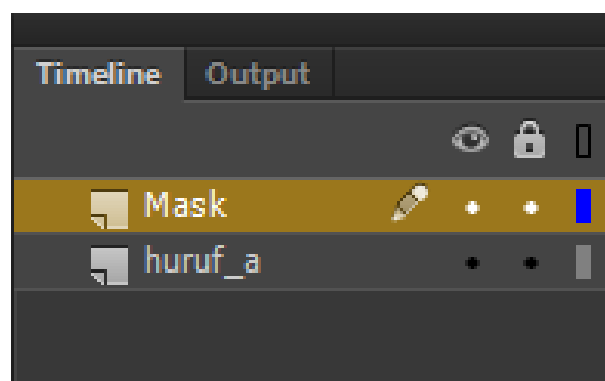
Gambar 5.9 Menyimpan potongan gambar

Proses pembuatan animasi huruf Jepang digambarkan pada gambar sebagai berikut. Proses pembuatan animasi huruf Jepang menggunakan Adobe Flash. Animasi ini dibuat dengan proses *Masking*. Proses *masking* adalah memunculkan object yang ingin kita lihat dan menyembunyikan bagian yang ingin disembunyikan dengan menggunakan *masking*. Pertama proses *masking* dilakukan dengan cara memasukkan objek huruf ke dalam canvas.



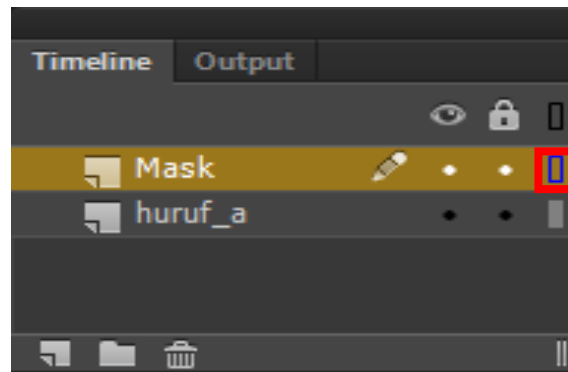
Gambar 5.10 Memasukkan objek ke dalam canvas

Kemudian mengubah *layer* 1 menjadi “huruf_a”. Lalu menambahkan *layer* baru dan memberi nama “Mask” untuk melakukan *Masking*.



Gambar 5.11 Proses tambah layer pada *masking*

Setelah itu menghilangkan *outline* pada objek yang akan di *masking* dengan cara menekan *outlines* pada layer “Mask”.



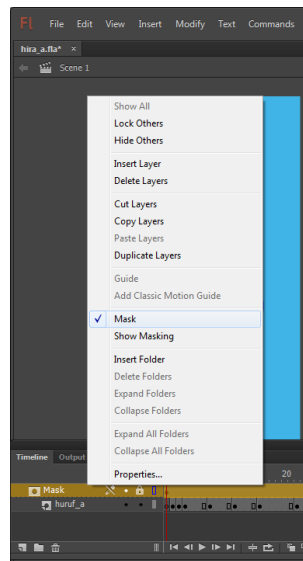
Gambar 5.12 Proses menghilangkan *Outlines*

Dibawah ini merupakan hasil dari menghilangkan *outlines* pada objek huruf Jepang.



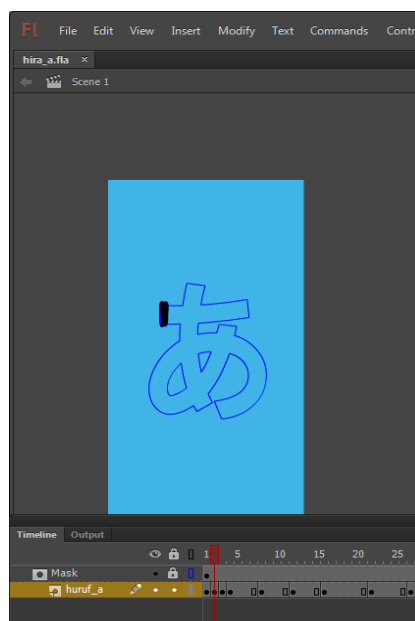
Gambar 5.13 Hasil dari menghilangkan *outlines* pada objek huruf Jepang

Kemudian mengklik kanan pada *layer* “Mask” lalu memilih “Mask” untuk melakukan proses *masking*.



