

**IDENTIFIKASI ASPEK UTAMA DALAM ULASAN *GAME* MENGGUNAKAN
*NEURAL NETWORK***

SKRIPSI

Oleh :
MUH MIFTAHUL KHAIR
NIM. 210605110024



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

**IDENTIFIKASI ASPEK UTAMA DALAM ULASAN *GAME*
MENGUNAKAN *NEURAL NETWORK***

SKRIPSI

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

Oleh :
MUH MIFTAHUL KHAIR
NIM. 210605110024

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

HALAMAN PERSETUJUAN

**IDENTIFIKASI ASPEK UTAMA DALAM ULASAN *GAME*
MENGUNAKAN *NEURAL NETWORK***

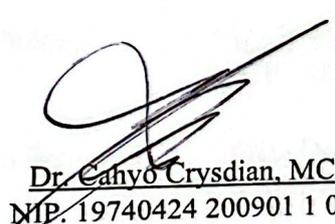
SKRIPSI

Oleh :
MUH MIFTAHUL KHAIR
NIM. 210605110024

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 19 Juni 2025

Pembimbing I,

Pembimbing II,


Dr. Cahyo Crysdiyan, MCS
NIP. 19740424 200901 1 008


Tri Mukti Lestari, M.Kom
NIP. 199111082020122005

Mengetahui,

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



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

HALAMAN PENGESAHAN

**IDENTIFIKASI ASPEK UTAMA DALAM ULASAN GAME
MENGUNAKAN NEURAL NETWORK**

SKRIPSI

Oleh :

MUH MIFTAHUL KHAIR
NIM. 210605110024

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 19 Juni 2025

Susunan Dewan Penguji

Ketua Penguji : Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

Anggota Penguji I : Okta Oमारuddin Aziz, M.Kom
NIP. 19911019 201903 1 013

Anggota Penguji II : Dr. Cahyo Crysdian, MCS
NIP. 19740424 200901 1 008

Anggota Penguji III : Tri Mukti Lestari, M.Kom
NIP. 199111082020122005

()
()
()
()



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


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

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Miftahul Khair
NIM : 210605110024
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Identifikasi Aspek Utama Dalam Ulasan *Game*
Menggunakan *Neural network*.

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, 19 Juni 2025
Yang membuat pernyataan,



Muhammad Miftahul Khair
NIM. 210605110024

MOTTO

... If You read this, Don't Mind me...

HALAMAN PERSEMBAHAN

Alhamdulillah rabbil 'alamin puji syukur kepada Allah SWT karena berkat rahmat dan petunjuk-Nya penulis dapat menyelesaikan skripsi ini. Shalawat serta salam selalu tercurahkan kepada Rasulullah SAW yang telah membawa kita dari zaman jahiliyah menuju addinul Islam.

Penulis mempersembahkan karya ini kepada diri penulis sendiri, orang tua, para dosen, kerabat, teman, sahabat, serta orang-orang yang telah membantu, mendoakan, serta menyemangati penulis dalam menuntaskan skripsi ini.

KATA PENGANTAR

Assalamualaikum wr wb.

Puji syukur saya panjatkan kepada Allah Swt, yang telah memberikan rahmat dan karunia-Nya sehingga saya dapat menyelesaikan penyusunan skripsi ini yang berjudul "Identifikasi Aspek Utama dalam Ulasan *Game* Menggunakan *Neural network*" dengan lancar dan tepat waktu.

Dalam proses penulisan skripsi ini, saya menyadari bahwa banyak pihak yang telah memberikan bantuan, bimbingan, serta dukungan, baik secara langsung maupun tidak langsung. Oleh karena itu, dengan segala kerendahan hati, saya ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. H.M. Zainuddin, MA., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan, M.MT., IPU, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Cahyo Crysdian, MCS selaku Dosen Pembimbing I yang telah memberikan banyak dukungan dan bimbingan dalam penulisan skripsi ini.
5. Tri Mukti Lestari, M.Kom selaku Dosen Pembimbing I yang telah memberikan banyak dukungan dan bimbingan dalam penulisan skripsi ini.
6. Dr. Irwan Budi Santoso, M.Kom selaku Ketua Penguji yang telah memberikan banyak saran untuk menyelesaikan skripsi ini.

7. Okta Qomarudding Aziz, M.Kom selaku dosen penguji I yang telah menguji serta memberikan masukan dalam penulisan skripsi ini.
8. Seluruh Dosen dan Jajaran Staf Program Studi Teknik Informatika yang telah memberikan banyak bantuan dalam skripsi ini.
9. Kedua orang tua, Bapak Suharju S.Kom dan Ibu Nurwahidah S.P serta saudara-saudara yang telah memberikan dukungan kepada penulis untuk menyelesaikan skripsi ini.
10. Teman-teman Angkatan 2021 Teknik Informatika “ASTER” yang telah memberikan banyak bantuan baik material maupun dukungan intelektual, semangat, serta motivasi dalam menyelesaikan penyusunan skripsi ini.
11. Semua pihak yang tidak dapat disebutkan satu per satu, termasuk responden yang telah memberikan kontribusi, saran, dan dukungan dalam perjalanan penulisan skripsi ini.

Akhir kata, penulis mengakui bahwa penulisan pada skripsi ini masih banyak kekurangan.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Malang, 19 Juni 2025

Penulis

DAFTAR ISI

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	xi
DAFTAR TABEL	xii
ABSTRAK	xiii
ABSTRACT	xiv
المخلص	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	5
1.5 Manfaat Penelitian	5
BAB II STUDI PUSTAKA	6
2.1 Klasifikasi Text	6
2.2 Metode Evaluasi	9
BAB III TAHAPAN PENELITIAN & DESAIN SISTEM	12
3.1 Tahapan Penelitian	12
3.2 Desain Sistem	14
3.3 <i>Preprocessing</i>	15
3.4 <i>Fast-Text</i>	17
3.5 <i>Neural Network</i>	21
3.5.1 <i>Proses Training</i>	24
BAB IV UJI COBA DAN PEMBAHASAN	30
4.1 Data Pengujian	30
4.2 Skenario Pengujian	33
4.3 Hasil Uji Coba	42
4.3.1 Hasil Uji Coba Skenario 1	42
4.3.2 Hasil Uji Coba Skenario 2	45
4.3.3 Hasil Uji Coba Skenario 3	48
4.3.4 Hasil Uji Coba Skenario 4	52
4.3.5 Hasil Uji Coba Skenario 5	56
4.4 Pembahasan	60
BAB V KESIMPULAN DAN SARAN	70
5.1 Kesimpulan	70
5.2 Saran	70
DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar 3.1 Alur Penelitian	13
Gambar 3.2 Alur Sistem	14
Gambar 3.3 Alur Preprocessing	15
Gambar 3.4 Arsitektur <i>Neural Network</i>	22
Gambar 4.1 Grafik <i>Loss</i> Skenario 1	42
Gambar 4.2 Grafik <i>Loss</i> Skenario 2.....	46
Gambar 4.3 Grafik <i>Loss</i> Skenario 3.....	50
Gambar 4.4 Grafik <i>Loss</i> Skenario 4.....	54
Gambar 4.5 Grafik <i>Loss</i> Skenario 5.....	58
Gambar 4.6 Grafik Perubahan Nilai Metrik Evaluasi.....	64
Gambar 4.7 Grafik <i>Loss K-Fold</i>	65

DAFTAR TABEL

Tabel 3.1 Contoh <i>Tokenizing</i>	15
Tabel 3.2 Contoh <i>Case Folding</i>	16
Tabel 3.3 Contoh <i>Punctual Removal</i>	16
Tabel 3.4 Contoh <i>Stemming</i>	17
Tabel 3.5 Contoh Kata Menjadi N-Gram Karakter.....	20
Tabel 3.6 Contoh <i>Subword Vector Embedding</i>	21
Tabel 4.1 Jumlah Ulasan Aspek Utama <i>Game</i>	30
Tabel 4.2 Skenario Pengujian	36
Tabel 4.3 <i>Confusion Matrix</i>	39
Tabel 4.4 <i>Confusion Matrix</i> Kelas <i>Gameplay</i>	40
Tabel 4.5 <i>Confusion Matrix</i> Kelas <i>Graphics</i>	40
Tabel 4.6 <i>Confusion Matrix</i> Kelas <i>Story</i>	40
Tabel 4.7 Jumlah Data Tiap Kelas Skenario 1	42
Tabel 4.8 <i>Confusion Matrix</i> Skenario 1	43
Tabel 4.9 Hasil Uji Coba Skenario 1	44
Tabel 4.10 Jumlah Data Tiap Kelas Skenario 2	45
Tabel 4.11 <i>Confusion Matrix</i> Skenario 2	47
Tabel 4.12 Hasil Uji Coba Skenario 2	48
Tabel 4.13 Jumlah Data Tiap Kelas Skenario 3	49
Tabel 4.14 <i>Confusion Matrix</i> Skenario 3	51
Tabel 4.15 Hasil Uji Coba Skenario 3	52
Tabel 4.16 Jumlah Data Tiap Kelas Skenario 4	53
Tabel 4.17 <i>Confusion Matrix</i> Skenario 4	55
Tabel 4.18 Hasil Uji Coba Skenario 4	56
Tabel 4.19 Jumlah Data Tiap Kelas Skenario 5	57
Tabel 4.20 <i>Confusion Matrix</i> Skenario 5	59
Tabel 4.21 Hasil Uji Coba Skenario 5	60
Tabel 4.22 Rata-Rata Hasil Uji Coba Skenario Pengujian	61
Tabel 4.23 <i>Confusion Matrix Fold</i> 1	66
Tabel 4.24 <i>Confusion Matrix Fold</i> 2.....	66
Tabel 4.25 <i>Confusion Matrix Fold</i> 3.....	66
Tabel 4.26 <i>Confusion Matrix Fold</i> 4.....	66
Tabel 4.27 <i>Confusion Matrix Fold</i> 5.....	66
Tabel 4.28 Rata-Rata Evaluasi Model Tiap <i>Fold</i>	68

ABSTRAK

Khair, Muh Miftahul. 2025. **Identifikasi Aspek Utama Dalam Ulasan *Game* Menggunakan *Neural network***. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Cahyo Crysdiyan, MCS (II) Tri Mukti Lestari, M. Kom.

Kata kunci: Identifikasi, *Neural network*, Ulasan *Game*.

Peran industri *game* sangat besar dalam ekonomi saat ini dengan memberikan kontribusi penting terhadap pertumbuhan ekonomi. Menurut *Fortune Business Insights*, pada tahun 2023 nilai pasar *game* global melampaui USD \$200 miliar, menunjukkan pertumbuhan yang sangat kuat melebihi banyak sektor hiburan tradisional. Ulasan atau *review* pengguna diperlukan untuk pertumbuhan industri *game* yang berkelanjutan. Namun, meskipun ulasan memberikan banyak wawasan, sifatnya yang tidak terstruktur dan subjektif sering kali menjadi tantangan bagi pengembang. Tujuan penelitian ini untuk mengidentifikasi aspek-aspek utama pada ulasan *game* yang terbagi menjadi 3 kategori utama, yaitu *gameplay*, *Graphics*, dan *Story* dari ulasan pengguna secara otomatis menggunakan *neural network*. Diantara 5 model yang dilatih berdasarkan skenario pengujian, Model 4 dengan rasio data *training* dan *testing* 60:40, *learning rate* 0,01, dan toleransi error $1e-10$ memberikan performa terbaik dengan presentase akurasi 90% dan hasil rata-rata presentase *precision* 89%, *recall* 91%, *F1-score* 90%.

ABSTRACT

Khair, Muh Miftahul. 2025. **Identification Of Key Aspects in *Game* Reviews Using *Neural networks***. Undergraduate Thesis. Department of Informatics Engineering Faculty of Science and Technology Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Dr. Cahyo Crysdiyan, MCS (II) Tri Mukti Lestari, M. Kom.

Key words: Identification, *Neural network*, *Game* Review.

The role of the gaming industry is immense in today's economy by making important contributions to economic growth. According to Fortune Business Insights In 2023, the value of the global gaming market surpassed USD \$200 billion, showing very strong growth exceeding many traditional entertainment sectors. User reviews are necessary for the continued growth of the gaming industry. However, while reviews provide many insights, their unstructured and subjective nature often poses a challenge for developers. The purpose of this research is to identify the main aspects of the *game* reviews which are divided into 3 main categories namely *gameplay*, *Graphics*, and *Story* from user reviews automatically using *neural networks*. Among the 5 models trained based on the test scenario, Model 4 with a *training* and *testing* data ratio of 60:40, a *learning rate* of 0,01, and an error tolerance of 1e-10 provides the best performance with an *Accuracy* percentage of 90% and an average percentage of *precision* 89%, *recall* 91%, *F1-score* 90%.

الملخص

خير، محمد مفتاحول. 2025. تحديد الجوانب الرئيسية في مراجعات الألعاب باستخدام الشبكات العصبية. رسالة جامعية. قسم هندسة المعلوماتية كلية العلوم والتكنولوجيا جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: الأول (د)

كاهيو كريسدیان, M. CS (الثاني) تري موكتي ليستاري, M. Kom.

الكلمات المفتاحية: تحديد، الشبكات العصبية، مراجعات الألعاب

إن دور صناعة الألعاب هائل في اقتصاد اليوم من خلال تقديم مساهمات مهمة في النمو الاقتصادي. وفقًا لـ فورتشن بيزنس إنسايتسفي عام 2023، تجاوزت قيمة سوق الألعاب العالمية 200 مليار دولار أمريكي، مما يدل على نمو قوي للغاية يتجاوز العديد من قطاعات الترفيه التقليدية. مراجعات المستخدمين ضرورية لاستمرار نمو صناعة الألعاب. ومع ذلك، على الرغم من أن المراجعات توفر العديد من الرؤى، إلا أن طبيعتها غير المنظمة والذاتية غالبًا ما تشكل تحديًا للمطورين. الغرض من هذا البحث هو تحديد الجوانب الرئيسية لمراجعات الألعاب التي تنقسم إلى 3 فئات رئيسية وهي طريقة اللعب والرسومات والقصة من مراجعات المستخدمين تلقائيًا باستخدام الشبكات العصبية، ومن بين النماذج الخمسة التي تم تدريبها بناءً على سيناريو الاختبار، يوفر النموذج 4 بنسبة بيانات تدريب واختبار 60:40، ومعدل تعلم 0,01، ومعدل خطأ بنسبة 00001,0 أفضل أداء بنسبة دقة 90% ومتوسط نسبة ضغط 89%، واستدعاء 91%، ودرجة $F1-score$ 90%.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Peran industri *game* sangat besar dalam ekonomi saat ini dengan memberikan kontribusi penting terhadap pertumbuhan ekonomi, penciptaan lapangan kerja, dan inovasi teknologi. Menurut *Fortune Business Insights* Pada tahun 2023, nilai pasar *game* global melampaui USD \$200 miliar menunjukkan pertumbuhan yang sangat kuat melebihi banyak sektor hiburan tradisional. Perkembangan ini tidak hanya menunjukkan peningkatan pengeluaran konsumen tapi juga melibatkan inovasi teknologi dan platform baru yang memperbaiki pengalaman serta interaksi pengguna. Contohnya, perkembangan dalam bidang grafis, kecerdasan buatan, dan realitas virtual telah mengubah permainan menjadi pengalaman yang lebih dalam, menarik berbagai audiens, dan membentuk ekosistem pengembang, pemasar, dan pembuat konten yang dinamis (Cheah et al., 2022).

