

**PENERAPAN ALGORITMA *NEURAL NETWORK* DALAM
MENENTUKAN TINGKAT KESULITAN PADA
PERMAINAN EDUKASI**

SKRIPSI

**Oleh:
NAILA NAHDIYAH
NIM. 18650050**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**PENERAPAN ALGORITMA *NEURAL NETWORK* DALAM
MENENTUKAN TINGKAT KESULITAN PADA
PERMAINAN EDUKASI**

SKRIPSI

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

**Oleh:
NAILA NAHDIYAH
NIM. 18650050**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

HALAMAN PERSETUJUAN

**PENERAPAN ALGORITMA *NEURAL NETWORK* DALAM
MENENTUKAN TINGKAT KESULITAN PADA
PERMAINAN EDUKASI**

SKRIPSI

**Oleh:
NAILA NAHDIYAH
NIM. 18650050**

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 18 Juni 2025

Pembimbing I,



Dr. Ir. Fachrul Kurniawan, S.T., M.MT., IPU
NIP. 19771020 200912 1 001

Pembimbing II,



Dr. M. Imamudin Lc, MA
NIP. 19740602 200901 1 010

Mengetahui,

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




Dr. Ir. Fachrul Kurniawan, S.T., M.MT., IPU
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

PENERAPAN ALGORITMA *NEURAL NETWORK* DALAM MENENTUKAN TINGKAT KESULITAN PADA PERMAINAN EDUKASI

SKRIPSI

Oleh:
NAILA NAHDIYAH
NIM. 18650050

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 23 Juni 2025

Susunan Dewan Penguji

Ketua Penguji : Dr. Yunifa Miftachul Arif, M.T
NIP. 19890515 201801 1 001

Anggota Penguji I : Shoffin Nahwa Utama, M.T
NIP. 19860703 202012 1 003

Anggota Penguji II : Dr. Ir. Fachrul Kurniawan, S.T., M.MT., IPU
NIP. 19771020 200912 1 001

Anggota Penguji III : Dr. M. Imamudin Lc, MA
NIP. 19740602 200901 1 010



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



Dr. Ir. Fachrul Kurniawan, S.T., M.MT., IPU
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Naila Nahdiyah

NIM : 18650050

Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Penerapan Algoritma *Neural Network* Dalam Menentukan Tingkat Kesulitan Pada Permainan Edukasi

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, 25 Juni 2025

Yang membuat pernyataan,



Naila Nahdiyah
NIM. 18650050

MOTTO

Never try does not equate to not being able to.

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT

Teruntuk Ayah dan Ibu tercinta, yang dipenuhi dengan kesabaran, cinta dan keikhlasan dalam membimbing, mendoakan, dan mendampingi setiap langkah saya. Terima kasih atas segala pengorbanan yang tak terhitung yang selalu diberikan sejak dalam kandungan hingga mencapai titik ini. Saya sangat bersyukur telah hadir dalam keluarga ini.

Semoga Allah SWT membalas setiap kebaikan dan kasih sayang yang telah Ayah dan Ibu selalu curahkan dengan keberkahan yang tiada putus.

Terima kasih.

KATA PENGANTAR

Puji syukur penulis ucapkan kepada Allah SWT yang telah memberikan rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan studi di Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Sholawat serta salam semoga tetap terlimpahkan kepada Rasulullah Muhammad SAW yang telah memberikan bimbingan dan petunjuk sehingga kita tetap dalam Iman, Islam, dan Ihsan. Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam terselesaikannya Skripsi ini. Oleh karena itu, dengan penuh ketulusan hati penulis mengucapkan terima kasih dikhususkan kepada:

1. Prof. Dr. M. Zainuddin, MA selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan ST., M.MT., IPU selaku Ketua Program Studi Teknik Informatika UIN Maulana Malik Ibrahim Malang dan Pembimbing I penulis yang telah membimbing dan mengarahkan penulis selama menulis dan menyusun skripsi.
4. Dr. M. Imamudin Lc. MA., selaku dosen Pembimbing II yang telah banyak membantu penulis dalam penulisan skripsi.
5. Hani Nurhayati, M.T, selaku dosen wali yang telah dukungan, bimbingan, dan arahan selama masa perkuliahan.

6. Nia Faricha, S. Si, selaku admin Program Studi Teknik Informatika yang selalu membantu penulis dalam proses pengurusan administrasi skripsi.
7. Bapak dan Ibu dosen Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan ilmu kepada penulis.
8. Keluarga penulis, terutama orang tua penulis yang selalu bersabar dan terus memberikan dukungan kepada penulis, baik emosional dan finansial. Kakak dan adik yang selalu memberikan semangat dan dukungan terhadap penulis.
9. Kurniyatul Ainayah, Faradilah Putri Damayanti, dan Chandra Gunawan selaku teman penulis yang telah memberikan dukungan dan bantuan luar yang biasa dalam memulai penulisan skripsi ini.
10. Teman-teman penulis, Seta, Zulda, Cahya, Diyah, Nisa, Ema, Rizki, Adam, Anisa, Lingga, Andy, Inna, Hudan dan teman-teman yang tidak dapat saya sebutkan satu per satu yang sudah membantu penulis selama perkuliahan dan memberi dukungan agar penulis dapat menyelesaikan skripsi.
11. Dan terakhir untuk diri saya sendiri.

Skripsi ini masih jauh dari kata sempurna, maka dari itu penulis sangat menghargai bila terdapat kritik dan saran. Semoga skripsi ini dapat membawa manfaat.

Malang, 23 Juni 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR SIMBOL	xiv
ABSTRAK	xvi
ABSTRACT	xvii
المخلص	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	6
BAB II STUDI PUSTAKA	7
2.1 Penelitian Terkait	7
2.2 Penyeimbang Kesulitan	9
2.3 <i>Game</i>	10
2.4 <i>Game</i> Edukasi	11
2.5 <i>Artificial Intelligence (AI)</i>	12
2.5.1 <i>Neural Network</i>	13
2.6 <i>Usability Testing</i>	17
BAB III DESAIN DAN IMPLEMENTASI	19
3.1 Pengumpulan Data	19
3.2 Perancangan Sistem	19
3.2.1 <i>Finite State Machine (FSM)</i>	21
3.2.2 <i>System Usability Scale (SUS)</i>	22
3.3 Perancangan Skenario	24
3.3.1 Deskripsi <i>Game</i>	24
3.3.2 <i>Storyline</i>	25
3.4 Perancangan Aplikasi	26
3.4.1 Pembuatan Objek 3D	26
3.5 Perancangan <i>User Interface</i>	27
3.5.1 Perancangan Desain UI <i>Title Menu Screen</i>	27
3.5.2 Perancangan Desain UI Kuis	28
3.5.3 Perancangan Desain UI <i>Score Screen</i>	29

3.6	Perancangan Karakter dan Objek	29
3.6.1	Karakter Pemain	30
3.6.2	<i>Items</i>	30
3.7	Perancangan <i>Single-layer Perceptron</i> (SLP)	32
3.7.1	<i>Input</i>	33
3.7.2	Normalisasi (<i>Min-Max Scaling</i>).....	34
3.7.3	Penjumlahan (Kombinasi Linear).....	34
3.7.4	Fungsi Aktivasi (<i>Softmax</i>).....	35
3.7.5	Seleksi Output (<i>Argmax</i>).....	35
3.8	Proses <i>Training Single-layer Perceptron</i>	36
3.8.1	Data Training SLP Level 1 (Per Soal)	36
3.8.2	Data Training SLP Level 2 (Per Kuis).....	46
BAB IV HASIL DAN PEMBAHASAN.....		54
4.1	Hasil Implementasi Tampilan <i>Game</i>	54
4.1.1	UI Menu Utama	55
4.1.2	UI Menu Petunjuk.....	55
4.1.3	UI <i>Gameplay</i>	56
4.1.4	UI Kuis.....	56
4.1.5	UI <i>Game Over</i>	57
4.2	Hasil <i>Data Training SLP</i>	57
4.2.1	Hasil <i>Data Training</i> Model SLP Level 1 (Per Soal).....	58
4.2.2	Hasil <i>Data Training</i> Model SLP Level 2 (Per Kuis)	60
4.3	Hasil <i>Data Testing</i>	62
4.3.1	Evaluasi SLP Level 1	62
4.3.2	Evaluasi SLP Level 2.....	64
4.4	Hasil Evaluasi Model Sederhana	65
4.4.1	Hasil <i>Training</i> dan Evaluasi Model Sederhana.....	66
4.5	Hasil <i>Usability Testing</i>	67
4.6	Integrasi dengan Islam.....	69
4.6.1	<i>Mu'amalah Ma'a Allah</i>	69
4.6.2	<i>Mu'amalah Ma'a An-Nas</i>	70
4.6.3	<i>Mu'amalah Ma'a Alam</i>	71
BAB V KESIMPULAN DAN SARAN		72
5.1	Kesimpulan.....	72
5.2	Saran	73
DAFTAR PUSTAKA		

DAFTAR GAMBAR

Gambar 2.1 Grafik Pengalaman Pemain dalam Penyeimbang Kesulitan	9
Gambar 2.2 Skema Ilustrasi <i>Single-layer Perceptron</i>	14
Gambar 2.3 Ilustrasi <i>Single-layer Perceptron</i>	15
Gambar 3.1 Diagram <i>Finite State Machine</i>	21
Gambar 3.2 Rentan Skor SUS.....	23
Gambar 3.3 Pembuatan Objek 3D	27
Gambar 3.4 Desain UI <i>Title Menu Screen</i>	28
Gambar 3.5 Desain UI Kuis	28
Gambar 3.6 Desain UI <i>Score Screen</i>	29
Gambar 3.7 Desain Karakter Pemain 3D.....	30
Gambar 3.8 Desain Labirin	30
Gambar 3.9 Desain Sampah 3D	31
Gambar 3.10 Desain Pintu 3D	31
Gambar 3.11 <i>Flowchart Single-layer Perceptron</i>	33
Gambar 3.12 <i>Flowchart Data Training Single-layer Perceptron</i>	36
Gambar 3.13 <i>Flowchart Training SLP Level 2</i>	47
Gambar 4.1 <i>User Interface Main Menu</i>	55
Gambar 4.2 <i>User Interface Petunjuk</i>	55
Gambar 4.3 <i>User Interface Gameplay “Save the Energy”</i>	56
Gambar 4.4 <i>User Interface Kuis</i>	56
Gambar 4.5 <i>User Interface Game Over Scene</i>	57
Gambar 4.6 <i>Training Loss</i> pada SLP Level 1	59
Gambar 4.7 Grafik Akurasi <i>Training Data</i> pada SLP Level 1	60
Gambar 4.8 <i>Training Loss</i> SLP Level 2	61
Gambar 4.9 Grafik Akurasi <i>Training</i> SLP Level 2	62
Gambar 4.10 <i>Confusion Matrix</i> SLP dan Distribusi Prediksi Level 1	63
Gambar 4.11 <i>Confusion Matrix</i> SLP Level 2 dan Distribusi Prediksi Level 2	64
Gambar 4.12 <i>Confusion Matrix</i> Model Sederhana	66

DAFTAR TABEL

Tabel 3.1 Parameter <i>Data Training</i>	19
Tabel 3.2 Komposisi Prediksi Tingkat Kesulitan	20
Tabel 3.3 Kuesioner <i>System Usability Score</i>	22
Tabel 3.4 Kategori Penilaian SUS Skor	23
Tabel 3.5 Parameter Input	33
Tabel 3.6 Data Performa Responden	37
Tabel 3.7 Penambahan Parameter Nyawa pada Data Responden	38
Tabel 3.8 Pelabelan Data Responden	39
Tabel 3.9 Penentuan Format <i>One-Hot Encoding</i>	40
Tabel 3.10 Data Responden Hasil Normalisasi	40
Tabel 3.11 Inisialisasi Bobot dan Bias Awal	42
Tabel 3.12 Pembaruan Bobot dan Bias	46
Tabel 3.13 Parameter pada <i>Training</i> SLP Level 2	48
Tabel 3.14 Data Normalisasi Performa pada Kuis 1	49
Tabel 3.15 Inisialisasi Bobot dan Bias pada <i>Training</i> SLP Level 2	50
Tabel 3.16 Perhitungan Delta Bobot dan Delta Bias	52
Tabel 3.17 Hasil Pembaruan Bobot dan Bias (SLP Level 2, Setelah <i>Epoch</i> 1)	53
Tabel 4.1 Distribusi Kelas Sampel Soal	58
Tabel 4.2 <i>Output</i> Model SLP Level 1 (Bobot dan Bias akhir)	58
Tabel 4.3 <i>Epoch Training</i> SLP Level 1	58
Tabel 4.4 <i>Epoch Training</i> Model SLP Level 2	61
Tabel 4.5 <i>Classification Report</i> SLP Level 1	63
Tabel 4.6 <i>Classification Report</i> SLP Level 2	65
Tabel 4.7 <i>Classification Report Training</i> Evaluasi Model Sederhana	67
Tabel 4.8 Hasil <i>Survey</i> Kuesioner <i>Usability Testing</i>	68
Tabel 4.9 Hasil Perhitungan <i>Usability Testing</i>	68
Tabel 5.1 Perbedaan Model SLP Sederhana dan SLP Hierarki	73

DAFTAR SIMBOL

Lambang Romawi

<i>Lambang</i>	<i>Kuantitas</i>	<i>Satuan</i>
x_i	Nilai input ke-i	-
Z	Nilai net input (hasil kombinasi linear sebelum aktivasi)	-
Y_{prediksi}	Nilai output prediksi model (setelah fungsi aktivasi Softmax)	-
W	Matriks bobot (weights)	-
b	Vektor bias (bias)	-
α	Learning rate (tingkat pembelajaran)	-
δ	Delta error (sinyal kesalahan untuk pembaruan bobot)	-
L	Loss (nilai kerugian, dihitung dengan CCE)	-
N_{output}	Jumlah neuron output (kelas)	Jumlah
e	Bilangan Euler (basis logaritma natural, ≈ 2.718)	Konstanta
Σ	Operator penjumlahan	Operator
Δ	Operator perubahan (misalnya, perubahan bobot ΔW)	Operator
N_{sampel}	Jumlah sampel data	Jumlah
x_{kuis}	Vektor fitur input untuk Model SLP Level 2 (berdimensi 6)	-
$P_{\text{avg_Mudah}}$	Rata-rata probabilitas 'Mudah' dari SLP Level 1 untuk satu kuis	-
$P_{\text{avg_Sedang}}$	Rata-rata probabilitas 'Sedang' dari SLP Level 1 untuk satu kuis	-
$P_{\text{avg_Sulit}}$	Rata-rata probabilitas 'Sulit' dari SLP Level 1 untuk satu kuis	-
\bar{A}_{kuis}	Rata-rata nilai NormAkurasi dari 4 soal dalam satu kuis	-
\bar{T}_{kuis}	Rata-rata nilai NormWaktu dari 4 soal dalam satu kuis	-

\bar{N}_{kuis}	Rata-rata nilai NormNyawa dari 4 soal dalam satu kuis	-
J_{benar}	Jumlah jawaban benar pemain (digunakan dalam pelabelan GroundTruth)	Soal
$T_{\text{pengerjaan}}$	Total waktu pengerjaan kuis (digunakan dalam pelabelan GroundTruth)	detik (s)
N_{sisa}	Sisa nyawa pemain (digunakan dalam pelabelan GroundTruth)	Nyawa
N_{max}	Jumlah nyawa maksimum di awal permainan	Nyawa
NormAkurasi	Nilai Akurasi performa pemain yang telah dinormalisasi	-
NormWaktu	Nilai Waktu pengerjaan soal yang telah dinormalisasi	-
NormNyawa	Nilai Sisa Nyawa yang telah dinormalisasi	-
GroundTruth	Label target sebenarnya dari data (Mudah, Sedang, Sulit)	Kategorikal (String)

Singkatan

SLP	<i>Single-Layer Perceptron</i>
UI	<i>User Interface</i>
FSM	<i>Finite State Machine</i>
CCE	<i>Categorical Cross-Entropy</i>

ABSTRAK

Nahdiyah, Naila. 2025. *Penerapan Algoritma Neural Network dalam Menentukan Tingkat Kesulitan pada Permainan Edukasi*. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Ir. Fachrul Kurniawan ST., M.MT., IPU (II) Dr. M. Imamudin. Lc., MA.

Kata Kunci: *Single Layer Perceptron, Game Edukasi, Neural Network.*

Penggunaan energi secara berlebihan sedang menjadi masalah pada akhir-akhir ini, padahal energi adalah kebutuhan esensial dalam keberlangsungan hidup manusia. sejalan dengan ajaran Islam yang menganjurkan penjagaan alam, edukasi tentang berhemat energi menjadi krusial. edukasi melalui media digital dinilai lebih efektif dan menarik, maka dari itu dirancanglah sebuah *game* edukasi berbentuk kuis bertemakan hemat energi. Untuk membuat *game* menjadi lebih menarik, perlu menambahkan penyesuaian tingkat kesulitan yang dinamis. Penyesuaian tersebut menggunakan *Single-layer Perceptron* (SLP) yang dinilai dinamis dan efektif. Parameter *input* yang digunakan meliputi akurasi, waktu pengerjaan, dan sisa nyawa pemain. Hasil evaluasi sistem dilakukan pada dua level SLP: Level 1 (per soal) dan Level 2 (per kuis), menggunakan aktivasi *Softmax* dan *loss Categorical Cross-Entropy* (CCE). Hasil pelatihan menunjukkan akurasi SLP 1 dan 2 sebesar 80%. Pada data uji, akurasi SLP Level 1 sebanyak 80% dan pada Level 2 hanya 60%, menunjukkan perlunya peningkatan dalam generalisasi model. *Usability testing* menampilkan presentase rata-rata 56.17% dengan kriteria "Cukup", hal tersebut dapat diartikan bahwa permainan cukup mudah digunakan. Penerapan SLP dalam permainan edukasi diharapkan dapat menjadi sarana edukasi dan dapat meningkatkan kesadaran hemat energi.

ABSTRACT

Nahdiyah, Naila. 2025. **Implementation of Neural Network Algorithm in Determining Difficulty Levels in an Educational Game.** Undergraduate Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Dr. Ir. Fachrul Kurniawan S.T., M.MT., IPU (II) Dr. M. Imamudin. Lc., MA.

Keywords: Single-Layer Perceptron, Educational Game, Neural Network.

The excessive use of energy has recently become a growing issue, despite energy being an essential need for human survival. In line with Islamic teachings that promote environmental stewardship, education on energy conservation is becoming increasingly crucial. Digital media is considered a more effective and engaging means of delivering such education; therefore, an educational quiz-based game themed on energy-saving was developed. To make the game more appealing, dynamic difficulty adjustment is necessary. This adjustment uses a Single-layer Perceptron (SLP), which is considered both dynamic and effective. The input parameters include accuracy, completion time, and the player's remaining lives. System evaluation was conducted at two levels of SLP: Level 1 (per question) and Level 2 (per quiz), using Softmax activation and Categorical Cross-Entropy (CCE) loss function. Training results showed that SLP Levels 1 and 2 achieved 80% accuracy. However, on test data, the accuracy is 80% for Level 1 and only 60% for Level 2, indicating the need for improved model generalization. Usability testing yielded an average score of 57.16%, categorized as "Ok," suggesting that the game is relatively easy to use. The implementation of SLP in the educational game is expected to serve as a learning tool and raise awareness about energy conservation.

المخلص

،مهدية، نيل. 2025. تطبيق خوارزمية الشبكة العصبية في تحديد مستوى الصعوبة في لعبة تعليمية. البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (١) الدكتور فخرول كورنيوان الماجستير(٢) الدكتور محمد إمام الدين.

الكلمات الأساسية: بيرسيبترون أحادي الطبقة ، لعبة تعليمية ، شبكة عصبية.