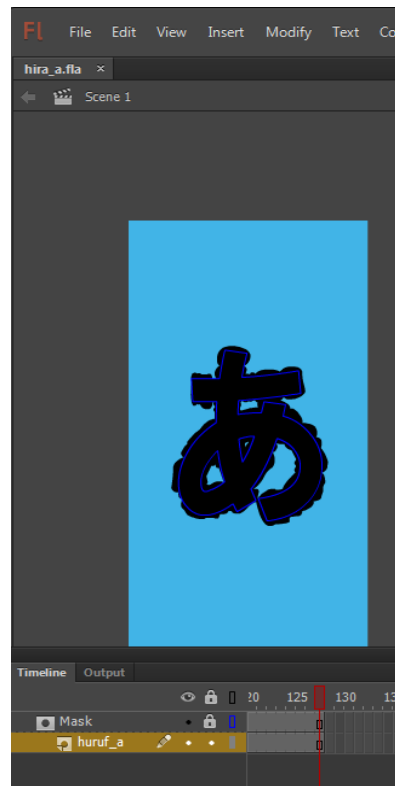
Gambar 5.14 Proses memilih Mask

Untuk melakukan animasi *masking* dilakukan dengan cara mengklik kanan -> *insert keyframe* pada layer “huruf_a”. Kemudian mengklik kanan-> *insert frame* pada layer “Mask”. Memilih *rectangle tool* atau *brush* untuk memberi *outline* pada objek huruf Jepang seperti gambar sebagai berikut. Setelah itu terapkan pada setiap *frame* sesuai dengan cara penulisan huruf Jepang.



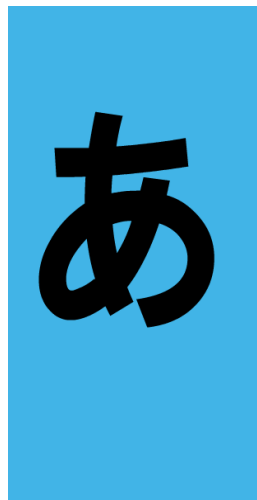
Gambar 5.15 Proses Mask

Setelah melakukan proses animasi *masking*, digambarkan pada gambar 5.16 sebagai berikut yang merupakan hasil dari proses animasi *masking*



Gambar 5.16 Proses animasi *masking*

Setelah melakukan animasi *masking* seperti langkah-langkah sebelumnya, menekan *Ctrl + Enter* untuk melihat hasil dari animasi yang telah dilakukan. Gambar dibawah ini merupakan hasil animasi yang telah dibuat.



Gambar 5.17 Hasil proses animasi *Masking*

5.2 Pembuatan (*Assembly*) Implementasi Antar Muka

Implementasi antar muka meliputi halaman pada setiap menu dan tampilan pada *user*.

5.2.1 Halaman Menu Utama

Gambar 5.18 merupakan tampilan menu utama dari aplikasi pembelajaran huruf *Hiragana* dan *Katakana*. Terdapat empat tombol yang terdiri dari Mulai untuk menuju halaman materi pembelajaran *Hiragana* atau *Katakana*. Tombol Keluar untuk menuju keluar dari aplikasi. Tombol Bantuan berfungsi menuju ke halaman bantuan yang berisi petunjuk cara pengoperasian aplikasi dan juga fungsi tombol. Tombol Informasi berisi informasi tentang aplikasi.

Tampilan dibuat dengan penggabungan objek yang telah dibuat menggunakan Unity dan pengaktifan tombol di dalamnya dilakukan dengan memanggil *method* yang diperlukan agar tombol tersebut dapat berfungsi dengan baik.



Gambar 5.18 Tampilan Menu Utama

5.2.2 Tampilan Menu Pembelajaran

Pada gambar 5.19 menunjukkan tampilan antar muka dari menu pembelajaran huruf Jepang *Hiragana* atau *Katakana*. Tombol *Hiragana* akan menuju ke halaman pembelajaran huruf *Hiragana*. Tombol *Katakana* akan menuju ke halaman pembelajaran huruf *Katakana*.



Gambar 5.19 Tampilan Menu Pembelajaran

5.2.3 Tampilan Menu Materi Pembelajaran *Hiragana*

Pada gambar 5.20 merupakan halaman daftar menu dari materi pembelajaran huruf Jepang *Hiragana*.



Gambar 5.20 Tampilan Menu Materi Pembelajaran *Hiragana*

5.2.4 Tampilan Menu Materi Pembelajaran *Katakana*

Pada gambar 5.21 merupakan halaman daftar menu dari materi pembelajaran huruf Jepang *Katakana*.



Gambar 5.21 Tampilan Menu Pembelajaran *Katakana*

5.2.5 Tampilan Menu Putar Video Pembelajaran

Pada gambar 5.22 terdapat tombol Putar Video untuk ilustrasi cara menulis huruf Jepang serta pengguna dapat melihat bentuk dari huruf Jepang. Ketika tombol Putar Video di tekan maka akan muncul suara yang dikeluarkan sesuai dengan huruf yang ditampilkan.