Terlebih lagi, peran industri *game* telah meningkat dalam mendorong aktivitas ekonomi dengan memengaruhi berbagai sektor seperti pendidikan, layanan kesehatan, dan interaksi sosial (Zeng et al., 2020). Penggunaan gamifikasi dalam bidang pendidikan telah terbukti meningkatkan prestasi belajar dan keterlibatan siswa, terutama di area STEM (Damaševičius et al., 2023). Permainan serius, yang diciptakan bukan untuk hiburan, semakin populer di dunia medis untuk meningkatkan kondisi pasien dan mempermudah proses pemulihan (Amorim et al.,

2020). Implikasi lintas sektoral ini menyoroti kemampuan industri *game* yang fleksibel dan potensinya dalam menangani masalah sosial yang rumit.

Ulasan atau review pengguna diperlukan untuk pertumbuhan industri *game* yang berkelanjutan. Pengembang *game* semakin bergantung pada ulasan pemain untuk melakukan perbaikan pada mekanisme *gameplay*, mengoptimalkan antarmuka pengguna, dan meningkatkan desain *game* secara menyeluruh (Alonso-Fernández et al., 2019). Berulangnya proses pengembangan *game* dengan menggunakan masukan dari pengguna untuk menjaga agar *game* tetap relevan dan menarik (Ciuchita et al., 2023). Contohnya, penggabungan tingkat kesulitan adaptif dan pengalaman bermain *game* yang disesuaikan telah terhubung dengan peningkatan kepuasan dan retensi pemain (Zayeni et al., 2020). Kemunculan platform *streaming* langsung juga memungkinkan komunitas terlibat dalam pengembangan *game* dengan berbagi pengalaman secara *real-time* (Li et al., 2020).

Namun, meskipun ulasan memberikan banyak wawasan, sifatnya yang tidak terstruktur dan subjektif sering kali menjadi tantangan bagi pengembang. Pengembang *game*, yang bergantung pada masukan ini untuk melakukan perbaikan dalam *gameplay* dan desain, sering merasa kewalahan dengan jumlah data yang harus diproses (Faric et al., 2019). Dengan beragamnya pengalaman pemain, sulit bagi pengembang untuk mengidentifikasi tema atau aspek yang perlu diperbaiki secara sistematis keragaman pendapat yang terungkap dalam ulasan dapat sangat berbeda-beda dan dipengaruhi oleh pengalaman serta harapan individu masing-masing pemain, yang semakin mempersulit tantangan tersebut (Tlili et al., 2022).

Agar bisa mengelola situasi ini dengan baik, pengembang permainan dapat memanfaatkan teknik analisis data tingkat lanjut seperti pemrosesan bahasa alami (NLP) dan teknologi jaringan saraf (Barnes & Prescott, 2018). Fungsionalitas alat-alat ini meliputi otomatisasi proses untuk mengekstraksi tema dan sentimen utama dari banyak data ulasan. Contohnya, dengan melatih jaringan saraf untuk mengenali elemen *gameplay*, grafis/visual, dan mekanik khusus, pengembang bisa mendapatkan pemahaman yang lebih baik tentang fitur yang disenangi oleh pemain dan aspek mana yang butuh perbaikan. Cara tersebut tidak hanya mempermudah analisis ulasan tetapi juga memungkinkan pengembang untuk memusatkan perhatian pada faktor terpenting yang mempengaruhi kepuasan dan keterlibatan pemain (Tao et al., 2021).

Neural network telah muncul sebagai alat yang ampuh untuk melakukan klasifikasi di berbagai bidang, terutama karena kemampuannya mempelajari pola kompleks dari kumpulan data besar. Dalam konteks NLP, integrasi teknik representasi kata tingkat lanjut, seperti *word embeddings*, meningkatkan kinerja ANN dalam tugas NLP. *Word Embedding* ini memungkinkan model memahami hubungan semantik antar kata, yang sangat penting untuk tugas seperti analisis sentimen dan klasifikasi topik. Penelitian Wang (Wang et al., 2020) mengusulkan metode klasifikasi teks *multi-label* yang memanfaatkan model representasi semantik dinamis bersama dengan jaringan saraf dalam, menunjukkan fleksibilitas model-model ini dalam menghadapi tantangan klasifikasi yang kompleks. Fleksibilitas ini sangat penting dalam NLP, di mana variasi bahasa dapat menyebabkan model mengalami *overfitting* jika tidak dikelola dengan baik.

Sebagai upaya memahami kebutuhan dan preferensi pemain dalam industri *game* yang dinamis, penelitian ini mengembangkan pendekatan "Identifikasi Aspek Utama dalam Ulasan *Game* Menggunakan *Neural network*". Dengan teknologi ini, diharapkan analisis ulasan *game* menjadi lebih terarah, efisien, dan mendalam, mendukung pengembang dalam menghadirkan pengalaman yang bermakna bagi pengguna. Pendekatan ini juga sejalan dengan prinsip Al-Qur'an tentang pentingnya menyimak dan memahami masukan dari orang lain secara bijak, seperti yang tercantum dalam Surah Az-Zumar ayat 18:

الَّذِينَ يَسْتَمِعُونَ الْقَوْلَ فَيَتَّبِعُونَ أَحْسَنَهُ ۗ أُولَٰئِكَ الَّذِينَ هَدَىٰ اللَّهُ وَأُولَٰئِكَ هُمُ الْأُولِيَاءُ ﴿١٨﴾

"(Yaitu) mereka yang mendengarkan perkataan lalu mengikuti apa yang paling baik di antaranya. Mereka itulah orang-orang yang telah diberi petunjuk oleh Allah dan mereka itulah ululalbab (orang-orang yang mempunyai akal sehat)." (Q.S. Az-Zumar: 18)".

Ayat ini mengingatkan agar kita mendengarkan dan memanfaatkan masukan untuk mengambil tindakan terbaik, selaras dengan tujuan pengembang *game* untuk memperbaiki dan menyesuaikan produk mereka demi kepuasan dan kebermanfaatan pengguna.

1.2 Pernyataan Masalah

Seberapa baik performa model *neural network* dalam mengidentifikasi atau mengklasifikasikan aspek-aspek utama dalam ulasan *game*?

1.3 Tujuan Penelitian

Mengukur performa *neural network* dalam mengidentifikasi aspek-aspek utama *game* dari ulasan pengguna secara otomatis.

1.4 Batasan Masalah

1. Penelitian ini hanya akan memproses **ulasan teks** dari *game*, bukan ulasan dalam bentuk audio atau video.
2. Ulasan yang digunakan hanya diambil dari *platform Steam*.
3. Ulasan yang diproses adalah ulasan dalam bahasa **Inggris** dan tidak mencakup ulasan dalam berbagai bahasa lain.
4. Banyak data ulasan dibatasi berdasarkan ulasan dari *game-game* populer yang terdaftar pada *platform Steam*.

1.5 Manfaat Penelitian

Berikut adalah manfaat dari dilakukannya penelitian ini, yaitu:

1. Penelitian ini dapat membantu para pengembang *game* dalam mengekstraksi informasi penting dari ulasan pengguna sehingga dapat digunakan untuk meningkatkan kualitas *game* yang mereka buat.
2. Penelitian ini bermanfaat bagi komunitas *gamer* (terutama untuk para developer *game*) dengan menciptakan *game* yang lebih sesuai dengan kebutuhan dan preferensi mereka berdasarkan masukan yang telah dianalisis secara mendalam.

BAB II

STUDI PUSTAKA

2.1 Klasifikasi Text

Pada bagian ini, berbagai penelitian terkait yang relevan dengan identifikasi aspek utama dalam ulasan *game* serta penggunaan *neural network* akan diulas.

Penelitian Merinda (2021), mengkaji kinerja RNN dan Naïve Bayes dengan menambahkan teknik TF-IDF (*Term Frequency-Inverse Document Frequency*) yang bertujuan untuk memberikan bobot pada hubungan kata (*term*) sebuah dokumen. Hasil pengujian menunjukkan RNN (TF-IDF) memiliki akurasi lebih besar yaitu 97,77% dibandingkan Naïve Bayes (TF-IDF) sebesar 80%.

Penelitian Fauzianto & Supatman (2023) memanfaatkan kemampuan Natural Language Processing (NLP) untuk menganalisis sentimen Masyarakat terhadap *Tech Winter*. Penelitian ini membandingkan efektifitas berbagai model NLP (Regresi Logistik, SVM, *Random Forest*, *Neural network*, dan Naïve Bayes) dalam menafsirkan sentiment yang diungkapkan dalam *tweet*. Temuan menawarkan pemahaman mengenai opini publik dan mengungkapkan model NLP paling efektif untuk analisis sentimen.

Penelitian Winda Kurnia Sari (2017) mengkaji penggunaan metode *Long Short-Term Memory* (LSTM) dalam klasifikasi teks *multilabel* dengan empat kategori artikel berita. Model LSTM menggunakan fitur *Word2Vec* 300 dimensi untuk meningkatkan akurasi klasifikasi. Setelah melakukan tuning *hyperparameter* dan pengujian terhadap delapan model LSTM, hasil terbaik menunjukkan akurasi

sebesar 95,38%, dengan nilai rata-rata presisi, *recall*, dan *F1-score* sebesar 95%. Model LSTM juga menunjukkan grafik yang mendekati *good-fit* pada model ketujuh dan kedelapan.

Penelitian Ilham (2023) membandingkan kinerja *word embedding Glove* dan *FastText* dalam klasifikasi teks menggunakan algoritma LSTM. Hasil eksperimen menunjukkan bahwa *FastText* lebih unggul dengan akurasi mencapai 90%, sementara perbedaan akurasi antar epoch tidak signifikan. Disimpulkan bahwa *FastText* lebih baik dibandingkan *Glove* dalam klasifikasi teks.

Penelitian Made Sujiwa Putra (2021) memanfaatkan beberapa model machine learning seperti *Artificial Neural network (ANN)*, *Latent Semantic Index (LSI)*, dan *Jaccard* untuk membuat sistem pendeteksi kesamaan atau *similarity checking* yang ada pada *system e-learning* untuk membantu mencegah *plagiarism* antar sesama siswa dalam tugas yang dikerjakan. Penelitian ini menggunakan hasil evaluasi dengan pendekatan *Mean Absolute Percentage Error (MAPE)* dimana metode *Jaccard* sebagai nilai aktualnya. Hasil menunjukkan bahwa metode ANN dalam pendeteksian kesamaan teks memiliki kinerja yang lebih mendekati metode *Jaccard* dibanding metode LSI. Hasil persentase MAPE pada metode ANN yang paling baik adalah 0,07% yang didapat dengan menggunakan fungsi aktivasi ReLU dan dengan model aritektur MLP. Sedangkan nilai MAPE metode LSI adalah 28,0%

Penelitian Made Mika Parwita (2019), bertujuan untuk klasifikasi ulasan user kedalam 3 kategori yaitu *bug*, *feature request*, dan *non-informative review* secara otomatis pada ulasan aplikasi pada *google play* dan *F-droid*. Hasil

menunjukkan bahwa model CNN (*Convolution Neural network*) mampu melakukan klasifikasi ulasan aplikasi dengan tingkat akurasi sebesar 79%, presisi 77%, dan *recall* 87%.

Penelitian Agung Pambudi (2021), melakukan analisis sentimen terkait pengaruh panjang kalimat pada komentar mahasiswa *training* ICT terhadap metode *Support Vector Machine* dan *Convolutional Neural network*. diperoleh kesimpulan bahwa panjang kalimat pada dataset akan mempengaruhi performansi metode SVM dan CNN jika dikombinasikan dengan *Word2vec*. Sedangkan untuk metode SVM+TFIDF, performansi tidak begitu dipengaruhi oleh panjang kalimat, meski demikian kombinasi metode ini memiliki waktu proses yang paling cepat jika dibandingkan metode lainnya. Adapun metode CNN+Word2vec menghasilkan performansi terbaik pada penelitian ini dengan nilai akurasi sebesar 0,94%, presisi sebesar 0,95%, *recall* sebesar 0,96%, dan *f1-score* sebesar 0,95%.

Penelitian Ameliah Riskiyah, menganalisis kepuasan pelanggan terhadap layanan transportasi *online* dengan menganalisis sentiment dari opini pengguna di platform *Twitter*. Untuk ekstraksi fitur menggunakan *Word2vec* dan model klasifikasi yang digunakan ialah ELM (*Extreme Learning Machine*) yang merupakan algoritma yang dirancang untuk *single hidden layer feedforward neural network* (SLFNs). Dengan skema pengujian dataset yang dibagi menjadi 20% untuk data uji dan 80% data latih diperoleh hasil akurasi sebesar 72%.

2.2 Metode Evaluasi

Pada bagian ini, berbagai penelitian terkait yang relevan dengan metode evaluasi *neural network* identifikasi aspek utama dalam ulasan *game* yang akan diulas.

Penelitian Sokolova & Lapalme (2009), mengkaji berbagai metode untuk mengevaluasi model *machine learning* dengan fokus pada penggunaan metrik evaluasi yang sesuai dengan konteks *machine learning* nya. Dalam studi mereka, untuk kasus klasifikasi *multi-class* mereka menggunakan metrik seperti *precision*, *recall*, dan *F1-score* untuk memberikan gambaran tentang kinerja model. *Precision* mengukur proporsi prediksi positif yang benar, *recall* mengevaluasi seberapa banyak *instance* positif yang terdeteksi oleh model, dan *F1-score* mengkombinasikan kedua metrik tersebut untuk memberikan nilai yang menyeimbangkan antara *precision* dan *recall*.

