

**IMPLEMENTASI *DECISION TREE* UNTUK MENENTUKAN MUSIK
ADAPTIF BERDASARKAN *PLAYER-CASE* DALAM *GAME* HIJAIYAH**

SKRIPSI

Oleh :
YENI LESTARI
NIM.19650120



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

**IMPLEMENTASI *DECISION TREE* UNTUK MENENTUKAN MUSIK
ADAPTIF BERDASARKAN *PLAYER-CASE* DALAM GAME HIJAIYAH**

SKRIPSI

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

Oleh :
YENI LESTARI
NIM. 19650120

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

HALAMAN PERSETUJUAN

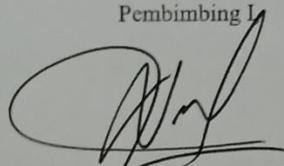
IMPLEMENTASI *DECISION TREE* UNTUK MENENTUKAN MUSIK
ADAPTIF BERDASARKAN *PLAYER-CASE* DALAM *GAME HIJAIYAH*

SKRIPSI

Oleh :
YENI LESTARI
NIM. 19650120

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 09 Juni 2023

Pembimbing I,



Dr. Fesy Nugroho, M.T
NIP. 19710722 201101 1 001

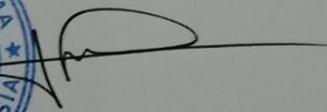
Pembimbing II,



Zainal Abidin, M.Kom
NIP. 19760613 200501 1 004

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




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

HALAMAN PENGESAHAN

IMPLEMENTASI *DECISION TREE* UNTUK MENENTUKAN MUSIK
ADAPTIF BERDASARKAN *PLAYER-CASE* DALAM *GAME HIJAIYAH*

SKRIPSI

Oleh :
YENI LESTARI
NIM. 19650120

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 13 Juni 2023

Susunan Dewan Penguji

Ketua Penguji : Hani Nurhayati, M.T
NIP. 19780625 200801 2 006

Anggota Penguji I : Puspa Miladin Nuraida Safitri A.
Basid, M.Kom
NIP. 19930828 201903 2 018

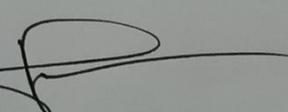
Anggota Penguji II : Dr. Fresy Nugroho, M.T
NIP. 19710722 201101 1 001

Anggota Penguji III : Zainal Abidin, M.Kom
NIP. 19760613 200501 1 004

()
()
()
()

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




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

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Yeni Lestari

NIM : 19650120

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Implementasi *decision tree* untuk menentukan adaptif musik berdasarkan *player-case* dalam *game* hijaiyah

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

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

Malang, 22 Juni 2023
Yang membuat pernyataan,



Yeni Lestari
NIM.19650120

MOTTO

” فَإِنَّ مَعَ الْعُسْرِ يُسْرًا ”

“Maka sesungguhnya bersama kesulitan ada kemudahan” . (Q.S Al-Insyirah:5)

HALAMAN PERSEMBAHAN

Karya ilmiah ini penulis persembahkan kepada kedua orang tua, kerabat terdekat, dosen dan untuk diri saya sendiri, sekaligus kepada seluruh orang yang berperan aktif dalam membantu penulis sehingga mampu menyelesaikan karya ilmiah ini dengan tepat waktu.

KATA PENGANTAR

Assalamualaikum wr wb.

Puji syukur penulis panjatkan kepada kehadiran Allah Swt yang telah melimpahkan nikmat serta karunia-Nya sehingga penulis mampu menyelesaikan penulisan Skripsi yang berjudul Implementasi *Decision tree* untuk Menentukan Musik Adaptif dalam *Game* Hijaiyah ini dengan keadaan sehat wal'afiat dan berhasil menyelesaikannya dengan baik dan tepat waktu.

Terdapat banyak pihak yang terlibat dalam penulisan Skripsi ini baik dalam proses membimbing penulisan dan juga memberikan semangat dan dukungan moril atau materiil. Untuk itu, dalam kesempatan ini, penulis ingin menyampaikan terima kasih kepada:

1. Prof. Dr. H.M. Zainuddin, MA selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Hariani, M.Si selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT. IPM selaku Ketua jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Fresy Nugroho, M.T selaku Dosen Pembimbing 1 yang telah memberikan banyak dukungan dan bimbingan dalam penulisan skripsi ini.
5. Zainal Abidin, M.Kom selaku Dosen Pembimbing 2 yang sudah memberikan banyak koreksi atas kesalahan penulis dan rela meluangkan waktunya.

6. Seluruh Dosen dan Staff Akademika Jurusan Teknik Informatika yang secara tidak langsung telah memberikan banyak bantuan demi penyelesaian skripsi ini.
7. Kedua orang tua saya, alm Bapak Sali dan juga ibu Muntamah yang telah memberikan dukungan terbesar dalam kehidupan penulis sehingga penulis bisa berada ditahap ini. Dan juga untuk kakak saya Edi Santoso dan keluarga kecilnya yang telah membantu penulis untuk tetap semangat. Sekaligus untuk adik-adik ponakan dan tetangga yang senantiasa memberikan dukungan moril kepada penulis.
8. Teman-teman satu angkatan Teknik Informatika 2019 dan juga teman-teman yang mengirimkan banyak semangat dan dukungannya.

Malang, 22 Juni 2023

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vii
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan	5
1.4 Manfaat	5
1.5 Batasan Masalah.....	6
1.6 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA.....	8
2.1 Penelitian Terkait	8
2.2 <i>Game</i>	10
2.3 <i>Game</i> 2D	11
2.4 <i>Game</i> PC	13
2.5 <i>Game Engine</i>	14
2.6 Hijaiyah	16
2.7 Musik	19
2.8 <i>Unity</i>	21
2.9 <i>Decision tree</i>	23
2.9.1 Pengertian <i>Decision tree</i>	23
2.9.2 Karakteristik <i>Decision tree</i>	25
2.9.3 Algoritma Induksi <i>Decision tree</i>	26
2.9.4 Tahapan dalam <i>Decision tree</i> Algoritma C4.5.....	27
2.9.5 Entropy	28
2.9.6 Information gain.....	28
BAB III METODE PENELITIAN	29
3.1 Penjelasan Umum <i>Game</i> 2D Belajar Hijaiyah.....	29
3.2 Finite State Machine	31
3.3 Perancangan <i>Decision tree</i>	33
3.3.1 Menentukan Tujuan	36
3.3.2 Menentukan Variabel.....	36
3.3.3 Data Training	38
3.3.4 Nilai Entropy.....	43
3.3.4.1 Entropy Musik.....	43

3.3.4.2	<i>Entropy</i> dan Nilai <i>Gain</i> Jarak <i>Player-NPC</i>	43
3.3.4.3	<i>Entropy</i> dan Nilai <i>Gain</i> Kecepatan <i>Player(Vp)</i>	44
3.3.4.4	<i>Entropy</i> dan Nilai <i>Gain</i> Jarak <i>Player-Finish</i>	45
3.3.5	Nilai Akurasi.....	48
3.3.6	<i>Cross validation</i> menggunakan K-Fold.....	49
3.3.7	Skenario Pengujian	51
BAB IV HASIL DAN PEMBAHASAN		52
4.1	Implementasi	52
4.1.1	Kebutuhan Perangkat Keras.....	52
4.1.2	Kebutuhan Perangkat Lunak.....	52
4.1.3	Implementasi <i>Decision tree</i>	53
4.1.4	Implementasi <i>Decision tree</i> dalam <i>Script Game</i>	61
4.1.5	Implementasi <i>Game 2D</i> Belajar Hijaiyah.....	66
4.2	Pengujian <i>Audio Game</i>	75
4.3	Integrasi dalam Islam	79
Integrasi Belajar dalam Islam.....		79
BAB V KESIMPULAN DAN SARAN		85
5.1	Kesimpulan	85
5.2	Saran.....	86
DAFTAR PUSTAKA		
LAMPIRAN		

DAFTAR GAMBAR

Gambar 2. 1 Tampilan antarmuka <i>Unity</i>	22
Gambar 2. 2 Layer Data Mining	24
Gambar 2. 3 Struktur <i>Decision tree</i>	26
Gambar 3. 1 <i>Environment</i> siang.....	29
Gambar 3. 2 <i>Environment</i> malam	30
Gambar 3. 3 Finite State Machine <i>Game</i>	32
Gambar 3. 4 Flowchart <i>Decision tree</i> Algoritma C4.5	34
Gambar 3. 5 Flowchart <i>Decision tree</i> dalam <i>game</i>	35
Gambar 3. 6 Fungsi untuk menghitung nilai akurasi	48
Gambar 3. 7 <i>Cross validation</i> K-Fold.....	50
Gambar 3. 8 Skenario Pengujian Sistem.....	51
Gambar 3. 9 Halaman Menu Utama	91
Gambar 3. 10. Halaman Bantuan	91
Gambar 3. 11. Halaman Utama <i>Game</i>	91
Gambar 3. 12 Halaman Informasi.....	92
Gambar 3. 13 Halaman time offer.....	92
Gambar 4. 1 Pseudocode <i>Decision tree</i> algoritma c4.5	54
Gambar 4. 2 Graph tree dari <i>decision tree</i>	57
Gambar 4. 3 Implementasi rules jarak jauh	62
Gambar 4. 4 Implementasi rules jarak sedang	64
Gambar 4. 5 Implementasi rules jarak dekat.....	65
Gambar 4. 6 Halaman Splash.....	66
Gambar 4. 7 Halaman Introduction <i>game</i>	67
Gambar 4. 8 Halaman Menu utama	67
Gambar 4. 9 Halaman Bantuan	68
Gambar 4. 10 Halaman <i>environment</i> siang	69
Gambar 4. 11 <i>Environment</i> malam	69
Gambar 4. 12 Karakter <i>player</i>	70
Gambar 4. 13 Karakter Musuh.....	70
Gambar 4. 14 Huruf hijaiyah	71
Gambar 4. 15 Halaman informasi time offer	71
Gambar 4. 16 Halaman menu <i>game</i>	72
Gambar 4. 17 Baling-baling penghalang	72
Gambar 4. 18 Halaman panel hijaiyah.....	73
Gambar 4. 19 Halaman resume	74
Gambar 4. 20 Halaman panel Winner.....	74
Gambar 4. 21 Halaman panel Lose <i>Game</i>	75

DAFTAR TABEL

Tabel 3. 3 Tabel Variabel Kategoris Jarak <i>player-NPC</i>	37
Tabel 3. 4 Tabel Atribut Kategoris Kecepatan <i>player</i>	37
Tabel 3. 5 Tabel variabel Kategoris Jarak <i>player-finish</i>	37
Tabel 3. 6 Data Training	39
Tabel 3. 7 Tabel Transform Data Training	43
Tabel 3. 8 Pembagian presentase akurasi	49
Tabel 4. 1 Kebutuhan perangkat keras	52
Tabel 4. 2 Kebutuhan perangkat lunak	53
Tabel 4. 3 Tabel Presentase pengujian akurasi <i>decision tree</i>	56
Tabel 4. 4 Tabel rata-rata akurasi perhitungan presentase	59
Tabel 4. 5 Pengujian K Fold Cross Validation	60
Tabel 4. 6 Tabel Detail Musik Adaptif	61
Tabel 4. 7 Tabel Keterangan istilah dalam code	63
Tabel 4. 8 Tabel Data Pengujian sistem	76
Tabel 4. 9 Tabel detail value dan nilai atribut	76
Tabel 4. 10 Tabel kecocokan pengujian sistem dengan rules <i>decision tree</i>	77

ABSTRAK

Lestari, Yeni. 2023. **Implementasi *Decision tree* untuk Menentukan Musik Adaptif Berdasarkan *Player-Case* dalam *Game Hijaiyah***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Fresy Nugroho, M.T. (II) Zainal Abidin, M.Kom.

Kata kunci: Musik Adaptif, *Decision tree*, *Game Hijaiyah*, Algoritma C4.5

Teknologi *game* di zaman ini mengalami perkembangan yang sangat pesat. Hal tersebut ditandai dengan munculnya berbagai jenis *game* dengan teknologi baru didalamnya. Diantara berbagai jenis teknologi *game*, *game* edukasi yang berbentuk 2D menjadi salah satu sarana yang penting dalam menunjang belajar anak usia dini. Sejak kecil, anak-anak sudah seharusnya diajarkan terkait pembelajaran dasar supaya anak-anak tidak teralihkan perhatiannya kepada *game* yang tidak bermanfaat. Oleh karena itu, membuat suatu *game* sederhana “Belajar Hijaiyah” menjadi salah satu solusi yang diberikan untuk menunjang proses belajar huruf al-quran kepada anak sejak usia dini. Tentunya, menciptakan *game* saja tidak cukup, sehingga dalam penelitian ini berfokus untuk menambah pengalaman pengguna dengan menambahkan musik adaptif dalam *game*. Untuk menentukan musik adaptif didalam *game*, terdapat beberapa atribut yang digunakan seperti jarak *player* dengan npc musuh, kecepatan *player*, dan jarak *player* dengan finish. Untuk melakukan penentuan musik adaptif di dalam penelitian ini digunakan metode *decision tree* dengan algoritma C4.5. Metode ini menggunakan parameter perhitungan nilai *entropy* dan nilai *gain* pada setiap atribut yang digunakan yaitu jarak *player* dengan npc musuh, kecepatan *player*, dan jarak *player* dengan finish untuk membentuk pohon keputusan. Setelah membentuk pohon keputusan, rules yang dihasilkan diimplementasikan dalam *game* sehingga mampu menentukan musik adaptif digunakan dalam *game*. Hasil dari penelitian ini adalah terciptanya *game* 2D “Belajar Hijaiyah” yang menggunakan musik adaptif didalamnya untuk menambah pengalaman pengguna dalam bermain *game*.

ABSTRACT

Lestari, Yeni. 2023. **Implementasi *Decision tree* untuk Menentukan Musik Adaptif Berdasarkan *Player-Case* dalam *Game Hijaiyah***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Promotor: (I) Dr. Fresy Nugroho, M.T. (II) Zainal Abidin, M.Kom.

Game technology in this era is experiencing very rapid development. This is marked by the emergence of various types of *games* with new technology in them. Among the various types of *game* technology, educational *games* in the form of 2D are an important means of supporting early childhood learning. Since childhood, children should be taught about basic learning so that children are not distracted by *games* that are not useful. Therefore, making a simple *game* "Learn Hijaiyah" is one of the solutions provided to support the process of learning the letters of the qu'ran to children from an early age. Of course, just creating a *game* is not enough, so in this research the focus is on enhancing the user experience by adding adaptive music in the *game*. To determine adaptive music in the *game*, several attributes are used, such as the distance between the *player* and the enemy NPC, the speed of the *player*, and the distance between the *player* and the finish. To determine adaptive music in this study, the *decision tree* method was used with the C4.5 algorithm. This method uses *entropy* value calculation parameters and *gain* values for each attribute used, namely the *player's* distance from the enemy NPC, *player's* speed, and *player's* distance to the finish to form a *decision tree*. After forming the *decision tree*, the resulting rules are implemented in the *game* so that they are able to determine the adaptive music used in the *game*. The result of this research is the creation of a 2D *game* "Learn Hijaiyah" which uses adaptive music in it to add to the user experience in playing *games*.

Keywords: Adaptive music, *Decision tree*, Arabic *Game*, C4.5 Algorithm.

البحث مستخلص

ليستاري، بيني. 2023. تنفيذ شجرة القرار لتحديد الموسيقى المتكيفة استنادًا إلى حالة اللاعب في لعبة الحجابة. رسالة جامعية. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرفون: (الأول) M.Kom.(الثاني) زينال عابدين، M.T.الدكتور فريسي نوغروهو،

تطورت تكنولوجيا الألعاب في هذا العصر بشكل سريع جدًا. ويتضح ذلك من ظهور أنواع مختلفة من الألعاب مع تقنيات جديدة داخلها. ومن بين مجموعة متنوعة من تقنيات الألعاب، تُعتبر الألعاب التعليمية ثنائية الأبعاد أحد الوسائل المهمة لدعم تعلم الأطفال في سن مبكرة. يجب أن يتم تعليم الأطفال في سن مبكرة المفاهيم الأساسية للتعلم لكي لا ينحرف اهتمامهم إلى الألعاب غير المفيدة. ولذلك، صنع لعبة بسيطة "تعلم الحروف الهجائية" يعد واحدة من الحلول المقدمة لدعم عملية تعلم حروف القرآن للأطفال في سن مبكرة. بالطبع، إن صنع اللعبة وحده ليس كافيًا، لذلك يركز هذا البحث على إضافة تجربة المستخدم من خلال إضافة موسيقى متكيفة داخل اللعبة. لتحديد الموسيقى المتكيفة في اللعبة، يتم استخدام عدة سمات مثل مسافة اللاعب من الشخصية غير لتحديد الموسيقى C4.5 الالعبة العدو، وسرعة اللاعب، ومسافة اللاعب من النهاية. يتم استخدام طريقة شجرة القرار بالخوارزمية المتكيفة في هذا البحث. تستخدم هذه الطريقة معلمات حساب قيمة الانتروبيا وقيمة الانتفاع في كل سمة تستخدم، وهي مسافة اللاعب من الشخصية غير الالعبة العدو، وسرعة اللاعب، ومسافة اللاعب من النهاية لتشكيل شجرة القرار. بعد تشكيل شجرة القرار، يتم تنفيذ القواعد المستخرجة في اللعبة لتحديد الموسيقى المتك

C4.5 الكلمات المفتاحية: الموسيقى المتكيفة، شجرة القرار، لعبة الحجابة، الخوارزمية

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada awal terciptanya *game*, *game* diciptakan dalam bentuk yang sederhana. Kesederhanaan *game* dilihat dari tampilan *game* dan musik efek sederhana yang ada didalamnya. Akan tetapi, seiring berkembangnya teknologi *game*, perkembangan tampilan dan unsur lain yang ada didalam *game* juga mengalami peningkatan. Namun, dalam perkembangan tersebut, penggunaan musik/audio dalam *game* kurang mendapat perhatian dari para pengembang *game* terutama dalam *game* edukasi. Untuk meningkatkan aspek penggunaan musik didalam *game*, musik adaptif menjadi salah satu hal yang bisa diterapkan dalam *game*. Menentukan musik adaptif berarti musik yang digunakan harus sesuai atau dapat beradaptasi dengan kondisi *player* dan lingkungan dalam *game*. Penentuan musik adaptif dalam penelitian ini dilakukan karena penggunaan musik adaptif dapat mempengaruhi pengalaman *player* dalam bermain. Selain itu, menentukan musik yang adaptif mampu memberikan penjelasan terhadap suasana demografis dalam lingkungan *game* (Collins, 2008). Oleh karena itu menentukan musik yang adaptif didalam *game* dapat memberi nilai tambah pada aplikasi *game*.

Menentukan musik yang adaptif dalam *game* ini, tentunya harus menggunakan kaidah yang benar dan sesuai dengan ajaran islam. Salah satunya yaitu dengan menggunakan musik yang diperbolehkan dalam islam. Musik dalam islam dibahas dalam 2 macam sudut pandang yaitu dari kajian ilmiah sebagai ilmu pengetahuan dan kajian hukum fiqhiyah. Dalam suatu hadist dari Nabi Muhammad SAW

bersabda yang artinya “*Allah itu indah dan mencintai keindahan*”. Ada beberapa tokoh muslim yang mengkaji tentang seni musik didalam islam salah satu nya adalah Ismail Raji Al-Faruqi. Menurut beliau, konsep musik memiliki beberapa point penting yaitu musik sebagai ekspresi estetis yang berkesinambungan dengan manusia, entitas tauhid sebagai elemen vital dari musik, dan musik merupakan khazanah ilmu yang bersumber dari tuhan. Selain itu, terdapat beberapa tokoh ulama lain yang memiliki pemikiran yang senada dengan pemikiran Al-Faruqi, diantaranya yaitu Quraish Shihab dan Al-Attas. Menurut Qurasih Shihab, musik merupakan seni yang mengandung keindahan didalamnya yang berasal dari ekspresi ruh yang mengandung keindahan dan didorong oleh fitrah yang dianugerahkan Allah swt kepada hamba-hambanya. Sedangkan menurut Al-Attas, musik merupakan implementasi dari ilmu Allah swt yang sampai pada manusia dan ditanggapi oleh akal sebagai realitas ruhani. (Jamil, 2022). Musik yang digunakan dalam penelitian ini menggunakan musik instrument dengan beberapa kriteria untuk menentukan sifat adaptifnya.

Menentukan musik adaptif dalam penelitian ini dilakukan dengan menyesuaikan keadaan *player* dengan Npc lain dan lingkungan dalam *game*. Dalam ajaran islam, memiliki kemampuan untuk menentukan pilihan merupakan suatu landasan yang digunakan dalam kehidupan. Seperti ketika manusia diperintahkan untuk memilih perlakuan yang baik dalam berkehidupan sehari-hari. Salah satu ayat al-quran tentang pentingnya memilih dan menentukan sesuatu dapat dilihat dari firman Allah swt tentang pemilihan seorang pemimpin. Dalam islam, untuk

memilih seorang pemimpin terdapat beberapa kriteria yang dijabarkan didalam al-quran salah satunya terdapat dalam surat Al-Qasas sebagai berikut:

قَالَتْ إِحْدَاهُمَا يَا أَبَتِ اسْتَجِرْهُ صَلَّى إِنَّ خَيْرَ مَنْ آسْتَجِرْتَهُ الْقَوِيُّ الْأَمِينُ

”Dan salah seorang dari kedua (perempuan) itu berkata, “Wahai ayahku! Jadikanlah ia sebagai pekerja (pada kita), sesungguhnya orang yang paling baik yang engkau ambil sebagai pekerja (pada kita) ialah orang yang kuat dan dapat dipercaya.” (Q.S Al-Qasas: Ayat 26, Mushaf Hafs).

Menurut Zubdatut tafsir min fathil qadir, ayat tersebut merupakan salah satu kriteria dalam memilih seorang yang dipekerjakan dalam islam yaitu yang memiliki karakter kuat dan dapat dipercaya. Karakter amanah pada ayat ini yaitu karakter yang tidak berkhianat terhadap barang yang diserahkan kepadanya. Sedangkan karakter kuat berarti kuat dalam menjalani pekerjaan termasuk pengalaman dan semangat yang dimiliki. Kedua karakter yang dijabarkan dalam ayat ini dimiliki oleh Nabi Musa as. Dalam hal ini, karakter amanah yang memiliki hubungan kepada Allah SWT sebagai pemberi amanah/hubungan vertikal dan hubungan kepada sesama manusia dan lingkungannya/hubungan horizontal. (Sarkawi & Fadli, 2021). Dari ayat ini dapat kita pahami bahwa perintah Allah swt dalam menentukan sesuatu selalu didasarkan pada keputusan yang tepat dengan melihat berbagai kondisi yang ada. Sama halnya dengan memilih seorang pemimpin, pemilihan musik yang adaptif dalam *game* juga harus disesuaikan dengan kondisi *player* yang bermain dalam *game* sehingga keputusan dalam pemilihan musik dapat diterapkan dan diterima dengan baik.