Gambar 5.22 Tampilan Menu Putar Video Pembelajaran

5.2.6 Tampilan Ilustrasi cara menulis huruf Jepang

Pada gambar 5.23 pengguna dapat melihat ilustrasi cara menulis huruf Jepang yang ditampilkan. Kemudian pengguna dapat melanjutkan ke halaman permainan *puzzle* dengan menekan tombol Puzzle. Ilustrasi cara menulis huruf Jepang ini harus dilihat dengan seksama bagaimana cara menulis huruf Jepang yang ditampilkan karena setelah permainan *puzzle* akan muncul menu Latihan untuk menulis huruf Jepang yang ditampilkan.



Gambar 5.23 Ilustrasi cara menulis huruf Jepang

5.2.7 Tampilan Menu Permainan *Puzzle*

Pada gambar 5.24 pengguna akan menyusun *puzzle* dengan cara menarik potongan gambar ke tempat yang telah disediakan. Selain itu terdapat juga waktu yang merupakan salah satu dari nilai input untuk menghitung nilai skor di akhir permainan.



Gambar 5.24 Tampilan permainan *puzzle*

5.2.8 Tampilan Menu *Puzzle* yang telah Diselesaikan

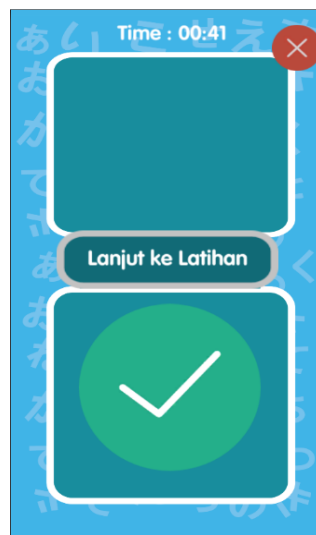
Pada gambar 5.25 pengguna telah menyusun potongan gambar hingga membentuk gambar huruf Jepang yang ditampilkan.



Gambar 5.25 Tampilan *Puzzle* yang telah diselesaikan

5.2.9 Tampilan Menu Lanjut ke Menu Latihan

Pada gambar 5.26 merupakan tampilan setelah pengguna menyelesaikan permainan puzzle apabila potongan gambar yang disusun tepat.



Gambar 5.26 Tampilan lanjut ke menu Latihan

5.2.10 Tampilan Menu Latihan

Pada gambar 5.27 pengguna akan melakukan latihan untuk menulis huruf Jepang yang di tampilkan.



Gambar 5.27 Tampilan Menu Latihan

5.2.11 Tampilan Latihan Menulis

Pada gambar 5.28 merupakan tampilan latihan cara menulis huruf Jepang. Pengguna dapat menulis huruf Jepang yang ditampilkan dengan menulis sesuai urutan huruf hingga tepat. Apabila pengguna tidak tepat dalam menulis urutan huruf maka pengguna akan mengulang pada menu Latihan. Pada menu ini Terdapat tombol “Selesai” untuk menuju halaman pengecekan latihan menulis.



Gambar 5.28 Tampilan Latihan Menulis

5.2.12 Tampilan Pengecekan Latihan Menulis

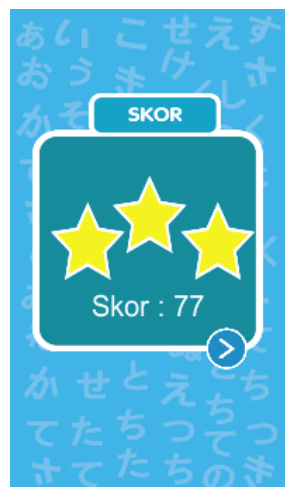
Pada gambar 5.29 merupakan tampilan pengecekan cara penulisan. Setelah pengguna melakukan latihan menulis dan menekan tombol “Selesai” pengguna akan ditunjukkan pada halaman pengecekan latihan menulis. Halaman ini berfungsi untuk mengetahui dimana letak kesalahan penulisan yang ditunjukkan dengan warna gelap. Kemudian apabila pengguna menekan tombol lanjut maka



Gambar 5. 29 Tampilan Pengecekan Latihan Menulis

5.2.13 Tampilan Menu Skor

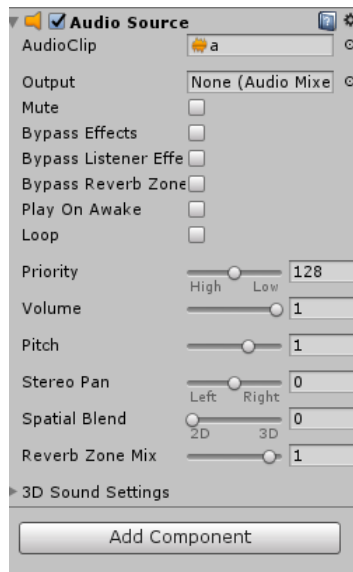
Tampilan skor bintang didapat dari implementasi metode *fuzzy tsukamoto* yang menggunakan 3 variabel yaitu point puzzle, point menulis dan waktu dengan *range* berbeda.