Penelitian Grandini et al (2020) bertujuan untuk menganalisis berbagai pilihan metrik yang digunakan untuk mengevaluasi kinerja algoritma klasifikasi pada dataset *multi-class*. Salah satu metrik yang paling dikenal adalah *Accuracy* yang memberikan ukuran keseluruhan tentang seberapa baik model dapat memprediksi klasifikasi individu dibandingkan dengan keseluruhan dataset. Selain itu, Grandini juga membahas *cross entropy*, yang mengukur kesepakatan antara distribusi probabilitas prediksi dan klasifikasi yang sebenarnya. Meskipun mengabaikan distribusi probabilitas pada kelas lain, metrik ini sering digunakan karena perhitungannya yang cepat.

Penelitian Naseer et al (2024) mengusulkan pendekatan baru dalam *object recognition* yang menggabungkan beberapa faktor penting untuk meningkatkan ketepatan pengenalan objek. Dalam penelitian ini, *Convolutional Neural network* (CNN) digunakan untuk mengidentifikasi objek dalam berbagai kondisi lingkungan. Penelitian ini menggunakan *precision*, *recall*, dan *F1-score* sebagai metrik evaluasi utama dalam memberikan gambaran mengenai kemampuan model dalam mengklasifikasikan objek. Metrik *Precision* menunjukkan bahwa sebagian besar prediksi positif yang dilakukan oleh model adalah benar, artinya model sangat akurat dalam mengidentifikasi objek yang relevan. Di sisi lain, *Recall* menunjukkan seberapa baik model dalam menemukan semua objek yang benar, termasuk yang lebih sulit atau kurang terwakili dalam dataset. *F1-score*, sebagai ukuran rata-rata harmonik dari *precision* dan *recall*, memberikan gambaran lebih seimbang mengenai kinerja model.

Penelitian Hashemi et al. (2023) mengusulkan penggunaan *weight-tuning* sebagai langkah praproses untuk menyeimbangkan pengaruh kelas fraud dan transaksi sah. Metrik evaluasi yang digunakan untuk menilai kinerja model adalah *precision*, *recall*, dan *F1-score*, yang relevan untuk masalah dengan distribusi kelas yang tidak seimbang. *Precision* dalam penelitian ini digunakan untuk mengukur seberapa akurat model dalam mengklasifikasikan transaksi sebagai fraud. Sebagai contoh, model dengan *precision* 0,79 pada *LightGBM* dan *XGBoost* menunjukkan bahwa sebagian besar transaksi yang diprediksi fraud memang benar-benar fraud. Sementara itu, *recall* di sini mengukur sejauh mana model dapat menemukan seluruh transaksi *fraud* yang ada, dengan nilai *recall* 0,80 pada *LightGBM* dan

XGBoost, yang menandakan bahwa model ini mampu menangkap sebagian besar transaksi *fraud* yang sebenarnya ada dalam dataset. *F1-score*, yang merupakan rata-rata harmonik dari *precision* dan *recall*, memberikan gambaran yang lebih seimbang mengenai kinerja model secara keseluruhan, dan nilai *F1-score* 0,79 pada *LightGBM* dan *XGBoost* menegaskan kinerja yang solid dengan keseimbangan yang baik antara keduanya.

BAB III

TAHAPAN PENELITIAN & DESAIN SISTEM

3.1 Tahapan Penelitian

Penelitian ini dimulai dengan pencarian dan pengumpulan referensi penelitian yang relevan. Pada tahap ini, berbagai literatur dan sumber ilmiah yang terkait dengan topik identifikasi aspek utama dalam ulasan *game* menggunakan *neural network* akan dikaji. Penelitian terdahulu yang membahas penerapan *machine learning*, khususnya *neural network* dalam klasifikasi teks dan evaluasi model, menjadi dasar teoritis untuk metodologi yang akan digunakan dalam penelitian ini.

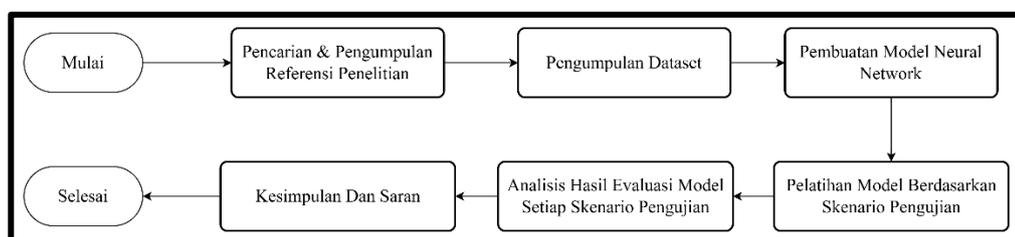
Setelah pengumpulan referensi, langkah selanjutnya adalah pengumpulan dataset yang akan digunakan dalam penelitian. Dataset yang relevan akan diambil dari *platform* seperti *Kaggle*, yang menyediakan berbagai dataset ulasan *game* yang dapat digunakan untuk mengidentifikasi aspek-aspek penting dalam ulasan tersebut. Proses ini juga mencakup pembersihan data dan *preprocessing*, termasuk langkah-langkah seperti tokenisasi dan *stemming* untuk mempersiapkan data agar dapat digunakan dalam model *machine learning*.

Dengan dataset yang telah siap, tahap berikutnya adalah pembuatan model *neural network*. Model *neural network* ini akan dibangun dengan tujuan untuk melakukan klasifikasi aspek utama pada ulasan *game*. Proses ini melibatkan pemilihan arsitektur yang sesuai untuk *neural network* dan konfigurasi parameter yang optimal agar model dapat memberikan performa terbaik.

Setelah model dibuat, langkah berikutnya adalah pelatihan model berdasarkan skenario pengujian. Pada tahap ini, model akan dilatih menggunakan dataset yang telah disiapkan, dengan berbagai skenario pengujian yang telah dirancang sebelumnya. Ini memungkinkan untuk melihat bagaimana model berperforma dalam berbagai kondisi, misalnya, dengan dataset yang lebih besar atau lebih kecil, atau dataset yang tidak seimbang antara kelas positif dan negatif.

Setelah model dilatih, dilakukan analisis hasil evaluasi model pada setiap skenario pengujian. Pada tahap ini, hasil dari pengujian berbagai skenario akan dianalisis untuk melihat seberapa baik model mengenali dan mengklasifikasikan ulasan dalam kondisi yang berbeda. Metrik evaluasi seperti *precision*, *recall*, dan *F1-score* akan digunakan untuk menilai kinerja model, serta untuk menganalisis *trade-off* antara kedua metrik tersebut.

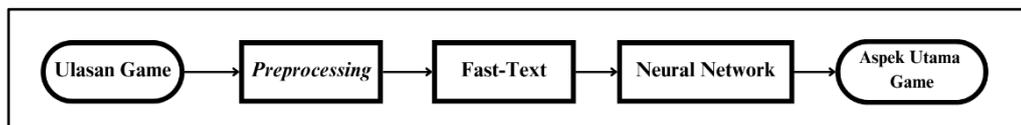
Pada akhirnya, penelitian ini akan diakhiri dengan kesimpulan dan saran yang didapatkan dari analisis hasil pengujian. Kesimpulan ini akan memberikan pemahaman mengenai model *neural network* dalam mengidentifikasi aspek utama dalam ulasan *game* dan saran untuk pengembangan lebih lanjut atau perbaikan model yang lebih optimal. Alur tahapan penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

3.2 Desain Sistem

Sistem dirancang agar mampu melakukan klasifikasi aspek-aspek utama dalam *game* dari data review yang diambil dari platform penyedia *game*. Desain sistem ini meliputi alur kerja yang akan diimplementasikan dalam tahapan-tahapan yang ditunjukkan pada Gambar 3.2.



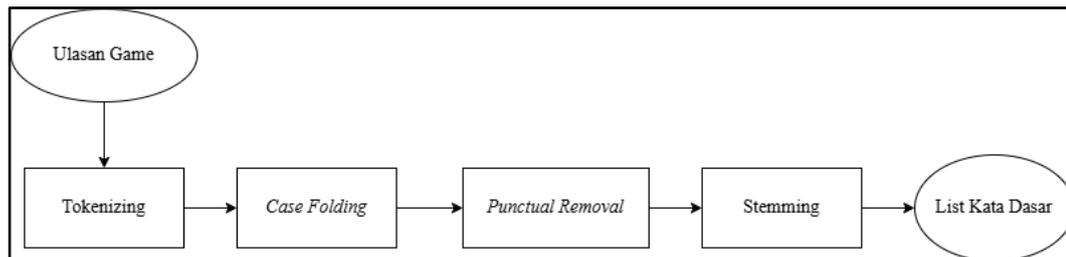
Gambar 3.2 Alur Sistem

Penelitian ini dimulai dengan pengumpulan data berupa ulasan *game* dari *platform* yang relevan. Setelah data terkumpul, tahap berikutnya adalah *preprocessing*, yaitu membersihkan dan menyiapkan teks agar siap untuk diproses oleh model. Pada tahap ini, berbagai teknik pembersihan data, seperti *tokenizing*, *stemming*, *case folding*, *stopword removal*, dilakukan untuk memastikan bahwa data dalam bentuk yang bersih dan konsisten.

Setelah *preprocessing*, dilakukan ekstraksi fitur menggunakan metode *FastText* yang berfungsi mengubah teks menjadi format numerik. Langkah ini penting agar data ulasan dapat diolah lebih lanjut dalam model klasifikasi. Setelah proses ekstraksi, data numerik ini akan digunakan untuk melatih model *neural network*. Model ini akan dilatih untuk mengklasifikasikan aspek-aspek utama dalam ulasan *game*, sehingga dapat mengidentifikasi kategori seperti *gameplay*, grafik, dan performa berdasarkan masukan dari pengguna.

3.3 *Preprocessing*

Tahapan *preprocessing* dalam pengolahan data teks ini sangat penting agar data dapat dianalisis dengan baik dan lebih efektif ketika masuk ke tahapan *training* model. Alur *preprocessing* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Alur Preprocessing

1. *Tokenizing*

Tahap pertama dalam *preprocessing* adalah *tokenizing*, yaitu proses memecah teks ulasan menjadi bagian-bagian yang lebih kecil yang disebut "token." Dalam konteks analisis teks, token ini biasanya berupa kata-kata individu dalam kalimat. Dengan memecah teks menjadi kata-kata terpisah, setiap kata dapat dianalisis secara independen, yang memudahkan pemrosesan lebih lanjut. Tahap *tokenizing* ini sangat penting karena merupakan langkah awal dalam ekstraksi informasi dari teks. Contoh *Tokenizing* ditunjukkan pada Tabel 3.1.

Tabel 3.1 Contoh Tokenizing

Kalimat asli:	"The <i>gameplay</i> is awesome!"
Hasil tokenizing:	["The", " <i>gameplay</i> ", "is", "awesome"]

2. *Case Folding*

Selanjutnya, proses *case folding* dilakukan untuk memastikan konsistensi data. Pada tahap ini, semua huruf dalam teks diubah menjadi huruf kecil (*lowercase*). Langkah ini dilakukan untuk menghindari duplikasi yang disebabkan

oleh perbedaan huruf besar dan kecil. Misalnya, kata "*Game*" dan "*game*" akan dianggap sebagai satu kata yang sama setelah *case folding*. Dengan begitu, proses analisis teks menjadi lebih efisien dan akurat, karena variasi huruf besar dan kecil tidak akan mempengaruhi hasil akhir. Contoh *Case Folding* ditunjukkan pada Tabel 3.2:

Tabel 3.2 Contoh *Case Folding*

Kalimat asli:	"The <i>Gameplay</i> Is Awesome!"
Hasil <i>Case Folding</i>:	"the <i>gameplay</i> is awesome!"

3. *Punctuation Removal*

Tahap berikutnya adalah *punctuation removal*, di mana tanda baca seperti titik, koma, tanda seru, dan tanda tanya dihapus dari teks. Tanda baca umumnya tidak memberikan kontribusi terhadap makna utama dalam konteks analisis teks, sehingga penghapusannya bertujuan untuk membersihkan data. Dengan menghilangkan karakter yang tidak relevan ini, teks menjadi lebih sederhana dan lebih mudah untuk dianalisis, terutama dalam model yang lebih bergantung pada kata-kata inti. Contoh *Punctual Removal* ditunjukkan pada Tabel 3.3:

Tabel 3.3 Contoh *Punctual Removal*

Kalimat asli:	"the <i>gameplay</i> is awesome!"
Hasil <i>Punctual Removal</i>:	"the <i>gameplay</i> is awesome"

4. *Stemming*

Setelah itu, proses *stemming* dilakukan untuk mengubah setiap kata menjadi bentuk dasarnya atau *root word*. Misalnya, kata "*gaming*" akan diubah menjadi "*game*." Tujuan dari *stemming* adalah untuk mengurangi variasi kata dan memastikan bahwa berbagai bentuk kata yang memiliki makna sama diperlakukan sebagai satu kata dasar. Langkah ini penting untuk menjaga konsistensi dalam data

dan memungkinkan model untuk fokus pada makna inti dari kata-kata dalam teks. Contoh *Stemming* ditunjukkan pada Tabel 3.4.

Tabel 3.4 Contoh *Stemming*

Kalimat asli:	"i was hooked by the <i>Graphics</i> "
Hasil <i>Stemming</i>:	"i be hook by the graphic"

Terakhir, setelah melalui tahap *tokenizing*, *case folding*, *punctuation removal*, dan *stemming*, teks ulasan kini berada dalam bentuk yang bersih dan siap untuk dianalisis lebih lanjut. *List* hasil *preprocessing* ini merupakan kumpulan kata-kata dasar yang telah diproses, yang akan menjadi *input* untuk langkah-langkah berikutnya, seperti ekstraksi fitur dan klasifikasi menggunakan model *machine learning* atau *neural network*. Hasil *preprocessing* ini berperan penting dalam memastikan bahwa data yang digunakan berkualitas tinggi dan sesuai dengan tujuan analisis.