Beberapa penelitian yang berhubungan dengan adaptif musik dalam *game* masih sedikit. Salah satu penelitian adaptif musik yang ditemukan yaitu penelitian

adaptif musik dalam *game* yang memengaruhi cerita narrative pada *game* dalam tesis yang berjudul *Adaptive Music Generation for Computer Games* (Prechtl, 2015). Dimana penelitian ini sendiri berfokus pada teori musik barat dengan akor barat. Sedangkan musik dan audio didalam *game* tidak hanya berpengaruh dalam menentukan narrative cerita *game*, tetapi juga harus mempertimbangkan suasana *player* dan bagaimana interaksi antar NPC dan *player* dalam dalam *game*. Pada penelitian sebelumnya, metode yang digunakan menggunakan Markov Model dengan melakukan transisi nilai matrix untuk menggenerasi akor yang digunakan dalam adaptive musik. Sedangkan metode yang digunakan, kami menggunakan metode *decision tree* untuk menentukan nilai keputusan adaptif musik yang digunakan berdasarkan beberapa variabel yang diinputkan. Alasan menggunakan *decision tree* adalah metode tersebut dapat menentukan keputusan untuk memilih adaptif musik dari beberapa variabel saja. Dalam *game* belajar hijaiyah ini, data variabel yang digunakan sebagai parameter menentukan musik adaptif diantaranya jarak *player* dengan NPC, kecepatan *player*, dan jarak *player* dengan garis finish yaitu huruf hijaiyah terakhir dalam berjalan.

Dalam *game* belajar hijaiyah ini, musik adaptif yang digunakan adalah musik yang mengalami perubahan volume dan pitch suara. Dengan memberikan musik yang adaptif, *game* yang dibuat menjadi lebih menarik, sehingga mampu menambah pengalaman pengguna dalam bermain. Selain itu, adanya musik adaptif dalam *game* belajar hijaiyah ini diharapkan dapat menggabungkan kepentingan dalam belajar hijaiyah dalam sebuah permainan yang menarik. Dengan begitu,

pengguna dapat dengan mudah memahami materi huruf hijaiyah dan bisa menerapkannya dalam kehidupan sehari-hari seperti membaca al-quran.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah:

1. Bagaimana implementasi metode *decision tree* dalam menentukan adaptif musik pada *game* 2D sederhana “Belajar Hijaiyah”?
2. Bagaimana analisa hasil implementasi metode *decision tree* dalam menentukan musik adaptif pada *game* 2D “Belajar Hijaiyah”?

1.3 Tujuan

Tujuan dari penelitian ini diantaranya:

1. Mengimplementasikan metode *decision tree* dalam menentukan musik dinamis yang terdapat dalam *game*.
2. Menganalisa hasil implementasi metode *decision tree* dalam menentukan musik adaptif pada *game* 2D “Belajar Hijaiyah”.

1.4 Manfaat

Manfaat dari penelitian ini adalah:

1. Mempermudah anak-anak usia dini dalam belajar huruf hijaiyah
2. Mengetahui hasil implementasi metode *decision tree* dalam menentukan adaptif musik.
3. Mampu menganalisa hasil dari implementasi metode *decision tree* dalam menentukan musik adaptif pada *game* 2D “Belajar Hijaiyah”.

1.5 Batasan Masalah

Pada penelitian ini, kami menetapkan beberapa batasan seperti:

1. Metode yang digunakan dalam mengimplementasikan adaptif musik adalah dengan menggunakan metode *decision tree* dengan algoritma C4.5.
2. *Game* yang digunakan adalah *game* 2D sederhana berbasis desktop/computer.
3. Hasil implementasi yang diharapkan adalah musik adaptif yang berfokus kepada adaptif *volume* dan *pitch* musik dalam *game* 2D.
4. Pembelajaran hijaiyah yang terdapat dalam *game* meliputi cara membaca, contoh bacaan, dan cara menggabungkan huruf hijaiyah.
5. *Game* Belajar Hijaiyah yang digunakan dalam penelitian ini ditujukan untuk anak-anak usia dini mulai dari 3 tahun -6 tahun.

1.6 Sistematika Penulisan

Secara garis besar, sistematika penulisan pada tugas akhir ini dapat dilihat sebagai berikut:

1. BAB 1 Pendahuluan

Pada bab 1 pendahuluan ini berisi tentang latar belakang dikerjakannya tugas akhir, alasan yang ada dibalik pembuatan projek, penelitian terdahulu yang pernah dilakukan, rumusan masalah, tujuan penelitian, manfaat penelitian, rumusan masalah dan sistematika penulisan tugas akhir.

2. BAB 2 Kajian Pustaka

Bab 2 kajian pustaka berisi tentang teori-teori yang mendukung adanya penelitian ini. Selain itu, pada Bab ini juga menjelaskan prinsip-prinsip yang mendukung teori-teori dalam penelitian ini.

3. BAB 3 Desain Penelitian

Pada Bab 3 desain penelitian menjelaskan tentang rancangan yang digunakan dalam penelitian ini, sekaligus menjelaskan *wireframe* dari desain interface *game* 2 dimensi yang dibuat serta alur *game* dan perhitungan *music* adaptif yang terjadi dengan menggunakan metode *decision tree* yang telah ditetapkan.

4. BAB 4 Hasil dan Pembahasan

Bab 4 hasil dan pembahasan membahas tentang hasil implementasi metode *decision tree* yang digunakan dalam menentukan musik adaptif pada *game* 2D belajar hijaiyah. Bab ini juga menampilkan hasil pengujian sistem untuk membuktikan kecocokan implementasi *decision tree*.

5. BAB 5 Kesimpulan dan Saran

Bab 5 kesimpulan dan saran berisi tentang kesimpulan yang didapat dari hasil implementasi yang sudah dibuat, dan juga berisi saran yang bertujuan untuk mengkritik sistem dalam pengembangan sistem kedepannya.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Salah satu penelitian *decision tree* yaitu pada tahun 2019 oleh Retnani Latifah, Emi Setia Wulandari, Priadhana Edi Kreshna berjudul model *decision tree* untuk prediksi jadwal kerja menggunakan scikit-learn. Pada penelitian ini, metode *decision tree* digunakan untuk dapat mengatasi data kategorikal dan numerikal yang sulit ditangani oleh metode lain. Library yang digunakan pada penelitian ini yaitu library *scikit-learn* yang diimplementasikan dalam bahasa pemrograman *python*. Data yang digunakan pada penelitian ini berjumlah 54 data dengan 3 variabel kategorikal dan 1 variabel numerikal. Presentase pemisahan terbaik yaitu dengan menggunakan 70% data latih dan 30% data uji. Pada penelitian ini penggunaan metode *decision tree* dibandingkan dengan KNN dan menyatakan hasil yang diperoleh dari metode *decision tree* memiliki akurasi yang lebih baik dibandingkan metode KNN. (Latifah, Wulandari, & Kreshna, 2019)

Penelitian terkait *decision tree* selanjutnya ditulis oleh Intan Dewi Pranastiti pada tahun 2016 berjudul penggunaan *decision tree* untuk *game* pemilihan jalur lintasan sepeda. Pada penelitian *game* pemilihan jalur lintasan sepeda menggunakan metode *decision tree* untuk menentukan jalur atau level terbaik untuk digunakan pemain dalam *game*. Variabel yang digunakan pada penelitian ini diantaranya denyut nadi sebagai root bersifat kategorikal, usia dan jenis kelamin. Sedangkan kelas target nya yaitu *scene*. Data training yang digunakan pada penelitian ini berjumlah 12 dan data yang digunakan pada saat pengujian berjumlah

5. Pada penelitian ini, implementasi metode *decision tree* digunakan pada saat sebelum *game* dimulai dimana *player* memasukkan terlebih dahulu data denyut nadinya, usia dan jenis kelamin nya. Setelah itu, *player* mendapat hasil *scene* terbaik yang dihasilkan dengan perhitungan *decision tree* oleh komputer. (Pranastiti, 2016)

Selain itu, terdapat penelitian mengenai *decision tree* yang ditulis oleh Dalin Zhang, Yi Xu, Yunjuan Peng dan kawan-kawanya berjudul *an interpretable station delay prediction model based on graph community neural network and time-series fuzzy decision tree*. Pada penelitian ini membuat model baru dengan mengkombinasikan beberapa metode yaitu *graph community neural network* dan *fuzzy decision trees*. Penelitian ini bertujuan untuk dapat memperbaiki sistem kereta api sehingga departemen-departemen yang bekerja dalam kereta api dapat bekerja sama dan mempersiapkan jadwal yang akurat dan efisien. Dengan menggunakan *time-series fuzzy decision tree*, hasil keputusan yang dihasilkan oleh perhitungan *fuzzy decision tree* dapat meningkatkan interpretabilitas model, transparansi dan realibilitas yang tinggi. Selain itu, pada penelitian ini juga membandingkan dengan 7 model lain yang sudah ada dan menghasilkan bahwa model baru ini dapat menghasilkan efek prediksi yang lebih tinggi atau akurat dibandingkan dengan model lain. (Zhang, Xu, Peng, Lu, & Liu, 2023)

Pada penelitian ini dilakukan pengimplementasian metode *decision tree* untuk menentukan musik adaptif pada *game* 2D belajar hijaiyah. Melihat dari penelitian sebelumnya, penelitian ini lebih berfokus untuk menentukan musik adaptif didalam *game* sehingga *game* menjadi lebih interaktif dan menyenangkan.

Algoritma yang digunakan pada metode *decision tree* penelitian ini yaitu algoritma *C4.5*.

2.2 *Game*

Game merupakan istilah yang berasal dari bahasa Inggris yang memiliki arti permainan. Berdasarkan arti *game* dalam kamus Oxford, pengertian *game* adalah suatu aktivitas yang dikerjakan untuk bersenang-senang yang didalamnya terdapat rules dan bisa terjadi menang atau kalah. Dari pengertian ini dapat disimpulkan bahwa *game* adalah suatu permainan yang memiliki aturan bermain dan dimainkan oleh seorang karakter dengan memberikan unsur kesenangan bagi pemainnya. *Game* diciptakan oleh seorang *game developer* dengan manfaat dan tujuan tertentu. Tujuan penciptaan *game* dapat digunakan untuk mengenalkan beberapa objek supaya lebih dikenali oleh masyarakat, ada juga *game* yang diciptakan untuk tujuan sebagai sarana refreshing atau memberikan kesenangan dan hiburan bagi para pemainnya, dan *game* yang diciptakan untuk memberikan edukasi baik akademik atau non-akademik kepada kalangan tertentu. Untuk dapat memainkan *game*, ada banyak media yang digunakan seperti *game* yang secara langsung dimainkan dilingkungan sekitar tempat tinggal, dan juga *game* digital yang populer di zaman sekarang.

Seiring perkembangan zaman, *game* menjadi salah satu bidang teknologi yang banyak diminati oleh masyarakat terutama kalangan anak-anak, remaja, dan juga golongan dewasa baik pria maupun wanita. Teknologi *game* berkembang pesat dengan adanya perpaduan lingkungan virtual dan reality dalam *game* seperti AR (*Augmented Reality*) dan VR (*Virtual Reality*). Salah satu contoh *game* populer

pada tahun 2018-2020 adalah Mobile Legend. *Game* tersebut seakan menyita seluruh perhatian masyarakat dengan memberikan permainan yang menarik. Dalam *game* tersebut terdapat komponen-komponen yang detail dalam *game* yang memberikan sentuhan tersendiri dalam *game* sehingga membuat pemain *game* menyukai permainan didalamnya. Dalam skripsi Muhammad Ashof disebutkan menurut Teresa Dillon (Dalam Aeny, 2010) elemen dasar dalam *game* terdiri dari *game rule*, plot, theme, character, object, text, grafik, *sound*, animasi dan *user interface*.

Didalam *game* juga terdapat jenis-jenis dan bentuk-bentuk *game*. Bentuk diantaranya dapat berupa *game* 3 dimensi, dan *game* 2 dimensi. *Game* 2 dimensi adalah *game* dimana *player* hanya dapat bergerak dalam 2 sisi (sumbu x dan sumbu y), sedangkan *game* 3 dimensi memungkinkan *player* untuk bergerak ke segala sisi. Jenis *game* yang ada menurut Restiana (2007) dalam suatu skripsi dari Universitas Muhammadiyah Semarang disebutkan jenis *game* meliputi shooting (tembak-tembakan), fighting (pertarungan), adventure (petualangan), simulasi, strategi, sport, puzzle, dan edukasi. Dalam penelitian ini, *game* yang akan dibuat merupakan *game* edukasi dan menggunakan bentuk 2 dimensi supaya lebih mudah digunakan oleh anak-anak.

2.3 Game 2D

Game 2D (2 dimensi) adalah *game* yang menggunakan 2 koordinat yaitu x dan y. dibandingkan dengan *game* 3 dimensi yang memiliki sumbu x, y, dan z, *game* 2 dimensi lebih sederhana dalam pembuatannya karena hanya memiliki tampilan dari satu sisi saja. *Game* 2D muncul pada masa pertama diciptakannya

game. Dimana *computer game* pada generasi pertama menampilkan *game* dalam bentuk 2d. Pada saat itu simulasi untuk membuat *game* 3d sangat terbatas. Kemudian *game* 2d terus mengalami perkembangan dari masa ke masa.

Walaupun hanya dapat dimainkan dari satu sisi saja, *game* 2d memiliki keunikan tersendiri yang tetap dapat menarik minat pemain untuk terus memainkan *game*. Salah satu hal yang perlu dilakukan untuk menambah keunikan dari *game* 2d adalah dengan memperhatikan setiap komponen dari *game* dan menampilkan keunikan dari alur cerita *game* dan beberapa komponen lainnya seperti audio dan object didalamnya. Jika melihat dari banyaknya aplikasi *game* di platform download seperti *playstore*, dapat kita lihat bahwa *game* 2d banyak digunakan pada *game-game* edukasi. Hal ini dikarenakan *game* 2d lebih mudah diaplikasikan sehingga edukasi yang ingin disampaikan dapat diterima dan dipahami dengan lebih baik oleh penggunanya. Selain itu, karena hanya memiliki 2 dimensi, pergerakan karakter didalam *game* lebih mudah dikontrol dan mudah dipelajari oleh para developer. Salah satu *game* 2d dengan jumlah download terbanyak adalah *game* kucing bernama TOM Friends. *Game* 2d kucing ini banyak didownload oleh kalangan anak kecil karena terdapat banyak aktivitas yang dapat dilakukan oleh karakter kucing dalam kehidupan sehari-hari. Edukasi yang ingin ditampilkan dalam *game* ini yaitu mengedukasi anak-anak untuk dapat memelihara hewan peliharaan kucing dengan baik dan mengarahkan untuk melakukan aktivitas yang positif dan menyenangkan.

2.4 Game PC

Game PC atau *game personal computer* adalah *game* yang dimainkan dari komputer. Dalam skripsi Intan Dewi Pranastiti disebutkan asal usul permainan *game PC* (*video game*) terletak pada awal tabung sinar katoda berbasis pertahanan peluru kendali sistem pada akhir tahun 1940-an. Kemudian, mengalami perkembangan menjadi *game-game* sederhana pada era tahun 1950-an. Dan pada akhirnya *game pc* terus mengalami perkembangan seiring perkembangan teknologi dengan tingkat kecanggihan dan kompleksitasnya yang mengalami peningkatan. Pada mulanya, *game pc* hanya dimainkan dikomputer mainframe, kemudian berkembang sehingga dapat dijalankan di arcade, mainframe, konsol, komputer pribadi dan juga telepon genggam.

Perkembangan *game* diikuti oleh beberapa factor yang saling memengaruhi satu dengan yang lainnya sehingga dapat menciptakan modern *games* yang ada pada saat ini. Diantara factor yang memengaruhi yaitu perubahan hardware yang digunakan, perubahan interaksi yang dilakukan dalam *game*, perubahan software yang digunakan untuk membuat *game*, perubahan kebutuhan bisnis *game*, perubahan demografik dari *player* didalam *game*, diversifikasi, dan perubahan design dalam *game* (Overmars, 2012). *Game* muncul pada tahun 1952 yaitu *game tic-tac-toe* yang diciptakan oleh Alexander Douglas. Kemudian muncul *game Tennis for Two* yang dikembangkan pada tahun 1958 dan diciptakan oleh William Higginbotham, seorang ahli fisika nuklir. *Game* ini menjadi *game* yang dianggap menjadi *game* pertama yang diciptakan walaupun sebelumnya sudah ada *tic-tac-toe* yang diciptakan. Menuju era tahun 60 kemudian muncul *game Spacewar* yang

dikembangkan oleh seorang mahasiswa di MIT pada tahun 1961. Kemudian *game* terus mengalami perkembangan dengan diciptakannya *game The Sims* pada tahun 2000. Pada saat itu, *game* yang dibangun untuk personal *computer* memiliki banyak kontra dari para developer karena beberapa masalah yang ditimbulkan. Hal itu membuat developer mulai berhenti membangun *game* untuk personal *computer* dan memilih untuk mempublikasikan console *game*. Salah satu permasalahan yang timbul saat itu adalah *game* yang dikembangkan untuk personal *computer* dapat mengalami pembajakan gratis diberbagai situs sehingga merugikan developer. Walaupun begitu, dengan segala keresahan yang ada, pada akhirnya *game* pc terus mengalami perkembangan dan bahkan bisa dioperasikan di beberapa perangkat yang berbeda.

2.5 Game Engine

Dalam buku karangan Jared Halpern yang berjudul *2d games with unity* disebutkan *game engine* adalah suatu software yang didesign untuk mengurangi biaya pengeluaran, kompleksitas dan waktu yang dibutuhkan dalam pembuatan atau pengembangan *games*. *Game engine* bertujuan untuk memberikan kemudahan kepada developer untuk membuat suatu *game*. Hal ini dikarenakan didalam *game engine* telah disediakan tools untuk membuat suatu komponen yang dibutuhkan dalam *game*. Dengan adanya *game engine*, kita lebih mudah membuat object atau interface karena didalam *game engine* menyediakan berbagai macam elemen sesuai dengan fungsionalitasnya dalam pembuatan *game*. Beberapa hal yang umum ada dalam *game engine* seperti berikut ini:

1. Graphics Rendering Engine

2. *Physics Engine*
3. *Audio Engine*
4. *Scripting support*
5. *A World Object Model*
6. *Animation Handling*
7. *Networking Code*
8. *Multithreading*
9. *Memory Management*
10. *Artificial Intelligence*

Selain itu, *game engine* juga dilengkapi dengan library yang memudahkan developer untuk membuat suatu *game*. Bahkan, ada beberapa jenis *game engine* yang memungkinkan untuk dapat membuat *game* dengan bahasa pemrograman yang disediakan sehingga developer tidak perlu menyusun banyak script untuk membuat suatu *game*. Dengan begitu, waktu yang dibutuhkan untuk membuat *game* menjadi lebih singkat dan hal yang menjadi fokus pembuatan menjadi lebih sedikit. Walaupun banyak *game engine* yang menyediakan library lengkap untuk membuat *game*, biasanya developer akan tetap memilih membuat script karena *game* yang diprogram dari script lebih powerful dan memuaskan developer karena developer bisa memastikan segala hal yang dibutuhkan dalam *game* dengan lebih rinci.

Sejarah *game engine* dimulai beriringan dengan adanya *game* itu sendiri. Pada tahun 1987, Ron Gilbert, dengan beberapa bantuan dari Chip Mansion, membuat *SCUMM* (*Script Creation Utility for Maniac Mansion*) ketika beliau sedang bekerja di LucasFilm. Pada saat itu, *game engine* ini mampu membuat *game*

dengan type yang sangat spesifik. Hal ini diketahui dari kata MM yaitu Maniac Mansion yang merupakan bagian dari *game adventure* (petualangan). *Game engine SCUMM* ini telah digunakan untuk membuat *game-game* terkenal seperti *Full Throttle*, *The Secret of Monkey Island*, *The Graphics Adventure* dan masih banyak lainnya. Seiring berkembangnya teknologi modern, saat ini banyak studio *game* yang memiliki *game engine* sendiri untuk mengembangkan *game* mereka. Seperti Bethesda *Game Studios* dan Blizzard Entertainment yang menggunakan *game engine* buatan sendiri dalam studio produksi *game* mereka.

Walaupun beberapa studio *game* modern menggunakan *game engine* sendiri, masih terdapat beberapa *game engine* yang digunakan oleh banyak pengembang *game*. Seperti *game engine* yang open source ataupun berbayar. Diantara *game engine* yang kita ketahui yaitu *Unity*, *corona*, *construct*, *unreal engine*, dan *godot game engine*.

2.6 Hijaiyah

Hijaiyah berasal dari kata *hajjaa* yang memiliki arti mengeja, menghitung huruf, membaca huruf demi huruf. Dalam pembahasan makalah materi agama dijelaskan huruf adalah sarana untuk menyatakan kehendak, cipta, dan rasa. Sebelum mengenal teknologi komunikasi seperti telepon, manusia menggunakan tulisan sebagai alat komunikasi antar sesama. Pada zaman dahulu, manusia menulis melalui surat dan mengirimkannya kepada sanak saudara untuk saling berkomunikasi dari jauh. Huruf yang menjadi bahasa tulis yang digunakan manusia memiliki beberapa fase perkembangan. Fase pertama yaitu *al-shauri al-dzati* dimana manusia mendeskripsikan suatu peristiwa melalui gambar. Fase kedua yaitu

al-shauri al-ramzi dimana manusia mendeskripsikan suatu peristiwa, waktu terjadinya, kondisi dan kejadian melalui makna lewat suatu gambar. Fase ketiga yaitu al-maqthoi, manusia menggunakan tanda-tanda untuk mendeskripsikan maksud dan tujuan mereka. Fase keempat yaitu al-hija'I, manusia mengakulturasi tanda-tanda menjadi huruf. Hal itu menunjukkan bahwa tulisan atau huruf sangat memiliki sejarah yang penting dalam kehidupan manusia. Sama seperti huruf latin, huruf hijaiyah memiliki sejarah yang sangat penting dalam agama islam.

Huruf pada mulanya berbentuk tanda-tanda, kemudian generasi selanjutnya melakukan penyederhanaan sehingga menjadi huruf. Begitu pula dengan huruf arab/ hijaiyah. Menurut penelitian sejarawan, tulisan arab yang digunakan sampai saat ini berasal dari Mesir Kuno yang disebut hieroglyph. Pada abad sebelum kedatangan islam, orang-orang Hijaz sudah belajar mengenai baca tulis di Syiria dan Irak. Kemudian dikenalkan kepada masyarakat Quraiys. Sehingga tulisan arab terus menyebar sampai pada abad ke-6 dan ke-7, banyak pemuda islam yang pandai baca dan tulis. Dalam makalah pembahasan materi hijaiyah disebutkan orang yang pertama kali mengurutkan huruf hijaiyah mulai alif sampai ya' adalah Nashr Bin 'Ashim Al-Laitsi. Huruf hijaiyah pada jaman dahulu dapat dilihat pada gambar dibawah ini:

Huruf Hijaiyah berjumlah 28 mulai dari alif sampai dengan ya'. Namun, jika huruf lam-alif dan hamzah yang merupakan huruf yang berdiri sendiri digabungkan, maka huruf hijaiyah berjumlah menjadi 30. Cara penulisan huruf Arab berbeda dengan huruf Latin. Bahasa Latin ditulis mulai dari kiri ke kanan, sedangkan bahasa Arab/ ditulis dari kanan ke kiri. Bahasa Arab memiliki sejarah

yang panjang dan penting dalam agama Islam. Bahasa Arab merupakan bahasa asli al-quran sehingga menjadi sangat penting untuk dipelajari oleh umat Islam. Oleh karena itu, sebagai umat Islam yang baik, seharusnya kita mempelajari bahasa Arab sebagai bahasa al-quran.