أصبح الاستخدام المفرط للطاقة مشكلة متزايدة في الآونة الأخيرة، على الرغم من أن الطاقة تُعدّ حاجة أساسية لاستمرار حياة الإنسان. وتماشياً مع تعاليم الإسلام التي تدعو إلى الحفاظ على البيئة، أصبحت التوعية بأهمية ترشيد استهلاك الطاقة أمراً بالغ الأهمية. وتُعتبر الوسائط الرقمية وسيلة أكثر فاعلية وجاذبية لنشر هذه التوعية؛ لذا تم تصميم لعبة تعليمية على شكل مسابقة تحمل موضوع "توفير الطاقة". ومن أجل جعل اللعبة أكثر تشويقاً، تم إدخال خاصية تعديل مستوى الصعوبة بشكل: الذي يُعدّ ديناميكياً وفعالاً. تشمل معلمات الإدخال "SLP) ديناميكي. يعتمد هذا التعديل على "بيرسيبترون أحادي الطبقة المستوى الأول (لكل سؤال): SLP دقة الإجابة، زمن الإجابة، وعدد الأرواح المتبقية للاعب. تم تقييم النظام على مستويين من Categorical Cross-Entropy (CCE) وخسارة Softmax والمستوى الثاني (لكل مسابقة)، باستخدام دالة التفعيل أظهرت نتائج التدريب دقة بلغت 80% في كل من المستوى الأول والثاني. ولكن على بيانات الاختبار، انخفضت الدقة إلى 80% في المستوى الأول و60% فقط في المستوى الثاني، مما يدل على الحاجة إلى تحسين قدرة النموذج على التعميم. أما اختبار قابلية الاستخدام، فقد أظهر نسبة متوسطها 57.16% ضمن تصنيف "متوسط"، مما يشير إلى أن اللعبة سهلة الاستخدام نسبياً. ومن المأمول أن يُسهّم استخدام SLP في اللعبة التعليمية في تقديم وسيلة فعالة للتعليم وزيادة الوعي بأهمية ترشيد استهلاك الطاقة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Energi merupakan salah satu hal yang paling dibutuhkan dalam keberlangsungan hidup manusia. Dengan meningkatnya perkembangan teknologi dan pertumbuhan penduduk, penggunaan energi terutama energi listrik dan air menjadi kewajiban yang harus dilakukan masyarakat untuk mengurangi energi terbuang menjadi terbuang dengan sia-sia. Energi listrik berasal dari batu bara 50%, gas bumi 29%, BBM 7% dan energi terbarukan 14% (Kementerian ESDM, 2019). Kebutuhan energi listrik di Indonesia pada tahun 2018 mengalami kenaikan sebanyak 3% dari tahun sebelumnya dengan jumlah populasi di Indonesia pada saat itu mencapai 264,2 juta penduduk menurut Badan Pusat Statistik (BPS). Kemudian pada tahun 2020, produksi air bersih yang disebarkan mencapai 5.262,1 juta m³ dengan jumlah penduduk 270,2 juta (BPS, 2020). Kuantitas permintaan energi yang digunakan di tiap tahunnya selalu meningkat sesuai dengan meningkat pesatnya jumlah penduduk dikarenakan seberapa pentingnya energi dalam keberlangsungan hidup manusia seperti penggunaan pada pabrik industri, komunikasi, transportasi hingga pekerjaan dalam rumah tangga.

Penggunaan energi yang berlebihan dapat menyebabkan pembengkakan biaya pengeluaran, kerusakan bumi seperti pemanasan global, tidak sesuainya energi yang disebarkan pada masyarakat hingga mengakibatkan menipisnya persediaan energi lebih cepat untuk kedepannya (Fitriani, et al. 2018). Maka dari itu

masyarakat perlu menghemat penggunaan energi supaya tidak melakukan pemborosan, hal ini sejalan dengan bagaimana Islam sangat memperhatikan tentang pemborosan dan menjaga lingkungan alam, sebagaimana tertuang dalam surat *Al-Qhasas* (28) : 77.

وَأَبْتَعْ فِيمَا آتَاكَ اللَّهُ الدَّارَ الْآخِرَةَ ۖ وَلَا تَنْسَ نَصِيبَكَ مِنَ الدُّنْيَا ۗ وَأَحْسِنَ كَمَا أَحْسَنَ اللَّهُ إِلَيْكَ ۖ
وَلَا تَبْغِ الْفُسَادَ فِي الْأَرْضِ ۖ إِنَّ اللَّهَ لَا يُحِبُّ الْمُفْسِدِينَ ۗ ٧٧

“Dan carilah pada apa yang telah dianugerahkan Allah kepadamu (kebahagiaan) negeri akhirat, dan janganlah kamu melupakan kebahagiaanmu dari (kenikmatan) duniawi dan berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik, kepadamu, dan janganlah kamu berbuat kerusakan di (muka) bumi. Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan.” (Q.S. Al-Qhasas: 28)

Dalam Tafsir Ibnu Katsir, *Darud Thayyibah Linnasyari wat Tauzi*, dijelaskan bahwa tidak bolehnya mempunyai hasrat merusak bumi yang padahal kamu hidup didalamnya dan tidak boleh berbuat buruk terhadap sesuatu yang Allah ciptakan yang sudah jelas mempunyai kemanfaatan kepada makhluknya dan Allah membenci kerusakan di bumi. Secara inheren, *game* edukasi dengan tema menghemat energi dapat membuat pemain lebih memahami konsep dan menerapkan prinsip hemat energi agar bumi tidak mengalami kerusakan. Maka dari itu manusia memiliki tanggung jawab tersendiri dalam menjaga keseimbangan dan kelestarian lingkungan hidup untuk tidak terjadi pengeksploitasian alam yang menyebabkan kerugian dan dampak buruk bagi kehidupan (Muhammad, 2022).

Penanganan penggunaan energi berlebih dapat dilakukan dengan melakukan edukasi kepada masyarakat dengan sosialisasi langsung. Namun dengan

berkembangnya teknologi, mengedukasi masyarakat dapat dilakukan dengan pendekatan melalui media digital. Pendekatan melalui teknologi atau media digital akan berdampak lebih baik dan lebih efektif daripada menggunakan pendekatan lain (Reeves, 1998). Penyampaian edukasi yang interaktif dan efektif dapat meningkatkan ketertarikan terhadap pentingnya melakukan pengelolaan penghematan energi. Salah satu solusi alternatif untuk mengedukasi masyarakat dengan mudah adalah melalui sebuah permainan edukasi atau *educational game*.

Educational game merupakan permainan berfokus pada konten pembelajaran dan peningkatan pemahaman pemain dalam bentuk permainan yang menghibur (Muhajarah & Rachmawati, 2019). Penyampaian materi dalam penggunaan *educational game* tidak terlepas dari metode penyampaian teks ataupun mekanisme permainan (Molnar & Kotskova, 2018). Untuk mendapatkan keseruan dan pemahaman saat memainkan permainan, penambahan tingkat kesulitan tentu sangat diperlukan untuk memberikan pengalaman yang berbeda dan manfaat dari pembelajaran kepada pemain (Schreiber & Romero, 2022).

Untuk menambahkan tingkat keseruan dan pemahaman dalam *game*, maka perlu menambahkan fitur yang membuat *game* menjadi lebih menarik dan dinamis. Maka dari itu perlu menambahkan kecerdasan buatan untuk menentukan materi yang sesuai dengan kemampuan pemainnya dengan memanfaatkan perilaku pemain sebagai input dan variabel hasil yang diperoleh saat memainkan *game* sebagai parameter kemampuan pemain, kemudian parameter tersebut digunakan untuk menentukan tingkat kesulitan permainan agar alur permainan dapat sesuai dengan tingkat pemahaman pemain. Hal tersebut akan memberikan pengalaman menarik

pada pemain dalam memahami topik yang akan dijelaskan. Dan diharapkan pemain mendapatkan pemahaman dan kemudahan pemahaman materi yang disampaikan.

Berdasarkan uraian diatas, penggunaan algoritma *neural network* dapat menjadi alternatif kecerdasan buatan untuk menjadi alat pengatur fitur tingkat kesulitan. Terdapat dua macam bentuk *neural network*, yaitu *single-layer neural network* dan *multilayer neural network*. Pada penelitian ini, *single-layer neural network* akan digunakan, dan metode yang dipakai adalah *single-layer artificial neural network* dengan model *perceptron*. Metode tersebut dipilih karena berdasarkan penelitian sebelumnya penerapan *single-layer perceptron* mampu mempelajari dan beradaptasi dengan sistem yang sederhana dan menghasilkan prediksi dalam waktu yang singkat (Karimboyevich & Nematullayevich, 2022). *Perceptron* adalah metode yang sangat efektif untuk menyelesaikan masalah linier karena strukturnya yang sederhana seperti mengambil keputusan dalam mengatur tingkat kesulitan permainan berdasarkan kondisi tertentu. Maka dari itu, logika *perceptron* dapat digunakan sebagai pendekatan yang efektif untuk menjadikan sistem permainan menjadi dinamis dan lebih relevan.

Permainan yang akan dikembangkan bernama *Save the Energy* yaitu sebuah permainan kuis bertemakan hemat energi yang berbentuk labirin 3 dimensi. Tujuan dari permainan ini adalah mengumpulkan item berupa kunci yang tersebar acak untuk membuka pintu labirin dengan cara menjawab semua kuis yang akan muncul ketika mendapatkan item.

Dengan adanya penerapan metode *perceptron* yang diterapkan pada permainan 3 dimensi *Save the Energy*, diharapkan dapat menjadi salah satu alat edukasi masyarakat dari mulai anak-anak hingga remaja, sehingga dapat menciptakan metode sosialisasi hemat energi yang menyenangkan dan mudah. Dan diharapkan untuk menambah wawasan dan tetap awas terhadap pentingnya menjaga pemakaian energi dalam kelanjutan hidup manusia.

1.2 Rumusan Masalah

Bagaimana menentukan tingkat kesulitan permainan edukasi berdasarkan kemampuan kognitif pemain?

1.3 Batasan Masalah

Berdasarkan identifikasi diatas, penelitian ini dibatasi oleh beberapa hal berikut:

1. Permainan yang dikembangkan berbentuk labirin 3 dimensi dan dimainkan oleh satu pemain.
2. Permainan berbasis desktop.
3. Parameter yang digunakan berasal dari performa pemain, yaitu jawaban benar, waktu pengerjaan dan sisa nyawa.
4. Pokok pembahasan meliputi konsep menjaga sumber energi terbarukan dan non-terbarukan tingkat SD sampai SMP.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membangun permainan edukasi yang dapat menentukan tingkat kesulitan berdasarkan kemampuan kognitif pemain.

1.5 Manfaat Penelitian

Penelitian ini bermanfaat sebagai sarana alternatif dalam memberikan edukasi kepada masyarakat dengan menggunakan *game* sebagai media edukasi yang menyenangkan dan dapat dimainkan oleh semua kalangan terutama anak-anak dan remaja. Sehingga diharapkan mampu mengatasi rasa jenuh saat mengedukasi pentingnya energi dan meningkatkan rasa sayang terhadap lingkungan sekitar. Selain itu, permainan ini dapat digunakan oleh pendidik untuk menyampaikan edukasi kepada siswa dan menilai pemahaman *moral science* para siswanya.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Terdapat penelitian tentang permodelan *neural network* untuk merancang kecerdasan buatan pada musuh dalam permainan yang berjenis *turn-based battle* yang diteliti oleh (Tsalitsy et al., 2013). Permodelan *neural network* ini menghasilkan nilai error sebesar 10%-20% yang berasal dari *data training*. Dapat diartikan bahwa algoritma ini cukup akurat dalam menyesuaikan tingkat kesulitan dengan kemampuan pemain.

Pada penelitian yang dilakukan oleh Saputra (2015), peneliti menambahkan algoritma *neural network* pada permainan *First Person Shooter (FPS)*. Peneliti menambahkan algoritma tersebut pada bot musuh, bot tersebut digunakan untuk menyesuaikan Tingkat kesulitan permainan sesuai data dari hasil *training neural network*. Tujuan penelitian ini adalah penambahan *neural network* untuk memberi pengalaman bermain yang lebih realistis dan menarik. Peneliti berhasil menunjukkan keberhasilan penggunaan *neural network* dalam menyeimbangkan Tingkat kesulitan sesuai dengan kemampuan pemain dan memberikan alur permainan yang lebih dinamis dan menantang.

Penelitian yang dilakukan oleh (Romero-Mendes et al.,2023) untuk menyesuaikan tingkat kesulitan permainan, peneliti menggunakan pendekatan deep learning. Penelitian ini mempunyai tujuan untuk mencegah ketidakpuasan pemain yang bisa saja terjadi saat tingkat kesulitan yang diperoleh pemain tidak sesuai

dengan keterampilan yang dipunya. Hasil dari penelitian ini menunjukkan penerapan algoritma deep learning berhasil meningkatkan keterlibatan yang tinggi, tingkat kesenangan dan kepuasan yang diperoleh pemain sebanyak 90%.

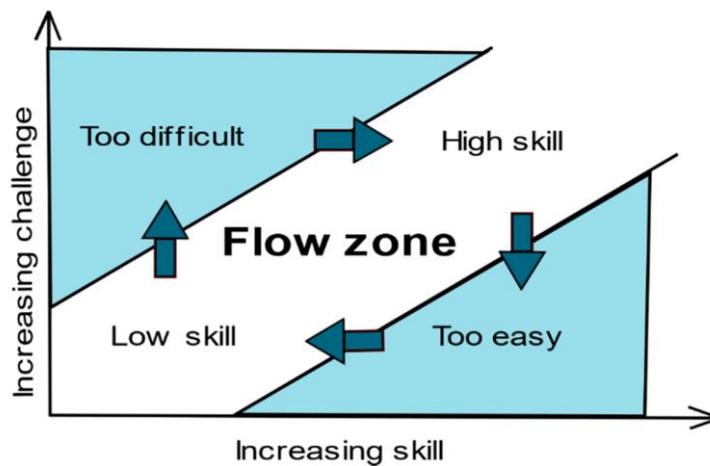
Dalam penelitian Robertson (2017) dalam pengambilan keputusan yang dilakukan oleh *neural network*, Robertson memilih *Multi-layer Perceptron* untuk mengontrol AI dalam permainan *racing 2D*. *Perceptron* digunakan untuk memprediksi bagaimana mobil harus berbelok atau tetap lurus berdasarkan *input* tertentu. Hasil dari penelitian ini menunjukkan bahwa penerapan algoritma tersebut menunjukkan akurasi yang relevan namun dengan waktu pelatihan yang cukup lama.

Dengan beberapa penelitian diatas, penelitian ini akan menggunakan algoritma *Single-layer Perceptron* dalam membuat permainan edukasi “*Save the Energy*” untuk membuat kesulitan permainan menjadi adaptif sesuai dengan kemampuan dan pemahaman pemain. *Single-layer perceptron* memiliki kemampuan yang efektif dalam memproses dan menganalisis data *input* yang sederhana. Sistem permainan akan menyesuaikan tingkat kesulitan pada kuis yang diberikan kepada pemain dengan waktu yang singkat, sehingga pengalaman bermain yang dirasakan akan lebih optimal dan seimbang. *Single-layer perceptron* memiliki beberapa *input* diantaranya jawaban pemain, waktu pengerjaan dan nyawa. Sedangkan *output* dari metode ini akan diolah menggunakan fungsi aktivasi *Softmax* untuk menghasilkan pertanyaan-pertanyaan yang sesuai dengan tingkat pengetahuan pemain dan skor yang didapat.

2.2 Penyeimbang Kesulitan

Menurut Adams (2014) dalam buku mereka yang berjudul “Fundamentals of Game Design”, salah satu elemen penting dalam pembuatan *game* adalah adanya penyeimbang kesulitan untuk memastikan bahwa permainan tetap menantang dan juga adil. Pemberian penyeimbang kesulitan juga dapat meningkatkan pengalaman pemain dan kontribusi dalam keseluruhan keberhasilan pada *game*.

Menurut Pamungkas (2022), ketika kuis terlalu mudah, pemain akan cepat merasa bosan. Apabila kuis dianggap terlalu sulit, pemain akan merasa frustrasi. Maka dari itu dalam permainan terdapat penjelasan materi diakhir permainan. penggunaan penyeimbang kesulitan dapat digambarkan dalam bentuk grafik seperti pada Gambar 2.1.



Gambar 2.1 Grafik Pengalaman Pemain dalam Penyeimbang Kesulitan

Dalam beberapa kasus, implementasi adaptivitas kesulitan pada permainan bisa didapatkan dalam berbagai cara, seperti melalui *non-player character* yang pernah dibuat oleh Peirce et. al (2008), dan menyesuaikan variabel dalam permainan itu sendiri seperti kesulitan pada tiap level atau varian *enemy* yang akan

ditemui (Hall-Tipping. 1991). Pada penelitian ini, variabel yang akan digunakan untuk adaptivitas kesulitan pada permainan kuis antara lain:

1. Jawaban benar per soal.
2. Waktu pengerjaan. Variabel ini berisi seberapa lama pemain mengerjakan satu soal tiap kuis.
3. Sisa nyawa.

Algoritma yang digunakan dalam penelitian ini untuk menjadikan *game* adaptif sesuai kemampuan pemain adalah *Single-layer Perceptron*. *Perceptron* akan memroses data *input* yang diantaranya adalah jawaban, nyawa dan waktu dalam tiap satu soal untuk diubah menjadi data yang bisa diolah kembali untuk menyeimbangkan kesulitan. Algoritma tersebut akan menghasilkan *output* berupa pertanyaan yang muncul pada kuis secara acak menyesuaikan kemampuan pemain dan mengubah variabel lainnya seperti, pengurangan nyawa, penambahan skor, bobot dan bias.

2.3 *Game*

Jika diartikan dalam bahasa Indonesia, *game* memiliki arti permainan. Dalam KBBI, permainan */per-ma-in-an/* merupakan sesuatu atau barang yang digunakan untuk bermain. Pada buku yang ditulis oleh Hubbell dan Kapell (2020) yang berjudul “*What Is a Game*”, pengertian “*game*” sendiri masih menjadi perhatian tersendiri dikarenakan luasnya makna dan makna dapat berubah tergantung orang yang menganalisis *game*. Salen dan Zimmerman (2004) dalam bukunya yang berjudul “*Rules of Play: Game Design Fundamentals*” juga

mengatakan bahwa terdapat keambiguan arti dari kata *play* dan *game*. Namun kesimpulan yang bisa didapat dari beberapa definisi yang dikutip dari berbagai subjek yang ditanya, *game* merupakan suatu sistem dimana pemain terlibat dalam konflik buatan yang ditentukan oleh aturan, dan menghasilkan sesuatu yang dapat diukur. Aturan yang dibuat dalam *game* bertujuan untuk membatasi perilaku pemain dan menentukan arah permainan.

Dalam buku “*Rules of Play: Game Design Fundamentals*” juga menjelaskan tentang *game digital* yang memiliki definisi sifat dasar dari *game* dan tantangan inti dalam desain *game* yang berlaku tanpa memandang media mana *game* tersebut dibuat. *Game digital* memiliki empat ciri khusus:

1. Interaksi secara langsung antara pemain dengan *gamenya*, namun dalam lingkup yang terbatas.
2. Melibatkan manipulasi dan pengolahan data yang kompleks.
3. Menggunakan sistem yang kompleks juga dalam menjalankan mekanika permainan.
4. Memanfaatkan jaringan untuk mengelola data dan memungkinkan interaksi antar pemain dari berbagai tempat.