Gambar 5. 30 Tampilan Menu Skor

5.3 Implementasi Audio

Dalam aplikasi ini, file suara huruf Jepang menggunakan format .mp3 didapatkan dari daftar suku kata bahasa Jepang yang dikeluarkan oleh lembaga penyiaran Jepang (NHK). File suara huruf Jepang ini digunakan dalam setiap bagian huruf Jepang yang ditampilkan ketika pengguna menekan tombol “Putar Video”. [14]



Gambar 5.31 Implementasi Audio

5.4 Implementasi Proses

Pada implementasi proses, dijalankan fungsi fungsi penting aplikasi, seperti *script* yang ada pada aplikasi.

5.4.1 Proses Pengacakan Pada Permainan *Puzzle*

Pengacakan pada permainan *puzzle* ini menggunakan metode *Fisher-Yates Shuffle* untuk pengacakan potongan gambar yang akan muncul secara acak. Dibawah ini merupakan *source code* metode *Fisher Yates-Shuffle*.

```
static void FisherYates_Shuffle<T>(T[] array)
{
    int n = array.Length;
    for (int i = 0; i < n; i++)
    {
        float nextValue =
        UnityEngine.Random.Range(0.0f, 1.0f);
        int r = i + (int)(nextValue * (n - i));
        T t = array[r];
        array[r] = array[i];
        array[i] = t; } }
```

5.4.2 Proses Nilai *Score*

Proses nilai *score* menggunakan metode *fuzzy tsukamoto* dengan 3 variabel *input* yaitu ketepatan puzzle, ketepatan menulis huruf, dan waktu yang memiliki 3 himpunan keanggotaan berbeda dengan *range* yang berbeda serta 1 *output* yaitu *score*. Dibawah ini merupakan *source code* dari point puzzle.

```
public float NormMinPointPuzzle(float x){
    float result = 0f;

    if(x >= 4f){
        result = 0f;
    } else if(x < 4f){
        result = (4f-x)/4f;
    } else if(x <= 0f){
        result = 1f;
    }

    return result;
}

public float NormMiddlePointPuzzle(float x){
    float result = 0f;

    if(x<=3f){
        result = 0f;
    } else if(x < 5f){
        result = (x-3f)/2f;
    } else if(x < 7f){
        result = (7f-x)/2f;
    }

    return result;
}

public float NormHighPointPuzzle(float x){
    float result = 0f;

    if(x<=6f){
        result = 0f;
    } else if(x < 9f){
        result = (x-6f)/3f;
    } else if(x >= 9f){
        result = 1f;
    }

    return result;
}
```

Dibawah ini merupakan implementasi *source code* dari variabel point menulis.

```
public float NormMinPointPuzzle(float x){
    float result = 0f;

    if(x >= 4f){
        result = 0f;
    } else if(x < 4f){
        result = (4f-x)/4f;
    } else if(x <= 0f){
        result = 1f;
    }
    return result;
}

public float NormMiddlePointPuzzle(float x){
    float result = 0f;

    if(x<=3f){
        result = 0f;
    } else if(x < 5f){
        result = (x-3f)/2f;
    } else if(x < 7f){
        result = (7f-x)/2f;
    }

    return result;
}

public float NormHighPointPuzzle(float x){
    float result = 0f;

    if(x<=6f){
        result = 0f;
    } else if(x < 9f){
        result = (x-6f)/3f;
    } else if(x >= 9f){
        result = 1f;
    }

    return result;
}
```

Dibawah ini merupakan implementasi *source code* dari variabel waktu.

```
public float NormMinTime(float x){
    float result = 0f;

    if(x >= 30f){
        result = 0f;
    } else if(x < 30f){
        result = (30f-x)/30f;
    } else if(x <= 0f){
        result = 1f;
    }
    return result;
}

public float NormMiddleTime(float x){
    float result = 0f;

    if(x <= 25f){
        result = 0f;
    } else if(x < 35f){
        result = (x-25f)/10f;
    } else if(x < 45f){
        result = (45f-x)/10f;
    }

    return result;
}

public float NormHighTime(float x){
    float result = 0f;

    if(x<=40f){
        result = 0f;
    } else if(x < 60f){
        result = (x-40f)/20f;
    } else if(x >= 60f){
        result = 1f;
    }
    return result;
}
```

Tahap Mesin *Inferensi*, pada mesin *inferensi* menerapkan fungsi *MIN* untuk setiap aturan pada aplikasi.

```
private void MinPredictCalc(){
    for (int i = 0; i < rulesCount; i++) {
        float result = 0f;

        predicateTemps [0] = timeRuleResults [i];
        predicateTemps [1] = pointPuzzleRuleResults [i];
        predicateTemps [2] = pointWritingRuleResults [i];
        predicateTemps [3] = starRuleResults [i];

        minPredicates [i] = predicateTemps.Min ();
    }
}
```