Untuk dapat memahami bahasa Arab, hal yang paling dasar untuk dipelajari adalah dengan mempelajari huruf hijaiyah. Huruf hijaiyah menjadi landasan manusia ketika mempelajari bahasa al-quran. Dewasa ini, banyak anak usia dini yang dimanjakan oleh orang tuanya untuk bebas bermain *gadget*. Fenomena ini seharusnya menjadi perhatian orang tua untuk mengajarkan anak nilai-nilai Islam sejak dini. Oleh karena itu, pengawasan orang tua terhadap aktivitas anak-anak dalam bermain harus diperhatikan. Dengan menciptakan *game* sederhana ini, diharapkan mampu menjadi sarana belajar anak-anak dengan menyenangkan. Dengan begitu, anak-anak dapat memahami huruf hijaiyah sejak usia dini.

DiIndonesia, anak-anak biasanya dikenalkan dengan buku iqro untuk mempelajari huruf hijaiyah. Sejak usia 3-6 tahun, anak-anak diajarkan untuk mempelajari huruf arab dengan iqro. Anak-anak diajarkan huruf Arab oleh guru mengaji untuk membaca iqro dan menuliskan huruf Arab sehingga anak-anak mudah memahami bacaan Arab. Akan tetapi, seiring perkembangan teknologi yang canggih dan juga melenakan membuat anak-anak menjadi malas untuk pergi mengaji dan belajar al-quran. Sehingga, kita harus menyesuaikan kondisi jaman saat ini dengan menciptakan media pengajaran yang lebih efektif dan disenangi oleh anak-anak.

2.7 Musik

Musik merupakan salah satu komponen penting yang ada dalam *game*. Dalam dunia *game* baik virtual reality ataupun *computer games*, *musik* menjadi salah satu komponen yang memiliki pengaruh yang penting dalam *game*. Pada jaman dahulu, musik menjadi komponen yang tidak terlalu menjadi pertimbangan dalam *game*, namun seiring berkembangnya teknologi dan munculnya berbagai variasi musik, musik menjadi salah satu komponen penting karena dapat memberikan kesan dalam *game*. Dalam tulisan Jasson Prestiliano, beliau menjabarkan perkembangan audio/musik yang terdapat dalam *game*. *Game spacewars* yang muncul pada tahun 1971 tidak memiliki *sound* didalam *game*. Kemudian, pada tahun 1977, muncul *game* Pong yang sudah mulai memiliki *sound*. *Sound* yang ada pada *game* Pong saat itu adalah *sound*, sekarang kita kenal dengan efek suara, “bleep” dan “bloop”. Kemudian seiring berkembangnya teknologi yang digunakan, yaitu munculnya chip, *sound* yang digunakan dalam *game* memiliki pokey chip dan 4 channel dengan pitch, volume, dan distorsi yang berbeda. *Sound* tersebut terdapat dalam *game* Atari 5200 yang diciptakan pada tahun 1982. *Sound* dalam *game* terus mengalami perkembangan dengan munculnya CD, DVD *player* dan teknologi high definition lainnya sebagai perangkat pemutar *sound*.

Musik yang terdapat dalam *game* memiliki fungsi yang beragam. Dalam jurnal karya Patrick Hutchings dan Jon McCormack disebutkan bahwa adanya audio/*sound* dalam *game* mampu menambah kesan pengalaman pengguna dan memperkuat rasa pengalaman yang terdapat dalam *game*. *Sounds* dan audio mampu memberikan kesan mendalam kepada pengguna dalam memainkan *game*. Hal ini

dikarenakan *sound* dan audio memberikan nyawa kepada *game* dengan menggambarkan suasana dan kondisi yang terdapat didalam *game*. Dalam dunia *game* sendiri, *sound*, musik, dan audio biasanya diciptakan oleh seorang composer atau *team* tersendiri yang memahami cita rasa musik. Seorang developer *game* juga dapat meminta bantuan kepada studio musik yang dapat membuat dan menentukan musik khusus untuk *game* mereka.

Modern ini, banyak developer yang mengembangkan audio adaptive dalam pembuatan *game*. *Game* yang memiliki audio yang aktif dan audio yang adaptif memiliki perbedaan. *Game* dengan audio aktif menampilkan *game* dengan audio didalamnya tanpa memerlukan interaksi dari *player* maupun objek. Audio aktif yang berada dalam *game* hanya berfungsi untuk memberikan kesan pengalaman pengguna tanpa memerhatikan proses yang berada didalam *game*. Sedangkan musik yang adaptif dalam *game* menampilkan suara dalam *game* dengan adanya beberapa variabel interaksi yang terdapat dalam *game*. Variabel interaksi tersebut diantaranya jumlah nyawa yang dimiliki *player*, kecepatan pergerakan *player*, bagaimana ketika suatu button diklik oleh *player* dan lain sebagainya. Dengan adanya musik yang adaptif didalam *game*, *game* menjadi lebih hidup dan *player* berhubungan dengan langsung dengan perjalanan *game* yang dimainkan. Selain itu, musik yang adaptif mampu menambah user experience sehingga pemain *game* lebih nyaman ketika menggunakan *game*.

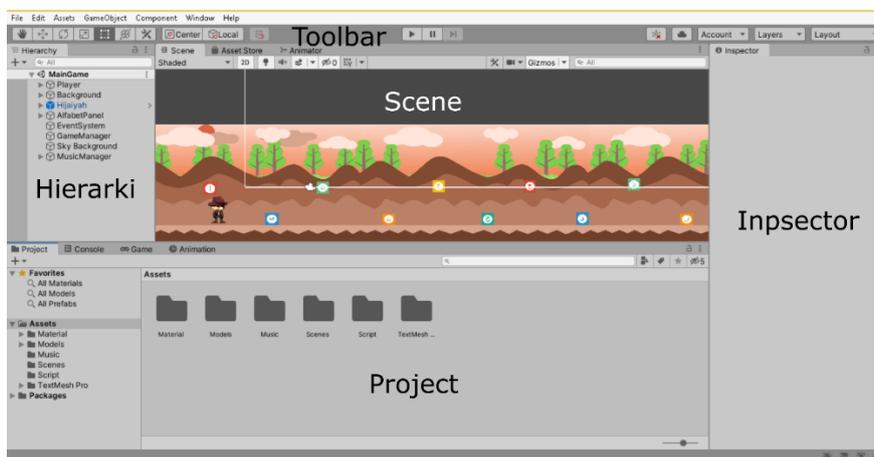
Dalam *game* sederhana belajar hijaiyah ini, kami mengimplementasikan bagaimana suatu *game* memiliki musik yang adaptif dan menarik. Musik yang adaptif dapat ditentukan dengan melihat kondisi *player* dalam bermain seperti

lingkungan yang menunjukkan posisi *player*, jarak *player* dengan NPC, waktu tersisa yang dimiliki oleh *player*, jarak *player* dengan garis finish, dan kecepatan berjalan yang dilakukan oleh *player*. Dengan menciptakan musik yang adaptif dalam *game* belajar hijaiyah ini diharapkan pemain dapat bermain dengan nyaman dan mampu menambah semangat pemain untuk terus belajar huruf hijaiyah sampai pemain mampu memahami hijaiyah dengan baik dan benar.

2.8 Unity

Unity merupakan salah satu *game engine* yang banyak digunakan dalam dunia pengembangan *game*. Dalam buku yang berjudul *Developing 2D Games with Unity* karya Jared Halpern, *unity* merupakan salah satu *game engine* yang menyediakan banyak keuntungan dibandingkan dengan *game engine* lain yang tersedia. *Unity* banyak memunculkan banyak aplikasi *game* baik untuk desktop, console, dan mobile. Dalam skripsi Intan Dewi Pranastiti disebutkan *unity technology* dibangun pada tahun 2004 oleh David Helgason, Nicholas Francis, dan Joachim Ante. (Pranastiti, 2016) Kemudian, *unity* dipublikasikan satu tahun setelahnya yaitu tahun 2005. Pembuatan *unity* ini didasari oleh rasa kepedulian mereka kepada para developer *game* yang tidak dapat membeli *game engine* sendiri karena harga yang mahal. Oleh karena itu *unity* dipublikasikan secara open source mulai tahun 2009 dan tahun 2012 berhasil mencapai popularitas tertinggi dengan lebih dari 1 juta developer yang tercatat menggunakannya diseluruh dunia. Namun sekarang *unity* menyediakan versi berbayar yaitu *Unity Plus Tier*, *Unity Pro* dengan beberapa tambahan fitur yang lebih menguntungkan.

Unity menyediakan berbagai tools dengan fungsinya masing-masing untuk membuat *game* dengan mudah. Dengan berbagai kelebihan yang dimiliki *unity* memudahkan developer untuk membuat *game*. Kelebihan yang terdapat didalam *unity* antara lain tampilan interface *unity* yang user-friendly seperti ditampilkan pada gambar 1, mendukung pembentukan *game* 2D dan 3D, merupakan *game* engine cross-platform yang memudahkan developer dalam membangun aplikasi dalam berbagai platform. Selain itu, didalam *unity* juga mendukung pembuatan script dengan bahasa C#, dimana bahasa ini sering digunakan oleh developer dalam mengembangkan *game*. Didalam *unity* juga terdapat asset store yang menyediakan berbagai macam asset object yang bisa digunakan oleh developer secara gratis maupun berbayar. Asset store ini dapat diakses secara online sehingga harus terhubung dengan internet.



Gambar 2. 1 Tampilan antarmuka *Unity*

Unity memungkinkan developer untuk membuat *game* dengan berbagai genre. Mulai dari genre *FPS (First Person Shoot)*, *Third person*, *VR*, *AR*, *Lego*, dan *Karting*. Selain itu, *unity* menyediakan layanan untuk para pemula atau pun pelajar

yang ingin belajar membuat *game* di *unity*. *Unity* merupakan *game engine* dengan sistem *license* sehingga harus mendaftarkan terlebih dahulu *license* yang dimiliki sebelum mulai menggunakan *unity*. *License* yang disediakan oleh *unity* memiliki beberapa pilihan, oleh karena itu developer yang menggunakan *unity* harus menyesuaikan dengan kebutuhannya. Semenjak dipublikasikan pada tahun 2005, *unity* sudah berhasil menjadi *game engine* yang digunakan untuk membuat beberapa *game* terkenal seperti *Thomas was alone*, *temple run*, *the room*, *rimworld*, *hearthstone*, *kerbal space program*, *pokemon go*, dan *cuphead*.

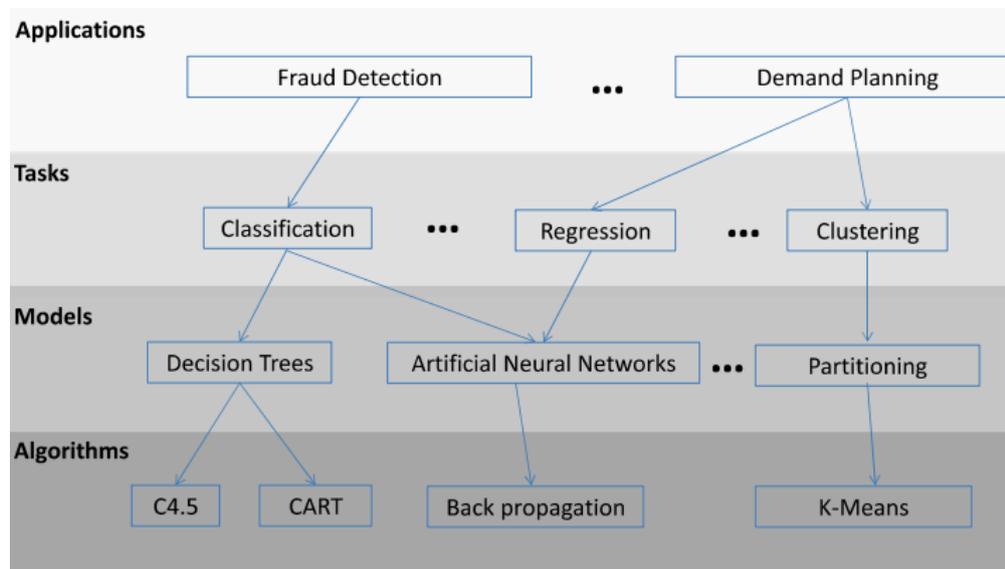
Unity adalah *cross-platform game engine* yang digunakan dalam membangun *game* 2d sederhana belajar hijaiyah ini. *Unity* menyediakan fasilitas untuk menambahkan script C#, Python, dan Mono Developpe yang bisa digunakan dalam proses penentuan audio adaptif dalam *game* 2d ini. Selain itu, *unity* mendukung berbagai varietas assets seperti animasi, suara, dan tekstur dan dapat digunakan untuk mengimport asset dari luar. Oleh karena itu, penggunaan *unity* sangat tepat untuk digunakan dalam membangun *game* sederhana ini.

2.9 Decision tree

2.9.1 Pengertian Decision tree

Decision tree memiliki arti pohon keputusan dalam bahasa Indonesia. Sedangkan jika dihubungkan dengan *data mining*, *decision tree* merupakan model prediktif yang dapat digunakan baik untuk klasifikasi atau regresi. Dalam operation research, *decision tree* menggambarkan kepada model hierarki suatu keputusan dan konsekuensi yang ditimbulkan. *Decision tree* biasa digunakan oleh pengambil keputusan seperti HRD atau yang lainnya untuk mencari strategi terbaik yang dapat

digunakan untuk mencapai tujuan. Sebelum mengetahui lebih jauh, berikut ini 4 layer model data mining dalam kaitannya dengan *decision tree*:



Gambar 2. 2 Layer Data Mining

Penggunaan *decision tree* dalam data mining sangat populer untuk mengambil keputusan. Para peneliti mengatakan bahwa metode *decision tree* banyak digunakan karena sifatnya yang sederhana dan transparan. Selain itu, dalam menentukan keputusan, metode *decision tree* memiliki beberapa kelebihan diantaranya:

1. *Decision tree* merupakan metode yang sangat jelas sehingga mudah dipelajari dan digunakan oleh pemula.
2. *Decision tree* ditampilkan dalam bentuk grafis yang memiliki struktur hierarkis sehingga mudah untuk ditafsirkan dibandingkan metode yang lain.

Akan tetapi, kelebihan *decision tree* dalam penggunaan berupa grafik dapat menjadi tidak berguna apabila pohon keputusan memiliki banyak node dalam proses klasifikasinya. Oleh karena itu, untuk permasalahan yang lebih kompleks, memiliki

banyak variabel, diperlukan pengembangan prosedur grafik sehingga mampu menyederhanakan interpretasi. Selain itu, algoritma ini memiliki kekurangan karena dapat menyebabkan overfitting ketika jumlah data yang digunakan terlalu banyak.

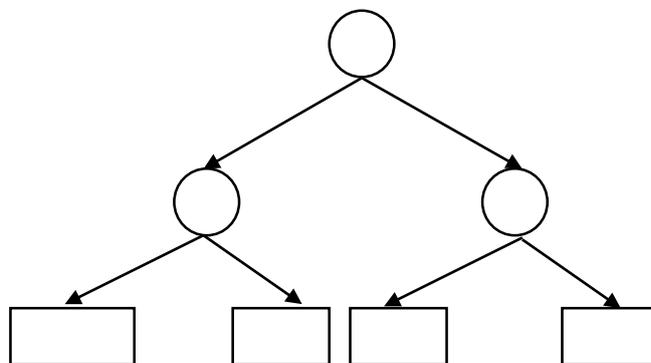
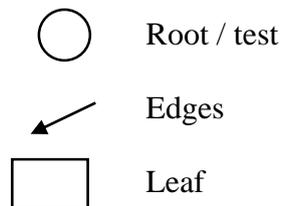
Algoritma C4.5

Dalam penelitian ini, perhitungan *decision tree* dilakukan dengan menggunakan algoritma C4.5. Algoritma C4.5 merupakan salah satu algoritma yang digunakan dalam membuat pohon keputusan. Algoritma ini merupakan pengembangan dari algoritma ID3 (*Iterative Dichotomiser 3*). Sehingga dalam pembentukan pohon keputusan, prinsip kerja dari algoritma C4.5 ini hampir sama dengan algoritma ID3. Akan tetapi, dalam tahapan penentuan atribut yang digunakan sebagai root, algoritma C4.5 menggunakan penentuan root dengan menghitung nilai *gain ratio* tertinggi dari atribut yang ada, sedangkan ID3 dalam menentukan root memilih atribut dengan nilai *gain* tertinggi saja (Defiyanti & Pardede, 2018). Alasan pemilihan algoritma *decision tree* C4.5 dalam penelitian ini adalah metode *decision tree* dengan algoritma C4.5 ini lebih baik dibandingkan dengan algoritma SVM, Neural Network, Naïve Bayes karena algoritma C4.5 lebih efisien dan sederhana (Defiyanti & Pardede, 2018).

2.9.2 Karakteristik *Decision tree*

Metode *decision tree* memiliki 3 simpul yaitu simpul akar/ simpul root, simpul test, dan simpul leaves/ simpul keputusan. Simpul root merupakan simpul yang terletak paling atas dalam pohon keputusan atau menjadi titik awal proses *decision tree*. Simpul test atau simpul perantara merupakan simpul yang

berhubungan dengan pengujian atau pertanyaan variabel, dan simpul leaf merupakan node terakhir yang berisi keputusan akhir *decision tree*. Struktur dari simpul-simpul pohon keputusan dapat dilihat dibawah ini:



Gambar 2. 3 Struktur *Decision tree*

Algoritma *decision tree* yang diketahui diantaranya C45, ID3, dan CART. Masing-masing algoritma memiliki tahapan masing-masing untuk mengambil keputusan dengan menggunakan *decision tree*.

2.9.3 Algoritma Induksi *Decision tree*

Metode *decision tree* memiliki beberapa aturan dalam menjalankan proses yang dilakukannya. Operasi induksi dalam *decision tree* merupakan suatu entitas yang mengolah data training dan membentuk model yang merepresentasikan hubungan diantara atribut input dan atribut hasil. Pohon keputusan dibangun dalam suatu metode yang disebut recursive top-down divide and conquer. Hal ini berarti

proses pengambilan keputusan dimulai dari root, kemudian variabel diuji dan mencabang ke variabel dijalur yang benar sampai memperoleh keputusan/ simpul leaf. Atribut yang terdapat dalam pohon keputusan memiliki 2 jenis atribut yaitu kategori dan non-kategori. Atribut kategori adalah atribut yang menentukan hasil keputusan akhir bernilai iya atau tidak, sedangkan atribut non-kategori adalah nilai atau value yang ada pada setiap kategori. Atribut yang memiliki nilai kontinu perlu didiskritkan terlebih dahulu untuk dapat digunakan. Atribut-atribut yang ada diuji dengan perhitungan heuristic atau pengukuran statistic, misalnya perhitungan *information gain*.

2.9.4 Tahapan dalam *Decision tree* Algoritma C4.5

Tahapan dalam *decision tree* dengan menggunakan algoritma C4.5 yaitu:

1. Menentukan tujuan / goal yang diinginkan (hasil keputusan). Menentukan data variabel yang digunakan.
2. Menentukan root yang digunakan dengan memilih nilai *gain ratio* tertinggi dari variabel. (Setio, Saputro, & Winarno, 2020)
3. Melakukan pengulangan pada proses perhitungan *entropy* pada atribut berikutnya sampai membentuk node 2.
4. Menemukan leaf node atau hasil keputusan sehingga tidak terdapat permasalahan dalam cabang yang kosong.

2.9.5 Entropy

Entropy merupakan suatu parameter yang digunakan untuk mengukur tingkat heterogenitas dari kumpulan data yang digunakan dalam data training. Untuk menentukan nilai *entropy*, rumus yang digunakan sebagai berikut:

$$Entropy (S) = \sum_i^c -P_i \log_2 P_i \quad (2.1)$$

Keterangan:

c = jumlah nilai yang ada dalam atribut target (kelas target).

P_i = Jumlah proporsi sampel (peluang) untuk kelas i.

2.9.6 Information gain

Information gain merupakan ukuran keefektifan suatu atribut dalam mengklasifikasi data. Rumus yang digunakan untuk menghitung informasi *gain* yaitu:

$$Information\ gain (S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{[Sv]}{[S]} Entropy (Sv) \quad (2.2)$$

Keterangan:

A= Atribut.

V= Nilai yang mungkin untuk atribut A.

Values (A) = Himpunan nilai yang mungkin untuk atribut A.

[Sv] = jumlah sampel untuk nilai v.

[S] = Jumlah seluruh sampel data.