3.4 *Fast-Text*

FastText adalah teknik pembelajaran representasi kata (*word embedding*) yang dikembangkan oleh tim **Facebook AI Research (FAIR)**. Teknik ini diperkenalkan untuk mengatasi beberapa kelemahan model *embedding* sebelumnya, seperti *Word2Vec*, terutama dalam menangani kata-kata yang jarang muncul (*rare words*) atau kata-kata baru yang tidak ada dalam data pelatihan (*out-of-vocabulary words*) (Bojanowski et al., 2017). *FastText* merepresentasikan kata-kata tidak hanya sebagai vektor unik, tetapi juga berdasarkan n-gram dari karakter-karakter yang menyusunnya. Pendekatan ini memungkinkan model untuk memahami struktur internal kata, sehingga lebih efektif dalam menangani kata-kata yang belum pernah dilihat selama pelatihan.

Dalam *FastText*, setiap kata direpresentasikan sebagai **jumlah (sum)** dari vektor embedding n-gram karakter yang menyusunnya. Sebagai contoh, kata "**gaming**" akan dipecah menjadi beberapa trigram karakter, seperti “*_ga*”, “*gam*”, “*ami*”, “*min*”, “*ing*”, dan “*ng_*” Setiap n-gram karakter ini memiliki embedding tersendiri, dan representasi akhir dari kata tersebut adalah jumlah dari semua embedding n-gram-nya. Secara matematis, representasi kata q dalam *FastText* dapat dinyatakan dengan rumus berikut:

$$Q = \sum_{g \in G(q)} z_g \quad (3.1)$$

Keterangan :

$G(q)$ = Himpunan *n-gram* karakter untuk kata q .
 z_g = Vektor embedding untuk setiap *n-gram* g .

Dalam beberapa implementasi praktis, setelah menjumlahkan vektor embedding n-gram, dilakukan normalisasi terhadap vektor hasil penjumlahan tersebut. Normalisasi biasanya dilakukan dengan membagi vektor hasil penjumlahan dengan jumlah n-gram $|N(q)|$, sehingga rumusnya menjadi:

$$X = \frac{1}{|N(q)|} Q \quad (3.2)$$

Keterangan :

Q = vektor embedding kata q .
 $|N(q)|$ = jumlah total n-gram dalam kata q .

Normalisasi ini dilakukan untuk menstabilkan skala vektor embedding yang dihasilkan. Tanpa normalisasi, kata-kata dengan jumlah n-gram yang lebih banyak (misalnya, kata yang lebih panjang) akan memiliki vektor embedding dengan magnitudo yang lebih besar dibandingkan kata-kata dengan jumlah n-gram yang

lebih sedikit. Hal ini dapat menyebabkan ketidakseimbangan dalam proses pembelajaran model, terutama ketika menggunakan metrik seperti *cosine similarity* yang sensitif terhadap magnitudo vektor (Goldberg, 2016). Selain itu, normalisasi membantu menghindari bias terhadap panjang kata, karena panjang kata tidak selalu berkorelasi dengan makna atau pentingnya kata tersebut (Jurafsky & Martin, 2020). Dengan normalisasi, vektor embedding dari kata-kata dengan panjang berbeda dapat dibandingkan secara lebih adil, karena magnitudo vektor telah diseragamkan.

FastText memiliki beberapa keunggulan yang membuatnya efektif dalam berbagai tugas pemrosesan bahasa alami (*NLP*). Pertama, *FastText* dapat menangani kata-kata langka atau kata-kata baru yang tidak ada dalam data pelatihan dengan baik, karena model ini menggunakan n-gram karakter untuk merepresentasikan kata. Kedua, *FastText* telah terbukti efektif dalam berbagai tugas *NLP*, seperti klasifikasi teks, analisis sentimen, dan pemodelan bahasa, menjadikannya alat yang serbaguna dalam penelitian dan aplikasi praktis.

Berikut adalah tahapan manual dalam proses *embedding* menggunakan *FastText*:

1. Pemisahan Kata Menjadi N-Gram Karakter

FastText memecah kata menjadi beberapa n-gram karakter. N-gram adalah urutan karakter berturut-turut dalam sebuah kata. Untuk kata "*gaming*", tambahkan simbol pembatas khusus di awal dan akhir kata, menjadi "<gaming>", untuk memudahkan pemahaman konteks kata yang dimulai dan berakhir dengan karakter tertentu. Dengan kata "*gaming*", proses

pemecahannya menjadi n-gram 3-karakter (misalnya trigram) ditunjukkan pada Tabel 3.5.

Tabel 3.5 Contoh Kata Menjadi N-Gram Karakter

Kata	N-gram	Hasil (Subword)
<gaming>	3	<ga,gam,ami,min,ing,ng>

2. Vektor Embedding untuk Setiap *Subword*

Vektor-vektor embedding untuk n-gram tidak diberikan secara langsung, tetapi dipelajari melalui model *FastText*.

- a. FastText menggunakan pendekatan skip-gram, di mana model berusaha memprediksi kata-kata konteks di sekitar sebuah kata target dengan memanfaatkan *subword* nya. Pada saat pelatihan, model akan memperhatikan *subword* seperti "gam", "ami", "min", dll., dan mencoba memprediksi kata-kata di sekitar kata tersebut.
- b. Model dimulai dengan inisialisasi vektor embedding acak untuk setiap n-gram karakter (misalnya, untuk "gam", "ami", dll.).
- c. Selama pelatihan, model mencoba meminimalkan kesalahan prediksi konteks kata (kata yang muncul di sekitar kata target).

Untuk kata "gaming", pada tahap ini *FastText* akan memperlakukan *subword* nya, seperti "gam", "ami", dan lainnya, sebagai unit yang terpisah dan akan mencoba belajar representasi vektor untuk *subword* tersebut.

3. Penjumlahan dan Normalisasi Vektor *Subword* untuk Representasi Kata

Vektor embedding untuk setiap *subword* telah dipelajari. Untuk mendapatkan representasi akhir dari kata "gaming", *FastText* akan menjumlahkan vektor-vektor embedding dari semua *subword* yang

membentuk kata tersebut. Vektor embedding untuk setiap subword (dengan dimensi 4 untuk contoh ini) mungkin ditunjukkan pada Tabel 3.6.

Tabel 3.6 Contoh *Subword Vector Embedding*

<i>Sub-word</i>	<i>Vektor Embedding</i>
<ga	[0,10,-0,30,0,25,0,05]
gam	[0,12,-0,43,0,56,0,11]
ami	[0,22,-0,14,0,31,-0,07]
min	[0,16,0,23,-0,21,0,15]
ing	[0,30,-0,22,0,10,0,04]
ng>	[0,11,0,24,-0,12,0,33]

Penjumlahan dari vektor-vektor ini menghasilkan representasi akhir untuk kata "gaming" yang nantinya akan dilakukan normalisasi dengan membaginya dengan jumlah n-gram pada kata "gaming":

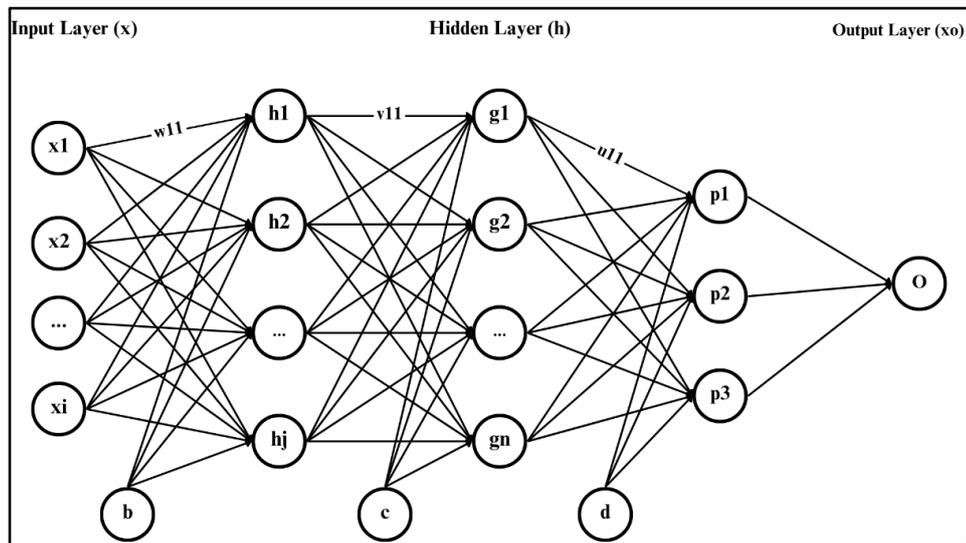
$$X_{gaming} = [(0,10 + 0,12 + 0,22 + 0,16 + 0,30 + 0,11), (-0,30 + (-0,43) + (-0,14) + 0,23 + (-0,22) + 0,24), \dots]$$

$$X_{gaming} = [1,01, -0,62, 0,89, 0,61] \times \frac{1}{6}$$

$$X_{gaming} = [0,17, -0,10, 0,15, 0,10]$$

3.5 *Neural Network*

Proses dimulai dari data *input* yang berupa representasi nilai dari fitur-fitur yang dihasilkan dari proses sebelumnya, seperti *FastText* atau representasi fitur lainnya, yang digunakan sebagai informasi awal untuk *neural network*. *Hidden layer* berada di antara *input layer* dan *output layer* dan berfungsi untuk memetakan informasi internal yang kompleks. Arsitektur *neural network* ditunjukkan pada Gambar 3.4.

Gambar 3.4 Arsitektur *Neural Network*

Pada *neural network*, dilakukan perhitungan berupa hasil kali matriks *input* dengan bobot, ditambahkan dengan bias, kemudian fungsi aktivasi diterapkan untuk menghasilkan *output* dari *layer* sebelumnya. *Output* ini kemudian dikirimkan sebagai *input* ke *layer* berikutnya. Struktur arsitektur *neural network* pada gambar dijelaskan sebagai berikut:

- Input Layer***: Terdiri dari i node, dengan i merupakan jumlah fitur yang dihasilkan dari proses ekstraksi fitur *FastText*. Setiap node merepresentasikan sebuah fitur unik.
- Hidden Layer 1***: Memiliki 14 node yang bertanggung jawab untuk mendeteksi pola-pola mendasar dalam data.
- Hidden Layer 2***: Memiliki 14 node yang bertanggung jawab untuk mendeteksi pola-pola mendasar dalam data.
- Hidden Layer 3***: Menggunakan sejumlah node yang direpresentasikan dengan simbol $p_1, p_2,$ dan p_3 yang sesuai dengan jumlah kelas yang ada dalam kasus klasifikasi (berjumlah 3 kelas).

- e. **Output Layer:** Memiliki 1 node yang berfungsi untuk mengambil nilai tertinggi dari node pada *layer* sebelumnya sebagai hasil dari klasifikasi

Pada *hidden layer* 1, Nilai *output* dari node h_j di *hidden layer* 1 dihitung

dengan rumus :

$$h_j = \max \left(0, \left(\sum_{i=1} (x_i \times w_{ij}) + b \right) \right) \quad (3.3)$$

Keterangan :

h_j = Hasil perhitungan fungsi aktivasi.

x_i = Nilai *input*.

w_{ij} = Nilai Bobot antara node *input* x_i dan node h_j di *hidden Layer* 1.

b = Nilai Bias *hidden Layer* 1.

Nilai dari hasil perhitungan bobot dan bias digunakan sebagai *input* fungsi aktivasi yang nanti hasil dari fungsi aktivasi akan menjadi nilai pada *node* di *hidden layer* 1. Fungsi aktivasi yang digunakan pada *hidden layer* 1 yaitu ReLU (*Rectified Linear Unit*) yang berfungsi sebagai mengganti semua nilai negatif menjadi nol, yang membantu jaringan belajar dengan lebih efisien.

Untuk perhitungan *hidden layer* 2 relatif mirip dengan perhitungan *hidden layer* 1 dimana nilai dari hasil perhitungan aktivasi ReLU pada *hidden layer* 1 digunakan sebagai *input* untuk perhitungan bobot dan bias pada *hidden layer* 2. Kemudian, hasil perhitungan bobot dan bias pada *hidden layer* 2 digunakan untuk untuk perhitungan aktivasi pada *hidden layer* 2 yang dimana fungsi yang digunakan sama, yaitu fungsi aktivasi ReLU. Berikut adalah rumus-rumus nya:

$$g_n = \max \left(0, \left(\sum_{j=1} (h_j \times v_{jn}) + c \right) \right) \quad (3.4)$$

Keterangan :

- g_n = Hasil perhitungan fungsi aktivasi.
- h_j = Nilai *input*.
- v_{ij} = Bobot antara *output* hidden layer 1 (h_j) dan node g_n di hidden layer 2.
- c = Nilai bias hidden layer 2.

Setelah itu dilakukan perhitungan bobot dan bias untuk *hidden layer* 3 yang kemudian hasil dari perhitungan tersebut dilakukan aktivasi menggunakan fungsi aktivasi *softmax*. Kemudian, nilai dari *node* yang memiliki nilai tertinggi yang ada pada *hidden layer* 3 akan di tampung pada *ouput layer*. Berikut adalah rumus-rumusya:

$$z_k = \sum_{n=1} (g_n \times u_{nk}) + d \quad (3.5)$$

Keterangan :

- z_k = Hasil perhitungan sebelum dilakukan fungsi aktivasi.
- g_n = Nilai *input*.
- u_{nk} = Nilai bobot.
- d = Nilai Bias hidden layer 3

$$p_k = \frac{\exp(z_k)}{\sum_{i=1}^k \exp(z_k)} \quad (3.6)$$

$$o = \operatorname{argmax}(p_k) \quad (3.7)$$

Keterangan :

- p_k = Nilai probabilitas dari kelas k .
- k = Jumlah node pada layer (sesuai dengan jumlah kelasnya).
- o = Kelas dengan probabilitas tertinggi.

3.5.1 Proses *Training*

Setelah melakukan perhitungan diatas, langkah selanjutnya ialah menghitung error di *output layer*. Error dihitung dengan membandingkan prediksi

model (y') dengan target sebenarnya (y) menggunakan fungsi *cross-entropy loss*.