Dibawah ini merupakan *source code* mencari nilai Z.

```
private void ZCalc(){
    for (int i = 0; i < 4; i++) {
        z [i] = Z4Calc (minPredicates [i]);
    }

    for (int i = 4; i < 6; i++) {
        z [i] = Z75Calc (minPredicates [i]);
    }

    for (int i = 6; i < 9; i++) {
        z [i] = Z70Calc (minPredicates [i]);
    }

    for (int i = 9; i < 10; i++) {
        z [i] = Z4Calc (minPredicates [i]);
    }

    for (int i = 10; i < 13; i++) {
        z [i] = Z75Calc (minPredicates [i]);
    }
    for (int i = 13; i < 15; i++) {
        z [i] = Z70Calc (minPredicates [i]);
    }

    for (int i = 15; i < 16; i++) {
        z [i] = Z75Calc (minPredicates [i]);
    }

    for (int i = 16; i < 18; i++) {
        z [i] = Z70Calc (minPredicates [i]);
    }

    for (int i = 18; i < 19; i++) {
        z [i] = Z4Calc (minPredicates [i]);
    }
}
```

Dibawah ini merupakan implementasi *source code* dari *defuzzifikasi*.

```
private void Defuzzyfication(){
    for (int i = 0; i < rulesCount; i++) {
        defuzzy [i] = z [i] * minPredicates [i];
    }
    result = defuzzy.Sum () / minPredicates.Sum ();
    if (result <= 40f) {
        star = Actions.Bintang1;
    } else if (result <= 75f) {
        star = Actions.Bintang2;
    } else if (result <= 100f) {
        star = Actions.Bintang3;
    }
}
```

BAB VI. PENGUJIAN DAN PEMBAHASAN

Bab VI berisi tentang *fase* pengujian (*testing*) yang termasuk dalam metode *Multimedia Development Life Cycle*. Tahapan ini dilakukan setelah proses analisa, perancangan dan implementasi telah selesai dilakukan.

6.1 Pengujian Aplikasi (*Testing*)

Pengujian aplikasi dimaksudkan untuk menguji semua elemen-elemen perangkat lunak yang dibuat apakah sudah sesuai dengan perancangan.

6.1.1 Hasil dan Kasus Pengujian Alpha

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian alpha pada aplikasi. Berikut hasil pengujian dari aplikasi pada tabel 6.1.

Tabel 6. 1 Kasus dan pengujian alpha

No	Kasus/diuji	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1.	Tampilan Awal	Membuka aplikasi	Ketika pertama membuka aplikasi, akan muncul tampilan <i>splashscreen</i> lalu berlanjut menampilkan tampilan Menu Utama	Berhasil
2.	Tombol Mulai	Memilih tombol mulai	Ketika memilih tombol menu mulai menampilkan menu materi huruf <i>hiragana</i> dan kata kana	Berhasil
3.	Tombol Informasi Aplikasi	Memilih tombol informasi aplikasi	Ketika memilih tombol informasi aplikasi menampilkan halaman informasi pembuat aplikasi	Berhasil
4.	Tombol Petunjuk	Memilih tombol petunjuk	Ketika memilih tombol informasi petunjuk menampilkan halaman informasi pengoperasian aplikasi	Berhasil
5.	Tombol Keluar	Memilih tombol keluar	Ketika memilih tombol keluar maka akan keluar dan menutup aplikasi	Berhasil
6.	Tombol <i>Hiragana</i>	Memilih tombol <i>hiragana</i>	Ketika memilih tombol <i>hiragana</i> maka menampilkan halaman memilih jenis huruf	Berhasil

7.	Tombol <i>Katakana</i>	Memilih tombol <i>katakana</i>	Ketika memilih tombol <i>katakana</i> maka menampilkan halaman memilih jenis huruf	Berhasil
8.	Tombol <i>Puzzle</i>	Memilih tombol <i>puzzle</i>	Ketika memilih tombol <i>puzzle</i> maka akan menuju halaman permainan <i>puzzle</i>	Berhasil
9.	Tombol <i>Practice</i>	Memilih tombol <i>practice</i>	Ketika memilih tombol <i>practice</i> maka akan menuju halaman permainan praktik menulis	Berhasil
10.	Tombol Lanjutkan Huruf Selanjutnya	Memilih tombol lanjutkan huruf selanjutnya	Ketika memilih tombol lanjutkan huruf selanjutnya maka akan menuju halaman huruf selanjutnya	Berhasil
11.	Tombol Kembali	Memilih tombol kembali	Ketika memilih tombol kembali maka akan menuju halaman sebelumnya	Berhasil
12.	Tombol Selanjutnya	Memilih tombol selanjutnya	Ketika memilih tombol selanjutnya maka akan menuju halaman selanjutnya	Berhasil
13.	Tombol Keluar (X)	Memilih keluar	Ketika memilih tombol keluar maka akan menuju ke halaman utama, mengulang, atau melanjutkan kembali	Berhasil

Berdasarkan hasil pengujian diatas maka dapat ditarik kesimpulan bahwa sistem aplikasi pengembangan *game puzzle* sebagai pembelajaran huruf jepang *hiragana* dan *katakana* menggunakan metode *Fisher-Yates Shuffle* dan *Fuzzy Tsukamoto* berbasis android sudah berjalan seperti yang diharapkan dan secara fungsional sudah dapat menghasilkan keluaran yang diharapkan.