2.4 Game Edukasi

Game edukasi merupakan permainan yang dirancang dengan tujuan membantu pemain mempelajari dan memahami konsep materi, mengembangkan keterampilan pemecahan masalah melalui media interaksi yang menyenangkan dan menarik, salah satunya dalam bentuk kuis (Ge & Ifenthaler, 2017). Terdapat

elemen-elemen desain penting dalam *game* edukatif (Marklund & Taylor, 2016), di antara lain:

- a. *Game mechanics*.
- b. Estetika visual yang berfungsi untuk meningkatkan daya tarik dalam *game*.
- c. Narasi.
- d. Insentif. Sesuatu yang akan didapatkan pemain ketika mencapai level tertentu untuk memotivasi pemain agar terus bermain.
- e. Musik dan efek suara.
- f. *Knowledge/skills*. Dalam bentuk materi pembelajaran yang ingin disampaikan melalui permainan.

2.5 *Artificial Intelligence (AI)*

Artificial intelligence dapat didefinisikan sebagai suatu sistem yang berperilaku cerdas dengan menganalisis lingkungan sekitarnya dan mengambil tindakan sesuai dengan keinginan sistem tertentu untuk mencapai tujuan yang sudah ditentukan (Sheikh et al., 2023).

Penggunaan AI pada *game* sudah mulai giat dilakukan pada beberapa dekade terakhir yang dibuktikan dengan banyaknya hasil penelitian yang dibagikan oleh para peneliti salah satunya yang dilakukan Lucas et al (2008). Lucas melakukan penelitian tentang penggunaan AI dalam pengembangan *game* (*Game AI*) dan AI yang mengembangkan *game* itu sendiri (*AI in Games*). Terdapat beberapa teknik utama yang biasa digunakan dalam pengembangan *game AI*, beberapa diantaranya adalah *pathfinding* yang digunakan untuk menemukan jalur

paling efisien dan teknik *decision making* yang digunakan untuk membuat keputusan NPC berdasar kondisi dalam *game* supaya menghasilkan perilaku yang realistis dan responsif bergantung pada tindakan pemain. Lucas juga mengidentifikasi beberapa hal yang menjadi kunci dalam penerapan *AI in Games*, seperti:

- a. *Adaptive AI*. AI yang dapat menyesuaikan strategi dan perilakunya berdasar perilaku pemain.
- b. *Opponent Modelling*. Teknik AI dalam memahami dan memprediksi tindakan lawan supaya permainan menjadi lebih menantang dan pemain merasa seimbang.

Tantangan-tantangan yang dihadapi dalam mengembangkan AI dalam *game* ada beberapa macam, seperti kompleksitas desain untuk membuat AI menjadi realistis dan responsif, adanya keterbatasan sumberdaya komputasi karena membutuhkan daya yang besar, dan juga keseimbangan antara sesuainya *game* dengan realita dan rasa seru dalam bermain.

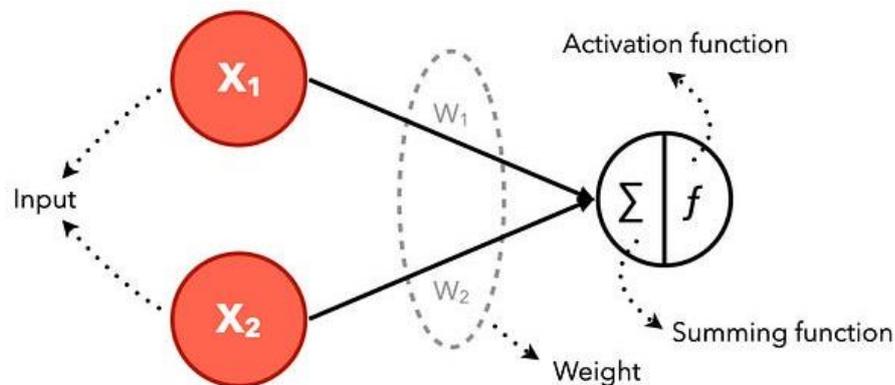
2.5.1 *Neural Network*

Neural network adalah algoritma terinspirasi dari gaya kerja pada otak manusia yang biasa disebut dengan neuron yang saling terhubung untuk memproses informasi (Roberts. 2021). Neuron menerima sinyal dari neuron-neuron lain yang berada pada tiap bagian tubuh kemudian informasi tersebut diproses dan akan menghasilkan *output* yang akan dikirimkan kembali ke neuron lain dalam jaringan. Algoritma *neural network* merupakan salah satu bagian dari *deep learning* yang

merupakan sub bidang dari *artificial intelligence* yang memungkinkan sistem komputer mempelajari data tanpa diprogram dengan eksplisit untuk tiap tugas. Pada buku “*Artificial Intelligence in Games*” yang ditulis oleh Paul Roberts, dijelaskan bahwa satu neuron dapat terhubung dengan ribuan neuron-neuron lain, dan jaringan-jaringan yang terhubung di tiap neuron disebut dengan *neural network*, dan dalam cakupan *artificial intelligence* disebut dengan “*artificial neural network*”. Menurut Loy (2019), model *artificial neural network* yang paling sederhana merupakan *single-layer neural network* atau *perceptron*.

2.5.1.1 *Single-layer Perceptron (SLP)*

Single-layer perceptron merupakan model paling sederhana dalam *artificial neural network* yang terkadang disebut juga dengan *single-layer neural network*. Dalam bagian tubuh manusia, *perceptron* digambarkan sebagai jaringan neuron yang terdapat pada manusia (Karimboyevich & Nematullayevich. 2022). *Single-layer* berarti tidak terdapat *hidden layer* dalam lapisan neuronnya. Berikut skema ilustrasi SLP.



Gambar 2.2 Skema Ilustrasi *Single-layer Perceptron*

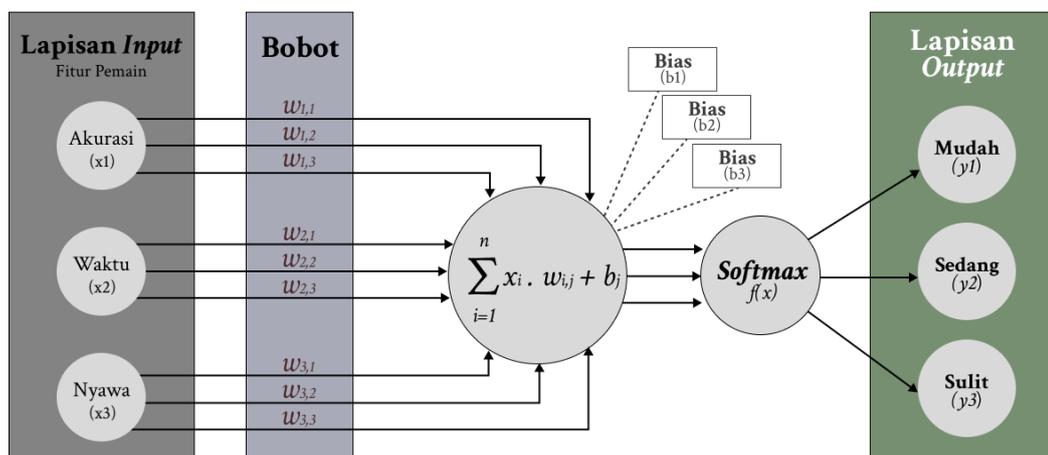
Pada Gambar 2.2 skema *single-layer perceptron* mempunyai dua lapisan yang terdiri dari: satu lapis *input* dan satu lapisan *output*.

1. Lapisan *input*. Lapisan yang bertugas menerima data dari luar sistem.
2. Lapisan *output*. Lapisan hasil dari proses pada lapisan *input* yang berisi prediksi ataupun klasifikasi.

Penggunaan *single-layer perceptron* pada penelitian ini difungsikan untuk memecahkan masalah klasifikasi linier, menggunakan satu lapisan neuron yang menerima data *input*, kemudian diolah dan menghasilkan *output* berdasar pada fungsi aktivasi tertentu. Perhitungan *single-layer perceptron* dapat dicari menggunakan formula berikut.

$$\sum_{i=1}^n x_i \cdot w_i \quad (2.1)$$

Penelitian ini menjadikan *single-layer perceptron* sebagai algoritma yang digunakan untuk menyeimbangkan tingkat kesulitan dalam permainan. Ilustrasi *single-layer perceptron* yang digunakan dalam penelitian ini adalah sebagai berikut.



Gambar 2.3 Ilustrasi *Single-layer Perceptron*

Fungsi aktivasi dalam *single-layer perceptron* terdapat tiga macam, yaitu *step function*, *sigmoid*, dan *softmax*.

1. *Step Function*

Fungsi aktivasi ini adalah fungsi paling sederhana yang mengubah hasil kombinasi linear menjadi *output* diskret (0 atau 1) tergantung pada nilai *threshold* (nilai ambang) dan *step function* tidak memiliki turunan, maka penggunaan fungsi *step* hanya untuk klasifikasi biner yang sederhana. Formulasi yang digunakan *step function* sebagai berikut.

$$f(x) = \begin{cases} 1. & \text{jika } z \geq \text{threshold} \\ 0. & \text{jika } z < \text{threshold} \end{cases} \quad (2.2)$$

2. *Sigmoid*

Fungsi aktivasi *sigmoid* adalah fungsi aktivasi kontinu yang menghasilkan *output* rentan dari 0 sampai 1 yang berguna untuk menangani probabilitas seperti peluang, maka dari itu digunakan pada model yang lebih data yang lebih kompleks. Berikut merupakan formulasi fungsi *sigmoid*.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

Sigmoid memiliki turunan (gradien). Berikut merupakan formula turunannya.

$$f'(x) = f(x)(1 - f(x)) \quad (2.4)$$

3. *Softmax*

Fungsi aktivasi Softmax adalah sebuah fungsi kontinu yang mentransformasi vektor nilai-nilai real arbitrer menjadi distribusi

probabilitas. Fungsi ini mengambil nilai dari output mentah (sering disebut *logits* atau *net input*) dari setiap neuron dan mengubahnya menjadi probabilitas yang berkisar antara 0 hingga 1 untuk setiap kelas, dengan total probabilitas dari semua kelas selalu berjumlah 1. Oleh karena sifatnya ini, fungsi Softmax sangat cocok digunakan untuk masalah klasifikasi multi-kelas, di mana suatu sampel hanya dapat termasuk ke dalam satu kategori pada satu waktu.

$$y_{prediksi,j} = \frac{e^{Z_j}}{\sum_{k=1}^{N_{output}} e^{Z_k}} \quad (2.5)$$

Keterangan:

$y_{prediksi,j}$: Probabilitas prediksi.

e : Bilangan eksponensial (≈ 2.718).

Z_j : Net *input* neuron *output* ke- j .

$\sum_{k=1}^{N_{output}} e^{Z_k}$: Penjumlahan eksponensial dari seluruh net *input* neuron *output* pada lapisan tersebut.

N_{output} : Jumlah total neuron *output*.

2.6 Usability Testing

Usability testing adalah evaluasi kualitatif dari bagaimana pemain menggunakan aplikasi dengan mudah. Menurut Rizky & Pudrianisa (2019), ketika fungsi aplikasi dapat berjalan dengan baik, efektif dan memuaskan untuk pemain, maka aplikasi dapat dianggap bisa digunakan. Riyadi & James (2021) mengungkap pendapat Jacob Nielsen tentang penggunaan 5 aspek dalam *usability testing*, yaitu:

1. *Learnability* merupakan evaluasi pemain tentang seberapa cepat pemain memahami sistem dan dengan mudah menjalankan fungsi-fungsi dalam sistem permainan.

2. *Efficiency* merupakan pengukur kemampuan menggunakan akurasi dan integritas dalam sejauh mana pemain mencapai tujuan tertentu.
3. *Memorability* adalah mengukur kemampuan pemain dalam mengingat dan mengimplementasikan sistem dalam jangka waktu tertentu.
4. *Errors* adalah kesalahan yang dilakukan pemain yang mempengaruhi representasi sistem yang sebenarnya.
5. *Satisfaction* merupakan penilaian subjektif dari pemain terhadap tingkat kepuasan dalam menggunakan sistem termasuk pandangan positif dan negatif yang dirasakan oleh pemain.

BAB III

DESAIN DAN IMPLEMENTASI

Penelitian ini bertujuan untuk membangun aplikasi permainan edukatif tentang menghemat energi terbarukan dan non-terbarukan berbasis desktop menggunakan *godot engine* sebagai kode editor, dan bahasa pemrograman *gdsript*. Pada bab ini akan dibahas mengenai desain dan implementasi metode dilakukan, mulai dari pengumpulan data, perancangan sistem, perancangan aplikasi yang didalamnya terdiri pembuatan skenario *game*, pembuatan *wireframe* aplikasi, proses pembuatan karakter dan objek 3D, perancangan metode dan proses *data training*.

3.1 Pengumpulan Data

Penelitian ini mengambil data hasil pengerjaan kuis dari responden sebanyak 44 siswa, mulai dari kelas 5 dan 6 SD dan seluruh jenjang SMP. Masing-masing responden mengerjakan 20 soal. Data tersebut digunakan sebagai sumber *data training* dan *data testing* untuk model klasifikasi tingkat kesulitan soal. Dari data tersebut, parameter utama yang digunakan adalah sebagai berikut.

Tabel 3.1 Parameter *Data Training*

Parameter	Deskripsi
Jawaban	1 jika benar, 0 jika salah
Waktu pengerjaan	$T/seconds$
Nyawa	Sisa nyawa pemain

3.2 Perancangan Sistem

Perancangan sistem pada penelitian ini bertujuan untuk membuat acuan model sistem yang akan digunakan sebelum diimplementasikan ke dalam aplikasi.

Sistem dirancang sebagai kuis edukatif yang memanfaatkan *Single-Layer Perceptron* (SLP) untuk mengklasifikasikan tingkat kesulitan kuis berdasarkan performa pemain. Setelah pemain menyelesaikan 1 kuis yang terdiri dari 4 soal, sistem akan mengevaluasi masing-masing soal berdasarkan performa pemain menggunakan model SLP, kemudian mengagregasi hasil prediksi tingkat kesulitan dari ke 4 soal tersebut untuk menentukan tingkat kesulitan kuis selanjutnya.

Terdapat 5 kuis yang berisi 4 pertanyaan dalam *game*, kuis pertama berfungsi sebagai acuan awal permainan. Kemudian berdasarkan prediksi performa pemain tersebut, sistem akan menampilkan soal dengan komposisi tingkat kesulitan seperti pada Tabel 3.3

Tabel 3.2 Komposisi Prediksi Tingkat Kesulitan

Prediksi Tingkat Kesulitan Kuis	Komposisi Soal
Mudah	4 soal mudah
Sedang	2 soal mudah + 2 soal sedang
Sulit	2 soal sedang + 2 soal sulit

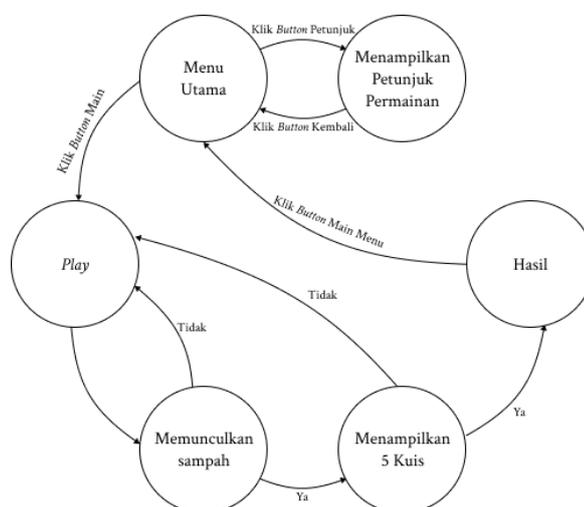
Jika pada arsip soal untuk tingkat kesulitan tertentu sudah habis, sistem akan menggunakan soal yang telah dikerjakan sebelumnya dari *cache* sesi permainan. *Cache* tersebut berisi kumpulan soal yang label kesulitannya diperbarui secara sementara berdasarkan performa pemain pada sesi itu. Proses ini memastikan keberlanjutan permainan tanpa mengubah data asli pada arsip soal.

Dalam perancangan sistem, terdapat dua macam desain sistem yang digunakan pada penelitian ini, yaitu *Finite State Machine* (FSM) yang digunakan menggambarkan logika sistem, dan *System Usability Scale* (SUS) yang digunakan

untuk mengevaluasi aplikasi secara keseluruhan berdasarkan pengalaman pemain memainkan *game* ini.

3.2.1 *Finite State Machine (FSM)*

FSM digunakan untuk memberikan gambaran sistem yang lebih terstruktur untuk sistem yang memiliki berbagai kondisi. FSM pada penelitian ini menggambarkan logika adaptasi tingkat kesulitan dalam *game*. Diagram FSM dijelaskan pada Gambar 3.2



Gambar 3.1 Diagram *Finite State Machine*

Lingkaran pada diagram diatas merupakan proses dari sistem. Ketika *player* menekan tombol Petunjuk yang berada didalam menu utama, tombol petunjuk akan aktif dan menampilkan kepada pemain *screen* petunjuk permainan, kemudian sistem akan kembali menampilkan menu utama dengan memilih tombol Kembali. Tombol Main pada menu utama akan membawa pemain menuju *scene* permainan. Saat sistem menampilkan *scene game*, sistem secara bersamaan akan menampilkan sampah secara acak didalam labirin. Ketika pemain menemukan sampah, sistem

akan menampilkan kuis. Jika kuis yang tampil sudah sebanyak 5 kali, maka sistem akan menampilkan hasil dari performa pemain tersebut. Dan sistem akan menampilkan kembali *main menu* Ketika pemain menekan tombol *Main Menu*.

3.2.2 *System Usability Scale (SUS)*

Tahap ini dilakukan untuk mengukur tingkat *usability* permainan menurut perspektif pemain. Pengujian dilakukan dengan memberikan pernyataan berupa kuesioner penilaian kepada beberapa pemain. Penilaian *usability* permainan *save the energy* menggunakan metode *system usability scale*.

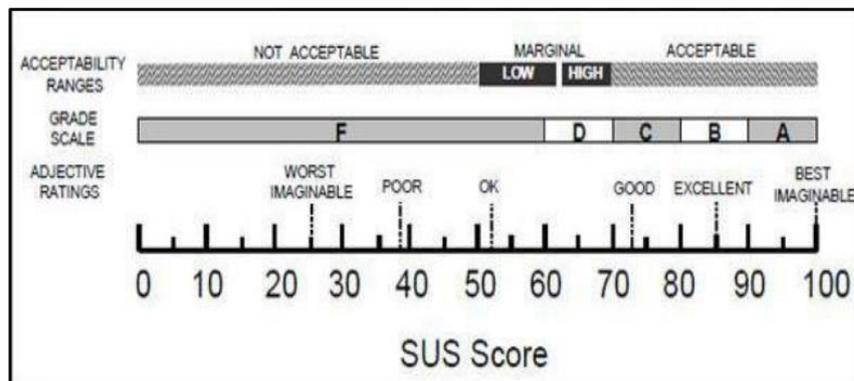
Metode ini menggunakan 5 poin skala Likert (sangat tidak setuju, tidak setuju, netral, setuju, sangat setuju) dari 10 item pernyataan sesuai dengan penilaian subjektif yang mencakup penilaian aspek positif dan negatif dari permainan. Dalam kuesioner ini terdapat struktur pernyataan positif dan negatif. Pernyataan positif berada pada nomor 1, 3, 5, 7, dan 9. Pernyataan negatif berada pada nomor 2, 4, 6, 8, dan 10. Berikut merupakan kuesioner yang digunakan untuk *usability testing*.