Entropy (Sv) = *Entropy* untuk sampel yang memiliki nilai v

BAB III

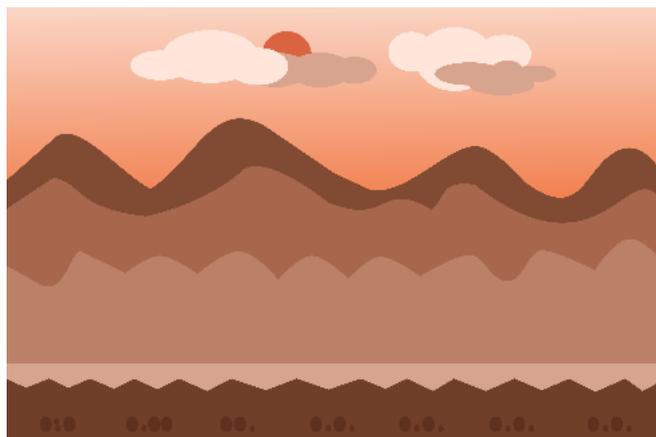
METODE PENELITIAN

3.1 Penjelasan Umum *Game* 2D Belajar Hijaiyah

Game belajar hijaiyah ini merupakan *game* yang memiliki tujuan untuk mengenalkan huruf-huruf hijaiyah mulai dari alif sampai ya'. *Game* hijaiyah ini hanya memiliki satu scene dengan dua *environment* yang berbeda. Berikut gambaran *environment* yang terdapat dalam *game* belajar hijaiyah:

1. *Environment* Pertama

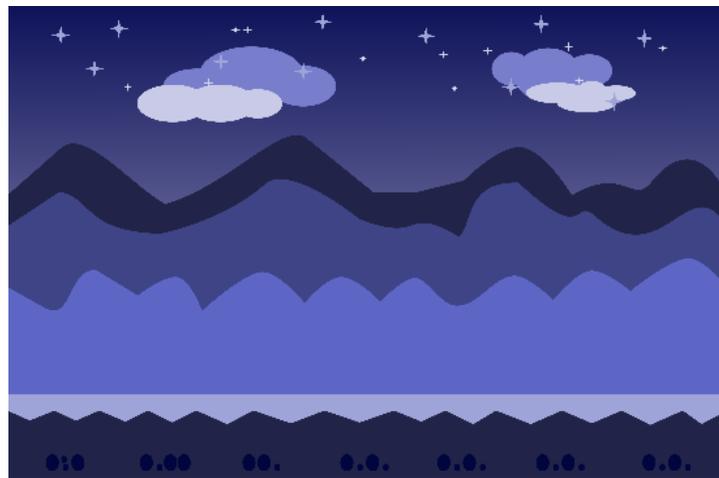
Pada *environment* pertama, background suasana yang akan ditampilkan adalah suasana ketika siang hari atau waktu beraktivitas. Dalam *environment* ini akan ditampilkan *player*, pohon-pohon, gunung dan matahari yang menunjukkan waktu semakin sore. Pada *environment* ini, NPC mulai bergerak mendekati *player* untuk dapat mengganggu perjalanan *player*. Tampilan *environment* pada suasana ini dapat dilihat pada gambar dibawah ini:



Gambar 3. 1 *Environment* siang

2. *Environment* Kedua

Pada *environment* kedua, background yang dilalui dalam perjalanan *player* adalah suasana malam hari disuatu gurun yang memiliki musim salju. Pada *environment* ini, *player* melanjutkan perjalanannya mempelajari huruf hijaiyah sampai menemukan huruf terkahir sebagai garis terakhir. Berikut tampilan interface pada *environment* kedua:



Gambar 3. 2 *Environment* malam

3. Nilai Edukasi

Dalam *game* hijaiyah ini, tentu saja nilai edukasi yang ingin disampaikan adalah pembelajaran mengenai huruf hijaiyah sebagai dasar untuk belajar al-quran. Penjelasan mengenai bacaan hijaiyah ini dapat dilihat oleh *player* ketika *player* menyentuh huruf hijaiyah yang ada dalam *game*. Dengan memanfaatkan *game* untuk pembelajaran huruf hijaiyah, *player* dapat belajar dengan lebih menyenangkan. Target utama dalam pembuatan *game* ini adalah anak-anak dengan usia rentang 3-12 tahun. Tujuan tersebut dikarenakan anak-anak usia dini seharusnya sudah mampu memahami hijaiyah sehingga dapat memahami al-quran.

Walaupun begitu, tidak menutup kemungkinan bahwa orang dewasa pun dapat menggunakan *game* ini sebagai sarana pembelajarn untuk anak-anak mereka.

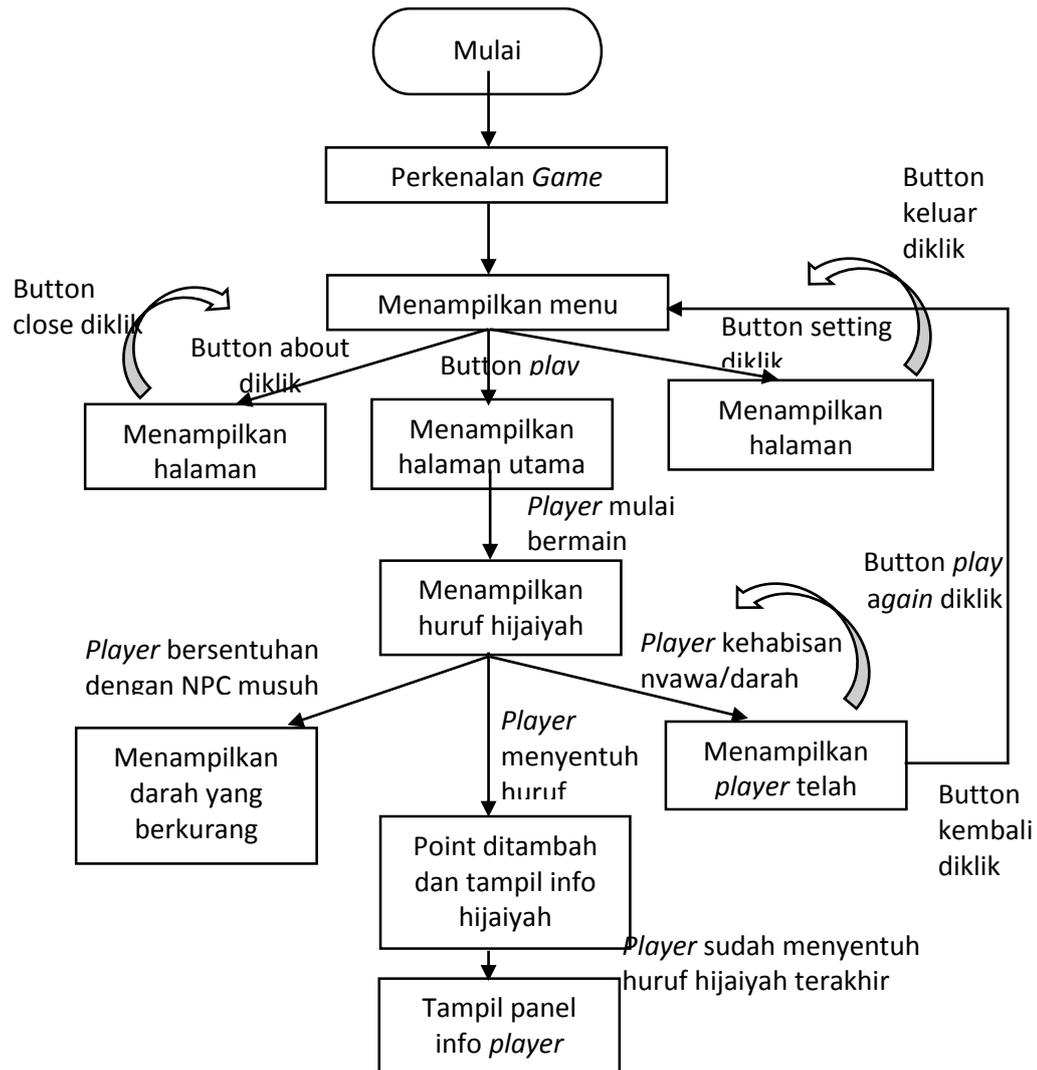
4. Objek Penelitian

Objek penelitian dalam *game* belajar hijaiyah ini adalah implementasi *decision tree*. Dengan tujuan membuat *game* yang menarik, maka dapat memanfaatkan salah satu komponen dalam *game* yaitu musik atau audio. Metode *decision tree* digunakan untuk menentukan bagaimana musik dan audio yang dijalankan dalam *game*. Untuk menggunakan metode *decision tree*, variabel yang digunakan antara lain jarak *player* dengan NPC, kecepatan *player*, dan jarak *player* dengan finish.

3.2 Finite State Machine

Finite state machine merupakan alur proses yang terjadi dari *game* pertama kali dimulai sampai *game* berakhir. Pada FSM ini menggambarkan rancangan alur perjalanan *game* yang dilalui oleh *player* yang dapat dilihat pada gambar 3.8. Terdapat beberapa objek yang memiliki beberapa pergerakan dalam bermain *game* belajar hijaiyah ini. Pada saat *player* mulai memasuki arena *game*, *player* berada ditempat yang identic dengan suasana gurun berwarna coklat. *Player* dapat berjalan dan melompat untuk menyusuri jalan dan mencari huruf-huruf hijaiyah. Selain *player*, terdapat juga karakter NPC musuh yang mengejar *player* dari belakang untuk dapat melukai *player*. Ketika NPC musuh ini menyentuh *player*, maka *player* akan kehilangan stok darah yang dimiliki *player*. *Player* memiliki 3 stok darah, sehingga *player* memiliki batas sejumlah tiga kali untuk menyentuh NPC musuh. Dalam menyusuri jalan, *player* mencari huruf hijaiyah yang ditemuinya dan menyentuhnya untuk mendapatkan score dan mendapatkan pelajaran. Setiap objek

huruf hijaiyah disentuh oleh *player*, maka sistem *game* menampilkan panel yang berisi pembahasan mengenai huruf hijaiyah yang disentuh dan *player* mendapat satu poin.



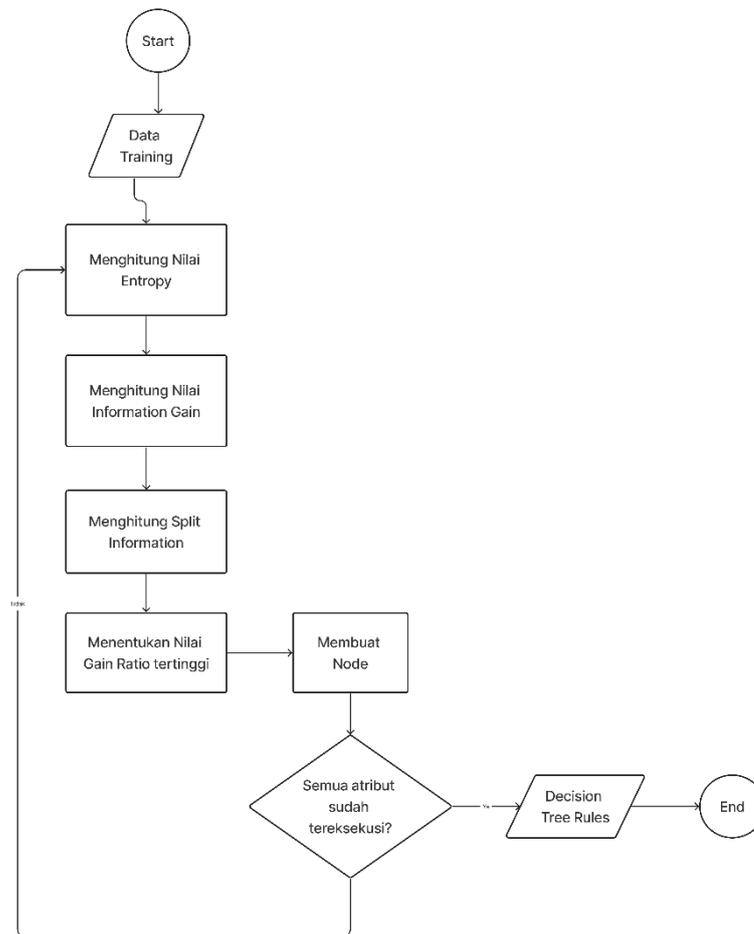
Gambar 3. 3 Finite State Machine Game

Pembelajaran yang ditampilkan seperti keterangan dari huruf hijaiyah, cara membacanya, dan contoh kata yang menggunakan huruf hijaiyah. Selain berfokus kepada mencari huruf hijaiyah untuk mengumpulkan score dan berfokus

menghindari musuh, *player* juga harus berhati-hati terhadap baling-baling yang terletak di jalan. Baling-baling tersebut menghalangi *player* untuk berjalan sehingga *player* harus memerhatikan waktu yang digunakan dalam bermain untuk dapat mencapai garis finish. Apabila *player* sudah berhasil mencapai huruf hijaiyah terakhir yaitu huruf ya, maka *player* berhasil memenangkan *game* dan mendapat score sesuai jumlah huruf hijaiyah yang sudah dipelajari oleh *player*. Ketika *player* sudah berhasil menyelesaikan *game*, maka sistem akan memberikan pilihan kepada *player* untuk bermain lagi atau keluar dari *game*.

3.3 Perancangan *Decision tree*

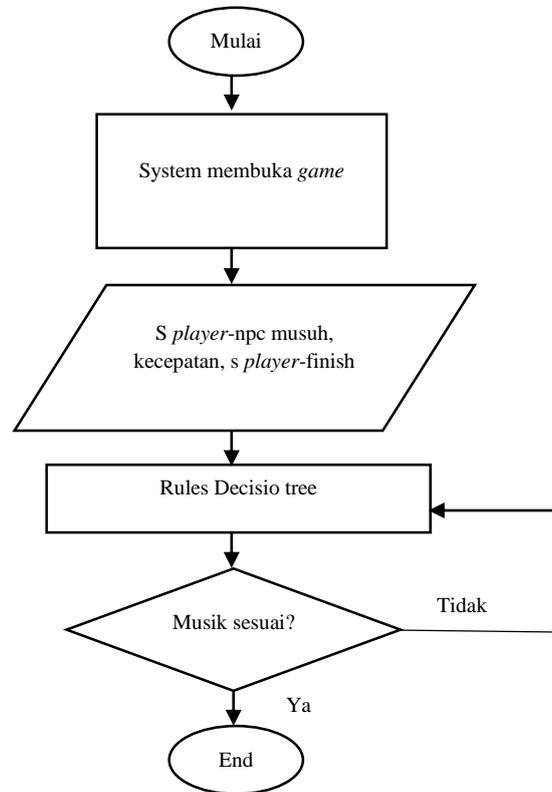
Sistem pengambilan keputusan dalam *game* belajar hijaiyah ini dengan menggunakan metode *decision tree*. Dalam menggunakan metode *decision tree* terdapat beberapa tahapan yang dilakukan yaitu menentukan tujuan/goal, menentukan variabel parameter, menghitung nilai *entropy*, mengolah data training, menghitung nilai keefektifan, dan mendapatkan keputusan dari perhitungan *decision tree*. Untuk lebih memahami implementasi *decision tree* dalam *game*, berikut *flowchart decision tree* dalam *game* belajar hijaiyah:



Gambar 3. 4 Flowchart *Decision tree* Algoritma C4.5

Untuk melakukan perhitungan *decision tree*, langkah pertama yang harus dilakukan adalah menginputkan data training yang akan digunakan. Setelah itu, menghitung nilai *entropy* dan nilai *gain* pada masing-masing atribut. Langkah selanjutnya yaitu menentukan nilai rasio *gain* tertinggi yang menjadi root dari *decision tree* sehingga mendapatkan node 1/ cabang pertama dalam *decision tree*. Untuk menentukan node berikutnya, dilakukan perhitungan nilai *entropy* dan nilai *gain* pada masing-masing atribut dengan kelas yang sama. Ketika semua atribut

sudah tereksekusi sehingga tidak ada cabang lagi, maka *decision tree* sudah terbentuk dengan menghasilkan *rules* yang dapat diimplementasikan dalam *game*.



Gambar 3. 5 Flowchart *Decision tree* dalam *game*

Implementasi *decision tree* dalam *game* dimulai ketika *player* memasuki halaman *game*. Ketika berada didalam *game*, *player* memiliki 3 variabel yaitu jarak posisi *player* dengan musuh, kecepatan *player*, dan jarak posisi *player* dengan finish. Tiga variabel ini kemudian disesuaikan dengan rules yang dihasilkan dari perhitungan *decision tree* sehingga mampu menentukan musik adaptif yang sedang dimainkan dalam *game*.

3.3.1 Menentukan Tujuan

Tujuan merupakan hasil yang ingin dicapai dengan mengimplementasikan metode *decision tree*. Tujuan dalam implelementasi *decision tree* ini adalah menentukan musik yang adaptif dalam *game*. Dengan menggunakan metode ini, dapat diketahui hasil keputusan dari beberapa variabel parameter sehingga dapat mengetahui musik adaptif yang dapat digunakan.

3.3.2 Menentukan Variabel

Setelah mengetahui tujuan, langkah selanjutnya adalah menentukan variabel yang digunakan sebagai parameter untuk menentukan keputusan. Variabel yang digunakan dalam perhitungan *decision tree* ini diantaranya:

1. Musik

Musik merupakan variabel yang menjadi tujuan yang ingin dihasilkan dalam implementasi *decision tree*. Musik yang digunakan dalam *game* adalah musik yang adaptif. Ada tiga kategori yang terdapat dalam variabel musik ini yaitu light, middle, dan high. Musik light menggambarkan musik yang digunakan dalam *game* menggunakan lagu 1 yaitu lagu sholawat. Musik middle menggambarkan musik dengan lagu yang digunakan menggunakan lagu sholawat kedua yang berbeda dengan musik pertama. Musik high menggambarkan musik dengan lagu sholawat ketiga yang berbeda dengan lagu pada saat musik light ataupun musik middle.

2. Jarak *player* dengan NPC

Jarak *player* dengan NPC (Sa) merupakan variabel yang menjadi root atau akar dalam implementasi *decision tree*. Jarak *player* dengan NPC memiliki 3

kategori yaitu jauh, sedang, dan dekat dengan value yang berbeda-beda. Berikut nilai dari masing-masing value jarak *player* dengan NPC:

Tabel 3. 1 Tabel Variabel Kategoris Jarak *player*-NPC

No	Value	Nilai Numerik	Keterangan
1.	Jauh	1	Jarak <i>player</i> -npc >14
2.	Sedang	2	$9 < \text{Jarak } \textit{player}\text{-npc} < 14$
3.	Dekat	0	$0 \leq \text{Jarak } \textit{player}\text{-npc} < 9$

3. Kecepatan *player* berjalan

Kecepatan *player* berjalan (V_p) menjadi variabel kedua yang digunakan dalam implementasi *decision tree*. Kecepatan *player* memiliki 3 kategori sebagai berikut:

Tabel 3. 2 Tabel Atribut Kategoris Kecepatan *player*

No	Value	Nilai Numerik	Keterangan
1.	Lambat	1	Kecepatan <i>player</i> ≤ 1
2.	Sedang	2	$1 < \text{Kecepatan } \textit{player} < 2$
3.	Cepat	0	$2 < \text{Kecepatan } \textit{player} < 3$

4. Jarak *player* dengan garis akhir

Jarak *player* dengan garis akhir(S_b) merupakan jarak *player* dengan huruf hijaiyah terakhir yaitu huruf Ya. Variabel ini memiliki 3 kategori sebagai berikut:

Tabel 3. 3 Tabel variabel Kategoris Jarak *player*-finish

No	Value	Nilai Numerik	Keterangan
1.	Jauh	1	Jarak <i>player</i> -finish ≥ 70
2.	Sedang	2	$40 < \text{Jarak } \textit{player}\text{-finish} < 70$
3.	Dekat	0	$0 \leq \text{Jarak } \textit{player}\text{-finish} < 40$

3.3.3 Data Training

Data yang digunakan sebagai data training pada penelitian ini merupakan data statistik yang berjenis data kategoris. Data kategoris merupakan data yang merepresentasikan karakteristik dari atributnya. Dalam penggunaannya, data kategoris juga memiliki nilai numerik yang biasa digunakan seperti pria/wanita (0/1). Pada penelitian ini data training yang digunakan dengan *decision tree* dibagi menjadi data latih dan data uji sesuai presentase yang diinginkan. Presentase penggunaan data uji dan data latih ada beberapa macam seperti 90:10, 80:20, 70:30, dan 60:40. Diagram statistic yang dapat memperjelas keseluruhan data training yang digunakan dapat dilihat pada gambar 3.1 dan 3.2.

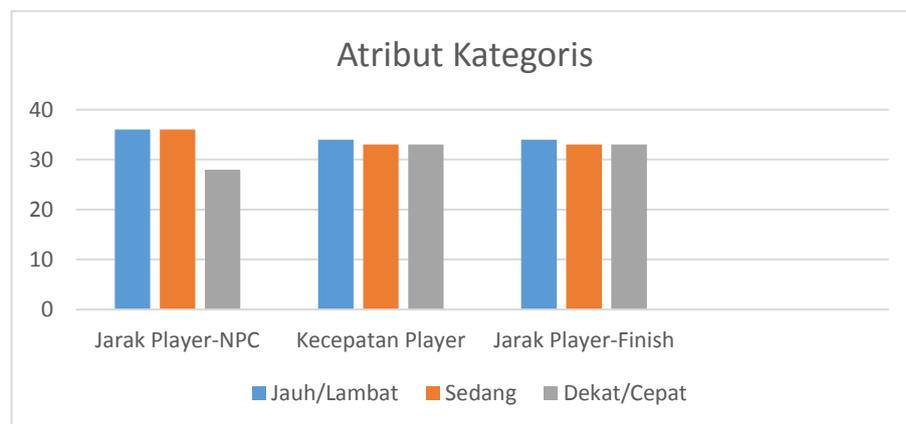


Diagram 3. 1 Atribut Kategoris dalam Data Training

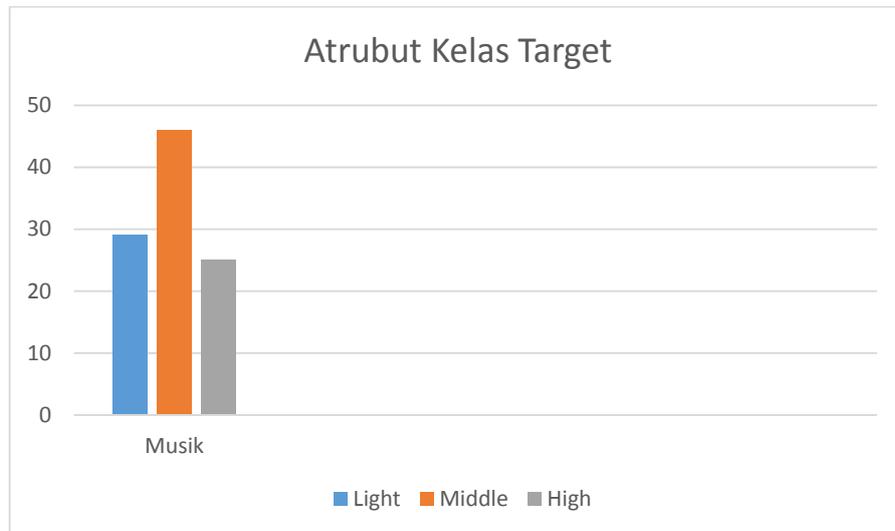


Diagram 3. 2 Atribut Kelas Target dalam Data Training

Data training yang digunakan pada implementasi *decision tree* ini yaitu menggunakan 100 data training. Data training ini diperoleh dari hasil wawancara yang dilakukan kepada beberapa kalangan mahasiswa dari Kota Malang dengan memberikan form Questioner yang berisi pertanyaan-pertanyaan berkaitan dengan musik adaptif. Untuk data hasil wawancara dan responden pada proses pengumpulan data training ini dapat dilihat didalam lampiran penelitian ini. Berikut ini data training yang digunakan:

Tabel 3. 4 Data Training

No	S Player-NPC	Vp	S Player-Finish	Musik
1	Jauh	Lambat	Jauh	Light
2	Jauh	Lambat	Sedang	Light
3	Jauh	Lambat	Dekat	Middle
4	Jauh	Sedang	Jauh	Light
5	Jauh	Sedang	Sedang	Middle
6	Jauh	Sedang	Dekat	Middle
7	Jauh	Cepat	Jauh	Middle