6.1.3 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana pengujian langsung melibatkan para pengguna aplikasi *game* secara langsung dengan menggunakan kuesioner mengenai manfaat yang dirasakan para pengguna terhadap aplikasi yang dibangun.

Uji coba aplikasi pengembangan *game puzzle* sebagai pembelajaran huruf jepang *hiragana* dan *katakana* menggunakan metode *Fisher-Yates Shuffle* dan

Fuzzy Tsukamoto berbasis android dilakukan pada masyarakat umum dengan segmen diatas 16 tahun dengan jumlah responden sebanyak 10 orang. Tabel 6.3 adalah kuesioner yang diisi oleh *user* setelah mencoba aplikasi ini.

Tabel 6. 2 Kuesioner Aplikasi

Keterangan :

SB = Sangat Baik

C= Cukup

SK = Sangat Kurang

B = Baik

K = Kurang

NO	PERNYATAAN	NILAI				
		SB	B	C	K	SK
APLIKASI						
1	Desain tampilan aplikasi (kesesuaian penggunaan warna tampilan, jenis dan ukuran tulisan, bentuk dan letak tombol)					
2	Berfungsinya tombol – tombol					
3	Kesesuaian huruf <i>Hiragana</i> dan <i>Katakana</i>					
4	Proses permainan <i>puzzle</i>					
5	Proses Permainan praktik menulis					
6	Durasi waktu dalam permainan sudah cukup					
PENGGUNA						
1	Penyampaian pengetahuan tentang pembelajaran huruf <i>hiragana</i> dan <i>katakana</i>					
2	Pemahaman terhadap materi yang disampaikan					
3	Anda merasa terbantu dengan aplikasi pembelajaran huruf <i>Hiragana</i> dan <i>Katakana</i>					

6.1.4 Hasil Pengujian Kuesioner

Tabel 6.3 Hasil Pengujian Kuisisioner yang telah dilakukan penyebaran kuisisioner kepada 10 responden siswa-siswi SMA di SMAN 7 Malang dengan responeden berusia 16 tahun ditampilkan pada Tabel 6.4.

Tabel 6.3 Hasil Kuesioner

NO	PERNYATAAN	NILAI				
		SB	B	C	K	SK
APLIKASI						
1	Desain tampilan aplikasi (kesesuaian penggunaan warna tampilan, jenis dan ukuran tulisan, bentuk dan letak tombol)	2	7	1	0	0
2	Berfungsinya tombol – tombol	1	3	3	3	0
3	Kesesuaian huruf <i>Hiragana</i> dan <i>Katakana</i>	4	5	1	0	0
4	Proses permainan <i>puzzle</i>	3	1	5	1	0
5	Proses Permainan praktik menulis	1	0	7	2	0
6	Durasi waktu dalam permainan sudah cukup	1	5	4	0	0
PENGGUNA						
1	Penyampaian pengetahuan tentang pembelajaran huruf <i>hiragana</i> dan <i>katakana</i>	4	4	2	0	0
2	Pemahaman terhadap materi yang disampaikan	5	3	2	0	0
3	Anda merasa terbantu dengan aplikasi pembelajaran huruf <i>Hiragana</i> dan <i>Katakana</i>	5	4	1	0	0

Keterangan :