Tabel 3.3 Kuesioner *System Usability Score*

No.	Pernyataan	Usability
1	<i>Game</i> ini mudah dipahami sejak pertama kali bermain	<i>Learnability</i>
2	Terdapat elemen yang membingungkan dalam <i>game</i> ini	
3	Saya dapat menyelesaikan teka-teki dengan cepat	<i>Efficiency</i>
4	Saya mengalami kesulitan dalam menavigasi labirin	
5	Jika saya bermain kembali, saya masih ingat cara bermain <i>game</i> ini	<i>Memorability</i>
6	Saya perlu mempelajari ulang cara bermain <i>game</i> ini jika sudah lama tidak memainkannya	
7	Saya jarang melakukan kesalahan saat menjawab kuis dan menyelesaikan labirin	<i>Errors</i>
8	Pertanyaan dalam kuis sulit dipahami	
9	Permainan ini menyenangkan	<i>Satisfaction</i>
10	<i>Game</i> ini membuat saya frustrasi	

Tiap pernyataan memiliki skor kontribusi rentan 0 sampai 4 yang dihitung berdasarkan skala yang didapat dari pemain. Untuk pernyataan positif skor kontribusi hasil dari skala dikurangi 1, pernyataan negatif skor kontribusi hasil dari 5 dikurangi skala. Skor perhitungan SUS berkisar dari 0 sampai 100, berikut rumus perhitungan skor:

$$SUS\ Score = \left(\sum_{i=1}^{10} \text{skor kontribusi} \right) \times 2,5 \quad (3.1)$$



Gambar 3.2 Rentan Skor SUS

Berdasarkan pada Gambar 3.3 interpretasi skor SUS dapat dikategorikan seperti pada Tabel 3.4

Tabel 3.4 Kategori Penilaian SUS Skor

Grade	SUS	Range	Adjective	Acceptability
A	78.9-100	85-100	<i>Best Imaginable</i>	<i>Acceptable</i>
B	72.6-78.8	65-84	<i>Excellent</i>	<i>Acceptable</i>
C	62.7-72.5	35-64	<i>Good</i>	<i>Marginal</i>
D	51.7-62.6	15-34	<i>OK</i>	<i>Marginal</i>
F	25.1-51.6	2-14	<i>Poor</i>	<i>Non Acceptable</i>
F	0-25	0-1.9	<i>Worst Imaginable</i>	<i>Non Acceptable</i>

Kategori penilaian SUS sebagai berikut:

1. *Grade A* (78.9-100): Menunjukkan bahwa pemain puas dengan sistem dan *game* yang telah dibuat.
2. *Grade B* (72.6-78.8): Permainan mudah dimainkan tetapi terdapat area yang memerlukan sedikit perbaikan.
3. *Grade C* (62.7-72.5): Permainan memiliki *usability* yang cukup baik, tetapi terdapat beberapa masalah yang membuat pemain kurang puas.
4. *Grade D* (51.7-62.6): Permainan memiliki *usability* dibawah standar. Pemain mungkin mengalami kesulitan menyelesaikan permainan dengan efisien.
5. *Grade F* (25.1-51.6): Permainan mempunyai *usability* yang buruk. Pemain cenderung frustrasi dan mungkin tidak ingin memainkan *game* ini kembali.
6. *Grade F*(0-25): Permainan memiliki *usability* yang sangat buruk, hampir tidak ada aspek *usability* yang memadai

3.3 Perancangan Skenario

Tahap perancangan skenario meliputi deskripsi permainan, mendesain alur cerita, dan mekanisme permainan.

3.3.1 Deskripsi *Game*

Pada penelitian ini, penulis akan mengembangkan sebuah *game* edukatif *Save the Energy* yang bervisualisasi 3 dimensi. *Save the Energy* adalah permainan interaktif yang memadukan antara tantangan kuis dan menjelajahi labirin yang rumit. Pemain akan disuguhkan dengan labirin yang penuh teka-teki di mana

pemain harus menemukan dan mengumpulkan sampah untuk dapat membuka pintu ujung labirin dan bebas. Syarat untuk dapat mengumpulkan sampah adalah pemain harus menjawab semua kuis tentang hemat energi. Permainan ini bertujuan untuk melihat kemampuan dan meningkatkan pemahaman dan kesadaran pemain dalam hemat berenergi. Dengan penggunaan metode *single-layer perceptron*, permainan *Save the Energy* akan menjadi lebih menyenangkan karena menyesuaikan dengan pemahaman pemain.

3.3.2 *Storyline*

Save the Energy menceritakan tentang seorang warga yang tidak pernah membuang sampah pada tempatnya. Warga tersebut diculik dan dikirim ke sebuah labirin karena sikapnya. Namun, untuk keluar dari labirin tersebut tidak lah mudah. Terdapat banyak tantangan yang harus dilewati. Setiap persimpangan terdapat sampah yang harus dikumpulkan, teka-teki pertanyaan dan rumitnya jalan untuk keluar. Warga harus melakukan tantangan tersebut dengan bijak dan mengumpulkan poin sebanyak mungkin agar dapat membuka pintu diujung labirin dan berhasil keluar.

Plot utama dalam permainan *Save the Energy*:

1. Awal Perjalanan

Warga tersadar dari tidurnya, dan sudah berada didalam labirin. Kemudian warga menemukan surat yang berisi cara untuk keluar dari labirin.

2. Tantangan

Warga diwajibkan untuk mengumpulkan sampah, dan sampah dapat diangkat apabila warga menjawab semua pertanyaan dari kuis yang muncul terkait hemat energi. Jika salah menjawab, maka poin yang didapat akan berkurang.

3. Puncak Cerita

Semakin banyak poin yang didapat, semakin sulit pertanyaan yang muncul saat mengumpulkan sampah. Jika warga berhasil mengumpulkan semua sampah, maka pintu keluar berhasil terbuka.

4. Akhir Cerita

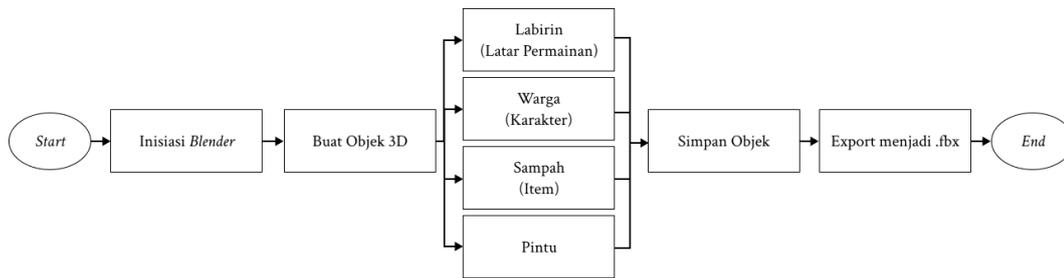
Ketika warga berhasil membuka pintu keluar dan keluar dari labirin, warga akan dapat kembali pulang ke rumahnya dan dia tidak akan lagi membuang sampah sembarangan untuk menjaga bumi dan tidak diculik kembali.

3.4 Perancangan Aplikasi

Perancangan aplikasi permainan *Save the Energy*, terdapat langkah-langkah pembuatan permainan mulai dari pembuatan objek 3D menggunakan *Blender* dan pembuatan aplikasi melalui *Godot Engine*.

3.4.1 Pembuatan Objek 3D

Terdapat beberapa langkah dalam pembuatan objek 3D yang digunakan dalam game ini, antara lain:



Gambar 3.3 Pembuatan Objek 3D

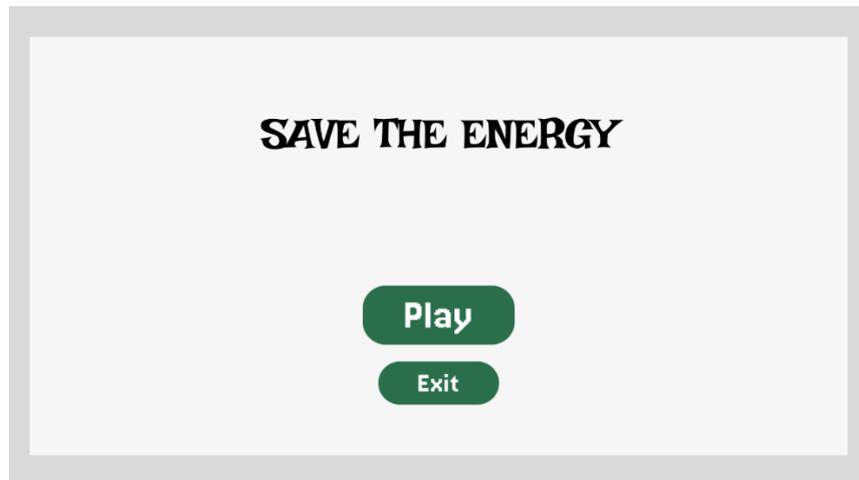
1. Pembuatan objek 3D melalui perangkat lunak *Blender*.
2. Objek 3D yang dibuat mencakup:
 - a. Labirin, sebagai latar permainan.
 - b. Karakter, sebagai warga.
 - c. Sampah, sebagai item yang dikumpulkan.
 - d. Pintu, sebagai garis *finish*.
3. Objek 3D yang dibuat di *export* dalam format *.fbx*

3.5 Perancangan *User Interface*

Tahap pembuatan *interface* bertujuan untuk membuat tampilan yang disajikan kepada pemain dan penghubung antara pemain dengan sistem dalam aplikasi. Pada game *Save the Energy* terdapat tiga bagian *interface* utama, yaitu *title menu screen*, *kuis screen*, *score screen*.

3.5.1 Perancangan Desain UI *Title Menu Screen*

Tampilan awal *main menu* terdapat pada Gambar 3.5.

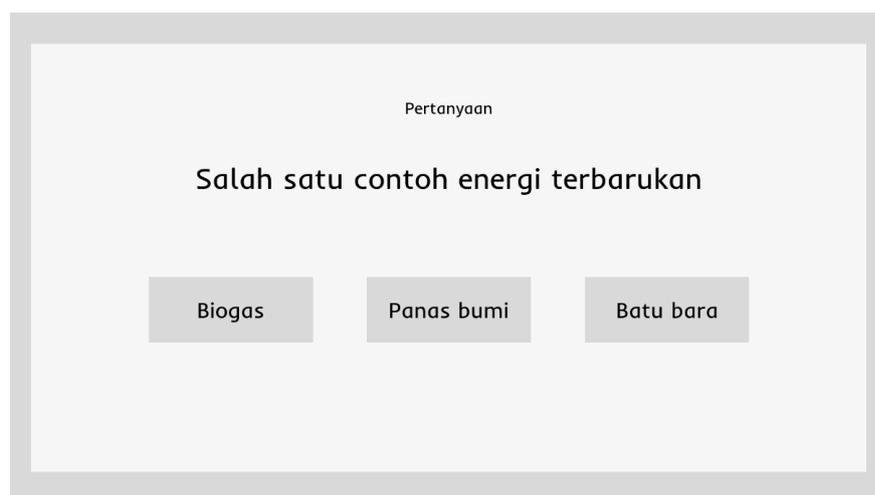


Gambar 3.4 Desain UI *Title Menu Screen*

Gambar 3.5 menunjukkan rancangan tampilan awal dari *main menu* permainan. Terdapat dua tombol aktif yaitu *Play* untuk memulai permainan, dan *Exit* untuk keluar dari aplikasi.

3.5.2 Perancangan Desain UI Kuis

Tampilan awal kuis terdapat pada Gambar 3.6.

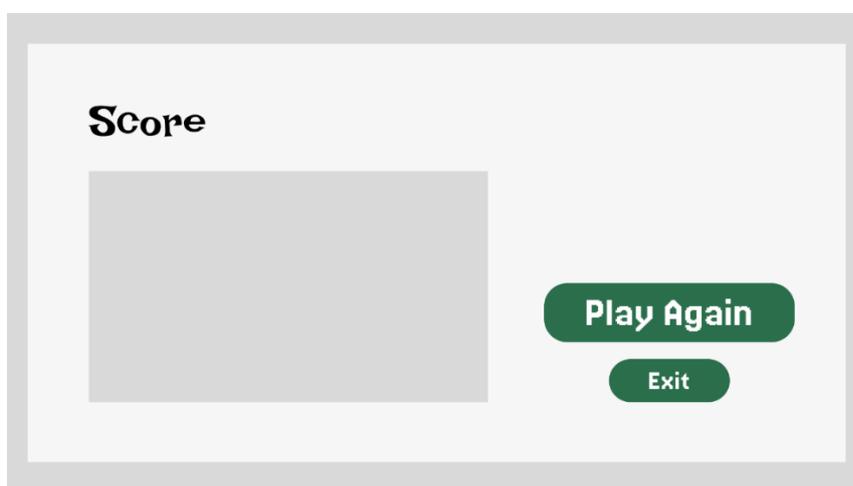


Gambar 3.5 Desain UI Kuis

Gambar 3.6 menunjukkan *user interface* pengerjaan kuis. Pada *scene* tersebut terdapat pertanyaan yang akan muncul secara acak, dan 3 tombol untuk jawaban dari pilihan ganda.

3.5.3 Perancangan Desain UI *Score Screen*

Tampilan layar untuk *score screen* terdapat pada Gambar 3.7.



Gambar 3.6 Desain UI *Score Screen*

Gambar 3.7 merupakan layar *score* yang didalamnya akan berisi hasil skor pemain, sisa nyawa, dan waktu bermain. Terdapat 2 tombol aktif, yaitu *Play Again* yang akan memulai kembali permainan ke *state Play*, dan tombol *Exit* yang berfungsi untuk menutup aplikasi.

3.6 Perancangan Karakter dan Objek

Dalam tahap perancangan karakter dan objek, penelitian ini menjelaskan tentang bagaimana konsep pembuatan karakter pemain dan objek lainnya, seperti sampah, pintu dan labirin sebagai *setting* dalam permainan.

3.6.1 Karakter Pemain

Karakter pemain dalam *game* ini adalah sebuah warga, seperti yang digambarkan pada Gambar 3.8



Gambar 3.7 Desain Karakter Pemain 3D

Gambar 3.8 merupakan desain 3D warga yang akan menjadi karakter pemain. Karakter berbentuk seperti *chibi* yang mempunyai postur tubuh kecil.

3.6.2 Items

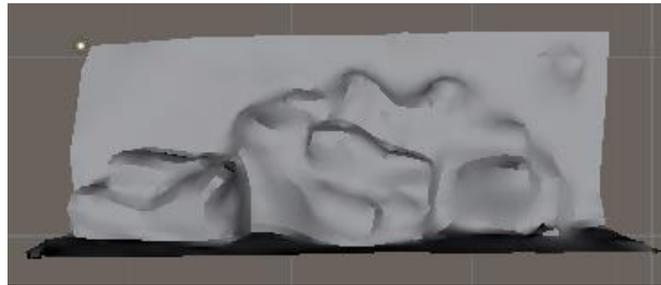
1. Labirin



Gambar 3.8 Desain Labirin

Gambar 3.9 merupakan *setting* permainan tempat warga terjebak dan harus mengumpulkan *item*. Dalam labirin hanya terdapat 1 pintu keluar dan tidak ada pintu masuk.

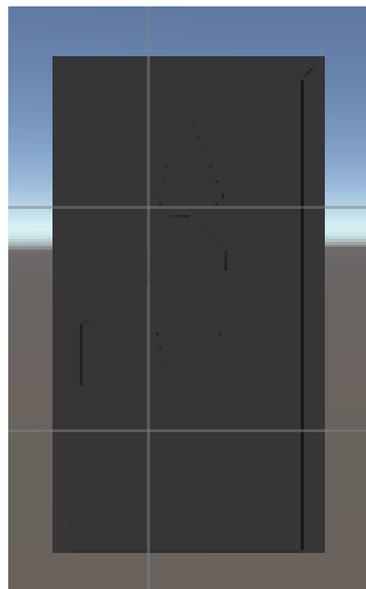
2. Sampah



Gambar 3.9 Desain Sampah 3D

Gambar 3.10 merupakan item sampah yang harus dikumpulkan oleh pemain melalui menjawab kuis yang diberikan. Posisi sampah berada tersebar didalam labirin

3. Pintu

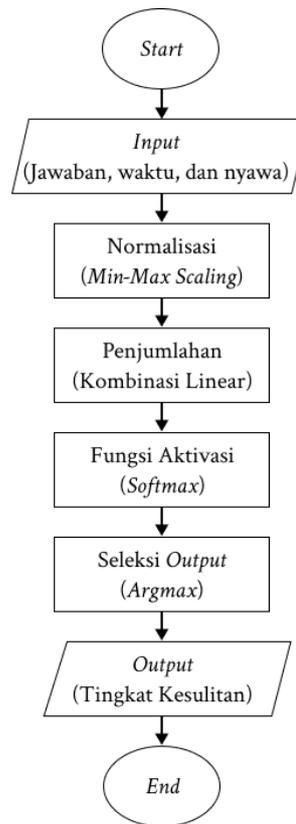


Gambar 3.10 Desain Pintu 3D

Gambar 3.11 merupakan item pintu yang berada pada ujung labirin. Pintu akan dapat dibuka jika pemain mengumpulkan seluruh sampah dalam labirin.

3.7 Perancangan *Single-layer Perceptron* (SLP)

Penggunaan metode *single-layer perceptron* dalam permainan *Save the Energy* diimplementasikan untuk menentukan tingkat kesulitan permainan berdasarkan performa yang bertujuan untuk meningkatkan pengalaman pemain. Sistem dapat menilai kemampuan pemain dan menentukan pertanyaan selanjutnya dengan tepat menggunakan data yang didapat dari pemain. Salah satu komponen utama yang digunakan pada metode ini adalah *input*. *Input* yang digunakan pada permainan ini adalah jawaban pemain, waktu pengerjaan, nyawa dan skor, dan terdapat satu variabel keluaran yaitu tingkat kesulitan. Arsitektur *single-layer perceptron* dalam penelitian ini dapat gambarkan dengan urutan sebagai berikut.



Gambar 3.11 Flowchart Single-layer Perceptron

3.7.1 Input

Tahap awal *single-layer perceptron* yaitu pengambilan data performa dari pemain yang terdiri dari 3 parameter utama, seperti yang dijelaskan pada Tabel 3.6. Data diperoleh dari hasil performa pemain pada permainan sebelumnya.

Tabel 3.5 Parameter Input

Input/Fitur	Deskripsi	Simbol
Jawaban	Jawaban benar/salah	x_1
Waktu	Waktu pengerjaan soal (detik)	x_2
Nyawa	Sisa nyawa	x_3

Dalam penerapannya, data performa akan diagregasi tiap sesi kuis. Satu sesi kuis berisi 4 soal, data yang digunakan mencakup total jawaban benar, total waktu pengerjaan, dan sisa nyawa pada akhir kuis.

3.7.2 Normalisasi (*Min-Max Scaling*)

Proses normalisasi dilakukan dengan tujuan untuk menyetarakan nilai tiap fitur. Metode yang digunakan adalah *min-max scaling* yang akan mengubah nilai kedalam rentang 0-1. Berikut formulasi *Min-Max Scaling*.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3.2)$$

Keterangan:

x : Nilai Asli

x_{min}, x_{max} = Nilai minimum dan maksimum dari data *input*

x_{norm} = Nilai yang telah dinormalisasi

Proses normalisasi juga dilakukan untuk membuat tidak ada fitur yang mendominasi karena perbedaan skala nilai.

3.7.3 Penjumlahan (Kombinasi Linear)

Proses penjumlahan menggunakan perhitungan kombinasi linear dilakukan setelah mendapatkan hasil dari proses normalisasi *input*. Setiap *input* akan dikalikan dengan bobot masing-masing. Hasil dari perkalian bobot akan dijumlah dan ditambah dengan nilai bias untuk menghasilkan *net input* (Z). Berikut formulasi kombinasi linear.

$$Z_j = \sum_{i=1}^n (w_{ij} * x_i) + b_j \quad (3.3)$$

Keterangan:

Z_j : Hasil kombinasi linear untuk neuron *output* ke- j

x_i : Nilai *input* ke- i yang telah dinormalisasi.

w_{ij} : Bobot yang menghubungkan *input* ke- i dengan neuron *output* ke- j .

n : Jumlah total *input*

b_j : Nilai bias untuk neuron *output* ke- j

Jumlah *output* pada penelitian ini sebanyak 3, yaitu tingkat kesulitan mudah, sedang, dan sulit. Tiap *input* dikalikan dengan bobot masing-masing dengan nilai masing-masing neuron. Nilai bias juga terdapat 3 macam sebanyak neuron *output*. Nilai bobot dan bias didapatkan setelah melakukan *data training*. Hasil dari perhitungan Z_j akan ada sebanyak 3, sesuai dengan neuron *output*, Z_{Mudah} , Z_{Sedang} , dan Z_{Sulit} .