Berikut adalah rumusnya :

$$l = \sum_{i=1}^3 y_i \log y_i' \quad (3.8)$$

Keterangan :

y_i = Label target untuk kelas ke- i

y_i' = Probabilitas prediksi dari fungsi Softmax untuk kelas ke- i

Kemudian dilakukan perhitungan *update* bobot dan bias agar hasil *training* model selanjutnya menjadi lebih baik. Untuk rumus-rumus sebagai berikut :

a. *Update* bobot pada *hidden layer 3*

Hitung gradien error pada *output layer* $\delta_k^{(3)}$:

$$\delta_k^{(3)} = \hat{y}_k - y_k \quad (3.9)$$

Perbarui bobot dan bias:

$$u_{nk(\text{baru})} = u_{nk(\text{lama})} - \eta \cdot \delta_k^{(3)} \cdot g_n \quad (3.10)$$

$$d_{(\text{baru})} = d_{(\text{lama})} - \eta \cdot \delta_k^{(3)} \quad (3.11)$$

b. *Update* bobot pada *hidden layer 2*

Hitung gradien error pada *hidden layer 2* $\delta_n^{(2)}$:

$$\delta_n^{(2)} = \left(\sum_k \delta_k^{(2)} \cdot u_{nk} \right) g_n' \quad (3.12)$$

Perbarui bobot dan bias:

$$v_{jn(\text{baru})} = v_{jn(\text{lama})} - \eta \cdot \delta_n^{(2)} \cdot h_i \quad (3.13)$$

$$c_{(\text{baru})} = c_{(\text{lama})} - \eta \cdot \delta_n^{(2)} \quad (3.14)$$

c. Update bobot pada *hidden layer* 1

Hitung gradien error pada hidden layer 1 $\delta^{(1)}$:

$$\delta_j^{(1)} = \left(\sum_n \delta_n^{(2)} \cdot v_{jn} \right) h_i' \quad (3.15)$$

Perbarui bobot dan bias:

$$w_{ij(\text{baru})} = w_{ij(\text{lama})} - \eta \cdot \delta_i^{(1)} \cdot x_i \quad (3.16)$$

$$b_{(\text{baru})} = b_{(\text{lama})} - \eta \cdot \delta_i^{(1)} \quad (3.17)$$

Keterangan :

- \hat{y}_k = Prediksi *output*
- y_k = Label target
- η = *Learning rate*
- u_{nk} = Bobot dari hidden layer 2 ke *output* layer
- v_{jn} = Bobot dari hidden layer 1 ke hidden layer 2
- w_{ij} = Bobot dari *input* ke hidden layer 1

Proses *backpropagation* ini dilakukan berulang kali untuk setiap *epoch* dalam proses *training* hingga model mencapai konvergensi atau error minimum. Algoritma proses *training* pada *neural network* di tunjukkan melalui *pseudo-code* berikut:

Algorithm 1 : *Training Neural network*

Input : X_train, Y_train, tolerance, *learning rate*, model

Output : loss_per_iteration, Accuracy_per_iteration

- 1 Initialize w, v, u \leftarrow Random Decimal Number
- 2 Initialize b, c, d \leftarrow Random Decimal Number
- 3 Set prev_loss \leftarrow 0
- 4 Set iteration \leftarrow 0
- 5 Set no_improvement_count \leftarrow 0
- 6 loss_per_iteration \leftarrow []
- 7 Accuracy_per_iteration \leftarrow []

```

8 while True do
9     z_h ← X_train * w + b
10    a_h ← ReLU(z_h)
11    z_g ← a_h * v + c
12    a_g ← ReLU(z_g)
13    z_p ← a_g * u + d
14    a_p ← Softmax(z_p)

15    dz_p ← a_p - Y_train
16    dw_p ← (a_g.T * dz_p) / m
17    dd ← sum(dz_p) / m
18    dz_g ← (dz_p * u.T) * ReLU_derivative(z_g)
19    dw_g ← (a_h.T * dz_g) / m
20    dc ← sum(dz_g) / m
21    dz_h ← (dz_g * v.T) * ReLU_derivative(z_h)
22    dw_h ← (X_train.T * dz_h) / m
23    db ← sum(dz_h) / m

24    u ← u - learning_rate * dw_p
25    d ← d - learning_rate * dd
26    v ← v - learning_rate * dw_g
27    c ← c - learning_rate * dc
28    w ← w - learning_rate * dw_h
29    b ← b - learning_rate * db

30    loss ← -sum(Y_train * log(a_p + 1e-15)) / m
31    Accuracy ← mean(argmax(Y_train, axis=1) == argmax(a_p, axis=1))

32    loss_per_iteration ← loss
33    Accuracy_per_iteration ← Accuracy

34    if iteration mod 10 = 0 then
35        Print("Iteration", iteration, "Loss:", loss, "Accuracy:", Accuracy)
36    end if

37    if |prev_loss - loss| < tolerance then
38        no_improvement_count ← no_improvement_count + 1
39    else
40        no_improvement_count ← 0
41    end if

42    if no_improvement_count ≥ 5 then
43        Print("Training stopped due to no improvement.")
44        break
45    end if
46    prev_loss ← loss

```

```

47     iteration ← iteration + 1
48 end while
49 return loss per iteration, Accuracy per iteration

```

Proses *feedforward* dalam *neural network* dimulai dengan mengalirkan data dari *input layer* menuju ke *hidden layers* dan akhirnya ke *output layer*. Rumus 3.3 diterapkan pada *hidden layer* 1, *input* x_i (X_{train}) dikalikan dengan bobot w_{ij} (w), kemudian ditambahkan bias b (b) dan diterapkan fungsi aktivasi *ReLU*, menghasilkan *output* h_j (a_h). perhitungan ini ditunjukkan pada baris 9 & 10 pada *pseudocode*. Proses serupa terjadi untuk Rumus 3.4 di *hidden layer* 2, di mana *output* dari *hidden layer* 1, h_j (a_h) dikalikan dengan bobot v_{jn} (v), ditambahkan bias c (c), dan kemudian fungsi aktivasi *ReLU* diterapkan untuk menghasilkan *output* g_n (a_g). Perhitungan ini ditunjukkan pada baris 11 & 12 pada *pseudocode*. Selanjutnya, *output* dari *hidden layer* 2 diproses dengan Rumus 3.5 di *hidden layer* 3, di mana setiap *output* g_n (a_g) dikalikan dengan bobot u_{nk} (u) dan ditambahkan bias d (d), menghasilkan *input* z_k (z_p) untuk *hidden layer* 3 dan dilakukan fungsi aktivasi *Softmax* menggunakan Rumus 3.6 diterapkan pada nilai z_k untuk menghasilkan *output* p_k (a_p), yang menunjukkan probabilitas masing-masing kelas. Perhitungan ini ditunjukkan pada baris 13 & 14 pada *pseudocode*. Proses *feedforward* ini menggambarkan alur dari *input* hingga *output* dengan melibatkan perhitungan bobot, bias, dan fungsi aktivasi, yang semuanya terhubung langsung dengan implementasi kode dalam *neural network*.

Pada proses *backpropagation*, dimulai dengan menghitung *error* pada *output layer*, yang disebut dengan $\delta_k^{(3)}$ (dz_p), menggunakan Rumus 3.9 yang merupakan selisih antara prediksi yang dihasilkan model \hat{y}_k (a_p) dan label target

y_k (Y_{train}). *Error* ini digunakan untuk menghitung gradien pada *output layer*, yang menunjukkan seberapa besar kontribusi masing-masing bobot terhadap kesalahan prediksi. Selanjutnya, error dari *output layer* diteruskan ke *hidden layer* 2, menggunakan Rumus 3.12 di mana *error* $\delta_n^{(2)}$ (dz_g) dihitung dengan mengalikan error *output layer* $\delta_k^{(3)}$ (dz_p) dengan bobot u_{nk} (u), dan kemudian diterapkan pada turunan fungsi aktivasi *ReLU hidden layer 2* (z_g). Proses yang sama dilakukan pada *hidden layer 1*, menggunakan Rumus 3.15 di mana *error* $\delta_j^{(1)}$ (dz_h) dihitung dengan mengalikan error dari *hidden layer 2*, $\delta_n^{(2)}$ (dz_g), dengan bobot v_{jn} (v), kemudian diterapkan pada turunan fungsi aktivasi *ReLU hidden layer 1* (z_h) untuk menghasilkan error pada hidden layer 1.

Setelah menghitung error di setiap layer, langkah berikutnya adalah memperbarui bobot dan bias di setiap lapisan menggunakan *gradient descent*. Gradien untuk masing-masing bobot dihitung dengan mengalikan *error* dengan *input* yang sesuai, lalu bobot dan bias diperbarui dengan mengurangi nilai gradien yang dikalikan dengan *learning rate*. Dengan melakukan *backpropagation* ini secara berulang pada setiap iterasi, model secara bertahap akan mengurangi kesalahan dalam prediksi dan meningkatkan akurasi melalui pembaruan parameter yang lebih baik. Proses ini memungkinkan model *neural network* untuk belajar dan mengoptimalkan prediksinya berdasarkan data yang diberikan.

BAB IV

UJI COBA DAN PEMBAHASAN

4.1 Data Pengujian

Dataset ulasan *game* yang digunakan dalam penelitian ini diambil dari *Kaggle* ([Steam Reviews](#)), sebuah platform yang menyediakan berbagai macam dataset untuk keperluan penelitian dan analisis data. Dataset ini berisi ulasan yang diberikan oleh pemain *game* pada berbagai platform *game* online, seperti Steam. Ulasan-ulasan ini mencakup beragam aspek pengalaman bermain, yang relevan untuk tujuan penelitian ini, seperti *gameplay*, grafik, dan cerita. Namun dataset tersebut belum memiliki kategori aspek utama *game* yang dijadikan sebagai label atau *Ground Truth*. Jumlah ulasan aspek utama *game* berdasarkan hasil survey kepada responden umum yang ditunjukkan pada table berikut 4.1:

Tabel 4.1 Jumlah Ulasan Aspek Utama *Game*

Label / Ground Truth	Jumlah
<i>Gameplay</i>	493
<i>Graphics</i>	309
<i>Story</i>	209
Total	1011

Kategori *Gameplay* mengacu pada berbagai aspek interaksi dan pengalaman pemain dalam sebuah *game*. Kategori ini mencakup mekanisme *gameplay*, kontrol, *challenges*, dan nuansa *gameplay* secara keseluruhan. Indikator utama untuk kategori ini melibatkan fokus pada pengalaman *gameplay*, seperti *challenges* dan *gameplay loop* juga merupakan faktor penting, dengan ulasan yang membahas kesulitan, desain level, dan dampak tindakan pemain (Fabricatore Carlo, 2007). Peran pemain dan dinamika tim dipertimbangkan, terutama dalam pengaturan

multi-player. Mode tertentu, seperti *gameplay single-player* dan kompetitif, relevan dengan kategori ini (Marshall et al., 2013). Komentar tentang masalah kontrol dan responsivitas, seperti kontrol atau kesulitan berinteraksi dengan elemen *game*, juga signifikan. Aspek-aspek ini secara kolektif berkontribusi pada estetika *Gameplay* pada *game* (Zagal & Tomuro, 2010).

Kategori *Graphics* mengacu pada elemen visual yang ditampilkan di layar, yang mewakili suasana dunia pada *game*, karakter, objek, dan efek visual. Hal ini dibuat menggunakan teknologi grafik komputer dan merupakan komponen inti dari bagaimana *player* merasakan dan berinteraksi dengan *game* (Tavares et al., 2011). Untuk komponen-komponen dari *Graphics* pada *game* itu meliputi:

1. **Asset Grafik:** Ini adalah komponen visual individual yang digunakan dalam permainan, seperti model karakter, tekstur, properti, dan elemen antarmuka pengguna (Fukaya et al., 2025).
2. **Animasi dan Grafik Gerak:** Animasi menghidupkan aset grafis, memungkinkan pergerakan karakter, objek, dan efek visual. Termasuk Grafik Gerak / *Motion Graphics* digunakan untuk efek visual dinamis, seperti ledakan, transisi antarmuka, dan elemen animasi lainnya yang meningkatkan pengalaman visual (Tavares et al., 2011).
3. **Performa dan *Rendering*:** *Rendering* grafis *real-time* sangat penting untuk permainan interaktif, yang mengharuskan *game* untuk menghasilkan visual dengan cukup cepat untuk merespons masukan pemain, biasanya pada *frame rate* di atas 30 *frame* per detik untuk interaktivitas yang lancar (Spjut et al., 2022).

Kategori *Story / Lore* ini mengacu pada merujuk pada elemen naratif yang memberikan konteks, latar belakang, dan makna pada dunia, karakter, dan peristiwa dalam *game*. Dimana Cerita / *Story* ialah rangkaian peristiwa dan alur karakter yang terstruktur yang terungkap seiring progress pemain, sementara pengetahuan / *Lore* mencakup sejarah yang lebih luas, mitos, dan detail mendasar yang memperkaya dunia *game*, sering kali disajikan melalui petunjuk lingkungan, dialog, atau materi tambahan (X. Wang et al., 2025). Untuk komponen-komponen *Story / Lore* pada *game* itu meliputi:

1. Pembangunan Dunia dan Pengetahuan (*Lore*): Penciptaan dunia permainan yang dapat dipercaya dan terperinci, termasuk sejarah, budaya, dan aturannya. Pengetahuan / *Lore* sering disampaikan melalui penceritaan lingkungan, deskripsi item, seni visual (seperti *splash art*), dan detail latar belakang yang mungkin tidak menjadi pusat plot utama tetapi menambah kedalaman dan kekayaan.
2. Struktur Naratif: Alur cerita yang menyeluruh, termasuk bagian awal, tengah, dan akhir, serta jalur yang bercabang atau beberapa akhir cerita dalam narasi interaktif (Gyllenback, 2009). Dapat mencakup penceritaan linier atau pendekatan *non-linier* yang terfragmentasi, tempat pemain menyusun cerita dari petunjuk yang tersebar.
3. Karakter dan Pengembangan Karakter: Kepribadian, motivasi, dan perkembangan karakter dalam *game*. Latar belakang dan hubungan karakter sering kali terungkap melalui dialog, *cutscene*, dan pilihan pemain (Kouratoras, 2020).

4.2 Skenario Pengujian

Setelah model *neural network* melalui tahap pelatihan (*training*) untuk mempelajari data, langkah selanjutnya adalah tahap *testing*. Tujuan utama dari tahap ini adalah untuk mengukur seberapa baik kinerja model setelah melewati proses pelatihan. Proses *testing* dilakukan dengan menggunakan data ulasan *game* yang sebelumnya tidak dikenali oleh model selama pelatihan.

Langkah pertama dalam proses *testing* adalah menyiapkan data ulasan *game* yang akan diuji dengan model *neural network*. Data ini berisi ulasan yang telah dilabeli dengan kategori aspek utama seperti *Gameplay*, *Graphic*, dan *Story*, namun data ini belum pernah digunakan dalam proses pelatihan sebelumnya. Model yang telah dioptimasi bobot dan bias-nya pada tahap pelatihan kemudian diterapkan untuk melakukan prediksi pada data *testing*.