SB = Sangat Baik

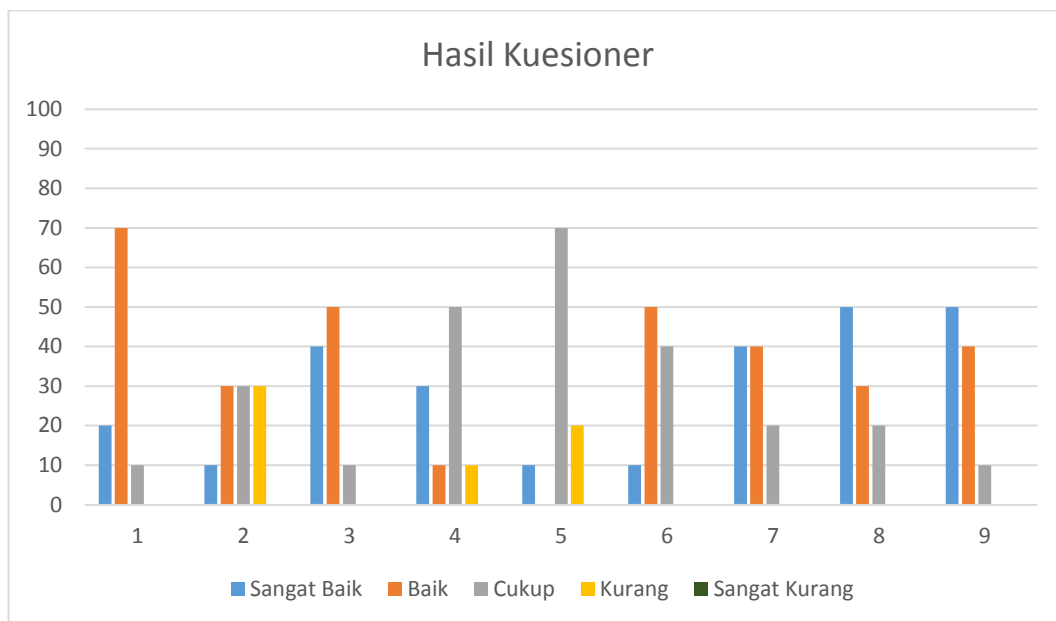
C= Cukup

SK = Sangat Kurang

B = Baik

K = Kurang

Presentasi hasil kuesioner pada Tabel 6.4 ditunjukkan pada gambar Grafik 6.1.



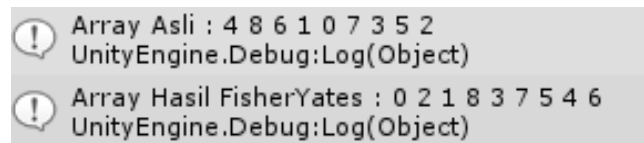
Gambar 6. 1 Grafik Hasil Kuesioner

Grafik 6.1 menunjukkan 20% responden memilih sangat baik, 70% responden memilih baik, dan 10% responden memilih cukup bahwa aplikasi memiliki tampilan yang sesuai. 10% responden memilih sangat baik, 30% responden memilih baik, 30% responden memilih cukup, 30% responden memilih kurang mengenai berfungsinya tombol-tombol. 40% responden memilih sangat baik, 50% responden memilih baik, 10% responden memilih cukup mengenai kesesuaian huruf *Hiragana* dan *Katakana*. 30% responden memilih baik, 50% responden memilih cukup, 10% responden memilih kurang mengenai proses permainan *puzzle*. 10% responden memilih sangat baik, 70% responden memilih cukup, 20% responden memilih kurang mengenai proses permainan praktik menulis. 10% responden memilih sangat setuju, 50% responden memilih baik, 40% responden memilih cukup mengenai durasi waktu dalam permainan. 40% responden memilih sangat baik, 40% responden memilih baik, 20% responden memilih cukup mengenai penyampaian pengetahuan tentang pembelajaran huruf *Hiragana* dan *Katakana*. 50% responden memilih sangat baik, 30% responden memilih baik, 20% responden memilih cukup mengenai pemahaman terhadap materi yang disampaikan. 50% responden memilih sangat baik, 40% responden

memilih baik, 10% responden memilih cukup bahwa aplikasi ini dapat membantu untuk belajar huruf *Hiragana* dan *Katakana*.

6.2 Pengujian Metode *Fisher Yates-Shuffle*

Untuk menguji metode *Fisher Yates-Shuffle* ini berjalan sesuai dengan hasil yang diharapkan, maka dibuatlah sebuah uji coba sederhana pada *game puzzle*, yaitu dengan menampilkan *console* untuk mengetahui proses dalam sistem yang telah dibuat. Pengujian metode dilakukan pada setiap pengacakan potongan gambar yang tersedia sebagai berikut.



Gambar 6. 2 Hasil *console* pengacakan potongan gambar huruf Jepang

6.3 Pengujian Metode Fuzzy Tsukamoto

Untuk menguji metode *Fuzzy Tsukamoto* ini berjalan sesuai dengan hasil yang diharapkan, maka dibuatlah sebuah uji coba sederhana pada nilai skor dengan menampilkan nilai input variabel untuk mengetahui proses dalam sistem.

Input Variabel	
Time	45
Point Puzzle	7
Point Writing	81

Gambar 6. 3 Hasil *inspector* Fuzzy Tsukamoto

Gambar 6.3 menunjukkan tentang *result* atau hasil nilai skor dari setiap variabel *fuzzy tsukamoto* dengan keluaran/output “bintang 3” yang mana hasil nilai skor didapatkan dari nilai tiap-tiap variabel yang ditunjukkan pada gambar 6.2.

Defuzzyfication	
Defuzzy	
Result	77.5
Star	Bintang 3

Gambar 6. 4 Hasil Skor

6.4 Distribusi (*Distribution*)

Pada tahap ini, *project* yang telah selesai diuji akan disimpan dalam bentuk file .apk dengan tujuan dapat didistribusikan dan digunakan oleh masyarakat dengan target usia 16 tahun.