3.7.4 Fungsi Aktivasi (*Softmax*)

Fungsi aktivasi mengambil nilai dari proses kombinasi linear. Penggunaan *softmax* bertujuan untuk mengubah nilai-nilai *net input* menjadi probabilitas. *Softmax* mengambil seluruh nilai *net input* tiap neuron dan mengubah menjadi distribusi probabilitas yang total untuk semua kelas adalah 1. Formulasi *Softmax* ada pada Persamaan 2.4.

3.7.5 Seleksi Output (*Argmax*)

Proses *argmax* mengambil nilai dari 3 *output* hasil proses aktivasi. Berikut formulasi *argmax*.

$$\text{argmax}(f(x)) = \text{argmax } f_i(x) \quad (3.5)$$

Keterangan:

$f_i(x)$ = Nilai *output* dari neuron ke- i

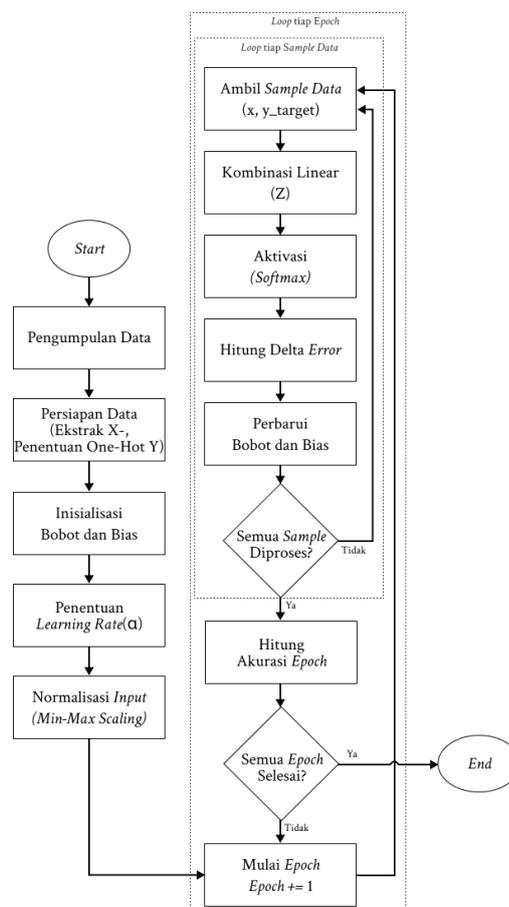
Proses ini bertujuan untuk mengubah probabilitas menjadi satu keputusan kelas dengan mengambil nilai probabilitas paling tinggi

3.8 Proses Training Single-layer Perceptron

Proses *training* untuk sistem mempelajari data dan mendapatkan bobot dan bias awal untuk proses data uji. Pada penelitian ini, terdapat dua macam *data training*, yaitu *training per soal*, dan *training per kuis*.

3.8.1 Data Training SLP Level 1 (Per Soal)

Proses *training* level 1 digunakan untuk menentukan bobot dan memiliki alur seperti pada Gambar 3.13



Gambar 3.12 Flowchart Data Training Single-layer Perceptron

Proses *data training* hampir sama dengan proses pengujian, terdapat beberapa tahap proses yang dilakukan hanya pada *data training*, seperti pembaruan bobot dan bias, perhitungan *delta error*, serta perulangan proses seluruh data dalam sejumlah *epoch*. Dalam penelitian ini, model yang dilatih menggunakan data performa 24 responden dari 20 soal yang dikerjakan.

3.8.1.1 Pengumpulan Data

Data performa dari 24 responden diproses untuk dapat menghasilkan label target (*ground truth*) untuk setiap kuis. Parameter yang diambil dari data performa adalah jawaban, dan waktu pengerjaan. Label target dikategorikan sebagai tingkat kesulitan soal. Untuk *data training*, penggunaan label target diubah menjadi format *one-hot encoding*. Berikut merupakan data performa dari 8 responden pertama pada Tabel 3.6

Tabel 3.6 Data Performa Responden

Responden	J1	J2	J3	...	J20	T1	T2	T3	...	T20
1	1	1	1		1	18	5	04		11
2	1	1	1		1	117	68	10		31
3	0	1	1		0	5	15	10		06
4	1	1	1		1	14	11	10		19
5	1	1	1		0	20	16	19		11
6	0	0	1		0	19	27	10		44
7	0	1	1		0	07	09	07		13
8	1	1	1		1	18	11	10		23

J1 sampai J2 merupakan parameter jawaban untuk 20 soal, dengan nilai 1 jika benar dan 0 jika salah. Dan T1 sampai T20 merupakan parameter waktu (per detik) dalam menjawab pertanyaan dengan soal.

3.8.1.2 Persiapan Data

Tahap ini merupakan tahap pengolahan data mentah menjadi format yang siap diproses untuk model SLP. Terdapat tiga langkah persiapan data, yaitu ekstraksi fitur *input* (x), pelabelan target (y), dan konversi label ke format *one-hot encoding*.

1. Ekstraksi Fitur *Input* (x)

Dalam data performa hanya terdapat dua parameter, penambahan parameter nyawa dilakukan untuk parameter kemampuan kognitif dalam *game*. Total nyawa pada *game save the energy* adalah sebanyak tujuh (7) nyawa. Nyawa akan berkurang satu ketika pemain menjawab salah pada satu soal, jika nyawa mencapai nol (0), maka permainan berakhir. Pada data responden, ketika responden sudah melakukan kesalahan menjawab soal sebanyak tujuh kali atau nyawa sama dengan habis, maka soal-soal berikutnya tidak terbaca oleh proses *training*. Berikut merupakan contoh data responden yang sudah ditambahkan parameter nyawa, terdapat pada Tabel 3.7

Tabel 3.7 Penambahan Parameter Nyawa pada Data Responden

ID_Pemain	ID_Soal	Jawaban	Waktu(s)	SisaNyawa
7	1	0	7	6
7	2	1	9	6
7	3	1	7	6
7	4	0	7	5
7	5	1	7	5
7	6	1	10	5
7	7	0	24	4
7	8	1	9	4

Nyawa awal sebanyak 7, namun pemain sudah menjawab salah soal nomor 1, maka dari itu nyawa tersisa 6, begitu pun data selanjutnya.

2. Pelabelan Target (Y) dan *One-Hot Encoding*

Proses pelabelan tingkat kesulitan bertujuan untuk menilai performa tiap soal secara individual. Berikut *pseudocode* yang digunakan dalam pelabelan.

```

FUNCTION determine_difficulty(jawaban, waktu, sisa_nyawa,
max_nyawa)
  IF jawaban == 1 AND waktu < 15 AND sisa_nyawa >=
max_nyawa * 0.75) THEN
    RETURN "Mudah"

  ELSE IF (jawaban == 1 AND waktu >= 15) OR
(jawaban == 0 AND sisa_nyawa >= (max_nyawa /2)
    RETURN "Sedang"

  ELSE
    RETURN "Sulit"
END

```

Tingkat kesulitan soal berlabel "Mudah" ketika responden menjawab benar, sangat cepat dibawah 15 detik, dan masih memiliki banyak nyawa tersisa. Tingkat kesulitan soal akan berlabel "Sedang" ketika responden menjawab benar namun tidak terlalu cepat, atau responden menjawab salah tetapi masih memiliki jumlah nyawa yang sedang saja. Tingkat kesulitan soal berlabel "Sulit" ketika responden menjawab salah, menjawab dengan lambat, dan memiliki sisa nyawa yang sedikit. Pada Tabel 3.8 memaparkan hasil dari pelabelan soal salah satu responden.

Tabel 3.8 Pelabelan Data Responden

ID Pemain	ID Soal	Jawaban	Waktu(s)	SisaNyawa	Label	<i>One-Hot Encoding</i>
7	1	0	7	6	Sedang	[0, 1, 0]
7	2	1	9	6	Mudah	[1, 0, 0]
7	3	1	7	6	Mudah	[1, 0, 0]
7	4	0	7	5	Sedang	[0, 1, 0]
7	5	1	7	5	Mudah	[1, 0, 0]
7	6	1	10	5	Mudah	[1, 0, 0]
7	7	0	24	4	Sulit	[0, 0, 1]
7	8	1	9	4	Sedang	[0, 1, 0]

Dapat dilihat bahwa responden nomor tujuh mengerjakan soal dengan beragam tingkat kesulitan sesuai dengan performa responden tersebut. Pelabelan *one-hot encoding* dibutuhkan karena fungsi aktivasi *softmax* membutuhkan target dalam bentuk biner. Berikut merupakan urutan kelas untuk *one-hot encoding* pada Tabel 3.9

Tabel 3.9 Penentuan Format *One-Hot Encoding*

Tingkat Kesulitan	<i>One-Hot Encoding</i>
Mudah	[1, 0, 0]
Sedang	[0, 1, 0]
Sulit	[0, 0, 1]

Elemen yang memiliki nilai 1 merupakan vektor biner yang menunjukkan bahwa mewakili salah satu tingkat kesulitan.

3.8.1.3 Normalisasi Data

Setelah melakukan labeling dan mengubahnya menjadi format *one-hot encoding*, data 24 responden diagregasi untuk mendapatkan tingkat kesulitan per soal. Setelah data diagregasi, dilakukan proses normalisasi seperti pada Tabel 3.10

Tabel 3.10 Data Responden Hasil Normalisasi

ID_Soal	Norm_Akurasi (x_1)	Norm_Waktu (x_2)	Norm_Nyawa (x_3)	<i>Ground Truth</i>
1	0.8	0,193277311	0,866666667	Sedang
2	0,933333333	0,208465608	0,833333333	Mudah
3	1	0,333333333	0,833333333	Mudah
4	0,866666667	0,254545455	0,844444444	Mudah
5	0,466666667	0,230434783	0,75	Sedang
6	0,733333333	0,201333333	0,683333333	Sedang
7	0,866666667	0,236419753	0,72	Sedang

Normalisasi parameter bertujuan untuk membuat nilai *input* menjadi seragam dengan skala 0-1.

3.8.1.4 Penentuan *Learning Rate* dan Inisialisasi Bobot, Bias

Learning rate (α) digunakan untuk memastikan model dapat belajar secara efektif dan mencapai konvergensi yang optimal. Parameter *learning rate* akan menentukan seberapa besar langkah penyesuaian bobot dan bias yang akan diambil di setiap iterasi pelatihan. Nilai *learning rate* yang digunakan pada penelitian ini adalah 0.1.

Sebelum *data training* dimulai, bobot dan bias awal untuk SLP harus diinisialisasi terlebih dahulu. Nilai bobot dan bias awal ditentukan secara acak dengan angka kecil. Bobot mempresentasikan tingkat pengaruh pada masing-masing *neuron output* terhadap kesulitan soal, dan bias adalah ambang batas untuk setiap *output neuron*. Berikut *pseudocode* inisialisasi bobot dan bias secara acak.

```

FUNCTION InisialisasiBobotBias(input_size, kelas, random_state):
  SET random_seed TO random_state
  APPLY random_seed TO RANDOM_NUMBER_GENERATOR

  SET this.input_size TO input_size
  SET this.kelas TO kelas

  SET this.bobot TO RANDOM_NORMAL_DISTRIBUTION(mean=0,
    std_dev=0.1, shape=(input_size, kelas))
  SET this.bias TO ARRAY_OF_ZEROS(size=kelas)

END

```

Penggunaan *random_state* memastikan bahwa inisialisasi ini dapat direproduksi, dapat diartikan bahwa ketika kode berjalan, angka bobot dan bias awal akan selalu sama. Pada Tabel 3.11 merupakan bobot dan bias awal yang digunakan pada penelitian ini

Tabel 3.11 Inisialisasi Bobot dan Bias Awal

Kelas	Bias	w_1	w_2	w_3
Mudah	-0.3087	0.0819	-0.0141	0.0328
Sedang	-0.3172	-0.014	-0.0237	0.0748
Sulit	-0.2047	0.0535	-0.0238	-0.0536

Bobot diinisialisasi menggunakan angka acak dari distribusi normal dengan rata-rata nol dan standar deviasi 0.1. Dan bias diinisialisasi dengan nilai acak antara -0.5 sampai 0.5.

3.8.1.5 Proses Iterasi

Training model SLP dilakukan selama beberapa *epoch*. 1 *epoch* merupakan satu siklus seluruh dataset pelatihan sudah diproses oleh model. Proses iterasi dapat berhenti ketika memenuhi salah satu dari beberapa kriteria, yaitu:

1. Jumlah *epoch* sudah mencapai maksimum.
2. Konvergensi *error* dikarenakan nilai *error* sudah sangat kecil dan tidak mengalami penurunan yang signifikan dalam beberapa *epoch* secara berturut-turun.
3. *Early stopping* ketika performa data validasi tidak membaik selama beberapa *epoch* untuk menghindari *overfitting*.
4. Akurasi mencapai 100%.

Untuk tiap sampel *data*, model akan melalui langkah-langkah berikut yang dijelaskan dalam *training* data soal nomor 1 pada *epoch* ke-1.

3.8.1.5.1 Pengambilan Sampel Data

Sampel data yang digunakan untuk proses iterasi adalah data responden yang sudah dinormalisasi, *ground truth*, bobot awal, dan bias awal.

3.8.1.5.2 Pembobotan dan Kombinasi Linear (Z)

Input yang sudah dinormalisasi dikalikan dengan bobot masing-masing dan dijumlahkan, kemudian ditambahkan dengan bias. Proses tersebut menghasilkan *net input* (Z). Berikut hasil perhitungan kombinasi linear.

$$\begin{aligned} Z_1 &= (0,0611 * 0,8) + (0,153 * 0,193277311) + (-0,1665 * 0,866666667) \\ &\quad + - 0.3087 \\ &= 0.04888 + 0.029571428 + (-0.1443) + (-0.3087) \\ &= -0.37455 \end{aligned}$$

$$Z_2 = -0.272873968$$

$$Z_3 = -0.2038$$

3.8.1.5.3 Fungsi Aktivasi (*Softmax*)

Hasil dari kombinasi linear diubah menjadi probabilitas, seperti pada perhitungan matematis dibawah ini.

$$y_{prediksi,Mudah} = \frac{2.225541}{2.225541+1.213219+2.378968} = 0.382544696$$

$$y_{prediksi,Sedang} = \frac{1.213219}{2.225541+1.213219+2.378968} = 0.208538319$$

$$y_{prediksi,Sulit} = \frac{2.378968}{2.225541+1.213219+2.378968} = 0.408917$$

$$\text{Total } y_{prediksi} = 0.382544696 + 0.208538319 + 0.408917 = 1$$

Hasil dari probabilitas *softmax* jika dijumlahkan akan menghasilkan nilai 1, maka dari itu *softmax* sangat cocok digunakan dalam sistem prediksi yang *outputnya* hanya akan dipilih 1.

3.8.1.5.4 Perhitungan Delta Error

Hasil aktivasi dibandingkan dengan target sebenarnya (y_{target}) untuk menghitung *error*. Dan menggunakan *Categorical Cross-Entropy* (CCE) untuk mengukur fungsi *loss*. Formulasi fungsi *loss* adalah sebagai berikut.

$$L = - \sum_{j=1}^{N_{output}} y_{target,j} * \log(y_{prediksi,j}) \quad (3.5)$$

Keterangan:

L : Nilai *loss* untuk satu sample

N_{output} : Neuron *output*

$y_{target,j}$: Nilai target sebenarnya pada kelas j

$y_{prediksi,j}$: *Output* aktivasi

Proses perhitungan untuk *epoch* 1 nomor soal 1:

$$L = -(y_{target,1} * \log(y_{prediksi,1})) + (y_{target,2} * \log(y_{prediksi,2})) + (y_{target,3} * \log(y_{prediksi,3}))$$

$$L = -[(1 * \log(0.382544696)) + (0 * \log(0.208538319)) + (0 * \log(0.408917))]$$

$$L = -[(1 * \log(0.382544696))]$$

$$L = -\log(0.382544696) \approx 0.417317815$$

Proses perhitungan *delta error* untuk memandu pembaruan bobot. *Delta error* adalah nilai selisih dari fungsi aktivasi dan nilai target. Berikut formulasi *delta error*.

$$\delta_j = y_{prediksi,j} - y_{target,j} \quad (3.6)$$

Keterangan:

δ_j : Neuron *output*

$y_{target,j}$: Nilai target sebenarnya pada kelas j

$y_{prediksi,j}$: *Output* aktivasi

Proses perhitungan *delta error*:

$$\delta_{Mudah} = 0.382544696 - 1 = -0,617455304$$

$$\delta_{Sedang} = 0.208538319 - 0 = 0.208538319$$

$$\delta_{Sulit} = 0.408917 - 0 = 0.408917$$

3.8.1.5.5 Pembaruan Bobot dan Bias

Pembaruan bobot dilakukan untuk menyesuaikan nilai bobot untuk mengurangi *error* pada prediksi SLP berikutnya. Bobot diperbarui menggunakan hasil dari *delta error*, *learning rate*, dan nilai *input*. Berikut formulasi pembaruan bobot.

$$\Delta w_{ij} = \alpha \cdot \delta_j \cdot x_i \quad (3.7)$$

Keterangan:

Δw_{ij} : Perubahan bobot *input* ke- i pada neuron *output* ke- j

α : *Learning rate*

δ_j : *Delta error* untuk neuron *output* ke- j

x_i : Nilai *input* ke- i

Pembaruan bias diperbarui untuk mengoptimalkan kinerja model. Bias memungkinkan untuk menyesuaikan ambang batas aktivasinya secara independen.