Selanjutnya, model akan melakukan feedforward pada data *testing* untuk menghasilkan prediksi kategori aspek utama dari setiap ulasan. Hasil prediksi yang diperoleh dari model selanjutnya akan dibandingkan dengan *ground truth* atau label yang sebenarnya, yang telah ditetapkan selama tahap pelabelan manual. Perbandingan ini akan menghasilkan nilai error yang menunjukkan sejauh mana ketepatan prediksi model. Hasil ini digunakan untuk mengevaluasi performa kinerja model *neural network* yang telah dibangun dalam mengklasifikasikan aspek ulasan *game* secara tepat. Algoritma proses *training* pada *neural network* di tunjukkan melalui *pseudocode* berikut:

Algorithm 2 : *Testing Neural network*

Input : X_test, Y_test, model

Output : test_Accuracy, confusion_matrix

```

1 y_pred ← model.Forward(X_test)
2 test_Accuracy ← Compute_Accuracy(y_test, y_pred)
3 Print("Test Accuracy:", test_Accuracy)

4 y_pred_classes ← argmax(y_pred)
5 y_true_classes ← argmax(y_test)
6 confusion_matrix ← Compute_Confusion_Matrix(y_true_classes,
y_pred_classes)
7 Print("Confusion matrix:", confusion_matrix)

8 return test_Accuracy, confusion_matrix

```

Pada proses *testing* dalam *neural network*, langkah pertama adalah melakukan *forward pass* pada data uji (X_test) untuk mendapatkan prediksi model (y_pred). Prediksi ini dihitung dengan meneruskan data uji melalui jaringan yang sudah dilatih, menghasilkan *output* yang berupa probabilitas untuk setiap kelas. Selanjutnya, akurasi model dihitung dengan membandingkan label yang diprediksi (y_pred) dengan label yang benar (Y_test) menggunakan fungsi *Compute_Accuracy*. Hasil perbandingan ini memberikan nilai akurasi yang menunjukkan seberapa baik model dalam mengklasifikasikan data uji. Akurasi ini kemudian dicetak sebagai informasi performa model pada data yang tidak dilatih sebelumnya.

Setelah itu, untuk mengevaluasi model dengan lebih mendalam, dihitunglah *confusion matrix*. Proses ini dimulai dengan mengambil kelas yang diprediksi (y_pred_classes) dan kelas yang benar (y_true_classes) menggunakan *argmax* pada hasil *output* model dan label asli. *Confusion matrix* kemudian dihitung dengan

membandingkan kedua kelas ini, memberikan gambaran rinci tentang jumlah prediksi yang benar dan salah untuk setiap kelas. Hasil *confusion matrix* ini juga dicetak untuk memberikan wawasan yang lebih detail mengenai bagaimana model mengklasifikasikan data pada level kelas-kelas individu. Semua langkah ini bersama-sama memberikan gambaran yang komprehensif tentang kinerja model pada data uji yang sebelumnya tidak pernah dilihat oleh model.

Skenario pengujian dilakukan untuk mengevaluasi performa model *neural network* yang telah dibangun. Proses pengujian dilakukan dengan mengatur parameter-parameter tertentu seperti jumlah *node* pada *hidden layer 1* dan *hidden layer 2*, *learning rate*. Tujuannya adalah untuk mengetahui konfigurasi terbaik yang menghasilkan akurasi tertinggi dengan *error* yang minimal. Parameter yang digunakan pada skenario pengujian meliputi:

1. **Learning rate (η):** Mengatur kecepatan pembaruan bobot selama proses backpropagation.
2. **Toleransi Error:** batas perubahan error yang diizinkan selama proses pelatihan. Jika perubahan *error* di bawah nilai toleransi, model akan berhenti melakukan pelatihan karena dianggap sudah mencapai konvergensi.

Adapun rasio pembagian data digunakan untuk melatih model dengan berbagai proporsi data untuk *training*, *validation*, dan *testing*.

Metode pengujian dilakukan dengan langkah-langkah berikut:

- a. **Training Model:** Model dilatih dengan parameter tertentu sesuai skenario pengujian.

- b. **Evaluasi Model:** Model diuji menggunakan data validasi untuk menghitung metrik performa (akurasi, *precision*, *recall*, dan *F1-score*).

Berikut adalah Tabel skenario pengujian pada penelitian ini:

Tabel 4.2 Skenario Pengujian

Skenario	Rasio Pembagian Data	Learning rate (η)	Toleransi Error
1	90:10	0,01	1e-5
2	80:20	0,01	1e-5
3	70:30	0,01	1e-5
4	60:40	0,01	1e-5
5	50:50	0,01	1e-5

Tabel 4.2 menunjukkan enam skenario pengujian yang dilakukan untuk mengevaluasi kinerja model dalam penelitian ini. Pada setiap skenario, jumlah data yang digunakan adalah sebanyak 1011 sampel, yang dipertahankan konstan agar pengujian lebih terkontrol dan hasil yang diperoleh dapat dibandingkan secara konsisten.

Pada model *FastText*, dimensi embedding yang digunakan dalam eksperimen ini diatur menjadi 100 dimensi. Hal ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa embedding dengan dimensi 100 telah terbukti efektif dalam banyak tugas *text classification*. Sebagai contoh, penelitian oleh Bojanowski et al. (2017) dalam pengembangan *FastText* menyarankan dimensi embedding 100-300 dimensi untuk menangkap makna semantik kata secara optimal, sementara tetap menjaga keseimbangan antara akurasi dan efisiensi komputasi. Dimensi embedding yang lebih kecil, seperti 100, biasanya cukup untuk mendapatkan representasi kata yang kuat tanpa memperlambat pelatihan secara signifikan.

Pada model *neural network* ini menggunakan 201 *neuron* pada *hidden layer* pertama dan 51 *neuron* pada *hidden layer* kedua. Jumlah *neuron* pada setiap *hidden layer* ini dipilih berdasarkan aturan praktis dan rumus teori yang digunakan dalam penelitian Syaharuddin et al., (2022), yang menawarkan rumus untuk menghitung batas atas jumlah *neuron* pada *hidden layers* 1 dan 2 dalam *multi-layer neural networks*.

$$N_z = 2 \times N_x + 1 \quad (4.1)$$

$$N_z = \frac{N_x + N_y}{2} \quad (4.2)$$

Dengan $N_x = 100$ (jumlah *neuron* pada *input layer*), dan $N_y = 3$ (jumlah kelas / label), dapat menghitung batas atas jumlah *neuron* yang optimal pada kedua *hidden layer*. Untuk rumus 4.1 digunakan untuk menentukan jumlah *neuron* pada *hidden layer* 1 dan rumus 4.2 digunakan untuk menentukan jumlah *neuron* pada *hidden layer* 2. Berdasarkan rumus ini, terhitung bahwa jumlah *neuron hidden layer* 1 ialah 201 dan jumlah *neuron* untuk *hidden layer* 2 ialah 51, yang dianggap cukup untuk menangkap pola kompleks dari dataset sambil menghindari *overfitting*

Selanjutnya, pada skenario pengujian, parameter yang diuji mencakup *learning rate*, toleransi error, dan rasio pembagian data. Pengujian dilakukan pada beberapa kombinasi rasio pembagian data. Penentuan *learning rate* berdasarkan praktik umum dalam penelitian sebelumnya tentang *text classification* dan *Neural network*, seperti yang dilakukan dalam penelitian Krizhevsky et al., (2012) yang menggunakan *learning rate* 0,01 untuk melatih CNN.

Rasio pembagian data *training* dan *testing* pada skenario pengujian ini dibagi menjadi pembagian yang umum digunakan dalam berbagai penelitian

sebelumnya, yaitu pembagian 90:10, 80:20, 70:30, 60:40, dan 50:50 seperti pada penelitian Umar Ibrahim et al., (2022).

Untuk toleransi error, nilai yang digunakan adalah $1e-5$, yang merupakan nilai standar yang banyak digunakan dalam eksperimen *deep learning* untuk menentukan kapan pelatihan dihentikan. Nilai ini memastikan bahwa pelatihan tidak berlanjut jika perubahan error antara iterasi sangat kecil, yang menunjukkan bahwa model telah mencapai konvergensi dan tidak lagi memperoleh manfaat dari pelatihan lebih lanjut.

Dari lima skenario pengujian yang diajukan, dilakukan proses pelatihan (*training*) dan pengujian (*testing*). Setelah mendapatkan nilai akurasi tertinggi pada setiap skenario, langkah selanjutnya adalah menghitung *confusion matrix*. *Confusion matrix* digunakan untuk memberikan gambaran yang lebih jelas mengenai performa klasifikasi model yang telah dilatih. *Confusion matrix* terdiri dari empat kategori utama yang digunakan untuk mengevaluasi hasil prediksi model.

- 1) ***True Positive (TP)*** adalah jumlah data ulasan yang berhasil diprediksi dengan benar sebagai kelas positif.
- 2) ***False Positive (FP)*** terjadi ketika model salah memprediksi kelas positif.
- 3) ***True Negative (TN)*** adalah jumlah data ulasan yang berhasil diprediksi dengan benar sebagai kelas negatif.
- 4) ***False Negative (FN)*** terjadi ketika model salah memprediksi kelas negatif.

Pada kasus klasifikasi multiclass, di mana terdapat lebih dari dua kelas kategori, kelas positif adalah label yang dihitung dalam perhitungan ini, sementara

kelas negatif adalah label lainnya (Ali et al., 2022). Evaluasi ini digunakan untuk menganalisis keakuratan model dalam mengklasifikasikan aspek utama dalam ulasan *game*, seperti **Gameplay**, **Graphic**, dan **Story**.

Confusion matrix digunakan untuk mengukur performa model klasifikasi aspek dalam ulasan *game*, di mana kategori A, B, dan C mewakili kelas-kelas aspek utama dalam ulasan *game*. Kelas A merujuk pada aspek **Gameplay**, kelas B pada aspek **Graphic**, dan kelas C pada aspek **Story**. *Confusion matrix* untuk penelitian ini disajikan dalam Tabel 4.3.

Tabel 4.3 *Confusion Matrix*

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay</i> (A)	<i>Graphics</i> (B)	<i>Story</i> (C)
Gameplay (A)	AA	AB	AC
Graphics (B)	BA	BB	BC
Story (C)	CA	CB	CC

Confusion matrix untuk setiap kelas memiliki empat kategori: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) dalam setiap kelas. Dimana *True Positive* (TP) Merupakan jumlah ulasan yang benar-benar termasuk dalam kategori *Gameplay*, *Graphic*, atau *Story*, dan diprediksi dengan tepat oleh model sebagai kategori yang sesuai, *True Negative* (TN) Merupakan jumlah ulasan yang tidak termasuk dalam kategori yang diprediksi, dan model memprediksi dengan benar bahwa ulasan tersebut tidak masuk dalam kategori tersebut. *False Positive* (FP) Merupakan jumlah ulasan yang sebenarnya tidak termasuk dalam kategori yang diprediksi, dan *False Negative* (FN) merupakan jumlah ulasan yang sebenarnya termasuk dalam kategori yang diprediksi, tetapi model salah memprediksi bahwa ulasan tersebut tidak masuk

dalam kategori yang tepat. *Confussion matrix* masing-masing kelas ditampilkan pada Tabel 4.4, Tabel 4.5, Tabel 4.6.

Tabel 4.4 *Confusion Matrix* Kelas *Gameplay*

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi	
	<i>Gameplay</i> (A)	Bukan <i>Gameplay</i>
<i>Gameplay</i> (A)	TP (AA)	FN (AB + AC)
Bukan <i>Gameplay</i>	FP (BA + CA)	TN (BB + CB + BC + CC)

Tabel 4.5 *Confusion Matrix* Kelas *Graphics*

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi	
	<i>Graphics</i> (B)	Bukan <i>Graphics</i>
<i>Graphics</i> (B)	TP (BB)	FN (BA + BC)
Bukan <i>Graphics</i>	FP (AB + CB)	TN (AA + AC + CA + CC)

Tabel 4.6 *Confusion Matrix* Kelas *Story*

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi	
	<i>Story</i> (C)	Bukan <i>Story</i>
<i>Story</i> (C)	TP (CC)	FN (CA + CB)
Bukan <i>Story</i>	FP (AC + BC)	TN (AA + AB + BA + BB)

Dari tabel *confusion matrix* dapat digunakan untuk mengukur nilai akurasi, presisi, *recall*, dan *F1-score* dari model dengan menggunakan persamaan berikut:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (4.4)$$

Accuracy adalah nilai persentase prediksi benar terhadap semua data. *Precision* adalah rasio prediksi benar kelas positif terhadap total semua kelas positif. *Recall* adalah proporsi kelas positif yang berhasil diprediksi dengan benar.

F1-score adalah rata-rata bobot dari presisi dan *recall*. Nilai *F1-score* yang lebih tinggi menunjukkan bahwa performa model yang lebih baik (Ali et al., 2022):

4.3 Hasil Uji Coba

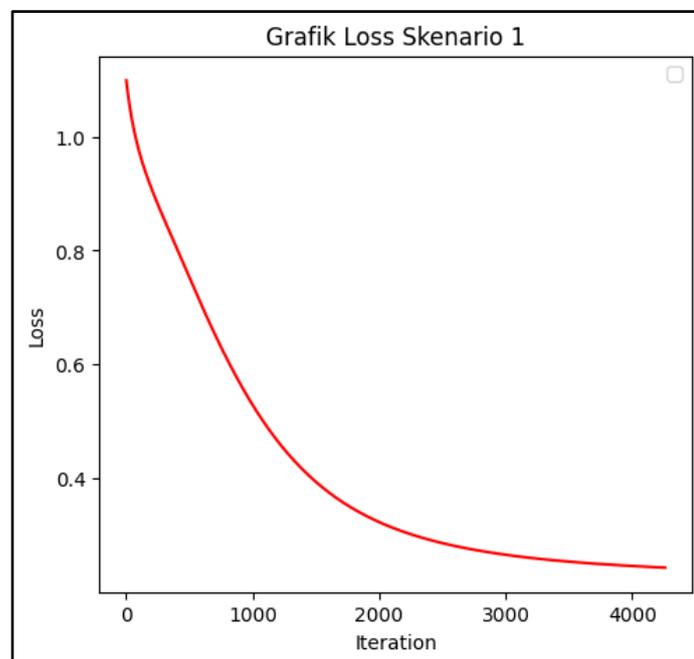
4.3.1 Hasil Uji Coba Skenario 1

Pada Skenario 1, model *neural network* diuji dengan rasio 90% *training* dan 10% *testing*, *learning rate* 0,1, dan toleransi error $1e-5$. Untuk Jumlah data untuk tiap-tiap kelas pada skenario 1 ditunjukkan pada tabel 4.7.