No	S Player-NPC	Vp	S Player-Finish	Musik
8	Jauh	Cepat	Sedang	Middle
9	Jauh	Cepat	Dekat	High
10	Sedang	Lambat	Jauh	Light
11	Sedang	Lambat	Sedang	Light
12	Sedang	Lambat	Dekat	Middle
13	Sedang	Sedang	Jauh	Light
14	Sedang	Sedang	Sedang	Light
15	Sedang	Sedang	Dekat	Middle
16	Sedang	Cepat	Jauh	Middle
17	Sedang	Cepat	Sedang	Middle
18	Sedang	Cepat	Dekat	High
19	Dekat	Lambat	Jauh	Light
20	Dekat	Lambat	Sedang	Light
21	Dekat	Lambat	Dekat	Middle
22	Dekat	Sedang	Jauh	Light
23	Dekat	Sedang	Sedang	Light
24	Dekat	Sedang	Dekat	Middle
25	Dekat	Cepat	Jauh	Middle
26	Dekat	Cepat	Sedang	Middle
27	Dekat	Cepat	Dekat	High
28	Jauh	Lambat	Jauh	Light
29	Jauh	Lambat	Sedang	Middle
30	Jauh	Lambat	Dekat	High
31	Jauh	Sedang	Jauh	Middle
32	Jauh	Sedang	Sedang	Middle
33	Jauh	Sedang	Dekat	High
34	Jauh	Cepat	Jauh	Light
35	Jauh	Cepat	Sedang	Middle
36	Jauh	Cepat	Dekat	Middle
37	Sedang	Lambat	Jauh	Light
38	Sedang	Lambat	Sedang	Light
39	Sedang	Lambat	Dekat	Middle
40	Sedang	Sedang	Jauh	Middle
41	Sedang	Sedang	Sedang	Middle
42	Sedang	Sedang	Dekat	Middle
43	Sedang	Cepat	Jauh	Light
44	Sedang	Cepat	Sedang	Middle
45	Sedang	Cepat	Dekat	Middle

No	S Player-NPC	Vp	S Player-finish	Musik
46	Dekat	Lambat	Jauh	Middle
47	Dekat	Lambat	Sedang	High
48	Dekat	Lambat	Dekat	High
49	Dekat	Sedang	Jauh	Middle
50	Dekat	Sedang	Sedang	Middle
51	Dekat	Sedang	Dekat	Middle
52	Dekat	Cepat	Jauh	Middle
53	Dekat	Cepat	Sedang	High
54	Dekat	Cepat	Dekat	High
55	Jauh	Lambat	Jauh	Light
56	Jauh	Lambat	Sedang	Light
57	Jauh	Lambat	Dekat	Middle
58	Jauh	Sedang	Jauh	Light
59	Jauh	Sedang	Sedang	Light
60	Jauh	Sedang	Dekat	Middle
61	Jauh	Cepat	Jauh	Middle
62	Jauh	Cepat	Sedang	Middle
63	Jauh	Cepat	Dekat	High
64	Sedang	Lambat	Jauh	Middle
65	Sedang	Lambat	Sedang	Middle
66	Sedang	Lambat	Dekat	Middle
67	Sedang	Sedang	Jauh	Middle
68	Sedang	Sedang	Sedang	High
69	Sedang	Sedang	Dekat	High
70	Sedang	Cepat	Jauh	High
71	Sedang	Cepat	Sedang	High
72	Sedang	Cepat	Dekat	High
73	Dekat	Lambat	Jauh	Middle
74	Dekat	Lambat	Sedang	High
75	Dekat	Lambat	Dekat	High
76	Dekat	Sedang	Jauh	High
77	Dekat	Sedang	Sedang	High
78	Dekat	Sedang	Dekat	High
79	Dekat	Cepat	Jauh	High
80	Dekat	Cepat	Sedang	High
81	Dekat	Cepat	Dekat	High
82	Jauh	Lambat	Jauh	Light
83	Jauh	Lambat	Sedang	Light
84	Jauh	Lambat	Dekat	Middle

No	S Player-NPC	Vp	S Player-finish	Musik
85	Jauh	Sedang	Jauh	Light
86	Jauh	Sedang	Sedang	Light
87	Jauh	Sedang	Dekat	Light
88	Jauh	Cepat	Jauh	Middle
89	Jauh	Cepat	Sedang	Middle
90	Jauh	Cepat	Dekat	High
91	Sedang	Lambat	Jauh	Light
92	Sedang	Lambat	Sedang	Light
93	Sedang	Lambat	Dekat	Middle
94	Sedang	Sedang	Jauh	Light
95	Sedang	Sedang	Sedang	Light
96	Sedang	Sedang	Dekat	Middle
97	Sedang	Cepat	Jauh	Middle
98	Sedang	Cepat	Sedang	Middle
99	Sedang	Cepat	Dekat	High
100	Dekat	Lambat	Jauh	Middle

Pada tabel diatas tersedia 100 data training yang digunakan dalam pengujian *decision tree* menggunakan bahasa python. Pada masing-masing value dari setiap atribut ditransform terlebih dahulu sehingga masing-masing data kategorikal yang ada dirubah menjadi data numerik. Contoh hasil transform atribut dan value pada data training dapat dilihat pada tabel 3.7, sedangkan untuk keseluruhan hasil transform dapat dilihat pada lampiran penelitian ini. Kemudian, nilai yang sudah ditransform dikombinasikan sehingga mendapat beberapa hasil musik kelas target yang berbeda-beda. Data training tersebut dibagi dengan presentase yang berbeda-beda kemudian diuji dengan metode *K-Fold Cross validation* sehingga mendapatkan nilai akurasi terbaik yang bisa digunakan untuk menentukan musik adaptif dalam *game*.

Tabel 3. 5 Tabel Transform Data Training

No	S Player-Npc	V Player	S player-finish	Musik
1.	1	1	1	1
2.	1	1	2	1
3.	1	1	0	2
4.	1	2	1	1
5.	1	2	2	2

3.3.4 Nilai Entropy

3.3.4.1 Entropy Musik

Jumlah class target pada penggunaan *decision tree* ini ada 3, yaitu light, middle, dan high. Berikut detail dari jumlah sampel pada masing-masing target:

- Sampel untuk kelas 1/light = 29 (S1)
- Sampel untuk kelas 2/middle = 46 (S2)
- Sampel untuk kelas 3/high = 25 (S3)

$$Entropy = -\frac{29}{100} * \log_2 \left(\frac{29}{100} \right) - \frac{46}{100} * \log_2 \left(\frac{46}{100} \right) - \frac{25}{100} * \log_2 \left(\frac{25}{100} \right) = 1,5332 \quad (3.1)$$

3.3.4.2 Entropy dan Nilai Gain Jarak Player-NPC

Didalam variabel jarak *player* dengan npc musuh terdapat 3 value yaitu jauh, sedang, dan dekat.

L= Light, M= Middle, H=High

S= [29L, 46M, 25H]

Sjauh = [14L, 17M, 5H]

$$Entropy(Sjauh) = -\frac{14}{36} * \log_2\left(\frac{14}{36}\right) - \frac{17}{36} * \log_2\left(\frac{17}{36}\right) - \quad (3.2)$$

$$\frac{5}{36} * \log_2\left(\frac{5}{36}\right) = 1,4366$$

Ssedang = [11L, 18M, 7H]

$$Entropy(Ssedang) = -\frac{11}{36} * \log_2\left(\frac{11}{36}\right) - \frac{18}{36} * \log_2\left(\frac{18}{36}\right) - \frac{7}{36} * \quad (3.3)$$

$$\log_2\left(\frac{7}{36}\right) = 1,4820$$

Sdekat = [4L, 11M, 13H]

$$Entropy(Sdekat) = -\frac{4}{28} * \log_2\left(\frac{4}{28}\right) - \frac{11}{28} * \log_2\left(\frac{11}{28}\right) - \frac{13}{28} * \quad (3.4)$$

$$\log_2\left(\frac{13}{28}\right) = 1,4445$$

$$Nilai Gain = 1,5332 - \left(\frac{36}{100}\right)(1,4366) - \quad (3.5)$$

$$\left(\frac{36}{100}\right)(1,4820) - \left(\frac{28}{100}\right)(1,4445) = 0,0780$$

3.3.4.3 Entropy dan Nilai Gain Kecepatan *Player*(Vp)

Didalam variabel kecepatan *player*(Vp) musuh terdapat 3 value yaitu lambat, sedang, dan cepat.

S = [29L, 46M, 25H]

Slambat = [15L, 14M, 5H]

$$Entropy(Slambat) = -\frac{15}{34} * \log_2\left(\frac{15}{34}\right) - \frac{14}{34} * \log_2\left(\frac{14}{34}\right) - \frac{5}{34} * \quad (3.6)$$

$$\log_2\left(\frac{5}{34}\right) = 1,4546$$

Ssedang = [12L, 15M, 6H]

$$Entropy (Ssedang) = -\frac{12}{33} * \log_2 \left(\frac{12}{33} \right) - \frac{15}{33} * \log_2 \left(\frac{15}{33} \right) - \frac{6}{33} * \quad (3.7)$$

$$\log_2 \left(\frac{6}{33} \right) = 1,4949$$

$$Scepat = [2L, 17M, 14H]$$

$$Entropy (Scepat)) = -\frac{2}{33} * \log_2 \left(\frac{2}{33} \right) - \frac{17}{33} * \log_2 \left(\frac{17}{33} \right) - \frac{14}{33} * \quad (3.8)$$

$$\log_2 \left(\frac{14}{33} \right) = 1,2628$$

$$Nilai Gain = 1,5332 - \left(\frac{34}{100} \right) (1,4546) - \left(\frac{33}{100} \right) (1,4949) - \quad (3.9)$$

$$\left(\frac{33}{100} \right) (1,2628) = 0,1285$$

3.3.4.4 Entropy dan Nilai Gain Jarak Player-Finish

Didalam variabel jarak *player* dengan finish/huruf hijaiyah terakhir musuh terdapat 3 value yaitu jauh, sedang, dan dekat.

$$S = [29L, 46M, 25H]$$

$$Sjauh = [16L, 15M, 3H]$$

$$Entropy(Sjauh) = \frac{16}{34} * \log_2 \left(\frac{16}{34} \right) - \frac{15}{34} * \log_2 \left(\frac{15}{34} \right) - \frac{3}{34} * \quad (3.10)$$

$$\log_2 \left(\frac{3}{34} \right) = 1,3416$$

$$Ssedang = [12L, 14M, 7H]$$

$$Entropy(Ssedang) = \frac{12}{33} * \log_2 \left(\frac{12}{33} \right) - \frac{14}{33} * \log_2 \left(\frac{14}{33} \right) - \frac{7}{33} * \quad (3.11)$$

$$\log_2 \left(\frac{7}{33} \right) = 1,5300$$

$$Sdekat = [1L, 17M, 15H]$$

$$Entropy(Sdekat) = \frac{1}{33} * \log_2 \left(\frac{1}{33} \right) - \frac{17}{33} * \log_2 \left(\frac{17}{33} \right) - \frac{15}{33} * \log_2 \left(\frac{15}{33} \right) \quad (3.12)$$

$$\log_2 \left(\frac{15}{33} \right) = 1,1628$$

$$Nilai Gain = 1,5332 - \left(\frac{34}{100} \right) (1,3416) - \left(\frac{33}{100} \right) (1,5300) - \left(\frac{33}{100} \right) (1,1628) = 0,1884 \quad (3.13)$$

Perhitungan Nilai Gain untuk Jarak Player-Finish jauh

Vp (Kecepatan player)

Slambat = [8L, 4M, 0H]

Ssedang =[6L, 4M, 1H]

Scepat = [2L, 7M, 2H]

$$\begin{aligned} Gain (Sjauh, Vp) &= Entropy (Sjauh) - 12/ \\ &34(Entropy (Vlambat)) - 11/34(Entropy(Vsedang)) - 11/ \\ &34(Entropy (Vcepat)) \quad (3.14) \\ &= 1,3416 - 12/34(0) - 11/34(1,3221) - 11/34(1,3092) \\ &= 0,4902 \end{aligned}$$

Jarak Player-Npc

Sjauh= [8L, 4M, 0H]

Ssedang =[6L, 5M, 1H]

Sdekat= [2L, 6M, 2H]

$$\begin{aligned}
Gain (Sjauh, Sb) &= Entropy(Sjauh) - \frac{12}{34} * Entropy(Sjauh) - \\
\frac{12}{34} * Entropy(Ssedang) - \frac{10}{34} * Entropy(Sdekat) &= 1,3416 - \frac{12}{34} * \\
0 - \frac{12}{34} * 1,3250 - \frac{10}{34} * 1,3709 &= 0,4707
\end{aligned} \tag{3.15}$$

Perhitungan Entropy dan Nilai Gain untuk Jarak Player dengan Finish sedang

Vp (Kecepatan Player)

Slambat = [7L, 2M, 2H]

Ssedang =[5L, 4M, 2H]

Scepat = [0L, 8M, 3H]

$$\begin{aligned}
Gain (Ssedang, Vp) &= Entropy(Ssedang) - \frac{11}{33} * \\
Entropy(Vlambat) - \frac{11}{33} * Entropy(Vsedang) - \frac{11}{33} * \\
Entropy(Vcepat) &= 1,5300 - \frac{11}{33} * 1,3092 - \frac{11}{33} * 1,4949 - \frac{11}{33} * \\
0 &= 0,5953
\end{aligned} \tag{3.16}$$

Jarak Player-Npc

Sjauh = [5L, 7M, 0H]

Ssedang =[5L, 5M, 2H]

Sdekat = [2L, 2M, 5H]

$$\begin{aligned}
Gain (Sjauh, Sb) &= Entropy(Sjauh) - \frac{12}{33} * Entropy(Sjauh) - \frac{12}{33} * \\
Entropy(Ssedang) - \frac{9}{33} * Entropy(Sdekat) &= 1,5300 - \frac{12}{33} * 0 - \\
\frac{12}{33} * 1,4833 - \frac{9}{36} * 1,4355 &= 0,5991
\end{aligned} \tag{3.17}$$

3.3.5 Nilai Akurasi

Dengan menggunakan algoritma C4.5, kita dapat menghitung akurasi nilai yang dihasilkan dari perhitungan pada data training. Untuk menghitung nilai akurasi, dapat digunakan dengan menggunakan fungsi yang disediakan library yaitu fungsi `predict`. Fungsi `predict` digunakan dengan melihat nilai `xtrain`, `xtest`, `ytrain`, dan `ytest` dari data training yang digunakan. Dengan menggunakan fungsi `train_test_split`, nilai variabel yang digunakan sebagai parameter dapat diinputkan. Setelah itu, dapat dihasilkan nilai akurasi yang dihasilkan. Berikut ini baris kode yang digunakan untuk menghitung nilai akurasi perhitungan *decision tree* dengan menggunakan algoritma C4.5:

```
//Identifikasi pembagian data uji dan latih dengan fungsi
train_test_split()

//Menggunakan fungsi DecisionTreeClassifier untuk membuat pohon
keputusan dengan kriteria entropy

tree_dataset.fit(xtrain, ytrain)

y_pred = tree_dataset.predict(xtest)

//Menghitung nilai confusion matrix

print("Confusion Matrix")

print(cm)

akurasi=classification_report(ytest,y_pred)

//Nilai akurasi =accuracy_score(ytest,y_pred)

print("Tingkat Akurasi : %d persen" %(akurasi*100))
```

Gambar 3. 6 Fungsi untuk menghitung nilai akurasi

Dari perhitungan tersebut didapatkan nilai akurasi pada data training musik adaptif untuk menentukan jenis musik adaptif yang akan digunakan dalam *game* ketika *player* sedang bermain. Dengan musik yang adaptif, *player* dapat secara cepat mengetahui seberapa jauh mereka bermain dan mengetahui keberadaan musuh yang mengejar. Berikut ini presentase yang digunakan dalam mencari nilai akurasi dalam implementasi *decision tree*:

Tabel 3. 6 Pembagian presentase akurasi

No	Presentase
1.	90:10
2.	80:20
2.	70:30
3.	60:40

Terdapat 3 presentase yang digunakan pada implementasi *decision tree* ini yaitu 90% data latih dan 10% data uji, 80% data latih dan 20% data uji, 70% data latih dan 30% data uji, dan 60% data latih dan 40% data uji. Dari pengujian beberapa presentase ini didapatkan bahwa presentase yang menghasilkan nilai akurasi terbaik adalah presentase 60% data latih dan 40% data uji.

3.3.6 *Cross validation* menggunakan K-Fold

Selain menghitung nilai akurasi dengan membagi data menjadi beberapa presentase, perhitungan *decision tree* dapat diuji validasinya dengan menggunakan *Cross validation* K-Fold. Untuk menggunakan *cross validation*, data training dibagi menjadi beberapa kelompok, kemudian masing-masing kelas dihitung nilai akurasinya dan akhirnya semua nilai akurasi dirata-rata sehingga menghasilkan nilai akurasi terbaik yang didapatkan dari data training yang digunakan. Berikut ini bentuk validasi menggunakan *cross validation*:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

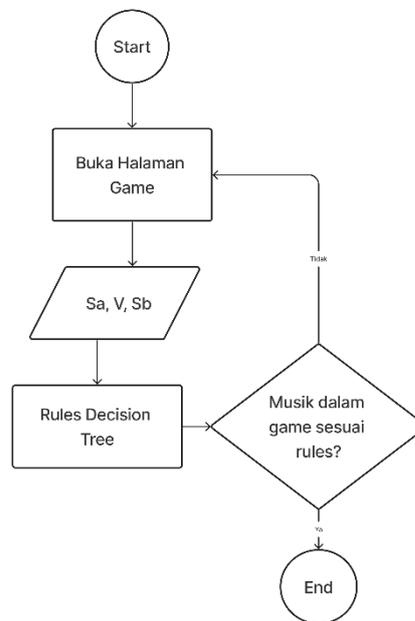
Gambar 3. 7 *Cross validation K-Fold*

	Data Latih
	Data Uji

Tahapan yang digunakan dalam menggunakan cross-validation sebagai berikut:

1. Jumlah data training yang digunakan dibagi menjadi k bagian
2. Melakukan perhitungan pada fold-1 yaitu pada fold pertama dimana bagian pertama data digunakan sebagai data uji dan sisa bagiannya, bagian 2-k menjadi data latih.
3. Melanjutkan proses pada bagian fold-2 dan seterusnya, dimana berarti bagian 2 menjadi data uji dan bagian yang lain sebagai data latih, begitu juga dengan fold-fold selanjutnya.
4. Setelah sampai pada iterasi ke-k, maka semua nilai akurasi dari perhitungan mulai dari fold-1 sampai fold-k dijumlahkan untuk mencari rata-rata akurasi. Rata-rata akurasi yang dihasilkan kemudian menjadi akurasi final (Mardiana, Kusnandar, & Satyahadewi, 2022).

3.3.7 Skenario Pengujian



Gambar 3. 8 Skenario Pengujian Sistem

Skenario pengujian implementasi *decision tree* dalam *game* ini dilakukan dengan menginputkan variabel seperti jarak *player* dengan NPC, kecepatan *player*, dan jarak *player* dengan garis finish. Data-data ini didapatkan secara otomatis didalam *game* menyesuaikan dengan posisi *player* berada. Sedangkan untuk variabel kecepatan, *player* bisa menambahkan kecepatan secara manual dengan tombol yang disediakan didalam halaman *game*. Setelah data-data didapatkan, kemudian mencocokkan rules dari *decision tree* dengan musik yang sedang digunakan dalam *game*. Ketika musik yang diputar sama dengan musik adaptif yang dihasilkan oleh *decision tree*, maka pengujian rules *decision tree* sesuai.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Pada bab ini membahas tentang implementasi dari desain penelitian yang sudah direncanakan pada bab 3. Pada bab ini membahas tentang perancangan *game* 2D belajar hijaiyah, implementasi *decision tree* dalam menentukan keputusan untuk penggunaan musik adaptif dan pengujian data testing terhadap hasil algoritma yang dihasilkan.

4.1.1 Kebutuhan Perangkat Keras

Perangkat keras merupakan hardware yang digunakan dalam implementasi *decision tree* untuk menentukan musik adaptif pada *game* belajar hijaiyah. Berikut ini daftar kebutuhan perangkat keras yang digunakan dalam proses implementasi:

Tabel 4. 1 Kebutuhan perangkat keras

No	Perangkat Keras	Spesifikasi
1.	Processor	Intel Inside Core i5
2.	RAM	4 Gb
3.	VGA	Intel Graphics HD Family
4.	Monitor	12
5.	Speaker	On
6.	Mouse	On
7.	Keyboard	On

4.1.2 Kebutuhan Perangkat Lunak

Perangkat lunak merupakan perangkat aplikasi dalam komputer yang digunakan dalam implementasi *game* belajar hijaiyah. Berikut daftar kebutuhan perangkat lunak yang digunakan:

Tabel 4. 2 Kebutuhan perangkat lunak

No	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	Windows 8 (64) bit
2.	<i>Game Engine</i>	Unity v
3.	Desain 2D	Inkscape
5.	Implementasi <i>Decision tree</i>	Google Collab, Python
6.	Script <i>Game</i>	C#, Monospace

4.1.3 Implementasi *Decision tree*

Implementasi metode *decision tree* dalam *game* ini dilakukan dengan menggunakan teknologi yang disediakan oleh google yaitu google collab. Google collab merupakan situs website yang memberikan kemudahan dalam mengimplementasikan *decision tree* dalam baris code. Untuk bahasa pemrograman yang digunakan dalam implementasi ini yaitu menggunakan bahasa python, dan algoritma *decision tree* yang digunakan yaitu algoritma C4.5. Dalam menentukan keputusan, algoritma C4.5 menggunakan *entropy* dan nilai *gain* sebagai parameter untuk membentuk suatu node. Karena didalam penelitian ini menggunakan bahasa pemrograman python, oleh karena itu perhitungan untuk mencari nilai *entropy* dan nilai *gain* tidak dilakukan secara manual, tetapi dengan menggunakan library yang sudah disediakan oleh *scikit-learn*. Untuk dapat memahami tahapan-tahapan yang dilakukan dalam membentuk *decision tree* dapat dilihat pada gambar 4.1.

```

<Program Decision Tree untuk Menentukan Musik
Adaptif>

Import library numpy, pyplot, pandas,
Model_Selection, library graphviz,
DecisionTreeClassifier;

Input Dataset Data Training -->
Data_TrainingMusik_LMH.csv

read--> Dataset

transform dataset kategori --> numerikal

Separate class target and attribute

Split data latih dan data uji -->
train_test_split()

Set Model Klasifikasi =
tree.DecisionTreeClassifier(criterion="gini",
random_state=500, )

Set Model_Klasifikasi =
Model.Klasifikasi.fit(xtrain,ytrain)

set nilai prediksi=ModelKlasifikasi.predict(data
uji)

set nilai matrix confusi =
confusion_matrix(ytest, nilai prediksi)

Akurasi = accuracy_score(ytest,y_prediksi)

export_graphviz --> Tree_Musik.dot

read --> Tree_Musik.dot as fig

Output Decision Tree Rules

```

Gambar 4. 1 Pseudocode *Decision tree* algoritma c4.5

Untuk melakukan perhitungan *decision tree*, pada penelitian ini menggunakan beberapa library yang sudah tersedia dalam Scikit-Learn. Beberapa *library* yang digunakan diantaranya numpy, pandas, sklearn, KFold yang digunakan dalam proses validasi, serta beberapa fungsi yang ada didalam masing-

masing library. Kemudian memasukkan dataset yang digunakan sebagai data training dalam proses perhitungan *decision tree*. Dataset yang digunakan memiliki data yang berjumlah 100. Setelah data berhasil dibaca oleh sistem, kemudian data yang memiliki value kategorikal diubah menjadi data numerik dengan menggunakan fungsi Encoder. Fungsi Encoder digunakan untuk mengubah value data kategorikal menjadi numeric.