Berikut formulasi pembaruan bias.

$$\Delta b_j = \alpha \cdot \delta_j \quad (3.8)$$

Keterangan:

Δb_j : Perubahan bias pada neuron *output* ke-*j*

α : *Learning rate*

δ_j : Delta *error* untuk neuron *output* ke-*j*

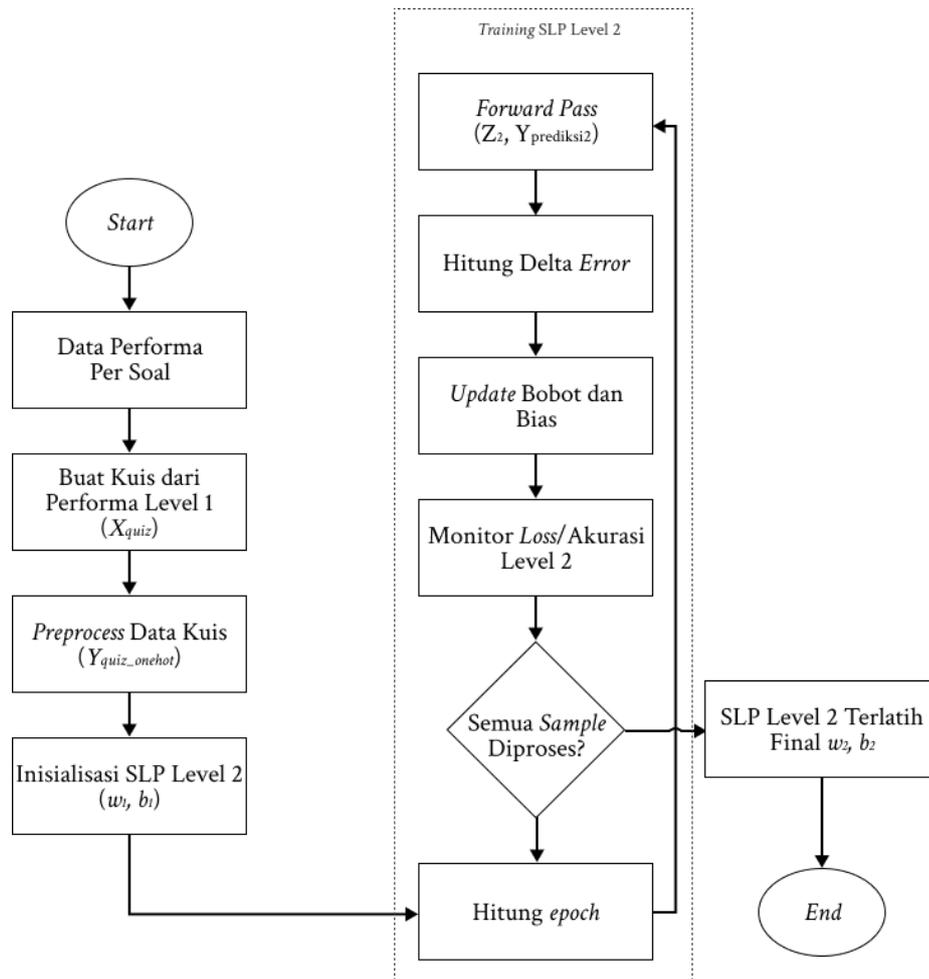
Berikut merupakan hasil proses pembaruan bobot dan bias yang dipaparkan pada Tabel 3.12

Tabel 3.12 Pembaruan Bobot dan Bias

Kelas	Bias	w_1	w_2	w_3
Mudah	-0.292276	0.081883	-0.014113	0.032844
Sedang	-0.31802	0.15359	-0.02422	-0.023896
Sulit	-0.220232	0.182064	0.070996	-0.065343

3.8.2 Data Training SLP Level 2 (Per Kuis)

Setelah berhasil melakukan *training* pada 20 soal hasil performa responden, serta nilai bobot dan bias berhasil dioptimalkan, proses selanjutnya adalah melatih dataset sesuai dengan konsep yang akan diimplementasikan dalam *game*. Alur dari *data training* level 2 dijelaskan pada Gambar 3.14.



Gambar 3.13 Flowchart Training SLP Level 2

Model SLP Level 2 digunakan untuk memprediksi tingkat kesulitan per kuis, sesuai dengan konsep dalam *game*. Sistem akan menentukan tingkat kesulitan kuis selanjutnya berdasarkan prediksi hasil performa kuis sebelumnya dengan menggunakan 6 parameter, yaitu 3 fitur rata-rata probabilitas prediksi dari SLP Level 1, dan 3 fitur rata-rata performa asli dari 4 soal dalam kuis. Dalam tiap kuis berisi 4 soal. Data performa keempat soal diproses terlebih dahulu oleh SLP Level 1 untuk mendapatkan probabilitas tingkat kesulitan, kemudian hasil tersebut dihitung rata-rata per kategori kelas. Dan fitur performa asli dari 4 soal dihitung rata-ratanya.

3.8.2.1 Persiapan Data

Dataset untuk *training* model SLP Level 2 tidak diambil secara langsung dari data mentah performa responden, melainkan dibangun secara dinamis dari proses SLP Level 1 dan agregasi dari data performa asli. Proses ini menggabungkan klasifikasi tingkat kesulitan tiap soal dengan metrik performa yang objektif.

Data pada SLP Level 2 menggunakan dataset hasil SLP Level 1, yaitu rata-rata probabilitas tingkat prediksi tiap kelas dan dataset performa responden. Berikut perincian parameter yang digunakan pada SLP Level 2 seperti pada Tabel 3.13

Tabel 3.13 Parameter pada *Training* SLP Level 2

Parameter	Keterangan
Rata-rata probabilitas prediksi kelas “Mudah”	$P_{avg,Mudah}$
Rata-rata probabilitas prediksi kelas “Sedang	$P_{avg,Sedang}$
Rata-rata probabilitas prediksi kelas “Sulit”	$P_{avg,Sulit}$
Rata-rata normalisasi akurasi	\bar{A}_{kuis}
Rata-rata normalisasi waktu	\bar{T}_{kuis}
Rata-rata normalisasi nyawa	\bar{N}_{kuis}
Kelas <i>Output</i> (<i>One-hot encoding</i>)	Mudah: [1, 0, 0] Sedang: [0, 1, 0] Sulit: [0, 0, 1]
Fungsi aktivasi	<i>Softmax</i>
Fungsi <i>loss</i>	CCE

Parameter *input* (3 rata-rata probabilitas prediksi kelas dan 3 rata-rata performa) akan digabung menjadi satu vektor fitur tunggal pada Kuis 1. Proses *training* diulang untuk tiap kuis hingga menghasilkan 5 sampel data kuis untuk pelatihan SLP Level 2.

Dalam 1 kuis terdapat 4 soal yang sudah diproses oleh model SLP Level 1 yang menghasilkan probabilitas dari fungsi aktivasi *softmax*. Nilai probabilitas dari 4 soal diambil kemudian dihitung rata-rata per kategori. Data performa pemain asli

dinormalisasi dan dihitung rata-rata dari 4 soal tiap fiturnya. Data performa responden pada kuis 1 dipaparkan pada Tabel 3.14.

Tabel 3.14 Data Normalisasi Performa pada Kuis 1

No. Soal	Akurasi	Waktu	Nyawa	Ground Truth
1	0.80	0.193277	0.866667	Mudah
2	0.933333	0.208466	0.833333	Mudah
3	1	0.333333	0.833333	Sedang
4	0.866667	0.254545	0.844444	Mudah

Dari data performa tiap soal dalam 1 kuis, dilakukan agregasi yang akan menjadi *input* model SLP Level 2. Hasil agregasi semua fitur (probabilitas soal dan performa) untuk model SLP Level 2 adalah sebagai berikut:

a. Rata-rata probabilitas soal pada model SLP Level 1:

- $P_{avg,Mudah} = 0.6611$
- $P_{avg,Sedang} = 0.1831$
- $P_{avg,Sulit} = 0.1558$
- $Ground\ truth = Mudah$

b. Rata-rata performa dalam 1 kuis:

- $\bar{A}_{kuis} = 0.8125$
- $\bar{T}_{kuis} = 0.1875$
- $\bar{N}_{kuis} = 0.8325$

3.8.2.2 Inisialisasi Bobot dan Bias

Nilai bobot dan bias awal diinisialisasi secara acak menggunakan fungsi *random_state* seperti pada model SLP Level 1. Hasil bobot dan bias awal disebutkan pada Tabel 3.15

Tabel 3.15 Inisialisasi Bobot dan Bias pada *Training* SLP Level 2

Parameter	Mudah	Sedang	Sulit
$P_{avg,Mudah}$	-0.108563	0.099735	0.028298
$P_{avg,Sedang}$	-0.150629	-0.05786	0.165144
$P_{avg,Sulit}$	-0.242668	-0.042891	0.126594
\bar{A}_{kuis}	-0.086674	-0.067889	-0.009471
\bar{T}_{kuis}	0.149139	-0.06389	-0.044398
\bar{N}_{kuis}	-0.043435	0.220593	0.218679
Bias	0.134401	0.349432	0.224455

Bobot diinisialisasi menggunakan angka acak dari distribusi normal dengan rata-rata nol dan standar deviasi 0.1. Dan bias diinisialisasi dengan nilai acak antara -0.5 sampai 0.5 untuk menghindari simetri dan memungkinkan setiap neuron memiliki bias awal yang berbeda.

3.8.2.3 Proses Iterasi

Proses iterasi pada SLP Level 2 melibatkan beberapa proses yang diulang selama beberapa *epoch*. Proses ini bertujuan untuk memperbarui bobot dan bias model untuk meningkatkan akurasi prediksi. Dalam tiap sampel data kuis yang diambil dari dataset *training* (5 sampel kuis), akan melalui tahap-tahap berikut:

1. Pengambil Sampel Data

Data yang digunakan berasal dari satu sampel data kuis beserta label target dalam format *one-hot encoding*.

2. Pembobotan dan Kombinasi Linear

Proses perkalian sampel data (x_i) dengan bobot (w_{ij}) masing-masing kelas dan dijumlahkan, kemudian ditambah dengan bias dan menghasilkan *net input* (Z_j). Berikut perhitungan pada sampel awal.

$$Z_1 = (0,0611 \times -0,108563) + (0,1831 \times -0,150629) + (0,1558 \times -0,242668) + (0,8125 \times -0,086674) + (0,1875 \times -0,063890) + (0,8325 \times 0,220593) + 0,134401 \approx -0,0764$$

$$Z_2 \approx 0,4475$$

$$Z_3 \approx 0,4137$$

Nilai *net input* pada tiap kelas adalah $Z \approx [-0.0764, 0.4475, 0.4137]$

3. Fungsi Aktivasi (*Softmax*)

Proses perubahan *net input* menjadi probabilitas prediksi ($Y_{prediksi,j}$) untuk memastikan bahwa total probabilitas semua kelas adalah 1. Berikut perhitungan lanjutan dari *net input*.

$$e^{-0.0763} \approx 0.9265$$

$$e^{0.4475} \approx 1.5644$$

$$e^{0.4137} \approx 1.5125$$

Jumlah eksponensial: $0.9265 + 1.5644 + 1.5125 = 4.0034$

$$Y_{prediksi,Mudah} = 0.9265/4.0034 \approx 0.2314$$

$$Y_{prediksi,Sedang} = 1.5644/4.0034 \approx 0.3908$$

$$Y_{prediksi,Sulit} = 1.5125/4.0034 \approx 0.3778$$

Probabilitas prediksi $Y_{prediksi} \approx [0.2314, 0.3908, 0.3778]$

4. Perhitungan Delta *Error*

Hasil dari probabilitas prediksi dibandingkan dengan label target sebenarnya (Y_{target}) untuk menghitung *error* menggunakan fungsi CCE

untuk mengukur loss. Berikut proses hitung *loss* dan delta *error* dari sampel sebelumnya.

$$Y_{target}[0] = [1.0, 0.0, 0.0]$$

$$L = -(1,0 \times \log(0,2314) + 0,0 \times \log(0.3908) + 0,0 \times \log(0.3778)) \\ \approx -(-1,464) = 1.464$$

$$\delta_{Mudah} = 0.2314 - 1 = -0,7686$$

$$\delta_{Sedang} = 0.3908 - 0 = 0.3908$$

$$\delta_{Sulit} = 0.3778 - 0 = 0.3778$$

5. Pembaruan Bobot dan Bias

Proses pembaruan bobot dan bias didasarkan pada perhitungan delta *error*, *learning rate*, dan nilai *input*. Pembaruan bias berdasarkan delta *error* dan *learning rate*. Berikut proses hitung pada *training* model SLP Level 2 untuk mendapatkan bobot dan bias baru dalam *epoch* 1 dan kuis ke-1. Perhitungan delta bobot dan delta bias ada pada Tabel 3.16.

Tabel 3.16 Perhitungan Delta Bobot dan Delta Bias

Parameter	w_{Mudah} (-0.6428)	w_{Sedang} (0.3283)	w_{Sulit} (0.3145)
x_1 (0.6611)	$0.2 * (-0.6428) *$ $0.6611 \approx$ -0.084994	0.043410	0.041584
x_2 (0.1831)	-0.023538	0.012022	0.011516
x_3 (0.1558)	-0.020027	0.010229	0.009798
x_4 (0.8125)	-0.104455	0.053350	0.051105
x_5 (0.1875)	-0.024105	0.012311	0.011793
x_6 (0.8325)	-0.107026	0.054663	0.052363
Δb	$0.2 * (-0,6428) \approx$ -0.128559	0.065661	0.062898

Setelah mendapatkan delta bobot dan bias, berikut perhitungan pembaruan bobot dan bias dengan hasil yang dipaparkan pada Tabel 3.17.

$$w_{baru} = w_{lama} + \Delta w \quad (3.10)$$

$$b_{baru} = b_{lama} + \Delta b \quad (3.11)$$

Tabel 3.17 Hasil Pembaruan Bobot dan Bias (SLP Level 2, Setelah *Epoch* 1)

Parameter	Mudah	Sedang	Sulit
x_1	-0.108563 + (-0.084994) = -0.193557	0.143145	0.069882
x_2	-0.174167	-0.045838	0.176660
x_3	-0.262695	-0.032662	0.136392
x_4	-0.191129	-0.014539	0.041634
x_5	0.125034	0.051579	-0.032605
x_6	-0.150461	0.275256	0.271042
Bias	0.1344401 + (-0.12856) = 0.005841	0.415092	0.287355

Tabel diatas menampilkan nilai-nilai bobot dan bias setelah satu *epoch* pertama pada *training* SLP Level 2. Nilai-nilai tersebut akan menjadi bobot dan bias baru untuk perhitungan pada *epoch* berikutnya hingga mencapai bobot dan bias yang optimal untuk kinerja model secara keseluruhan.

BAB IV

HASIL DAN PEMBAHASAN

Dalam bab ini akan dibahas tentang implementasi metode, hasil dan pembahasan sistem dalam aplikasi. Implementasi hasil terdiri dari hasil implementasi *user interface*/tampilan. Implementasi metode terdiri dari hasil *data training* model *single-layer perceptron* level 1 (per soal), hasil *data training* model *single-layer perceptron* level 2 (per kuis), hasil *data testing*, dan hasil *usability testing*.

4.1 Hasil Implementasi Tampilan *Game*

Implementasi permainan edukasi *Save the Energy* telah berhasil menampilkan tampilan *user interface* (UI) *game*. Tampilan pada *game* terdiri dari menu utama, menu petunjuk, *interface gameplay*, tampilan kuis dan tampilan *gameover*. Dalam UI tersebut terdapat tombol-tombol yang dapat berinteraksi dengan pemain, seperti tombol mulai, petunjuk, kembali, keluar, *paused*, dan *restart*.

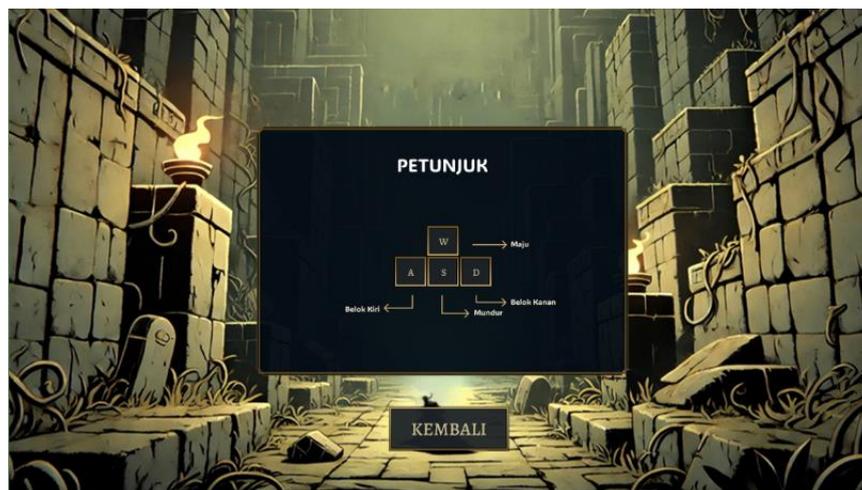
4.1.1 UI Menu Utama



Gambar 4.1 User Interface Main Menu

Menu utama seperti pada Gambar 4.1 ini terdapat tiga tombol aktif, yaitu “Mulai” untuk memasuki permainan, “Petunjuk” yang berisi panduan permainan, dan “Keluar” untuk keluar dari aplikasi.

4.1.2 UI Menu Petunjuk



Gambar 4.2 User Interface Petunjuk

Menu petunjuk seperti yang tertera pada Gambar 4.2 berisi panduan penggunaan tombol dalam aplikasi.

4.1.3 UI Gameplay



Gambar 4.3 User Interface Gameplay "Save the Energy"

Pada tampilan *gameplay* pada Gambar 4.3 terdapat layar utama yang menampilkan permainan, tombol *pause* untuk jeda, *timer* untuk mengetahui seberapa lama player ada didalam labirin, dan ikon sampah terkumpul.

4.1.4 UI Kuis



Gambar 4.4 User Interface Kuis

Tampilan kuis pada Gambar 4.4 muncul ketika pemain mendekati *item* sampah. Dalam tampilannya, terdapat pertanyaan, jawaban dengan pilihan ganda, tombol *pause* untuk jeda, dan sisa nyawa yang dimiliki pemain.

4.1.5 UI Game Over



Gambar 4.5 User Interface Game Over Scene

Tampilan *scene game over*, seperti pada Gambar 4.5 akan muncul ketika nyawa pemain habis sebelum pemain berhasil keluar dari labirin. Terdapat tombol aktif pada *scene* ini, tombol *restart* untuk mengulang permainan, dan *main menu* untuk kembali ke menu awal.

4.2 Hasil Data Training SLP

Proses data training menyajikan hasil dari proses pelatihan model *Single-layer Perceptron* (SLP) yang diimplementasikan secara hierarkis. Pelatihan dilakukan untuk dua level model: SLP Level 1 yang memprediksi tingkat kesulitan per soal, dan SLP Level 2 yang memprediksi tingkat kesulitan per kuis. Tujuan pelatihan ini adalah untuk mengoptimalkan bobot dan bias kedua model agar mampu mengklasifikasikan performa pemain secara akurat.

4.2.1 Hasil Data Training Model SLP Level 1 (Per Soal)

Model SLP Level 1 dilatih menggunakan dataset yang berisi 20 sampel data performa per soal. Distribusi kelas pada sampel data tertera pada Tabel 4. Proses pelatihan dilakukan sebanyak 2000 *epoch* dengan *learning rate* sebesar 0.1.

Tabel 4.1 Distribusi Kelas Sampel Soal

Kelas	Total
Mudah	9
Sedang	6
Sulit	5

Berikut nilai optimal bobot dan bias hasil dari *training* SLP Level 1 tertera pada Tabel 4.2.

Tabel 4.2 Output Model SLP Level 1 (Bobot dan Bias akhir)

Parameter	Mudah	Sedang	Sulit
Akurasi	3,336036	0,90119	-4,13661
Waktu	-3,22102	-0,34063	3,667122
Nyawa	1,920619	-1,74002	0,007115

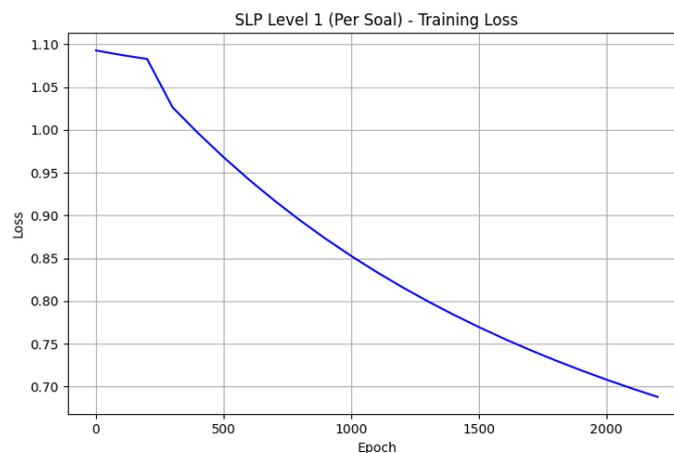
Pada *epoch* pertama, model menunjukkan nilai *loss* awal sebesar 1.092780 dan akurasi 0.4500. seiring berjalannya *epoch*, *loss* model konsisten mengalami penurunan meskipun melambat pada *epoch* terakhir. Data *epoch* dapat dilihat pada Tabel 4.3

Tabel 4.3 Epoch Training SLP Level 1

Epoch	Loss	Accuracy
100	1,09278	0,45
200	1,087482	0,45
300	1,082905	0,45
400	1,026325	0,45
500	0,996163	0,45
600	0,967989	0,45
700	0,941677	0,45
800	0,917099	0,5
900	0,894135	0,55
1000	0,872666	0,6
1100	0,852583	0,7

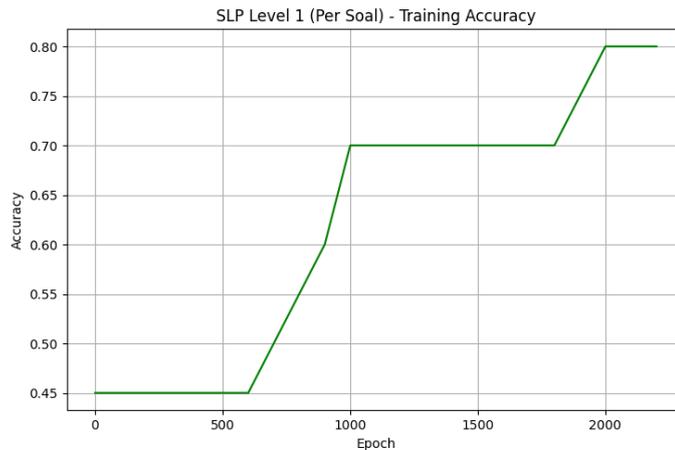
<i>Epoch</i>	<i>Loss</i>	<i>Accuracy</i>
1200	0,833781	0,7
1300	0,81616	0,7
1400	0,799629	0,7
1500	0,784104	0,7
1600	0,769506	0,7
1700	0,755763	0,7
1800	0,742808	0,7
1900	0,730581	0,7
2000	0,719026	0,75
2100	0,708093	0,8
2200	0,697733	0,8
2300	0,687905	0,8

Akurasi model juga menunjukkan peningkatan yang stabil. Pada Gambar 4.6 adalah grafik *training loss* yang membuktikan bahwa model mengalami penurunan *loss* yang konsisten.