Tabel 4.7 Jumlah Data Tiap Kelas Skenario 1

Label / Ground Truth	Data Training	Data Testing
<i>Gameplay</i>	436	57
<i>Graphics</i>	284	25
<i>Story</i>	189	20
Total	909	102

Hasil pengujian bahwa model memiliki performa yang cukup baik dalam mengklasifikasikan ulasan *game* ke dalam tiga kategori utama, yaitu *Gameplay*, *Graphics*, dan *Story/Lore*, dapat dilihat pada grafik *loss* Gambar 4.1 .



Gambar 4.1 Grafik *Loss* Skenario 1

Grafik *training loss* menunjukkan tren yang positif selama proses pelatihan. Pada grafik *training loss* kita dapat melihat penurunan yang signifikan pada awal pelatihan, diikuti dengan konvergensi yang stabil setelah sekitar 1000 iterasi. Hal ini menunjukkan bahwa model berhasil mengurangi error seiring berjalannya waktu dan mencapai konvergensi. Selanjutnya hasil *confusion matrix* untuk uji coba skenario 1 dapat dilihat pada tabel 4.8 sebagai berikut.

Tabel 4.8 *Confusion Matrix* Skenario 1

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay</i> (A)	<i>Graphics</i> (B)	<i>Story</i> (C)
<i>Gameplay</i> (A)	41	4	4
<i>Graphics</i> (B)	3	21	1
<i>Story</i> (C)	1	0	19

Confusion matrix menunjukkan bagaimana model memprediksi masing-masing kategori dalam data pengujian. Berikut adalah detail hasil *confusion matrix*:

1. ***Gameplay***: Model dengan akurat memprediksi 41 sampel sebagai *Gameplay*, dengan hanya 4 sampel yang terklasifikasi salah sebagai *Graphics* dan 4 sebagai *Story/Lore*.
2. ***Graphics***: Model memprediksi 21 sampel dengan benar sebagai *Graphics*, tetapi 3 sampel salah diklasifikasikan sebagai *Gameplay* dan 1 sebagai *Story/Lore*.
3. ***Story/Lore***: Model berhasil memprediksi 19 sampel sebagai *Story/Lore* dengan 1 sampel yang salah terklasifikasikan sebagai *Gameplay* dan 0 salah terklasifikasikan sebagai *Graphics*.

Selanjutnya ialah hasil *presicion*, *recall*, dan *f1-score* untuk masing – masing kelas beserta hasil akurasi model untuk hasil uji coba skenario 1 ditunjukkan pada tabel 4.9 sebagai berikut.

Tabel 4.9 Hasil Uji Coba Skenario 1

Label / Ground Truth	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
<i>Gameplay</i>	0,92	0,86	0,89
<i>Graphics</i>	0,84	0,84	0,84
<i>Story / Lore</i>	0,79	0,95	0,86
<i>Accuracy</i>	0,87		

Berdasarkan Tabel 4.3 yang disediakan, berikut adalah hasil evaluasi untuk masing-masing kategori:

1. ***Gameplay***: Model memiliki *precision* 0,92, *recall* 0,86, dan *f1-score* 0,89. Hal ini menunjukkan bahwa model cukup baik dalam memprediksi kategori *gameplay*, dengan jumlah prediksi yang benar lebih banyak dibandingkan kesalahan.
2. ***Graphics***: *Precision* untuk *Graphics* adalah 0,84, *recall* 0,84, dan *f1-score* 0,84. Meskipun akurasi prediksi sedikit lebih rendah dibandingkan dengan *gameplay*, model masih memberikan hasil yang memadai untuk kategori ini.
3. ***Story/Lore***: Untuk kategori *Story/Lore*, model memiliki *precision* 0,79, *recall* 0,95, dan *f1-score* 0,86. Model menunjukkan *recall* yang sangat tinggi (95%), yang berarti model sangat baik dalam mendeteksi *Story/Lore* meskipun jumlah *precision*-nya sedikit lebih rendah.

Rata-rata *Accuracy* model adalah 0,86, yang menunjukkan performa keseluruhan yang sangat baik dengan tingkat error yang rendah.

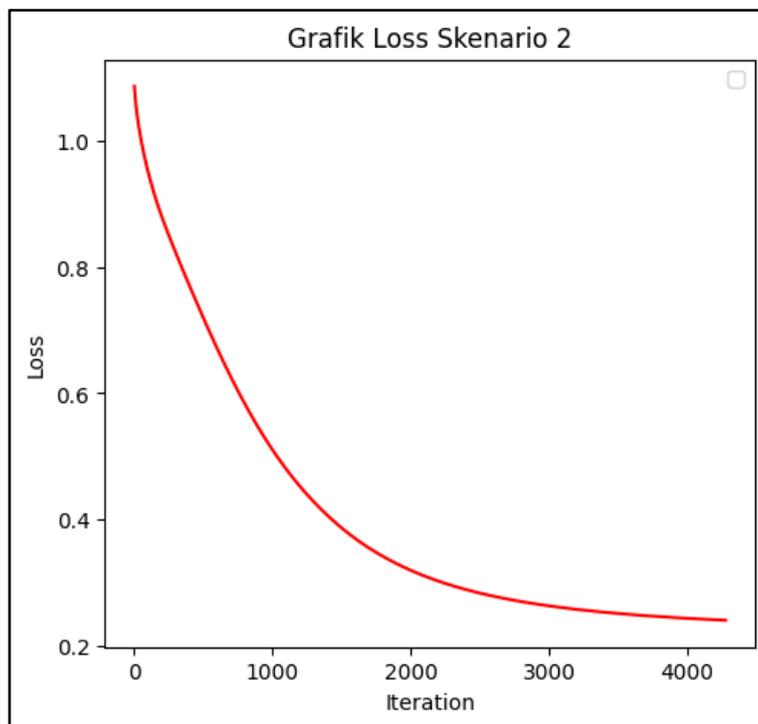
4.3.2 Hasil Uji Coba Skenario 2

Pada Skenario 2, , model *neural network* diuji dengan rasio 80% data untuk pelatihan dan 20% untuk pengujian, dengan *learning rate* (η) = 0,01 dan toleransi error = $1e-5$. Untuk Jumlah data untuk tiap-tiap kelas pada skenario 2 ditunjukkan pada tabel 4.10,

Tabel 4.10 Jumlah Data Tiap Kelas Skenario 2

Label / <i>Ground Truth</i>	Data <i>Training</i>	Data <i>Testing</i>
<i>Gameplay</i>	381	112
<i>Graphics</i>	256	53
<i>Story</i>	171	38
Total	808	203

Berdasarkan hasil pengujian, model menunjukkan performa yang baik dalam mengklasifikasikan ulasan *game* ke dalam tiga kategori utama, yaitu *gameplay*, *Graphics*, dan *Story/lore*. Dimulai dari nilai *loss* yang cukup tinggi, grafik menunjukkan kurva yang sangat tajam menurun pada awal iterasi dan kemudian stabil menuju nilai yang lebih rendah setelah mencapai sekitar 2000 iterasi. Penurunan nilai *loss* ini menunjukkan bahwa model berhasil melakukan pembelajaran dengan sangat baik dalam proses pelatihan, dimana kemampuan model untuk memperbaiki prediksinya terhadap data uji semakin meningkat. Hal ini dapat dilihat pada grafik *loss* Gambar 4.2.

Gambar 4.2 Grafik *Loss* Skenario 2

Grafik *training loss* menunjukkan penurunan *loss* yang signifikan seiring bertambahnya iterasi. *Loss* dimulai di sekitar 1,0 dan turun mendekati 0,2 setelah sekitar 4000 iterasi. Ini menunjukkan bahwa model berhasil belajar dari data pelatihan dan menurunkan error secara bertahap. Keberhasilan dalam reduksi *loss* ini menandakan bahwa model semakin mengoptimalkan bobot selama pelatihan. Hal ini mengindikasikan bahwa model telah belajar dengan efektif untuk membedakan dan mengklasifikasikan ulasan *game* berdasarkan fitur yang relevan dengan masing-masing kategori. Semakin kecil nilai *loss* yang tercatat, semakin baik pula kemampuan model dalam meminimalkan error dan kesalahan prediksi.

Selanjutnya hasil *confusion matrix* untuk uji coba skenario 2 dapat dilihat pada Tabel 4.11 sebagai berikut.

Tabel 4.11 *Confusion Matrix* Skenario 2

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay</i> (A)	<i>Graphics</i> (B)	<i>Story</i> (C)
<i>Gameplay</i> (A)	103	5	4
<i>Graphics</i> (B)	7	44	3
<i>Story</i> (C)	4	0	34

Confusion matrix memberikan gambaran yang lebih jelas mengenai prediksi model untuk masing-masing kategori. Berikut adalah penjelasan dari *confusion matrix*:

1. ***Gameplay***: Model berhasil memprediksi 103 sampel sebagai *gameplay* dengan 5 kesalahan yang terklasifikasikan sebagai *Graphics* dan 4 kesalahan sebagai *Story/lore*.
2. ***Graphics***: 44 sampel berhasil diprediksi dengan benar sebagai *Graphics*, namun 7 sampel salah terklasifikasikan sebagai *gameplay* dan 3 sampel sebagai *Story/lore*.
3. ***Story/Lore***: 34 sampel berhasil diprediksi dengan benar sebagai *Story/Lore*, dengan 4 sampel yang salah terklasifikasikan sebagai *gameplay* dan 0 yang salah terklasifikasikan sebagai *Graphics*.

Secara keseluruhan, model lebih baik dalam mengklasifikasikan kategori *gameplay* dan *Story/lore* dibandingkan *Graphics*, di mana ada kesalahan dalam mengklasifikasikan *Graphics* sebagai *gameplay*. Ini mungkin menunjukkan bahwa model kesulitan membedakan antara kedua kategori ini dalam beberapa kasus.

Selanjutnya ialah hasil *presicion*, *recall*, dan *F1-score* untuk masing – masing kelas beserta hasil akurasi model untuk hasil uji coba skenario 2 ditunjukkan pada tabel 4.12 sebagai berikut.

Tabel 4.12 Hasil Uji Coba Skenario 2

Label / Ground Truth	Precision	Recall	F1-score
<i>Gameplay</i>	0,90	0,92	0,91
<i>Graphics</i>	0,90	0,83	0,86
<i>Story / Lore</i>	0,85	0,89	0,87
<i>Accuracy</i>	0,89		

Berdasarkan Tabel 4.12, berikut adalah performa model untuk setiap kategori:

1. **Gameplay:** Model memiliki *precision* sebesar 0,90, *recall* 0,92, dan *f1-score* 0,91. Angka ini menunjukkan bahwa model cukup efektif dalam memprediksi kategori *gameplay* dengan *precision* yang tinggi, tetapi masih ada beberapa kesalahan prediksi.
2. **Graphics:** *Precision* untuk kategori *Graphics* adalah 0,90, dengan *recall* 0,83 dan *f1-score* 0,86. Model menunjukkan performa yang sedikit lebih rendah dalam memprediksi *Graphics*, dengan *precision* yang cukup baik meskipun ada kesalahan.
3. **Story/Lore:** Model memiliki *precision* 0,85, *recall* 0,89, dan *f1-score* 0,87. Model menunjukkan *recall* yang tinggi (89%), yang berarti model sangat efektif dalam mendeteksi *Story/lore*, meskipun *precision* agak rendah.

Secara keseluruhan, *Accuracy* model adalah 0,89, yang menunjukkan bahwa model dapat mengenali kategori-kategori utama dengan cukup baik, meskipun ada sedikit kesalahan pada kategori *Graphics*.

4.3.3 Hasil Uji Coba Skenario 3

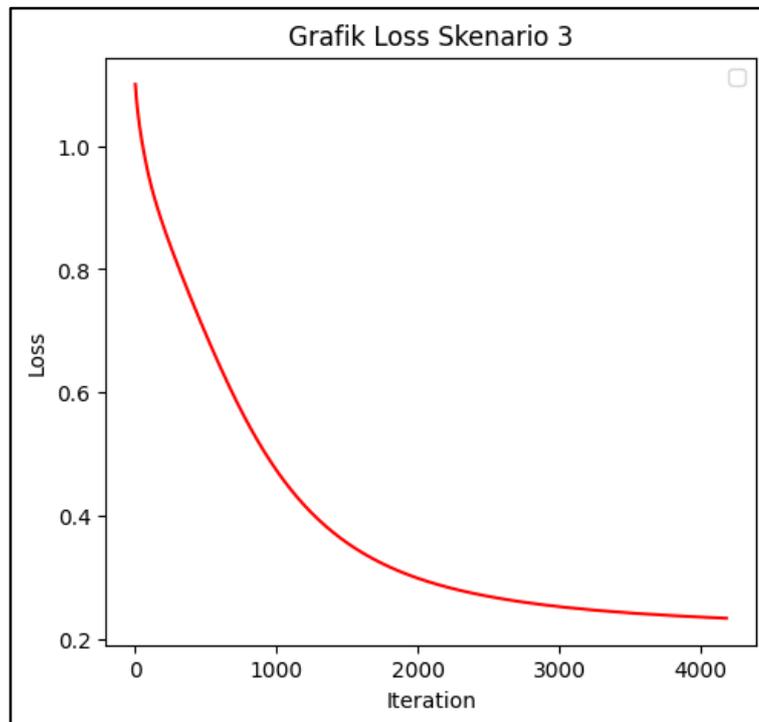
Pada Skenario 3, model *Neural network* diuji dengan rasio 70% data untuk pelatihan dan 30% untuk pengujian, *learning rate* (η) = 0,01, dan toleransi error =

1e-5. Untuk Jumlah data untuk tiap-tiap kelas pada skenario 3 ditunjukkan pada tabel 4.13.