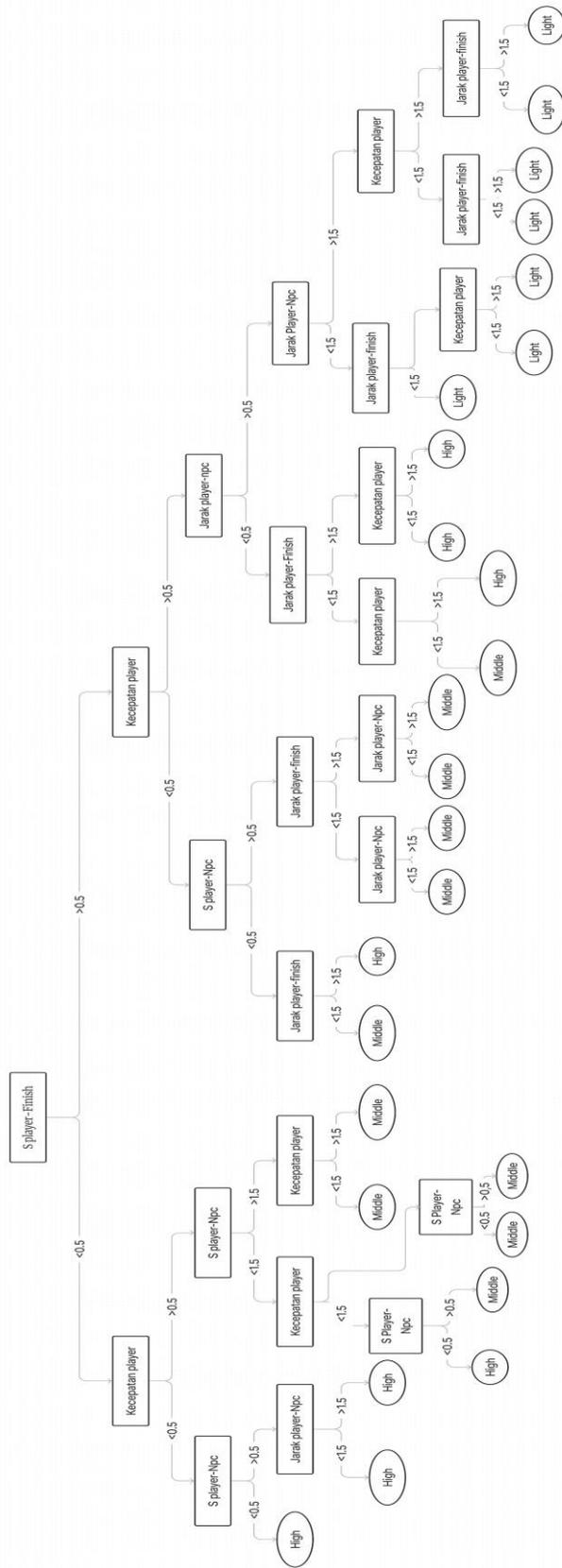
Langkah selanjutnya setelah data berhasil dibaca oleh sistem yaitu memisahkan atribut target dengan atribut yang lainnya. Atribut target pada penelitian ini yaitu atribut musik. Sedangkan atribut yang digunakan sebagai parameter yaitu jarak *player* dengan npc, kecepatan *player*, dan jarak *player* dengan finish. Setelah itu menggunakan fungsi `train_test_split` untuk membagi data latih dan data uji. Pada gambar diatas terjadi proses import fungsi `train_test_split` dari sklearn untuk proses pembagian data, kemudian terdapat `accuracy_score` untuk menghitung nilai akurasi yang didapat, dan juga `tree` untuk membentuk rules keputusan.

Langkah selanjutnya yaitu melakukan inisialisasi data latih dan data training serta presentase pembagiannya. Pada penelitian ini presentase yang digunakan yaitu 90% data latih dan 10% data uji. Pemilihan presentase ini didasarkan oleh nilai akurasi terbaik yang dihasilkan pada masing-masing presentase. Sebelum menghitung nilai akurasi, kita menghitung nilai confusion matrix. Setelah itu, nilai akurasi dihitung dengan menggunakan fungsi `accuracy_score`. Nilai akurasi yang dihasilkan dari presentase 90:10 adalah 70%. Berikut ini tabel perbandingan nilai akurasi dari 4 presentase yang digunakan:

Tabel 4. 3 Tabel Presentase pengujian akurasi *decision tree*

No	Presentase	Nilai Akurasi
1.	90:10	70%
2.	80:20	35%
2.	70:30	40%
3.	60:40	45%

Setelah melakukan perhitungan *decision tree* dengan menggunakan algoritma C4.5, kita telah memiliki graph tree yang berisi rules-rules yang digunakan dalam *game*. Untuk melihat rules-rules yang dihasilkan dari *decision tree* dalam bentuk graph, dapat digunakan library graphviz yang disediakan. Setelah melakukan perhitungan *decision tree* dengan menggunakan python, kemudian didapatkan pohon keputusan yang berisi rules-rules yang bisa diimplementasikan dalam *game*. Struktur pohon keputusan berbentuk graph tree yang dihasilkan dari perhitungan *decision tree* dapat dilihat pada gambar 4.2.



Gambar 4. 2 Graph tree dari *decision tree*

Rules yang dihasilkan dengan perhitungan *decision tree* menggunakan algoritma C4.5 dengan bahasa python sebagai berikut:

1. If sa dekat, sb dekat, dan vp cepat maka musik high
2. If sa dekat, sb dekat dan vp lambat maka musik high
3. If sa dekat, sb jauh dan vp cepat maka musik middle
4. If sa dekat, sb jauh dan vp lambat maka musik middle
5. If sa dekat, sb jauh dan vp sedang maka musik high
6. If sa dekat, sb sedang dan vp cepat maka musik high
7. If sa dekat, sb sedang dan vp lambat maka musik high
8. If sa dekat, sb sedang dan vp sedang maka musik high
9. If sa dekat, sb dekat dan vp sedang maka musik middle
10. If sa jauh, sb dekat dan vp cepat maka musik high
11. If sa jauh, sb dekat dan vp lambat maka musik middle
12. If sa jauh, sb dekat dan vp sedang maka musik middle
13. If sa jauh, sb jauh, dan vp cepat maka musik middle
14. If sa jauh, sb jauh dan vp lambat maka musik light
15. If sa jauh, sb jauh dan vp sedang maka musik light
16. If sa jauh, sb sedang dan vp cepat maka musik middle
17. If sa jauh, sb sedang, dan vp lambat maka musik light
18. If sa jauh, sb sedang, dan vp sedang, maka musik light
19. If sa sedang, sb dekat, dan vp cepat maka musik high
20. If sa sedang, sb dekat dan vp lambat maka musik middle
21. If sa sedang, sb dekat dan vp sedang maka musik middle

22. If sa sedang, sb jauh dan vp cepat maka musik middle
23. If sa sedang, sb jauh dan vp sedang maka musik light
24. If sa sedang, sb sedang, dan vp sedang, maka musik light
25. If sa sedang, sb sedang dan vp cepat maka musik middle
26. If sa sedang, sb jauh dan vp lambat maka musik light.
27. If sa sedang, sb sedang, dan vp lambat, maka musik light

Rata-Rata Nilai Akurasi dari beberapa Presentase

Tabel 4. 4 Tabel rata-rata akurasi perhitungan presentase

No	Presentase	Nilai Akurasi
1.	90:10	70%
2.	80:20	35%
2.	70:30	40%
3.	60:40	45%
Akurasi		47,5%

Nilai akurasi yang didapatkan dengan mencari rata-rata akurasi dari setiap presentase adalah sebesar 47,5% atau 0,47. Nilai akurasi ini didapatkan dari presentase 90:10, 80:20, 70:30, dan 60:40. Dari masing-masing nilai akurasi yang diperoleh, maka presentase nilai akurasi yang digunakan untuk membuat pohon keputusan adalah presentase dengan nilai akurasi terbaik yaitu 90% data latih dan 10% data uji.

K fold Cross validation

Untuk mendapatkan nilai akurasi, pada penelitian ini menggunakan *K fold Cross validation*. *K fold Cross validation* digunakan dengan membagi seluruh data training yang ada menjadi beberapa bagian. Dalam penelitian ini, terdapat 100 data training yang digunakan dan data dibagi dalam 10 bagian. Pada masing-masing bagian memiliki nilai akurasi masing-masing yang akan dijumlahkan untuk mencari

nilai rata-rata dari akurasi yang didapatkan. Berikut ini tabel penjelasan mengenai perhitungan menggunakan *K fold Cross validation*:

Tabel 4. 5 Pengujian K Fold Cross Validation

No	Data	Nilai Akurasi
1.	1-10	0.8
2.	11-20	0.6
3.	21-30	0.1
4.	31-40	0.5
5.	41-50	0.7
6.	51-60	0.2
7.	61-70	0.7
8.	71-80	0.4
9.	81-90	0.7
10.	91-100	0.7
Rata-Rata Akurasi		0.54

Dari perhitungan nilai akurasi menggunakan *K fold Cross validation*, hasil nilai akurasi rata-rata yang didapatkan adalah 54%. Berdasarkan hasil perhitungan ini, dapat dilihat bahwa dengan menggunakan *K-Fold Cross validation* mampu melebihi 50%, sehingga rules yang dihasilkan dari perhitungan *decision tree* dapat diimplementasikan dalam *game* belajar hijaiyah.

Setelah mencari nilai akurasi dengan menggunakan presentase pembagian data latih dan data uji dan menggunakan *K fold Cross validation*, nilai akurasi terbaik yang digunakan merupakan nilai akurasi dari perhitungan rata-rata akurasi menggunakan presentase pembagian data latih dan data uji. Akan tetapi, nilai akurasi akhir yang didapat dari kedua cara tersebut tidak memiliki perbedaan yang

signifikan sehingga dapat disimpulkan bahwa akurasi yang dihasilkan sudah sesuai dan mampu melebihi 50% kebenaran data.

4.1.4 Implementasi *Decision tree* dalam *Script Game*

Pada *game* 2D belajar hijaiyah, rules yang dihasilkan dari perhitungan *decision tree* digunakan dalam menentukan musik adaptif yang digunakan dalam permainan *game*. Berikut ini value musik dan detail musik yang digunakan dalam *game*:

Tabel 4. 6 Tabel Detail Musik Adaptif

No	Musik	Volume	Pitch
1.	Light	0.4	1
2.	Middle	0.8	1.5
3.	High	1	2

Implementasi penentuan musik adaptif yang terdapat dalam *game* dengan menggunakan visual studio code dengan menggunakan bahasa C# dapat dilihat pada gambar dibawah ini:

```

else if( sa>=14){
    if(v<=1.9f){
        if(sb>=40 && sb<=69){
            VolLight();
        }else if(sb>=0 && sb<=39){
            VolLight();
        }
    }else if(v>=2 && v<=2.9f){
        if(sb>=40 && sb<=69){
            VolLight();
        }else if(sb>=0 && sb<=39){
            VolMiddle();
        }
    }else if(v>=3 && v<=3.9f){
        if(sb>=40 && sb<=69){
            VolMiddle();
        }else if(sb>=0 && sb<=39){
            VolMiddle();
        }
    }else if(sb>=70){
        VolLight();
    }
    musuh.mute=false;
}

```

Gambar 4. 3 Implementasi rules jarak jauh

Pada gambar diatas merupakan implementasi rules yang sudah dihasilkan dari perhitungan *decision tree* dalam script *game*. Script *game* yang digunakan pada implementasi ini yaitu bahasa C# dan dibuat disoftware visual studio code. Gambar

diatas merupakan rules untuk rules dengan jarak *player* dengan NPC jauh. Berikut ini keterangan istilah yang digunakan dalam code implementasi dalam *game*:

Tabel 4. 7 Tabel Keterangan istilah dalam code

No	Istilah	Keterangan
1.	Sa	Jarak <i>player</i> dengan NPC Musuh
2.	V	Kecepatan <i>player</i>
3.	Sb	Jarak <i>player</i> dengan finish
4.	VolLight()	Fungsi untuk musik light
5.	VolMiddle()	Fungsi untuk musik middle
6.	VolHigh()	Fungsi untuk musik high
7.	VolNormal()	Fungsi untuk musik normal

Pada perhitungan *decision tree*, atribut target dari data training yang digunakan adalah kolom musik. Kolom musik sendiri memiliki 3 value seperti light, middle, dan high seperti yang sudah dijelaskan dalam Bab 3. Jenis musik adaptif ini ditentukan dari rules yang dihasilkan dari perhitungan *decision tree*. Akan tetapi, dikarenakan terdapat beberapa kondisi dimana rules tidak dapat memenuhi kondisi keseluruhan *player*, maka pada penelitian ini digunakan musik adaptif keempat yaitu normal yang digunakan ketika semua rules dalam *decision tree* tidak dapat menentukan musik adaptif. Musik normal ini digunakan dalam fungsi VolNormal yang berisi serangkaian instruksi yang dijalankan pada saat fungsi tersebut dipanggil.

```

else if( sa>=9 && sa<=13.9f){
    if (v<=1.9f){
        if (sb>=70){
            VolLight();
            //lagu.text="Light";
        }else if(sb>=40 && sb<=69){
            VolLight();
            //lagu.text="Light";
        }else if(sb>=0 && sb<=39){
            VolMiddle();
            //lagu.text="Middle";
        }}else if(v>=2 && v<=2.9f){
if (sb>=70){
            VolLight();
            // lagu.text="Light";
        }else if(sb>=40 && sb<=69){
            VolMiddle();
            //lagu.text="Middle";
        }else if(sb>=0 && sb<=39){
            VolMiddle();
            //lagu.text="Middle";
        }}else if(v>=3 && v<=3.9f){
if (sb>=70){
            VolLight();
            //lagu.text="Middle";
        }else if(sb>=40 && sb<=69){
            VolMiddle();
        }else if(sb>=0 && sb<=39){
            VolHigh(); }; }musuh.mute=true; }

```

Gambar 4. 4 Implementasi rules jarak sedang

Pada gambar tersebut merupakan kelanjutan implementasi rules didalam *game* belajar hijaiyah. Implementasi pada gambar tersebut digunakan ketika jarak *player* dengan NPC musuh (*sa*) sedang. Value sedang ini berada diantara 8-13. Didalam gambar tersebut terdapat 9 rules yang diterapkan dalam *game*.

```

if( sa>=0 && sa<=12.9f){
    if (v<=1.9f){
        if (sb>=70){
            VolMiddle();
        }else if(sb>=40 && sb<=69){
            VolMiddle();
        }else if(sb>=0 && sb<=39){
            //lagu.text="Middle";
            VolHigh();
        }
    }else if(v>=2 && v<=2.9f){
        if (sb>=0 && sb<=39){
            VolHigh();
        }else if(sb>=40 && sb<=69){
            VolHigh();
        }else if(sb >=70 ){
            VolMiddle();
        }
    }else if(v>=3 && v<=3.9f){
        if (sb>=70){
            VolMiddle();
        }else if(sb>=40 && sb<=69){
            VolHigh();
        }else if(sb>=0 && sb<=39){
            VolHigh();};musuh.mute=true;}

```

Gambar 4. 5 Implementasi rules jarak dekat

Pada gambar diatas merupakan implementasi lanjutan dari code sebelumnya. Pada gambar tersebut merupakan implementasi untuk jarak *player* denga NPC musuh dekat. Pada implementasi ini terdapat 9 rules yang digunakan. Selain itu, apabila keseluruhan rules yang dihasilkan *decision tree* tidak memenuhi kondisi *player*, maka musik yang digunakan yaitu musik normal dengan memanggil fungsi VolNormal.

4.1.5 Implementasi Game 2D Belajar Hijaiyah

a. Halaman Selamat Datang



Gambar 4. 6 Halaman Splash

Tampilan *user interface* pada gambar diatas merupakan tampilan pertama yang ditampilkan kepada user ketika membuka *game*. Tampilan ini berisi ucapan selamat datang kepada user ketika membuka *game* supaya user memiliki ketertarikan untuk bermain *game*. Untuk menuju halaman selanjutnya user dapat menekan tombol button next yang diarahkan menuju halaman pengenalan *game*.

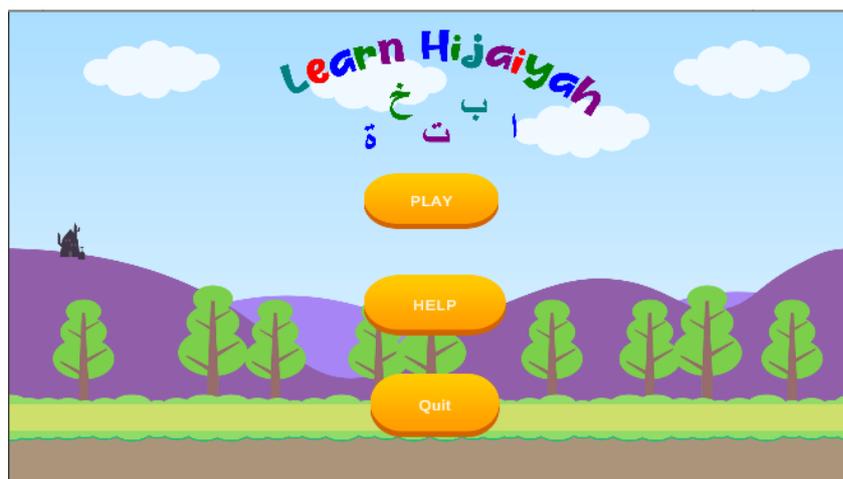
b. Halaman Pengenalan Game



Gambar 4. 7 Halaman Introduction *game*

Tampilan *user interface* pada gambar diatas merupakan halaman pengenalan *game*. Dalam halaman tersebut, user diberi informasi terkait *game* belajar hijaiyah secara umum. Untuk melanjutkan ke halaman berikutnya, user dapat menekan tombol button *let's play*.

c. Halaman Menu



Gambar 4. 8 Halaman Menu utama

Halaman interface diatas merupakan tampilan halaman utama yang berisi menu-menu yang dapat dijalankan oleh user. Terdapat 3 menu yang ditampilkan yaitu menu *play* yang membawa user menuju halaman *game*, kemudian menu *help* yang membawa user menuju halaman peraturan dalam *game*, dan terdapat menu *quit* yang membawa user untuk keluar dari permainan belajar hijaiyah.

d. Halaman Bantuan

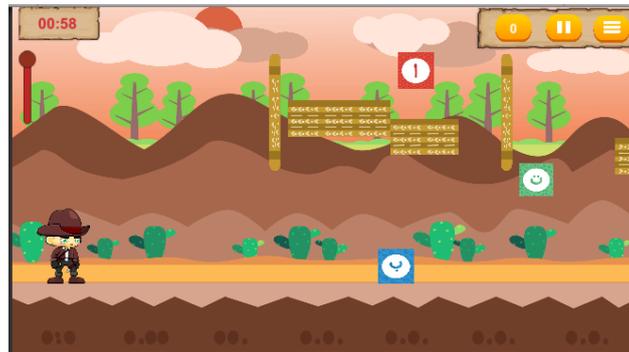


Gambar 4. 9 Halaman Bantuan

Halaman bantuan berisi beberapa peraturan dalam permainan yang harus dilakukan oleh *player* ketika bermain *game* belajar hijaiyah. Setelah user selesai membaca peraturan, user dapat menekan tombol button ikon back untuk kembali menuju halaman menu.

e. Halaman *game*

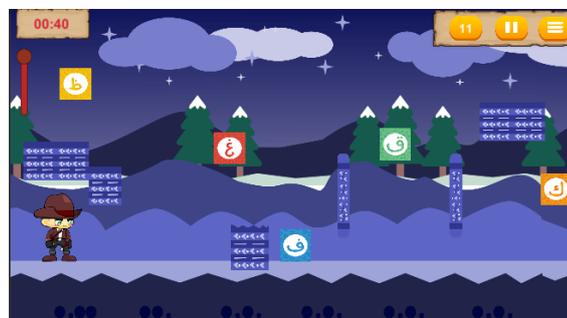
1. *Environment* satu



Gambar 4. 10 Halaman *environment* siang

Tampilan *user interface* diatas merupakan tampilan pada halaman *game*. Pada halaman *game* terdapat 2 *environment* yang ditampilkan dan tampilan pada gambar diatas merupakan tampilan pada *environment* pertama. Pada *environment* satu *player* berjalan ditempat pegunungan ketika siang menjelang petang. Suasana dalam *environment* satu dapat didukung dengan adanya matahari dan warna awan dan pegunungan yang dibuat cerah namun sedikit gelap. Pada halaman permainan, terdapat beberapa tampilan seperti waktu yang berjalan, nyawa/darah yang dimiliki user, score yang didapat user, tombol button untuk menghentikan *game* sementara/ *pause*, dan hamburger menu untuk menampilkan menu kepada user.

2. *Environment* dua



Gambar 4. 11 *Environment* malam

Tampilan *user interface* diatas merupakan tampilan *environment* kedua yang dilalui oleh *player* ketika bermain. Dalam *environment* dua, *player* berada dalam suasana malam hari. Suasana malam hari didukung dengan adanya bintang-bintang, warna objek yang gelap dan beberapa pohon salju yang menunjukkan suasana dingin pada malam hari.

3. Karakter *player*



Gambar 4. 12 Karakter *player*

Gambar diatas merupakan gambar karakter *player* yang dimainkan oleh user. Karakter tersebut dipilih dengan sifat karakternya yang dapat dilihat dari bajunya yang merupakan seorang petualang. Dengan begitu, karakter tersebut cocok untuk merepresentasikan *player* yang sedang berpetualang untuk mencari huruf hijaiyah.

4. Karakter NPC/musuh



Gambar 4. 13 Karakter Musuh

Gambar diatas merupakan gambar dari karakter NPC musuh dalam *game* belajar hijaiyah ini. Karakter musuh dalam *game* melakukan aksi dengan mengejar

player dari belakang dengan kecepatan yang meningkat seiring bertambahnya waktu. NPC musuh dapat mengurangi darah *player* setiap kali menyentuh *player*.

5. Huruf hijaiyah



Gambar 4. 14 Huruf hijaiyah

Gambar diatas merupakan gambar huruf hijaiyah yang merupakan objek yang dicari oleh *player*. Setiap *player* menyentuh huruf hijaiyah, sistem akan menampilkan panel yang berisi pengetahuan terkait huruf hijaiyah yang disentuh oleh user.