BAB VII. KESIMPULAN DAN SARAN

Pada bab ini membahas mengenai kesimpulan yang diperoleh dengan melihat hasil pengujian aplikasi yang telah dibuat dan saran yang dapat digunakan untuk kepentingan pengembangan selanjutnya.

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan pada aplikasi pengembangan *game puzzle* sebagai pembelajaran huruf Jepang *hiragana* dan *katakana* menggunakan metode *Fisher-Yates Shuffle* dan *Fuzzy Tsukamoto* berbasis android maka dapat diambil kesimpulan sebagai berikut:

1. Penerapan metode *Fisher-Yates Shuffle* yang diterapkan pada pengacakan potongan gambar huruf Jepang di *game puzzle* dapat mengacak potongan gambar huruf Jepang yang muncul tidak dapat terulang kembali di sesi yang sama. Metode *Fuzzy Tsukamoto* yang diterapkan pada nilai skor dengan perhitungan variabel *input* point, point menulis, dan waktu menghasilkan *output* berupa nilai skor dan gambar bintang di akhir permainan. Dengan demikian penggunaan metode *Fisher-Yates Shuffle* dan metode *Fuzzy Tsukamoto* dapat menghasilkan keluaran yang diharapkan.
2. Berdasarkan hasil kuesioner yang disebarkan kepada 10 responden, aplikasi ini dapat memberikan pengetahuan mengenai bentuk dan cara menulis huruf Jepang *Hiragana* dan *Katakana* sebesar 64,4 %, sehingga aplikasi ini dapat digunakan sebagai media pembelajaran huruf Jepang *Hiragana* dan *Katakana*.

7.2 Saran

Penelitian masih dapat dikembangkan menjadi aplikasi yang lebih kompleks dengan memperbanyak pembelajaran huruf *Hiragana* dan *Katakana* Daku on, *Hiragana* dan *Katakana* Handaku on, dan *Hiragana* dan *Katakana* Yo on serta dapat memperbaiki desain dan animasi cara menulis huruf Jepang agar lebih menarik.

DAFTAR PUSTAKA

- [1] Rendra, M. Ridwan. 2013. "*Perancangan Game Puzzle Sebagai Media Pembelajaran Aksara Sunda Untuk Remaja*". Jurnal Komunikasi Visual & Multimedia. Institut Teknologi Bandung.
- [2] Nugraha, Exridores, dkk. 2014. "*Penerapan Algoritma Fisher-Yates Pada Aplikasi The Lost Insect Untuk Pengenalan Jenis Serangga Berbasis Unity 3D*". Laporan Akhir Program Studi Informatika STMIK Global Informatika MDP Palembang.
- [3] Haditama, Imam, dkk.2016."*Implementasi Algoritma Fisher-Yates Dan Fuzzy Tsukamoto Dalam Game Kuis Tebak Nada Sunda Berbasis Android*". Laporan Penelitian Jurusan Teknik Informatika Universitas Islam Negeri Sunan Gunung Djati Bandung.
- [4] Safaat, Nazruddin. 2012. "*Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*". Bandung: Informatika.
- [5] Roedavan, Rickman. 2014. "*Unity Tutorial Game Engine*". Bandung: Informatika Bandung.
- [6] Kamus Bahasa Indonesia. 2003. "*Kamus Bahasa Indonesia Edisi ketiga*". Jakarta Balai Pustaka
- [7] Andang, Ismail. 2009. "*Education Games*". Pilar Media.Yogyakarta.
- [8] Renariah, Dra., M.Hum. 2002."*Artikel dalam Jurnal Sastra Jepang Fakultas Sastra Universitas Kristen Maranatha vol 1 No.2 edisi Februari 2002*". Bandung.
- [9] Riswono, Nonok Eko Wahyu.2016."*3D Adventure Game Untuk Belajar Membaca Al-Qur'an Dengan Menggunakan Metode Fuzzy Tsukamoto*".Skripsi Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- [10] Madcoms. 2013."*Kupas Tuntas Adobe Illustrator*". Andi. Yogyakarta.
- [11] Enterprise, Jubilee. 2010. "*Trik Cepat Menguasai Photoshop CS5*". Jakarta. PT.Elex Media Kompetindo.
- [12] Madcoms. 2008. "*Adobe Flash CS3 untuk Pemula*". Andi. Yogyakarta.

- [13] Kusumadewi, S., dan Purnomo, S. 2010. “*Aplikasi Logika Fuzzy untuk Pendukung Keputusan*”. Graha Ilmu. Yogyakarta.
- [14] NHK (Perusahaan Penyiaran Jepang). (2015). Daftar Suku Kata Bahasa Jepang [Online]. Tersedia : <https://www.nhk.or.jp/lesson/indonesian/syllabary/>