Gambar 4.6 *Training Loss* pada SLP Level 1

Pada akhir pelatihan, yaitu setelah 2000 epoch, Model SLP Level 1 berhasil mencapai nilai *loss* sebesar 0.687905 dan akurasi pada data pelatihan sebesar 0.8000 (80%) seperti yang dipaparkan pada Gambar 4.7



Gambar 4.7 Grafik Akurasi *Training Data* pada SLP Level 1

Peneliti menghentikan proses *epoch* diangka 2000 dikarenakan akurasi stagnan dipersentase 80%, bahkan terjadi penurunan sedikit di akhir *epoch* 4000. Nilai *loss* hanya menurun perlahan, yang menandakan model sudah mencapai titik jenuh yang menyebabkan tidak akan ada peningkatan signifikan dengan bertambah banyaknya *epoch*. Akurasi ini menunjukkan kemampuan model dalam mengidentifikasi tingkat kesulitan performa pemain pada setiap soal secara individual.

4.2.2 Hasil *Data Training* Model SLP Level 2 (Per Kuis)

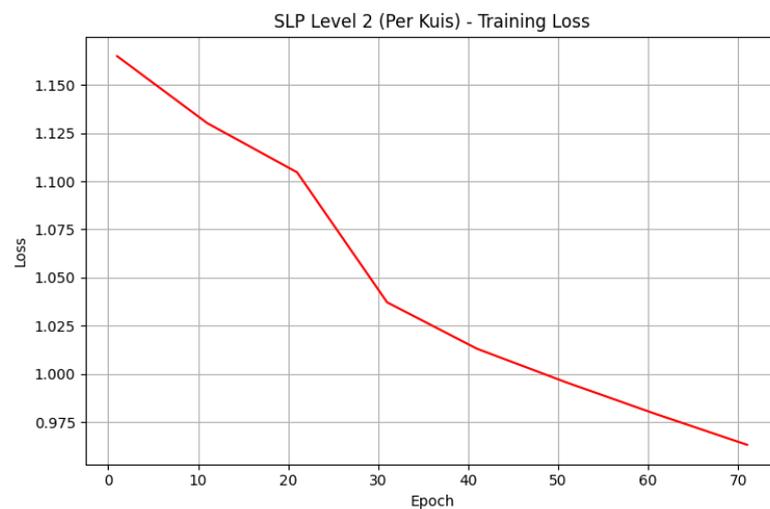
Setelah Model SLP Level 1 terlatih, output (probabilitas prediksi) dari model tersebut digunakan untuk membentuk dataset pelatihan bagi Model SLP Level 2. Dataset untuk level 2 ini terdiri dari 5 sampel data kuis, dengan distribusi kelas 'Mudah': 2, 'Sedang': 2, dan 'Sulit': 1. Input bagi SLP Level 2 terdiri dari 6 fitur: rata-rata probabilitas prediksi dari SLP Level 1 untuk tiga kelas, dan rata-rata performa asli (NormAkurasi, NormWaktu, NormNyawa) dari keempat soal dalam satu kuis.

Model SLP Level 2 dilatih selama 50 epoch dengan *learning rate* sebesar 0.2. Pada epoch pertama, model memiliki nilai loss awal sebesar 1.164836 dan akurasi 0.2000, seperti pada Tabel 4.4.

Tabel 4.4 *Epoch Training* Model SLP Level 2

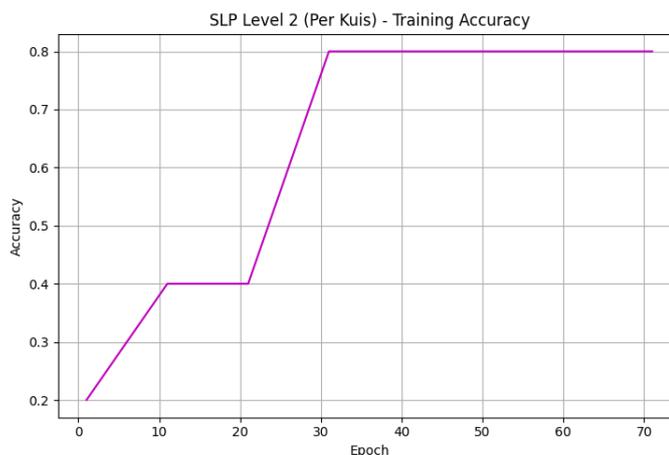
<i>Epoch</i>	<i>Loss</i>	<i>Accuracy</i>
10	1,164836	0,2
20	1,130141	0,4
30	1,104588	0,4
40	1,037195	0,8
50	1,013096	0,8
60	0,995533	0,8
70	0,979052	0,8
80	0,963315	0,8

Meskipun dataset untuk Level 2 sangat kecil, model menunjukkan kemampuan untuk belajar. Hasil dari *training loss* SLP Level 2 dapat dilihat pada Gambar 4.8



Gambar 4.8 *Training Loss* SLP Level 2

Pada akhir pelatihan, yaitu setelah 50 epoch, Model SLP Level 2 mencapai nilai loss sebesar 0.963315 dan akurasi pada data pelatihan sebesar 0.8000 (80%) seperti yang tertera pada Gambar 4.9.



Gambar 4.9 Grafik Akurasi *Training* SLP Level 2

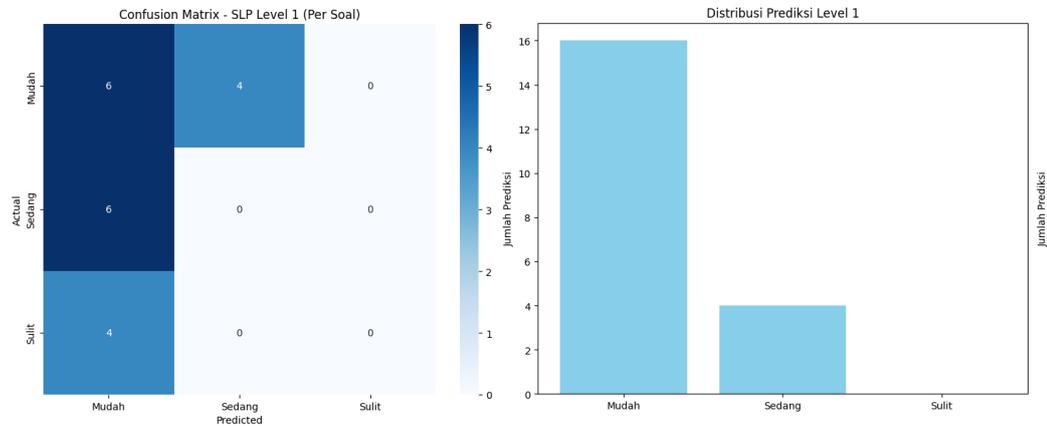
Pelatihan data *training* berhenti di *epoch* 100 dikarenakan akurasi sudah stagnan berada di persentase 80% sejak *epoch* 30 dan hampir tidak ada peningkatan setelah *epoch* 30. Meskipun *loss* masih terjadi penurunan, namun penurunannya cenderung lambat. Maka dari itu *training* diberhentikan pada *epoch* 100 karena akurasi yang sudah stabil.

4.3 Hasil *Data Testing*

Data uji dilakukan sebagai evaluasi sistem penyesuaian tingkat kesulitan. Data uji menggunakan data terpisah dari data *training*, yaitu menggunakan data responden baru sebanyak 20 orang. Hasil performa responden digunakan untuk memproses *data testing* dalam uji coba metode SLP.

4.3.1 Evaluasi SLP Level 1

Model SLP Level 1 dievaluasi menggunakan 20 sampel data uji soal. Hasil akurasi menunjukkan 0.8000 (80.00%). Hasil tersebut mengindikasikan kinerja yang baik dalam prediksi tingkat kesulitan. Detail prediksi per soal disajikan per soal dan laporan klasifikasi disajikan pada Gambar 4.10 dan Tabel 4.7



Gambar 4.10 *Confusion Matrix* SLP dan Distribusi Prediksi Level 1

Confusion matrix pada Gambar 4.10 (kiri) menunjukkan performa model SLP Level 1 dalam mengklasifikasikan tingkat kesulitan per soal. Dari total 10 soal dengan tingkat kesulitan 'Mudah', model berhasil memprediksi 6 di antaranya dengan benar, tetapi 4 lainnya salah prediksi menjadi “Sedang” atau “Sulit”. Untuk 6 soal 'Sedang' dan 4 soal “Sulit” model gagal memprediksi satu pun dengan benar. Ini menegaskan bahwa bias model dalam memprediksi kelas “Mudah”. Distribusi prediksi SLP Level 1 pada Gambar 4.10 (kanan) mengkonfirmasi hal ini, di mana sebagian besar prediksi jatuh pada kategori 'Mudah'.

Tabel 4.5 *Classification Report* SLP Level 1

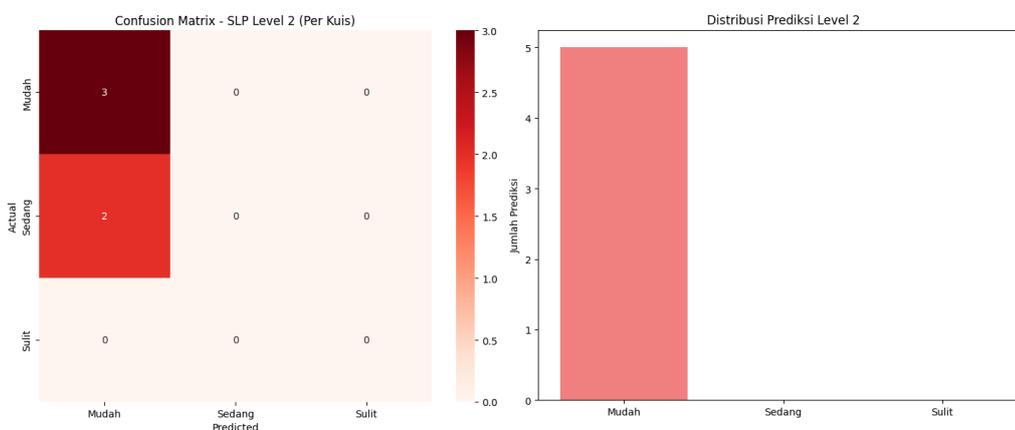
	<i>Precision</i>	<i>Recall</i>	<i>F1 score</i>	<i>Support</i>
Mudah	0.89	0.89	0.89	18
Sedang	0	0	0	0
Sulit	0	0	0	2
<i>Accuracy</i>			0.80	20
<i>Macro avg</i>	0.30	0.30	0.30	20
<i>Weighted avg</i>	0.80	0.80	0.80	20

Akurasi keseluruhan SLP Level 1 yang hanya 30% pada data uji mengindikasikan bahwa model perlu perbaikan signifikan dalam kemampuan

generalisasinya. Model menunjukkan bias yang kuat dalam memprediksi kelas 'Mudah', seperti yang terlihat dari nilai precision 0.38 untuk 'Mudah' dan recall 0.60, sementara recall untuk kelas 'Sedang' dan 'Sulit' adalah 0.00. Hal ini berarti model sering kali mengklasifikasikan soal dari kategori 'Sedang' dan 'Sulit' sebagai 'Mudah'. Hasil ini mengategorikan performa Level 1 sebagai “Perlu Perbaikan”.

4.3.2 Evaluasi SLP Level 2

Data kuis untuk pengujian SLP Level 2 dibuat berdasarkan agregasi hasil prediksi SLP Level 1 dan performa asli dari 4 soal dalam setiap kuis. Terdapat 5 data kuis yang digunakan untuk pengujian yang dijelaskan pada Gambar 4.11 dan Tabel 4.6.



Gambar 4.11 *Confusion Matrix* SLP Level 2 dan Distribusi Prediksi Level 2

Pada Gambar 4.11 (kiri) menunjukkan bahwa dari 5 kuis 'Mudah', model berhasil memprediksi keduanya dengan benar. Namun, untuk kuis 'Sedang' dan kuis 'Sulit', model gagal memprediksi dengan benar, malah mengklasifikasikannya sebagai 'Mudah'. Ini konsisten dengan bias prediksi 'Mudah' yang terlihat pada SLP

Level 1. Distribusi prediksi SLP Level 2 pada Gambar 4.11 (kanan) juga secara dominan menunjukkan prediksi untuk kelas 'Mudah'.

Tabel 4.6 *Classification Report* SLP Level 2

	<i>Precision</i>	<i>Recall</i>	<i>F1 score</i>	<i>Support</i>
Mudah	0.60	1	0.75	3
Sedang	0	0	0	2
Sulit	0	0	0	1
<i>Accuracy</i>			0.60	5
<i>Macro avg</i>	0.30	0.50	0.38	5
<i>Weighted avg</i>	0.36	0.60	0.45	5

Akurasi keseluruhan SLP Level 2 pada data uji mencapai 60%, yang secara angka tampak cukup baik. Namun, hal ini perlu dianalisis lebih dalam karena *precision*, *recall*, dan *f1-score* untuk kelas “Sedang” dan “Sulit” sama-sama bernilai 0.00. Ini menunjukkan bahwa model sama sekali gagal mengenali kuis dengan tingkat kesulitan menengah dan tinggi. Sebaliknya, hanya kelas 'Mudah' yang berhasil diprediksi dengan baik, ditandai dengan *precision* 0.60 dan *recall* sempurna 1.00. Kondisi ini menyebabkan *macro average precision* hanya mencapai 0.30 dan *f1-score* sebesar 0.38, mencerminkan ketimpangan performa antar kelas. Dengan demikian, model SLP Level 2 menunjukkan kecenderungan kuat untuk memprediksi semua kuis sebagai “Mudah”, dan belum mampu melakukan klasifikasi yang seimbang terhadap seluruh kelas kesulitan. Performa seperti ini menegaskan perlunya pendekatan model yang lebih sederhana dan stabil untuk menghasilkan prediksi yang lebih representatif dan akurat.

4.4 Hasil Evaluasi Model Sederhana

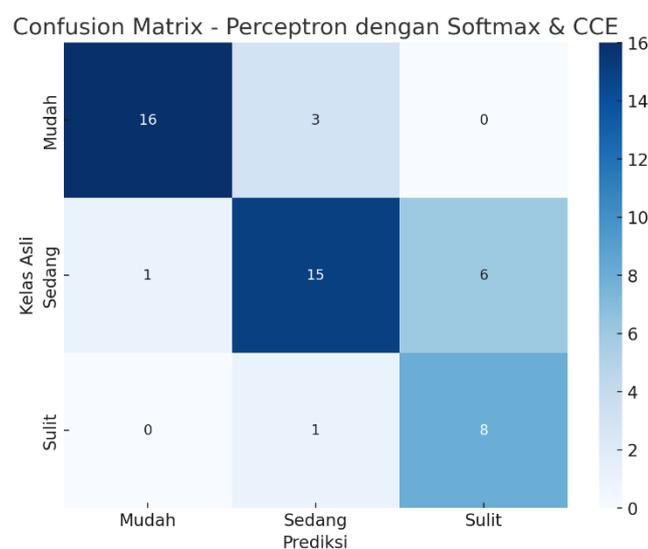
Penelitian ini semula menggunakan pendekatan hierarkis dua level dalam penerapan metode SLP, yakni untuk prediksi tingkat kesulitan per soal dan prediksi

tingkat kesulitan per kuis berdasarkan agregasi. Namun hasil evaluasi menunjukkan bahwa model tersebut kurang optimal dengan data uji SLP Level 1 adalah 80% dan SLP Level 2 turun menjadi 60%, serta bias kuat terhadap prediksi kelas “Mudah”.

Berdasarkan hasil tersebut, peneliti melakukan penyederhanaan pendekatan dengan menggunakan satu model klasifikasi langsung dengan *input* yang lebih ringkas dan objektif dari performa pemain, yaitu *score* (jumlah benar dari 4 soal), waktu rata-rata pengerjaan soal dan jumlah salah. Model evaluasi sederhana ini tetap menggunakan fungsi aktivasi *Softmax* dan *loss function Categorical Cross Entropy* (CCE). Data performa dari 44 responden yang mengerjakan 20 kuis akan dibagi menjadi 80% data pelatihan dan 20% data pengujian, dengan total sesi sebanyak 220 sesi kuis.

4.4.1 Hasil *Training* dan Evaluasi Model Sederhana

Berikut merupakan hasil evaluasi *data testing* menggunakan konfigurasi yang lebih sederhana yang tertera pada Gambar 4.12 dan Tabel 4.7.



Gambar 4.12 *Confusion Matrix* Model Sederhana

Gambar 4.12 menunjukkan bahwa model SLP sederhana berhasil memprediksi dari 19 kelas “Mudah”, 16 prediksi benar dan 3 salah prediksi menjadi “Sedang” dan tidak terdapat salah prediksi ke kelas “Sulit”. Dalam kelas “Sedang”, terdapat 22 kelas “Sedang”, 15 prediksi benar dan 1 salah prediksi menjadi “Mudah” dan 6 salah prediksi menjadi kelas “Sulit”. Terakhir, dalam kelas sulit, SLP berhasil memprediksi kuis “Sulit” sebanyak 8 kuis, dan salah prediksi menjadi “Sedang” sebanyak 1 kuis. Dari gambar tersebut dapat disimpulkan bahwa model sederhana cukup baik, seimbang, dan memiliki akurasi yang relative bagus. Model sering kali melakukan salah prediksi diantara kelas Sedang dan Sulit.

Tabel 4.7 *Classification Report Training* Evaluasi Model Sederhana

	<i>Precision</i>	<i>Recall</i>	<i>F1 score</i>	<i>Support</i>
Mudah	0.94	0.84	0.89	19
Sedang	0.79	0.68	0.73	22
Sulit	0.57	0.89	0.70	9
<i>Macro avg</i>	0.77	0.80	0.77	-
<i>Weighted avg</i>	0.81	0.78	0.78	50

Dari tabel tersebut dapat disimpulkan bahwa model sangat baik dalam mengenali soal dengan tingkat kesulitan “Mudah” dikarenakan mempunyai presisi “Mudah” sangat tinggi, yaitu 0.94. Sedangkan tingkat kesulitan “Sedang” dan “Sulit” terjadi kesalahan prediksi dan mempunyai nilai presisi yang cukup rendah, namun mempunyai nilai *recall* tinggi, yang artinya model sering mengenali kuis dengan tingkat kesulitan “Sulit” namun juga salah menebak sebagai kelas lain.