6. Panel Informasi Waktu Habis



Gambar 4. 15 Halaman informasi time offer

Tampilan *user interface* pada gambar diatas merupakan tampilan panel ketika *player* sudah kehabisan waktu dalam bermain. Waktu yang disediakan dalam bermain *game* adalah 1 menit. Ketika *player* tidak dapat menyelesaikan waktu yang disediakan, maka *player* kalah. Dalam panel yang ditampilkan ketika waktu telah

habis ditampilkan pula score yang didapatkan oleh *player* dan terdapat tombol button untuk *player* dapat bermain lagi dari awal dan button quit untuk kembali menuju halaman awal permainan.

7. Panel Menu dalam *Game*



Gambar 4. 16 Halaman menu *game*

Tampilan *user interface* diatas merupakan panel menu yang disediakan untuk *player* ketika sedang bermain *game*. Terdapat tiga tombol button yang disediakan yaitu tombol button setting yang membawa user menuju halaman setting, button info yang membawa user menuju halaman informasi tentang *game*, dan tombol button quit untuk keluar dari *game*.

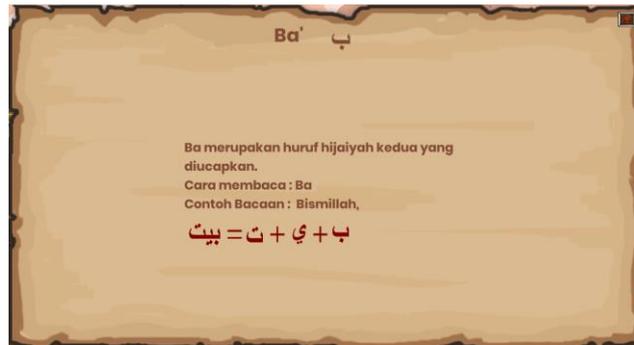
8. Baling-baling penghalang



Gambar 4. 17 Baling-baling penghalang

Gambar diatas merupakan gambar baling-baling yang terdapat dalam *game*. Baling-baling tersebut bergerak naik turun untuk menghalangi pergerakan *player* sehingga *player* harus berhati-hati dalam menggunakan waktu yang dia miliki.

9. Panel pengetahuan tentang huruf hijaiyah



Gambar 4. 18 Halaman panel hijaiyah

Tampilan *user interface* diatas merupakan tampilan panel pengetahuan terkait huruf hijaiyah. Didalam panel, terdapat pengetahuan mengenai huruf hijaiyah, cara membacanya, contoh kata dan cara menggabungkan huruf hijaiyah dalam bahasa arab. Panel tersebut ditampilkan ketika *player* menyentuh huruf hijaiyah. Setelah selesai membaca dan mempelajari huruf hijaiyah yang ditampilkan, *player* dapat menekan button close yang ditandai dengan ikon exit (X) yang terletak dipojok kanan atas panel.

10. Panel resume



Gambar 4. 19 Halaman resume

Tampilan *user interface* diatas merupakan tampilan panel ketika *player* menekan button resume untuk menghentikan permainan sementara. Didalam panel resume terdapat 3 tombol button yang disediakan yaitu tombol button resume untuk kembali melanjutkan permainan, tombol button bantuan yang membawa *player* keluar dari *game* dan menuju halaman bantuan, dan tombol button quit untuk keluar dari permainan dan menuju halaman menu.

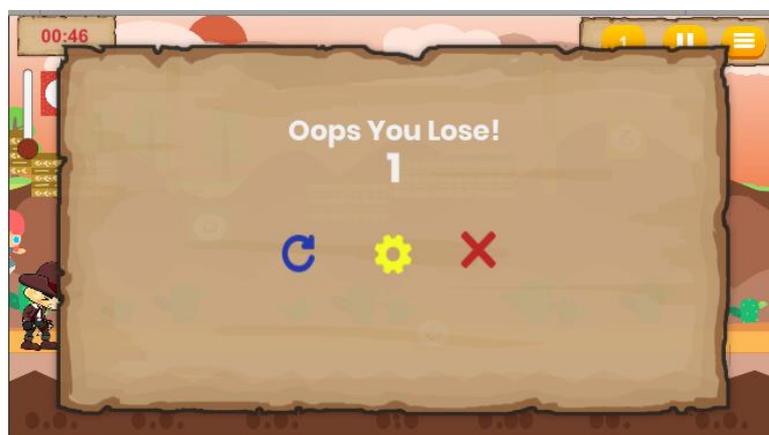
11. Panel menang



Gambar 4. 20 Halaman panel Winner

Tampilan *user interface* diatas merupakan tampilan panel ketika *player* berhasil menyelesaikan *game* dengan menyentuh huruf hijaiyah ya'. Dalam panel menang, *player* dapat mengetahui score yang telah dikumpulkannya. Selain itu, juga terdapat tombol button *play again* untuk bermain *game* lagi dan tombol button menu untuk menuju halaman awal *game*.

12. Panel kalah



Gambar 4. 21 Halaman panel Lose *Game*

Tampilan *user interface* diatas merupakan tampilan panel ketika *player* mengalami kekalahan dalam permainan yang disebabkan karena darah yang dimilikinya sudah habis. Pada panel kalah, *player* mengetahui score yang didapat, terdapat button untuk bermain kembali, menuju halaman setting, atau menuju halaman awal kembali.

4.2 Pengujian Audio *Game*

Pengujian dalam *game* ini dilakukan secara manual dengan memainkan *game* dan memperhatikan variabel yang digunakan dalam *game*. Terdapat 8 pengujian yang dilakukan dalam *game* belajar hijaiyah. Detail penjelasan mengenai pengujian

yang dilakukan dalam menguji musik adaptif yang digunakan dapat dilihat pada table dibawah ini:

Tabel 4. 8 Tabel Data Pengujian sistem

No	Jarak <i>Player-NPC</i>	Vp	Jarak <i>Player-Finish</i>	Musik
1.	16,8	1	105	Light
2.	9,40	2,67	88	Light
3.	6,39	1	72	Middle
4.	10,45	3,7	7,11	High
5.	9	1	52	Light
6.	6	3	30	High
7.	8	1	36	High
8.	6	3,3	92	Middle
9.	7	2	94	High
10.	7	3,3	67	High
11.	4	1	67	High
12.	6	2	60	High
13.	4	2	35	Middle
14.	18	1	30	Middle
15.	17	2	35	Middle
16.	70	3	76	Middle
17.	15	1	105	Light
18.	15	2	94	Light
19.	15	1	46	Light
20.	9	3	19	High
21.	11	1	37	Middle
22.	10	2	35	Middle
23.	12	3,3	100	Middle
24.	10	2	87	Light
25.	10	2	54	Light
26.	11	3,3	67	Middle
27.	11	1	90	Light

Penjelasan pengujian sistem:

Berikut ini tabel yang menunjukkan value yang berada pada setiap atribut:

Tabel 4. 9 Tabel detail value dan nilai atribut

No	Atribut	Value	Nilai
1.	Jarak <i>player-npc</i> (Sa)	Jauh	sa>=14
		Sedang	9<=sa<14

		Dekat	$0 \leq sa < 9$
2.	Kecepatan (v)	Lambat	$v < 2$
		Sedang	$2 \leq v < 3$
		Cepat	$3 < v < 4$
3.	Jarak <i>player-finish</i> (Sb)	Jauh	$Sb \geq 70$
		Sedang	$40 \leq sb < 70$
		Dekat	$0 \leq sb < 40$

Untuk membuktikan kecocokan musik adaptif yang dihasilkan dengan rules dihasilkan dari perhitungan *decision tree*, berikut ini tabel yang menjelaskan kecocokan sistem dengan rules *decision tree*:

Tabel 4. 10 Tabel kecocokan pengujian sistem dengan rules *decision tree*

No	Sistem Game	Rules Decision tree	Kesesuaian
1.	Uji 1	“If sa jauh dan sb jauh maka musik light” (Rule 16)	Sesuai
2.	Uji 2	“If sa sedang, sb jauh dan vp sedang maka musik light” (Rule 18)	Sesuai
3.	Uji 3	“If sa dekat, sb jauh dan vp lambat maka musik middle” (Rule 4)	Sesuai
4.	Uji 4	“If sa sedang, sb dekat dan vp cepat maka musik high” (Rule 10)	Sesuai
5.	Uji 5	“If sa sedang, sb sedang dan vp lambat maka musik light” (Rule 27)	Sesuai
6.	Uji 6	“If sa dekat, sb dekat dan vp cepat maka musik high” (Rule 1)	Sesuai
7.	Uji 7	“If sa dekat, sb dekat dan vp lambat maka musik high” (Rule 2)	Sesuai
8.	Uji 8	“If sa dekat, sb jauh dan vp cepat maka musik middle” (Rule 3)	Sesuai
9.	Uji 9	“If sa dekat, sb jauh dan vp sedang maka musik high” (Rule 5)	Sesuai

10.	Uji 10	“If sa dekat, sb sedang dan vp cepat maka musik high” (Rule 6)	Sesuai
11.	Uji 11	“If sa dekat, sb sedang dan vp lambat maka musik high” (Rule 7)	Sesuai
12.	Uji 12	“If sa dekat, sb sedang dan vp sedang maka musik high” (Rule 8)	Sesuai
13.	Uji 13	“If sa dekat, sb dekat dan vp sedang maka musik middle” (Rule 9)	Sesuai
14.	Uji 14	“If sa jauh, sb dekat dan vp lambat maka musik middle” (Rule 11)	Sesuai
15.	Uji 15	“If sa jauh, sb dekat dan vp sedang maka musik middle” (Rule 12)	Sesuai
16.	Uji 16	“If sa jauh, sb jauh dan vp cepat maka musik middle” (Rule 13)	Sesuai
17.	Uji 17	“If sa jauh, sb jauh dan vp lambat maka musik light” (Rule 15)	Sesuai
18.	Uji 18	“If sa jauh, sb jauh dan vp sedang maka musik light” (Rule 14)	Sesuai
19.	Uji 19	“If sa jauh, sb sedang dan vp lambat maka musik light” (Rule 17)	Sesuai
20.	Uji 20	“If sa sedang, sb dekat dan vp cepat maka musik high” (Rule 19)	Sesuai
21.	Uji 21	“If sa sedang, sb dekat dan vp lambat maka musik middle” (Rule 20)	Sesuai
22.	Uji 22	“If sa sedang, sb dekat dan vp sedang maka musik middle” (Rule 21)	Sesuai
23.	Uji 23	“If sa sedang, sb jauh dan vp cepat maka musik middle” (Rule 22)	Sesuai

24.	Uji 24	“If sa sedang, sb jauh dan vp sedang maka musik light” (Rule 23)	Sesuai
25.	Uji 25	“If sa sedang, sb sedang dan vp sedang maka musik light” (Rule 24)	Sesuai
26.	Uji 26	“If sa sedang, sb sedang dan vp cepat maka musik middle” (Rule 25)	Sesuai
27.	Uji 27	“If sa sedang, sb jauh dan vp lambat maka musik light” (Rule 26)	Sesuai

Dari hasil pengujian sistem dalam *game* seperti yang dijelaskan pada tabel 4.10 diatas, dapat diketahui bahwa dengan pengujian sistem *game* yang dilakukan sebanyak 27 kali sesuai 100% dengan rules-rules yang dihasilkan dari perhitungan *decision tree*. Hal ini membuktikan bahwa rules yang dihasilkan dari perhitungan *decision tree* dapat menentukan adaptif musik dengan efektif dan rules yang dihasilkan dapat digunakan untuk menentukan keputusan dari setiap value atribut yang berbeda.

4.3 Integrasi dalam Islam

Integrasi penelitian ini dalam islam ditulis dalam beberapa sub-bab untuk memetakan integrasi sehingga lebih mudah dipahami. Dalam penelitian ini, integrasi islam didalamnya meliputi dalam hal belajar, berperilaku adaptif, menentukan pilihan, dan mengaplikasikan dalam teknologi *game*.

Integrasi Belajar dalam Islam

Pendekatan metode belajar pada penelitian ini dikhususkan untuk belajar huruf hijaiyah sebagai huruf al-quran. Dengan memahami huruf hijaiyah, anak-anak

mendapat bekal untuk mempelajari al-quran secara lebih mendalam. Perintah belajar dalam islam dapat dilihat dalam surat al-alaq ayat pertama yang artinya “*bacalah*”. Berdasarkan tafsir al-muyassar, ayat ini memerintahkan kepada Nabi Muhammad saw untuk membaca apa yang telah diwahyukan oleh Allah swt kepadanya. Sedangkan menurut tafsir al-mukhtashar ayat ini memerintahkan Nabi Muhammad saw untuk membaca hal yang diwahyukan oleh Allah swt dimulai dengan menyebut nama Allah swt yang menciptakan seluruh makhluk (Tafsir Web, 2023). Dari penjelasan tersebut, dapat diketahui pentingnya belajar dalam islam. Sehingga, jika dikaitkan dengan penelitian yang telah dilakukan, memberikan sarana edukasi belajar hijaiyah dapat menjadi kemudahan anak untuk belajar agama islam.

Integrasi Perilaku Adaptif dalam Islam

Namun, dalam memudahkan pembelajaran, tentunya memerlukan adaptasi nilai-nilai yang disesuaikan dengan usia anak-anak dalam menggunakan sarana belajar. Untuk itu, dalam penelitian ini menampilkan kemampuan beradaptasi sehingga mampu berperilaku adaptif terhadap lingkungannya. Kemampuan adaptasi terhadap lingkungan ini dapat dilihat dengan kondisi seseorang yang harus menyesuaikan dengan lingkungan budaya masyarakat dimana dia tinggal. Dalam al-quran, perilaku adaptif terhadap lingkungan ini telah dicontohkan oleh Rasulullah saw sebagai *Rahmatan Lil Alamin* ketika melakukan hijrah ke Madinah bersama kaum Muhajirin. Berikut ini firman Allah swt dalam Al-Quran:

وَمَنْ يُهَاجِرْ فِي سَبِيلِ اللَّهِ يَجِدْ فِي الْأَرْضِ مُرْعَمًا كَثِيرًا وَسَعَةً جَ وَمَنْ يُخْرِجْ مِنْ بَيْتِهِ < مُهَاجِرًا إِلَى اللَّهِ
وَرَسُولِهِ > ثُمَّ يَدْرِكْهُ الْمَوْتُ فَقَدْ وَقَعَ أَجْرُهُ, عَلَى اللَّهِ قَلْبِي وَكَانَ اللَّهُ غَفُورًا رَحِيمًا

”Barang siapa yang berhijrah di jalan Allah swt, niscaya mereka mendapati dimuka bumi ini tempat hijrah yang luas dan rezki yang banyak. Barang siapa keluar dari rumahnya dengan maksud berhijrah kepada Allah swt dan Rasul-Nya, kemudian kematian menimpanya(sebelum sampai ke tempat yang dituju), maka sungguh telah tetap pahalanya disisi Allah swt. Dan adalah Allah Maha Pengampun lagi Maha Penyayang.”(Q.S An-Nisa: Ayat 100)

Dalam kisah Nabi Muhammad saw yang berhijrah pada ayat diatas dilakukan sebagai langkah strategis bagi kebangkitan islam dan umat islam didunia. Setelah berada diMadinah, salah satu langkah yang dilakukan adalah mempersaudarakan kaum muhajirin dan anshar. Pada saat awal hijrah ke Madinah, kaum muhajirin yang terbiasa dengan keadaan Mekah yang gersang harus menyesuaikan diri di Madinah yang merupakan tempat yang subur. Selain itu, kaum muhajirin harus melakukan adaptasi cara bekerja yang ada diMadinah yaitu bertani, padahal sebelumnya pekerjaan kebanyakan mereka ketika diMekah adalah berdagang. Dengan modal kaum muhajirin yang sedikit dan belum mengetahui cara bekerja disana sehingga mereka merasa kesulitan, akan tetapi, kemudian kaum Anshar sebagai kaum yang sudah lama tinggal memuliakan atau mendahulukan kebutuhan kaum muhajirin diatas kebutuhan mereka sendiri sehingga kaum muhajirin mampu mengatasi permasalahannya. Hal ini terdapat dalam Al-Quran, Allah swt berfirman:

وَالَّذِينَ تَبَوَّءُوا الدَّارَ وَالْآيَمَانَ مِنْ قَبْلِهِمْ يُحِبُّونَ مَنْ هَاجَرَ إِلَيْهِمْ وَلَا يَجِدُونَ فِي صُدُورِهِمْ حَاجَةً مِمَّا أُوتُوا وَيُؤْتُونَ عَلَى أَنْفُسِهِمْ وَلَوْ كَانَ بِهِمْ خَصَاصَةٌ وَمَنْ يُوقِ شُحَّ نَفْسِهِ فَأُولَئِكَ هُمُ الْمُفْلِحُونَ

”Dan orang-orang yang telah menempati kota Madinah dan telah beriman (Anshar) sebelum kedatangan mereka(Muhajirin), mereka mencintai orang yang berhijrah kepada mereka. Dan mereka tiada menaruh keinginan dalam hati mereka terhadap apa-apa yang diberikan kepada mereka(muhajirin), dan mereka mengutamakan (orang-orang muhajirin) atas diri mereka sendiri, sekalipun mereka memerlukan (apa yang mereka berikan itu).” (Q.S. Al-Hasyr: Ayat 9)

Kemampuan beradaptasi yang dicontohkan dalam dua ayat diatas menceritakan bahwa kaum muhajirin harus beradaptasi dengan melihat beberapa kondisi seperti cara bekerja, letak geografisnya, kebiasaan kaum disana, serta lingkungan supaya pendatang mampu beradaptasi dengan baik. Begitu pula dengan menentukan adaptif musik pada penelitian ini, dalam menentukan atau memilih musik yang adaptif harus menyesuaikan dengan beberapa kriteria yang digunakan sebagai parameter yang digunakan sekaligus dengan melihat lingkungan yang memengaruhi permainan didalamnya.

Integrasi Menentukan Pilihan dalam Islam

Untuk menentukan kriteria-kriteria yang tepat dalam menentukan sesuatu, Allah swt berfirman:

وَأْتَلُ عَلَيْهِمْ نَبَأَ ابْنَيْ آدَمَ بِالْحَقِّ إِذْ قَرَّبَا قُرْبَانًا فَتُقْبِلَ مِنْ أَحَدِهِمَا وَمَ يُتَقَبَّلُ مِنَ الْآخَرِ قَالَ لَأَقْتُلَنَّكَ ۗ قَالَ إِنَّمَا يَتَقَبَّلُ اللَّهُ مِنَ الْمُتَّقِينَ

”Ceritakanlah kepada mereka kisah kedua putera Adam (Habil dan Qabil) menurut yang sebenarnya, ketika keduanya mempersembahkan korban, maka diterima dari salah seorang dari mereka berdua (Habil) dan tidak diterima dari yang lain (Qabil). Ia berkata(Qabil):”Aku pasti membunuhmu!”. Berkata Habil:”Sesungguhnya Allah hanya menerima (korban) dari orang-orang yang bertakwa.” (Q.S. Al-Maidah:Ayat 27)

Berdasarkan *tafsir Jalalain*, pada ayat ini diceritakan kisah Habil dan Qabil yang saling mempersembahkan kurban kepada Allah swt. Habil mempersembahkan kurban berupa domba sedangkan Qabil mempersembahkan hasil dari tanaman. Dalam kisah ini, Allah swt menerima kurban Habil sedangkan kurban Qabil ditolak karena Allah swt hanya menerima kurban dari orang-orang yang bertaqwa. Selain berdasarkan *tafsir Jalalain*, menurut Ibnu Abbas dan Ibnu Ummar, ayat ini bercerita

mengenai kisah Qabil yang mempunyai ladang pertanian dan Habil mempunyai peternakan kambing. Pada masa itu, Allah swt memerintahkan Qabil untuk menikah dengan saudara kembarnya Habil. Akan tetapi, Qabil tidak senang dengan perintah tersebut karena beliau merasa saudara kembarnya lebih cantik daripada saudara kembarnya Habil. Kemudian Nabi Adam menyuruh mereka untuk berkorban kepada Allah swt guna mengetahui kurban siapa yang akan diterima. Qabil saat itu berkorban dengan hasil ladang pertaniannya dengan mutu yang rendah, sedangkan Habil berkorban dengan kambing peternakannya dengan kambing pilihan terbaik. Kemudian Allah swt menerima kurban Habil sehingga beliaulah yang berhak menikah dengan saudara kembarnya Qabil (Tafsir Al-Quran, 2020). Seperti kisah tersebut dimana Allah swt memutuskan menerima kurban dari Habil karena beberapa sebab seperti kambing yang diberikan Habil merupakan kambing kualitas terbaik, dan Habil merupakan salah satu orang yang bertakwa karena menuruti perintah Allah swt. Dalam penentuan musik adaptif yang digunakan dalam permainan juga harus memiliki kriteria yang dapat menentukan musik adaptif yang digunakan yaitu dengan melihat kondisi keadaan NPC dan lingkungannya.

Integrasi Teknologi *Game* dalam Islam

Dalam menentukan musik adaptif didalam sarana pembelajaran huruf hijaiyah tentunya harus diadaptasikan dengan perkembangan zaman sekarang. Oleh karena itu, dengan kondisi zaman sekarang dimana teknologi berkembang sangat pesat, maka penerapan pembelajaran bisa menggunakan teknologi yang tersedia, salah satunya adalah *game*. Jika dihubungkan dengan penelitian ini, metode pembelajaran

diwujudkan dalam bentuk *game* 2D supaya mudah dipahami oleh anak-anak usia dini sehingga penyampaian materi lebih mudah dipelajari. Penggunaan teknologi dalam suatu aktivitas ini sudah diberitahukan oleh Allah swt lewat firmanNya:

وَعَلَّمْنَاهُ صَنْعَةَ لَبُوسٍ لَّكُمْ لِنُحِصِنَكُمْ مِّنْ بَأْسِكُمْ ۗ فَهَلْ أَنْتُمْ شَاكِرُونَ
وَلَسَلِّمُنَا رِيحًا غَاصِفَةً تَجْرِي بِأَمْرِنَا إِلَى الْأَرْضِ الَّتِي بَرَكْنَا فِيهَا ۗ وَكُنَّا بِكُلِّ شَيْءٍ عَالِمِينَ

Artinya: " Dan telah kami ajarkan kepada Daud membuat baju besi untuk kamu, guna memelihara kamu dalam peperanganmu, maka hendaklah kamu bersyukur kepada Allah swt. Dan telah kami tundukkan kepada Sulaiman angin yang sangat kencang tiupannya yang berhembus dengan perintahnya ke negeri yang kami telah memberkatinya. Dan adalah kami Maha mengetahui segala sesuatu." (Q.S Al-Anbiya :Ayat 80-81)

Pada ayat tersebut Allah swt memberi pengetahuan kepada Nabi Daud untuk dapat membuat baju besi sehingga beliau ahli dibidang teknologi. Dan Allah swt memberi pengetahuan kepada Nabi Sulaiman cara menggerakkan angin sehingga Nabi Sulaiman menguasai teknologi angin (Harahap, Ginting, & Indriyani, 2023). Begitu pula halnya dengan teknologi yang ada dizaman sekarang seperti *game*, *game* dapat memuat hal-hal adaptif untuk dapat menyesuaikan dengan kondisi karakter dan lingkungan yang ada dalam *game*. Dengan mengimplementasikan teknologi *game* dalam menunjang pembelajaran huruf hijaiyah disertai dengan perilaku adaptif yang ada didalam *game*, maka *game* yang dibuat mampu menerapkan nilai-nilai agama islam untuk berperilaku adaptif terhadap keadaan baik lingkungan ataupun objek lain didalamnya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dengan mengimplementasikan metode *decision tree*, *game* sederhana ini mampu menciptakan *game* edukasi dengan musik adaptif didalamnya. Musik adaptif yang digunakan dalam *game* merupakan lagu instrumen yang adaptif terhadap volume dan pitch lagu. Untuk menentukan musik adaptif ini, perhitungan *decision tree* dilakukan dengan menggunakan bahasa pemrograman python dan software yang digunakan menggunakan visual studio code atau google collab. Dalam penelitian ini, metode *decision tree* diimplementasikan dengan menggunakan algoritma C4.5. Dengan menggunakan algoritma C4.5, kriteria atau atribut yang digunakan terdapat 3 kriteria. Kriteria tersebut diantaranya yaitu jarak antara *player*/pemain dalam *game* dengan karakter NPC musuh, kecepatan *player* dalam berjalan (*Vp*), dan jarak *player* dengan garis finish. Dalam penelitian ini, musik menjadi atribut *class target* dalam perhitungan dan memiliki 3 value yaitu *light*, *middle*, dan *high*.

Setelah melakukan perhitungan *decision tree*, didapatkan 27 *rules* yang diimplementasikan dalam *game* untuk menentukan musik adaptif. Rules berupa pohon keputusan yang dihasilkan dari perhitungan ini memiliki nilai akurasi rata-rata 54% berdasarkan perhitungan cross validation dan nilai akurasi sebesar 70% dari presentase 90% data latih dan 10% data uji. Kemudian, sistem *game* yang telah diimplementasikan rules didalamnya diuji sebanyak 27 kali sesuai dengan total rules yang dihasilkan dari *decision tree*. Dari pengujian sistem yang dilakukan pada

penelitian ini, dapat diketahui bahwa sistem musik adaptif yang ada didalam *game* sudah sesuai dengan semua rules yang ada. Sehingga, hal ini membuktikan bahwa rules *decision tree* yang dihasilkan efektif dan dapat dengan tepat diterapkan dalam *game*. Dengan begitu, musik adaptif didalam *game* belajar hijaiyah ini berhasil mengimplementasikan rules dari perhitungan *decision tree* algoritma C4.5 sehingga *game* mampu digunakan oleh pengguna dengan lebih menarik.

5.2 Saran

Terdapat beberapa hal dalam penentuan musik adaptif ini yang perlu ditingkatkan sebagai acuan untuk memperbaiki beberapa kekurangan yang ada dalam pengimplementasian. Berikut ini beberapa hal yang perlu diperhatikan:

1. Sistem dalam *game* lebih baik apabila terdapat penambahan fitur untuk menyimpan nilai *player* secara bertahap dalam bermain.
2. Untuk dapat menambah ketertarikan terhadap masyarakat terutama anak-anak, ada baiknya apabila sistem memiliki suara karakter yang dapat berinteraksi secara aktif dan dapat dicontoh dan dipraktikkan oleh *player* secara langsung.
3. Untuk dapat menambah kualitas *game*, diperlukan level yang lebih dari satu untuk dapat menambah modul pembelajaran terkait huruf hijaiyah dan huruf al-quran sehingga mampu memberikan motivasi kepada pengguna *game*. Dengan begitu, *player* tidak hanya mengetahui dasar huruf hijaiyah, namun juga mampu memahami secara lebih dalam huruf al-quran.

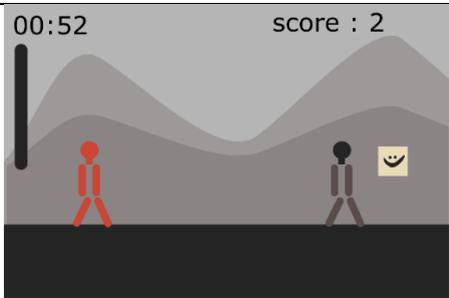
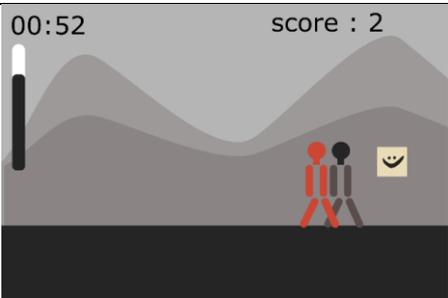
DAFTAR PUSTAKA

- Alex, S., Dhanaraj, K., & Deepthi, P. P. (2022). Private and Energy-Efficient *Decision tree*-Based Disease Detection for Resource-Constrained Medical User in Mobile Healthcare Network. *IEEE*, 17098-17112.
- Collins, K. (2008). *Game Sound :an Introduction to the history, theory, and practices of video game music and sound design*. Cambridge: MIT Press.
- Damanhuri, & Yacub, J. (2022). Mengenalkan Huruf Hijaiyah Pada Anak Usia Dini. *Jurnal Pendidikan Anak Usia Dini Azzahra*, 10(1), 49-59.
- Defiyanti, S., & Pardede, D. C. (2018). Perbandingan Kinerja Algoritma ID3 dan C4.5 Dalam Klasifikasi Spam Mail. *Gunadarma University Repository*, 1-5.
- Harahap, Y. S., Ginting, S. S., & Indriyani, N. K. (2023). Pendidikan Teknologi dalam Al-Quran. *Jurnal Pendidikan Tambusai*, 1898-1906.
- Hutchings, P., & McCormack, J. (2019). *Adaptive Music Composition for Games*.
- Jamil, S. (2022). Musik dalam Pandangan Islam. *Musikolastika* , 26-36.
- Kalsum, U. (2009). *Penggunaan Pohon Keputusan (Decision tree) untuk Pengambilan Keputusan Dalam Penerimaan Pegawai*. Pekanbaru: Informatika UIN Sultan Syarif Kasim Riau.
- Kozak, J. (2019). *Decision tree and Ensemble Learning Based On Ant Colony Optimization*. Switzerland: Springer International Publishing.
- Kuo, P.-H., S.Li, T.-H., Ho, Y.-F., & Lin, C.-J. (2015). Development of an Automatic Emotional Music Accompaniment System by Fuzzy Logic and Adaptive Partition Evolutionary Generic Algorithm. *IEEE*, 3(10), 815-824.
- Latifah, R., Wulandari, E. S., & Kreshna, P. (2019). Model *Decision tree* untuk Prediksi Jadwal Kerja Menggunakan Scikit Learn. *Seminar Nasional Sains dan Teknologi*, (pp. 1-6). Jakarta.
- Mahmud, A. (2017). Musik; Antara Halal dan Haram; kajian Ma'ani al-Hadist. *Jurnal Ilmu Al-Quran dan Tafsir*, 251-284.
- Mardiana, L., Kusnandar, D., & Satyahadewi, N. (2022). Analisis Diskriminan dengan KFold Cross Validation untuk Klasifikasi Kualitas Air diKota Pontianak. *UNTAN*, 97-102.
- Nierhaus, G. (2009). *Algorithmic Composition*. Germany: SpringerWienNewYork.
- Overmars, M. (2012, January 30). A Brief History Of Computer *Games*. pp. 1-35.
- Pranastiti, I. D. (2016). *Penggunaan Decision tree untuk Game Pemilihan Jalur Lintasan Sepeda*. Malang: Informatika UIN Malang.

- Prechtl, A. (2015). *Adaptive Music Generation for Computer Games*. United Kingdom: Open University UK.
- Robinson, C. (2019). *Game Audio With FMOD And Unity*. New York: Routledge Imprint of the Taylor and Francis Group.
- Rodrigues, M. F., & Coutinho, F. R. (2021). A Tool to Implement Adaptive Audio Techniques on *Unity Games*. *Proceedings of SBGames*. Brazil.
- Rokach, L., & Maimon, O. (2014). *Data Mining With Decision trees Theory and Application*. Singapore: World Scientific Publishing.
- Sangyong, L., Chulhee, L., Kwon, G. M., & Dohyun, K. (2022). Decision tree Algorithm Considering Between Classes. *IEEE*, 69750-69756.
- Sarkawi, & Fadli, A. (2021). Memilih Pemimpin dalam Islam. *Jurnal Kajian Manajemen Dakwah*, 198-215.
- Setio, P. B., Saputro, D. R., & Winarno, B. (2020). Klasifikasi dengan Pohon Keputusan Berbasis Algoritme C4.5. *Prisma*, 64-71.
- Smith, J., Musharraf, M., Veitch, B., & Khan, F. (2022). Pilot Study Using *Decision tree* to Diagnose The Efficacy of Virtual Offshore Egress Training. *IEEE*, 812-826.
- Tafsir Al-Quran*. (2020, September 25). Retrieved from tafsiralquran: <https://tafsiralquran.id/tafsir-surat-al-maidah-ayat-27-32/>
- Tafsir Web*. (2023, 05 31). Retrieved from tafsirweb.com: <https://tafsirweb.com/12867-surat-al-alaq-ayat-1.html>
- Thaniket, R., Kusriani, & Luthfi, E. T. (2020). Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Algoritma Support Vector Machine. *Jurnal Fateksa*, 5(2), 20-29.
- Thorn, A. (2013). *Learn Unity for 2D Game Development*. California: Apress.
- Zhang, D., Xu, Y., Peng, Y., Lu, L., & Liu, J. (2023). An Interpretable Station Delay Prediction Model Based on Graph Community Neural Network and Time-Series Fuzzy *Decision tree*. *IEEE*, 421-433.
- Zhang, J., Mechtaev, S., Beschastnikh, I., & Roychoudhury, A. (2022). Fair Decision Making via Automated Repair of *Decision trees*. *FireWare* (pp. 9-16). Pittsburgh, PA, USA: ACM.

LAMPIRAN

1. Story board

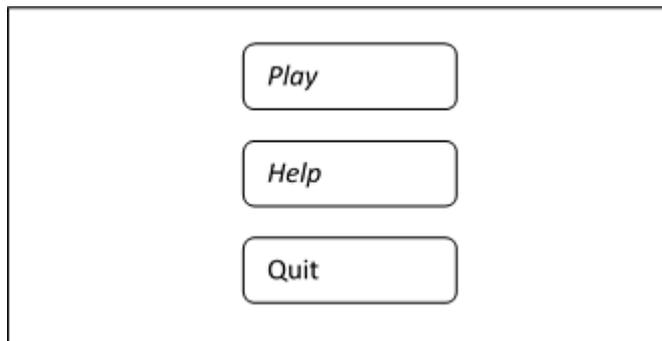
 <p><i>Player</i> dalam <i>game</i> belajar hijaiyah mulai bermain ketika button <i>play</i> diklik. Setelah button <i>play</i> diklik, <i>player</i> diarahkan menuju scene utama <i>game</i>.</p>	 <p><i>Player</i> berada di scene utama untuk memulai memainkan <i>game</i>.</p>
 <p><i>Player</i> mulai berjalan dan mencari huruf-huruf hijaiyah. Pada saat <i>game</i> dimulai, waktu dan score dalam <i>game</i> akan berjalan</p>	 <p>Ketika <i>player</i> berhasil menyentuh huruf hijaiyah, maka akan muncul informasi pengetahuan berkaitan dengan huruf hijaiyah yang disentuh dan score akan bertambah satu setiap <i>player</i> menyentuh hijaiyah.</p>
 <p>Disaat <i>player</i> mulai berjalan, NPC juga mulai berlari secara bertahap dari belakang <i>player</i>.</p>	 <p>Darah <i>player</i> akan berkurang satu setiap kali NPC bersentuhan dengan <i>player</i>.</p>

	
<p>Apabila NPC menyentuh <i>player</i> sudah mencapai 3 kali, maka <i>player</i> akan mati.</p>	<p>Selain itu, ketika waktu yang berjalan didalam <i>game</i> sudah habis, maka permainan akan berakhir dan kalah dalam <i>game</i>.</p>
	
<p>Gambar diatas merupakan panel yang muncul ketika <i>player</i> sudah kalah dalam <i>game</i> baik karena waktu habis atau dia sudah dikalahkan oleh NPC.</p>	<p>Ketika <i>player</i> sudah kalah, dan ingin bermain <i>game</i> kembali, maka <i>player</i> dapat menekan button <i>play again</i> untuk memulai <i>game</i> dari awal kembali.</p>

2. Rancangan Wireframe *User interface*

Dalam *game* sederhana belajar hijaiyah ini, terdapat beberapa tampilan antar muka yang digunakan oleh *player* ketika bermain. Berikut ini rancangan tampilan antarmuka dalam *game* belajar hijaiyah:

1. Halaman menu utama



Gambar 3. 9 Halaman Menu Utama

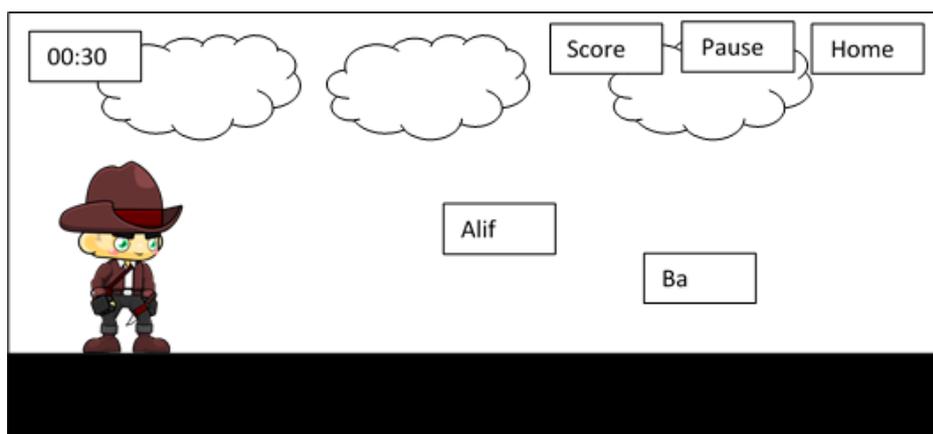
2. Halaman bantuan



Game ini mengharuskan *player* untuk menemukan seluruh huruf hijaiyah untuk mendapatkan score. *Player* menambah kecepatan dengan menekan tombol shift, dan *player* harus menghindari musuh yang mengejar dibelakang.

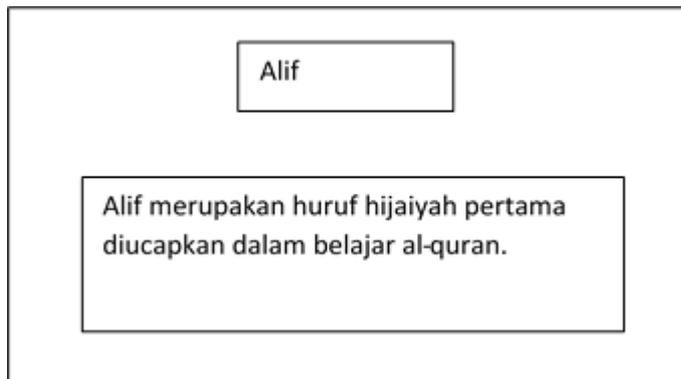
Gambar 3. 10. Halaman Bantuan

3. Scene utama dalam *game*



Gambar 3. 11. Halaman Utama *Game*

4. Halaman informasi



Gambar 3. 12 Halaman Informasi

5. Panel time offer dan *game* lost.

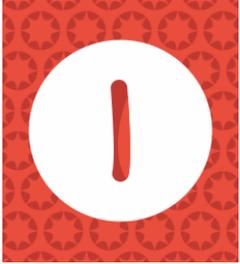


Gambar 3. 13 Halaman time offer

3. Desain Object

Tabel 3. 7 Desain Object dalam *Game*

No	Objek	Keterangan
1.	 <p data-bbox="475 1839 667 1872"><i>Environment 1</i></p>	<p data-bbox="1007 1514 1356 1800"><i>Environment</i> satu merupakan <i>environment environment</i> yang menunjukkan waktu siang hari. Suasana ditunjukkan dengan matahari dan suasana yang cerah seperti siang hari menjelang sore.</p>

2.	 <p>Environment 2</p>	<p><i>Environment</i> kedua merupakan <i>environment</i> yang menunjukkan waktu pada malam hari. Suasana digambarkan dengan awan dan bintang serta suasana yang gelap.</p>
3.	 <p>Karakter</p>	<p>Gambar disamping merupakan <i>player</i> yang bermain didalam <i>game</i>. <i>Player</i> memiliki kemampuan untuk berjalan ke kiri dan ke kanan. Selain itu, <i>player</i> dapat menambah kecepatan berjalan dengan menekan tombol shift.</p>
4.	 <p>Huruf Hijaiyah</p>	<p>Huruf hijaiyah yang digunakan dalam <i>game</i> sederhana ini dimulai dari huruf alif sampai dengan ya. Setiap <i>player</i> menyentuh huruf hijaiyah, <i>player</i> mendapat satu poin sehingga apabila <i>player</i> menyelesaikan semua huruf, <i>player</i> mendapat nilai 29.</p>
5.	 <p>NPC</p>	<p>Karakter NPC merupakan seorang perempuan. Npc ini dapat berlari dan melakukan loncatan berulang untuk dapat mengejar <i>player</i>. Setiap NPC menyentuh <i>player</i>, darah yang dimiliki <i>player</i> berkurang satu.</p>

4. Survey Data Training

Data Responden

Timestamp	Nama	Pekerjaan	Seberapa sering anda bermain game?
5-16-2023 8:04:07	Lidiana Chofifah Rizky	Mahasiswa	Lumayan (5-8 kali)
5-16-2023 8:23:08	Saraswati	Mahasiswa	Lumayan (5-8 kali)
5-16-2023 8:26:13	Pamukti	Mahasiswa	Lumayan (5-8 kali)
5-16-2023 8:40:41	Robiatul	Mahasiswa	Jarang (<3 kali)
5-16-2023 8:58:11	Almaas Rozikin Herawan	Mahasiswa	Jarang (<3 kali)
5-16-2023 9:24:01	livia	mahasiswa	Lumayan (5-8 kali)
5-16-2023 11:17:06	Eka	Mahasiswa	Sering (Lebih dari 10 kali)
5-16-2023 13:43:20	Amalia Dieni Tsabita	Mahasiswa	Sering (Lebih dari 10 kali)
5-17-2023 3:11:47	Miftah	Mahasiswa	Lumayan (5-8 kali)

Data Questioner Musik Adaptif

Keterangan Nilai Numerik:

1 = Light

2 = Middle

3 = High

Responden	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7
1	2	2	2	2	2	2	3
2	1	2	2	1	1	2	2
3	1	1	2	1	2	2	2
4	2	2	2	2	2	2	2
5	1	2	3	2	2	3	1
6	1	1	2	1	1	2	2
7	2	1	2	2	1	2	3
8	1	1	2	1	1	1	2
9	1	1	1	2	2	2	2

Responden	Rule8	Rule9	Rule10	Rule11	Rule12	Rule13	Rule14
1	2	2	2	2	2	2	2
2	2	3	1	1	2	1	1
3	2	3	1	1	2	1	1
4	2	2	2	2	2	2	2
5	2	2	1	1	2	2	2
6	2	3	2	2	2	2	3
7	2	2	2	3	3	3	3
8	2	3	1	1	2	1	1
9	2	2	2	3	3	2	2

Responden	Rule15	Rule16	Rule17	Rule18	Rule19	Rule20	Rule21
1	2	2	2	2	2	2	2
2	2	2	1	3	1	1	2
3	2	2	2	3	1	1	2
4	2	2	2	2	2	2	2
5	2	1	2	2	2	3	3
6	3	3	3	3	2	3	3
7	2	1	2	3	2	3	3
8	2	2	2	3	2	2	3
9	3	2	2	2	3	3	3

Responden	Rule22	Rule23	Rule24	Rule25	Rule26	Rule27
1	2	2	2	2	2	2
2	1	1	2	2	2	3
3	1	1	2	2	2	3
4	2	2	2	3	2	3
5	2	2	2	2	3	3
6	3	3	3	3	3	3
7	3	3	3	3	1	1
8	2	3	3	3	3	3
9	3	3	3	3	3	3

5. Transform Data Training

No	S Player-NPC	Vp	S Player-Finish	Musik
1	1	1	1	1

2	1	1	2	1
3	1	1	0	2
4	1	2	1	1
5	1	2	2	2
6	1	2	0	2
7	1	0	1	2
8	1	0	2	2
9	1	0	0	0
10	2	1	1	1
11	2	1	2	1
12	2	1	0	2
13	2	2	1	1
14	2	2	2	1
15	2	2	0	2
16	2	0	1	2
17	2	0	2	2
18	2	0	0	0
19	0	1	1	1
20	0	1	2	1
21	0	1	0	2
22	0	2	1	1
23	0	2	2	1
24	0	2	0	2
25	0	0	1	2
26	0	0	2	2
27	0	0	0	0
28	1	1	1	1
29	1	1	2	2

30	1	1	0	0
31	1	2	1	2
32	1	2	2	2
33	1	2	0	0
34	1	0	1	1
35	1	0	2	2
36	1	0	0	2
37	2	1	1	1
38	2	1	2	1
39	2	1	0	2
40	2	2	1	2
41	2	2	2	2
42	2	2	0	2
43	2	0	1	1
44	2	0	2	2
45	2	0	0	2
46	0	1	1	2
47	0	1	2	0
48	0	1	0	0
49	0	2	1	2
50	0	2	2	2
51	0	2	0	2
52	0	0	1	2
53	0	0	2	0
54	0	0	0	0
55	1	1	1	1
56	1	1	2	1
57	1	1	0	2

58	1	2	1	1
59	1	2	2	1
60	1	2	0	2
61	1	0	1	2
62	1	0	2	2
63	1	0	0	0
64	2	1	1	2
65	2	1	2	2
66	2	1	0	2
67	2	2	1	2
68	2	2	2	0
69	2	2	0	0
70	2	0	1	0
71	2	0	2	0
72	2	0	0	0
73	0	1	1	2
74	0	1	2	0
75	0	1	0	0
76	0	2	1	0
77	0	2	2	0
78	0	2	0	0
79	0	0	1	0
80	0	0	2	0
81	0	0	0	0
82	1	1	1	1
83	1	1	2	1
84	1	1	0	2
85	1	2	1	1

86	1	2	2	1
87	1	2	0	1
88	1	0	1	2
89	1	0	2	2
90	1	0	0	0
91	2	1	1	1
92	2	1	2	1
93	2	1	0	2
94	2	2	1	1
95	2	2	2	1
96	2	2	0	2
97	2	0	1	2
98	2	0	2	2
99	2	0	0	0
100	0	1	1	2