4.5 Hasil *Usability Testing*

Usability testing digunakan untuk mengevaluasi aplikasi secara keseluruhan oleh pengguna dalam seberapa mudah aplikasi *game save the energy* ini. Setelah

tahap uji coba, pemain diminta untuk memberikan pendapat terkait *game* ini berdasarkan *usability testing*. Pemain mengerjakan kuesioner tentang performa permainan *Save the Energy*. Pertanyaan tersebut berisi tentang kemudahan bermain, efisiensi dalam bermain, terjadi error saat bermain, kepuasan pemain dan daya ingat pemain dalam mengingat fitur-fitur aplikasi. Tabel 4.2 merupakan hasil *survey* kuesioner *usability testing*

Tabel 4.8 Hasil *Survey* Kuesioner *Usability Testing*.

Pemain	No. Butir Tanggapan									
	<i>Learnability</i>		<i>Efficiency</i>		<i>Memorability</i>		<i>Errors</i>		<i>Satisfaction</i>	
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
P1	5	2	5	1	5	2	1	2	4	1
P2	4	2	4	4	2	2	2	1	4	2
P3	4	3	4	3	3	3	2	4	3	3
P4	4	2	5	3	4	2	1	4	3	3
P5	3	1	5	3	4	2	1	5	3	1
P6	4	4	2	3	2	4	4	2	3	4
P7	4	3	3	2	5	1	3	3	3	3
P8	3	3	3	2	4	2	3	3	4	3
P9	4	3	4	2	5	2	3	3	5	2
P10	4	2	3	5	2	1	4	2	3	4
P11	5	4	3	4	3	3	2	4	4	2
P12	3	4	2	3	2	4	2	3	4	3
P13	4	3	5	4	3	1	3	4	5	4
P14	4	2	3	4	4	2	4	4	3	3
P15	3	3	4	4	4	3	3	5	2	4

Tabel 4.9 Hasil Perhitungan *Usability Testing*

No.	Aspek/Kategori	Jumlah Skor Rata-rata	Skor Maksimal	Persentase Skor Rata-rata (%)	Kriteria
1	<i>Learnability</i>	5	8	62.5	Cukup
2	<i>Efficiency</i>	4.53	8	56.67	Cukup
3	<i>Memorability</i>	3.53	8	44.17	Buruk
4	<i>Errors</i>	4.73	8	59.17	Cukup
5	<i>Satisfaction</i>	4.67	8	58.33	Cukup
Total Skor Usability Testing		22.47	40	56.17	Cukup
Kriteria Usability Testing		Cukup			

Berdasarkan hasil perhitungan nilai indeks pada Tabel 4.9, menunjukkan bahwa tingkat *usability* secara keseluruhan menunjukkan persentase rata-rata sebesar 56.17%. Oleh karena itu, dapat disimpulkan bahwa sistem tersebut cukup mudah digunakan oleh pengguna, mudah dipahami dan digunakan, aplikasi cukup disukai oleh pengguna, dan bekerja dengan baik. Dengan persentase keseluruhan 56.17%, dapat disimpulkan bahwa sistem telah memenuhi standar *usability* yang cukup baik dan siap untuk digunakan oleh pengguna, meskipun masih ada ruang untuk perbaikan minor pada beberapa aspek.

4.6 Integrasi dengan Islam

4.6.1 *Mu'amalah Ma'a Allah*

Game yang dirancang bertemakan menjaga alam. Allah memberikan energi yang melimpah, namun tidak semua manusia ingat bahwa energi bukanlah milik manusia, melainkan hanya titipan dari Allah yang harus dijaga dan dipelihara dengan benar. Banyak manusia yang mengeksploitasi sumber daya alam yang akhirnya menyebabkan rusaknya alam sekitar dan banyak makhluk hidup yang punah. Maka dari itu manusia harus tetap berusaha untuk menjaga alam dengan baik. Pada QS. *Al-Isra'* ayat 26 dijelaskan.

إِنَّ الْمُبَدِّرِينَ كَانُوا إِخْوَانَ الشَّيَاطِينِ ۗ وَكَانَ الشَّيْطَانُ لِرَبِّهِ كَفُورًا ۚ ٢٧

“*Sesungguhnya orang-orang yang pemboros itu adalah saudara setan dan setan itu sangat ingkar kepada Tuhannya..*” (*Q.S Al-Isra': 26*)

Allah menginginkan umat manusia tidak melakukan kerusakan di bumi karena dapat mengganggu ekosistem untuk kehidupan digenerasi mendatang. Allah

SWT akan murka kepada manusia yang pemboros, termasuk energi yang biasa kita pakai, seperti lampu, kipas angin, dan lain sebagainya. (Rahmad. 2025)

Pada ayat diatas menegaskan bahwa segala sumber daya alam, termasuk energi, adalah ciptaan Allah yang harus dimanfaatkan secara bijak tanpa berlebihan. Dalam konteks game "*Save the Energy*", ayat ini mengajarkan nilai penghematan dan penggunaan energi secara proporsional, menghindari pemborosan yang dapat merusak lingkungan.

4.6.2 *Mu'amalah Ma'a An-Nas*

Merawat alam selain menjaga kelestarian energi, juga dapat menjadi sumber pahala karena membantu orang. Dalam hadis yang diriwayatkan Imam Muslim.

صحيح مسلم ٢٩٠٠: حَدَّثَنَا ابْنُ نُمَيْرٍ حَدَّثَنَا أَبِي حَدَّثَنَا عَبْدُ الْمَلِكِ عَنْ عَطَاءٍ عَنْ جَابِرٍ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ مَا مِنْ مُسْلِمٍ يَغْرِسُ غَرْسًا إِلَّا كَانَ مَا أَكَلَ مِنْهُ لَهُ صَدَقَةٌ وَمَا سُرِقَ مِنْهُ لَهُ صَدَقَةٌ وَلَا يَرْزُوهَ أَحَدٌ إِلَّا كَانَ لَهُ صَدَقَةٌ إِلَى يَوْمِ الْقِيَامَةِ

*“Shohih Muslim 2900: Telah menceritakan kepada kami (Ibnu Numair), telah menceritakan kepada kami (ayahku) telah menceritakan kepada kami (Abdul Malik) dari (‘Atha’). Dari sahabat Jabir ra, ia berkata, Rasulullah saw. bersabda, ‘Tiada seorang muslim yang menanam pohon kecuali apa yang dimakan bernilai sedekah, apa yang dicuri juga bernilai sedekah. Tiada pula seseorang yang mengurangi buah (dari pohon-)nya melainkan akan bernilai sedekah bagi penanamnya sampai hari Kiamat, ’” (Imam Zakiyuddin Abdul Azhim Al-Mundziri, *At-Tarhib wat Tarhib minal Haditsisy Syarif*, [Beirut, Darul Fikr: 1998 M/1418 H], Juz III, halaman 304).*

Hadis ini mengajarkan bahwa usaha menjaga dan merawat alam, termasuk sumber energi seperti pohon dapat membantu perkembangan manusia dan makhluk hidup lain, maka dari itu hal tersebut termasuk perbuatan mulia dan mendapat pahala jariyah (Aroyandini, dkk., 2021). Dalam game edukasi, ini bisa menjadi

motivasi agar pemain merasa memiliki peran positif dalam menjaga lingkungan yang didalamnya terdapat sumber energi.

4.6.3 *Mu'amalah Ma'a Alam*

Dampak dari menjaga lingkungan selain hanya kepada manusia, pastinya berpengaruh terhadap lingkungan kita, seperti sungai, pantai, laut hingga hutan. Permainan ini mengajak pemain lebih memahami tugas sebagai manusia, makhluk hidup berakal, untuk lebih menjaga alam dan lingkungan. Seperti pada QS. *Ar-Rahman* ayat 7 sampai 9, Allah berfirman.

وَالسَّمَاءَ رَفَعَهَا وَوَضَعَ الْمِيزَانَ ۗ أَلَّا تَطْغَوْا فِي الْمِيزَانِ ۚ ۸ وَأَقِيمُوا الْوَزْنَ بِالْقِسْطِ وَلَا تُخْسِرُوا الْمِيزَانَ ۙ ۹

“Dan langit telah ditinggikan-Nya dan Dia ciptakan keseimbangan, agar kamu jangan merusak keseimbangan itu. Dan tegakkanlah keseimbangan itu dengan adil dan janganlah kamu mengurangi keseimbangan itu.” (Q.S Ar-Rahman: 7-9)

Pada ayat ini menjelaskan bahwa alam semesta diciptakan dengan seimbang ukurannya, tidak hanya dalam arti fisik melainkan keadilan, kesetaraan, dan keteraturan yang mempunyai sifat global (Syauqiah, dkk. 2025). Ketika manusia tidak ikut andil dalam menjaga lingkungan, maka akan menciptakan ketidakseimbangan dalam bumi yang dapat menciptakan bencana alam. Hal tersebut akan sangat berdampak kepada kehidupan manusia. Dalam permainan *“Save the Energy”* terdapat poin memberi keterangan bahwa bagaimana alam harus bekerja dengan seimbang, jika manusia mulai merusak keseimbangan alam maka rusaklah alam tersebut.

BAB V

KESIMPULAN DAN SARAN

Bab ini menyajikan kesimpulan dari penelitian yang telah dilakukan mengenai penerapan algoritma *Single-layer Perceptron* (SLP) dalam menentukan tingkat kesulitan pada permainan edukasi *Save the Energy*. Selain itu, bab ini juga memuat saran untuk pengembangan lebih lanjut yang dapat dilakukan di masa mendatang.

5.1 Kesimpulan

Penelitian ini bertujuan untuk membangun dan mengevaluasi algoritma *single-layer perceptron* dalam menilai tingkat kesulitan pada permainan edukasi. Dari penjelasan sebelumnya, terdapat beberapa hal yang dapat disimpulkan.

1. Hasil evaluasi menunjukkan bahwa SLP Level 1 mencapai akurasi 80% pada data latih dan data uji, namun cenderung bias terhadap kelas “Mudah”, dan sistem kesulitan mendeteksi atau mengidentifikasi kelas “Sedang” dan “Sulit”. Pada SLP Level 2 mencapai akurasi 80% untuk data testing, 60% untuk data uji, dan sistem juga gagal mengidentifikasi kelas “Sedang” dan “Mudah”, tetap terjadi bias pada “Mudah”.
2. Sebagai pembanding, model SLP Sederhana yang menggunakan tiga fitur utama, yaitu jumlah benar, waktu pekerjaan, dan jumlah salah. Model pembanding menunjukkan hasil yang lebih stabil dan seimbang dengan akurasi uji sebesar 78%, serta mempunyai nilai *F1-score* rata-rata sebesar 0.77. Model ini terbukti lebih ringan dan mudah diterapkan.

Secara keseluruhan, penelitian ini menunjukkan bahwa algoritma *Single-layer Perceptron* dapat digunakan untuk membangun sistem adaptasi tingkat kesulitan yang mampu menyesuaikan tantangan permainan berdasarkan kemampuan kognitif pemain. Hal ini berkontribusi pada peningkatan pengalaman bermain dan efektivitas pembelajaran dan juga dalam menjadi salah satu cara sosialisasi hemat energi dalam konteks permainan edukatif. Meskipun metode hierarki menunjukkan keterbatasan dalam generalisasi, pendekatan sederhana memberikan hasil yang menjanjikan untuk diterapkan secara praktis.

Perbedaan model SLP Hierarki dan SLP Sederhana terdapat pada Tabel 5.1

Tabel 5.1 Perbedaan Model SLP Sederhana dan SLP Hierarki

Model	Arsitektur	Akurasi Testing	Catatan
SLP 2 Level	Per soal + agregasi	60%	Kompleks, hasil bias dominan ke kelas "Mudah"
SLP Sederhana	Diproses langsung per kuis	78%	Lebih stabil, ringan, dan mudah diterapkan

Secara keseluruhan, hasil evaluasi menunjukkan bahwa penerapan algoritma *Single-layer Perceptron* mampu membangun sistem penyesuaian tingkat kesulitan yang adaptif, memungkinkan permainan *Save the Energy* untuk menyajikan tantangan yang sesuai dengan kemampuan kognitif pemain, sehingga meningkatkan pengalaman bermain yang personal dan efektif dalam edukasi.

5.2 Saran

Berdasarkan hasil dan keterbatasan penelitian ini, beberapa saran untuk pengembangan lebih lanjut di masa mendatang adalah sebagai berikut:

1. Peningkatan Ukuran dan Kualitas Dataset

- a. Mengumpulkan data performa pemain dalam jumlah yang jauh lebih besar dan beragam akan sangat membantu meningkatkan akurasi dan kemampuan generalisasi model, terutama untuk SLP Level 2 yang saat ini dilatih dengan dataset yang sangat kecil.
 - b. Memastikan data performa per soal memiliki variasi yang cukup untuk setiap tingkat kesulitan (Mudah, Sedang, Sulit).
2. Eksplorasi Arsitektur *Neural Network* yang Lebih Kompleks

Meskipun SLP efektif untuk masalah yang dapat dipisahkan secara linier, pertimbangkan untuk mencoba arsitektur jaringan saraf yang lebih kompleks, seperti *Multi-layer Perceptron* (MLP) dengan lapisan tersembunyi (hidden layers). MLP dapat menangani hubungan non-linier yang mungkin ada dalam data performa pemain dan berpotensi memberikan akurasi yang lebih tinggi.

3. Penyempurnaan Logika Adaptasi *Game*
- a. Meskipun FSM sudah diimplementasikan, logika transisi tingkat kesulitan dapat disempurnakan. Misalnya, mempertimbangkan *confidence score* dari prediksi SLP secara lebih mendalam (tidak hanya sebagai threshold biner), atau menambahkan mekanisme "memori" yang lebih panjang untuk performa pemain (misalnya, rata-rata performa dari beberapa kuis terakhir).
 - b. Eksplorasi strategi pemilihan soal yang lebih cerdas dari *question pool* (misalnya, memastikan keseimbangan topik atau jenis soal)

DAFTAR PUSTAKA

- Adams, Ernest. (2014). *Fundamentals of Game Design*. New Riders.
- Antonova, A., Kos, M., & Zaletelj, J. (2023). Designing and learning in virtual spaces. In *Advances in Game-Based Learning* (pp. 13–25). Springer. https://doi.org/10.1007/978-3-031-21448-6_2
- Aroyandini, E. N., Krismawati, R. A., & Anas, A. R. (2021). Menanam Tumbuhan dalam Perspektif Islam dan Sains Sebagai Upaya Preventif untuk Mengurangi kerusakan Lingkungan. *Prosiding Konferensi Integrasi Interkoneksi Islam dan Sains*, 3, 121–126.
- Fitriani, R., Wati, R., Hanifah, P., & Misriyanti, M. (2019). Kampanye hemat listrik terhadap efisiensi energi pada ibu rumah tangga yang bekerja. *Psikostudia: Jurnal Psikologi*, 7(2), 71. <https://doi.org/10.30872/psikostudia.v7i2.2407>
- Ge, X., & Ifenthaler, D. (2017). Designing Engaging Educational Games and Assessing Engagement in Game-Based Learning. In *Advances in Game-Based Learning*. IGI Global: 253–270. <http://dx.doi.org/10.4018/978-1-5225-0513-6.ch012>
- Hall-Tipping, J. (1991). Video Game Difficulty Level Adjuster Dependent Upon Player's Aerobic Activity Level During Exercise. *Google Patents*.
- Hubbell, G. S., & Kapell, M. W. (2020). *What Is a Game?*. New York: Bloomsbury Academic.
- Karimboyevich, S. E., & Nematullayevich, A. O. (2022). Single Layer Artificial Neural Network: Perceptron. *European Multidisciplinary Journal of Modern Science*, 5, 230–238.
- Kelleher, J. D. (2019). *Deep Learning*. MIT Press.
- Loy, J. (2019). *Neural Network Projects with Python*. Packt Publishing.
- Lucas, S. M., et al. (2008). The Use of Artificial Intelligence in Games: Game AI and AI in Games. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(1), 1–9. DOI: 10.1109/IWCMC.2020.9148327.
- Marklund, B. B., & Taylor, A.-S. A. (2016). Educational Games in Practice: The Challenges Involved in Conducting a Game-Based Curriculum. *14(2)*, 121–135.
- Molnar, A., & Kostkova, P. (2018). Learning about hygiene and antibiotic resistance through mobile games. *Proceedings of the 2018 International Conference on Digital Health*, 95–99. <http://dx.doi.org/10.1145/3194658.3194682>
- Muhajarah, K., & Rachmawati, F. (2019). Game Edukasi Berbasis Android: Urgensi Penggunaan, Pengembangan dan Penguji Kelayakan. *2*, 29–36. <https://doi.org/10.31764/justek.vXiY.3733>

- Muhammad, A. (2022). Urgensi pelestarian lingkungan hidup dalam Al-Qur'an. *Jurnal Pilar: Jurnal Kajian Islam Kontemporer*, 13(1), 67–78.
- Muhtarisah, N., & Anam, W. (2024). Fenomena hedonisme: Kajian tafsir dan hermeneutika Paul Ricoeur terhadap QS. Al-Isra' ayat 27 bagi gaya hidup mahasiswa masa kini. *An-Natiq Jurnal Kajian Islam Interdisipliner*, 4(2), 194–208. <https://dx.doi.org/an-natiq.v4i2.22103>
- Pamungkas, S. M. (2022). Personalisasi Penyeimbang Kesulitan dan Skenario Berbasis Profil Pemain pada Game Edukasi Bencana Gunung Meletus Menggunakan Metode Feedforward Neural Network. Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Rahmad. (2025). Fiqih Muamalah dan Fiqih Lingkungan dalam Menjawab Tantangan Ekonomi Digital dan Pelestarian Alam. *Jurnal Tahqiq*, 19(1), 67–75.
- Reeves, T. C. (1998). The Impact of Media and Technology in Schools. *Journal of The Journal of Art and Design Education*, 2, 58–63.
- Rizky, & Pudrianisa, S. L. G. (2019). Pengujian Usability Pada Tangible Game Sebagai Media Promosi Candi. *Journal INFOS*, 2(1).
- Roberts, P. (2021). *Artificial Intelligence in Games*. London: CRC Press.
- Robertson, D. G. R. (2017). An evaluation of fast multi-layer perceptron training techniques for games. *Computers in Entertainment*, 15(2), 1–18. <https://doi.org/10.1145/3136275.3136277>
- Schreiber, I., & Romero, B. (2022). *Game Balance* (First edition). CRC Press.
- Sobociński, M. D. (2019). Quality of Video Games: Introduction to a Complex Issue. In *Quality Production Improvement - QPI* (pp. 487–494). Sciendo. <http://dx.doi.org/10.2478/9783110680591-066>
- Suliswanto, T. (2014). Implementasi Algoritma Neural Network dalam AI Bot Game FPS. Skripsi. Jurusan Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia.
- Syauqiah, Z., Alfalah, A. H. S., & Nasrulloh. (2025). Keseimbangan Alam dalam Perspektif Al-Qur'an: Tafsir Tematik Tentang Lingkungan dan Implikasinya dalam Kehidupan Modern. *Jurnal Media Akademik (JMA)*, 3(6), Hal. xx–xx. <https://doi.org/10.62281>
- Tsalitsy, R., & Priyanta, S. (2013). Implementasi Machine Learning Untuk Menentukan Tingkat Kesulitan dalam Game yang Berjenis Turn Based Battle Studi Kasus Terhadap Game Monster Arena. Skripsi. Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada.

Wibisono, A., Wijayanto, & Menarianti, I. (2017). Pengembangan Game Edukasi Teka-teki Silang Ragam Istilah Teknologi Informasi Berbasis Android.