Tabel 4.13 Jumlah Data Tiap Kelas Skenario 3

Label / Ground Truth	Data Training	Data Testing
<i>Gameplay</i>	337	156
<i>Graphics</i>	227	82
<i>Story</i>	143	66
Total	707	304

Hasil pengujian menunjukkan bahwa model masih mampu memberikan performa yang baik. Pada awal pelatihan, grafik menunjukkan penurunan tajam pada nilai *loss*, yang kemudian melambat dan lebih stabil saat mencapai sekitar 2000 iterasi. Model tetap dapat menurunkan nilai *loss* dengan efektif. Ini menunjukkan bahwa meskipun proses pembelajaran sedikit lebih lambat, model masih mampu mengoptimalkan performanya dengan baik tanpa *overfitting*, yang sering terjadi ketika *learning rate* terlalu besar. Penurunan *loss* dilihat pada grafik *loss* Gambar 4.3.



Gambar 4.3 Grafik *Loss* Skenario 3

Grafik *training loss* menunjukkan penurunan *loss* yang konsisten sepanjang iterasi. Pada awal pelatihan, grafik menunjukkan penurunan tajam pada nilai *loss*, yang kemudian melambat dan lebih stabil saat mencapai sekitar 2000 dan terus menurun hingga mendekati 0,2 setelah sekitar 4000 iterasi. Proses pelatihan menunjukkan bahwa model mengurangi error secara signifikan selama pelatihan. Hasil ini memberikan indikasi positif bahwa model tetap efektif dalam menangani data dan dapat menyelesaikan tugas klasifikasi dengan baik.

Selanjutnya hasil *confusion matrix* untuk uji coba skenario 3 dapat dilihat pada Tabel 4.14 sebagai berikut.

Tabel 4.14 *Confusion Matrix* Skenario 3

Kelas Aktual/<i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay</i> (A)	<i>Graphics</i> (B)	<i>Story</i> (C)
<i>Gameplay</i> (A)	145	7	4
<i>Graphics</i> (B)	7	74	1
<i>Story</i> (C)	5	6	55

Confusion matrix memberikan gambaran lebih jelas tentang kinerja model dalam mengklasifikasikan masing-masing kategori. Berikut adalah penjelasan dari *confusion matrix*:

1. ***Gameplay***: Model dengan benar memprediksi 145 sampel sebagai *Gameplay* dengan hanya 7 kesalahan yang terklasifikasikan sebagai *Graphics* dan 4 kesalahan sebagai *Story/Lore*.
2. ***Graphics***: Model berhasil memprediksi 74 sampel dengan benar sebagai *Graphics*, namun 7 sampel salah diklasifikasikan sebagai *Gameplay* dan 1 sampel sebagai *Story/Lore*.
3. ***Story/Lore***: Model berhasil memprediksi 55 sampel dengan benar sebagai *Story/Lore*, dengan hanya 5 sampel yang salah terklasifikasikan sebagai *Gameplay* dan 6 sampel sebagai *Graphics*.

Secara keseluruhan, model lebih akurat dalam mengklasifikasikan *Gameplay* dan *Story/Lore*, dengan kesalahan lebih banyak terjadi pada kategori *Graphics*. Meskipun demikian, hasilnya tetap menunjukkan performansi yang cukup baik, dan kesalahan tersebut relatif kecil.

Selanjutnya ialah hasil *presicion*, *recall*, dan *f1-score* untuk masing – masing kelas beserta hasil akurasi model untuk hasil uji coba skenario 3 ditunjukkan pada tabel 4.15 sebagai berikut.

Tabel 4.15 Hasil Uji Coba Skenario 3

Label / Ground Truth	<i>Presicion</i>	<i>Recall</i>	<i>F1-score</i>
<i>Gameplay</i>	0,92	0,93	0,93
<i>Graphics</i>	0,85	0,90	0,88
<i>Story / Lore</i>	0,92	0,83	0,87
<i>Accuracy</i>	0,90		

Berdasarkan tabel hasil uji coba scenario 3, hasil evaluasi model adalah sebagai berikut:

1. ***Gameplay***: Model memiliki *precision* 0,92, *recall* 0,93, dan *f1-score* 0,93. Model sangat akurat dalam mengklasifikasikan kategori *Gameplay*, dengan *precision* yang sangat tinggi dan *recall* yang juga baik.
2. ***Graphics***: *Precision* untuk kategori *Graphics* adalah 0,85, *recall* 0,90, dan *f1-score* 0,88. Meskipun model masih memadai dalam memprediksi *Graphics*, ada sedikit penurunan dibandingkan dengan *Gameplay*.
3. ***Story/Lore***: Untuk kategori *Story/Lore*, model memiliki *precision* 0,92, *recall* 0,83, dan *f1-score* 0,87. *Recall* yang tinggi (83%) menunjukkan bahwa model baik dalam mendeteksi *Story/Lore* dengan *precission* yang tinggi.

Dengan *Accuracy* 0,90, model ini menunjukkan performa yang baik secara keseluruhan.

4.3.4 Hasil Uji Coba Skenario 4

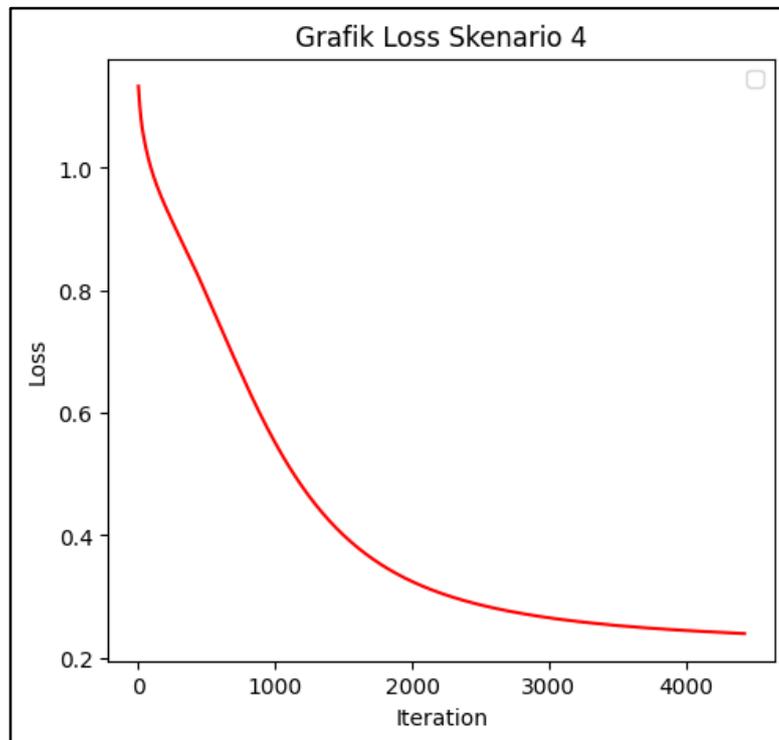
Pada Skenario 4, model *Neural network* diuji dengan rasio 60% data untuk pelatihan dan 40% untuk pengujian, *learning rate* (η) = 0,01, dan toleransi error =

1e-5. Untuk Jumlah data untuk tiap-tiap kelas pada skenario 3 ditunjukkan pada tabel 4.16.

Tabel 4.16 Jumlah Data Tiap Kelas Skenario 4

Label / Ground Truth	Data Training	Data Testing
<i>Gameplay</i>	289	204
<i>Graphics</i>	194	115
<i>Story</i>	123	86
Total	606	405

Hasil pengujian ini menunjukkan bahwa model masih memberikan performa yang cukup baik dengan akurasi yang tinggi dan *loss* yang relatif rendah. Pada awal pelatihan, grafik menunjukkan penurunan tajam pada nilai *loss*, yang kemudian melambat. Meskipun terjadi penurunan yang lebih halus setelah beberapa iterasi, model menunjukkan kemampuan yang stabil dalam melakukan **optimisasi**. Nilai *loss* yang semakin rendah menunjukkan bahwa model berhasil mengklasifikasikan ulasan *game* ke dalam kategori yang ditentukan. Grafik *loss* skenario 4 ditunjukkan pada Gambar 4.4.



Gambar 4.4 Grafik *Loss* Skenario 4

Pada grafik *training loss*, kita dapat melihat penurunan *loss* yang bertahap seiring berjalannya proses pelatihan model. Pada awalnya, nilai *loss* dimulai sekitar 1.1, dan secara bertahap turun mendekati 0,2 setelah sekitar 4000 iterasi. Penurunan yang konsisten ini menunjukkan bahwa model berhasil melakukan pembelajaran dengan baik, mengurangi kesalahan prediksi secara bertahap. Hal ini mengindikasikan bahwa model mampu mempelajari pola-pola data dengan efektif dan semakin mengurangi kesalahan klasifikasi yang terjadi selama pelatihan. Penurunan *loss* yang stabil juga menunjukkan bahwa model tidak mengalami *overfitting*, dan seiring berjalannya iterasi, model menjadi lebih terlatih dan lebih mampu mengklasifikasikan ulasan *game* ke dalam kategori yang ditetapkan dengan akurat.

Selanjutnya hasil *confusion matrix* untuk uji coba skenario 3 dapat dilihat pada Tabel 4.17 sebagai berikut.

Tabel 4.17 *Confusion Matrix* Skenario 4

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay</i> (A)	<i>Graphics</i> (B)	<i>Story</i> (C)
<i>Gameplay</i> (A)	191	8	5
<i>Graphics</i> (B)	10	102	3
<i>Story</i> (C)	6	7	73

Confusion matrix menggambarkan hasil prediksi model untuk masing-masing kategori. Berikut adalah detail hasil *confusion matrix*:

1. ***Gameplay***: Model memprediksi 191 sampel dengan benar sebagai *gameplay*. Namun, terdapat 8 kesalahan yang terklasifikasikan sebagai *Graphics* dan 5 kesalahan sebagai *Story/lore*.
2. ***Graphics***: Model memprediksi 102 sampel dengan benar sebagai *Graphics*, meskipun ada 10 kesalahan yang terklasifikasikan sebagai *gameplay* dan 3 kesalahan sebagai *Story/Lore*.
3. ***Story/Lore***: Model memprediksi 73 sampel dengan benar sebagai *Story/Lore*, dengan hanya 6 kesalahan yang terklasifikasikan sebagai *Gameplay* dan 7 kesalahan untuk *Graphics*.

Secara keseluruhan, model berhasil mengklasifikasikan kategori *Gameplay* dengan sangat baik, namun ada kesalahan dalam mengklasifikasikan *Graphics* sebagai *Gameplay*.

Selanjutnya ialah hasil *presicion*, *recall*, dan *f1-score* untuk masing – masing kelas beserta hasil akurasi model untuk hasil uji coba skenario 4 ditunjukkan pada Tabel 4.18 sebagai berikut.

Tabel 4.18 Hasil Uji Coba Skenario 4

Label / Ground Truth	Precision	Recall	F1-score
<i>Gameplay</i>	0,92	0,94	0,93
<i>Graphics</i>	0,87	0,89	0,88
<i>Story / Lore</i>	0,90	0,90	0,90
<i>Accuracy</i>	0,90		

Berdasarkan tabel hasil uji coba skenario 4, hasil evaluasi model adalah sebagai berikut:

1. **Gameplay:** Model menunjukkan *precision* sebesar 0,92, *recall* 0,94, dan *f1-score* 0,93, yang menunjukkan bahwa model sangat efektif dalam mengenali *Gameplay*, dengan *precision* dan *recall* yang sangat tinggi.
2. **Graphics:** *Precision* untuk *Graphics* adalah 0,87, *recall* 0,89, dan *f1-score* 0,88, yang menunjukkan performa yang baik meskipun sedikit lebih rendah daripada *Gameplay*. Kesalahan sebagian besar terjadi ketika *Graphics* diklasifikasikan sebagai *Gameplay*.
3. **Story/Lore:** Untuk kategori *Story/Lore*, model memiliki *precision* 0,90, *recall* 0,90, dan *f1-score* 0,90, menunjukkan bahwa model sangat baik dalam mendeteksi *Story/Lore*, dengan *recall* dan *precision* yang tinggi.

Rata-rata *Accuracy* model adalah 0,90, yang menunjukkan bahwa model secara keseluruhan dapat mengklasifikasikan ulasan *game* dengan baik.

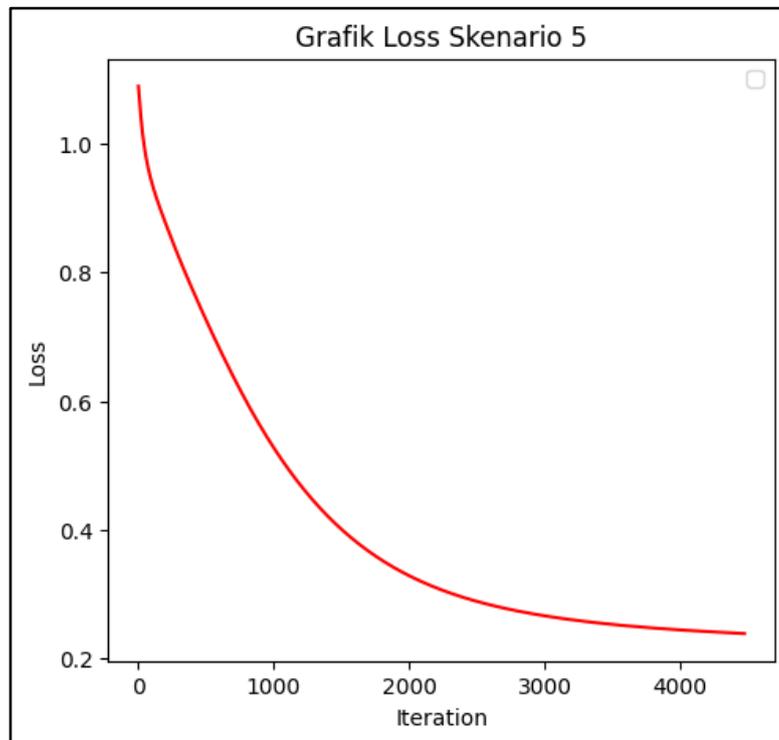
4.3.5 Hasil Uji Coba Skenario 5

Pada Skenario 5, model *Neural network* diuji dengan rasio 50% data untuk pelatihan dan 50% untuk pengujian, *learning rate* (η) = 0,01, dan toleransi error = $1e-5$. Untuk Jumlah data untuk tiap-tiap kelas pada skenario 5 ditunjukkan pada Tabel 4.19.

Tabel 4.19 Jumlah Data Tiap Kelas Skenario 5

Label / Ground Truth	Data Training	Data Testing
<i>Gameplay</i>	236	257
<i>Graphics</i>	163	146
<i>Story</i>	106	103
Total	505	506

Hasil pengujian ini menunjukkan bahwa model masih memberikan performa yang cukup baik dengan akurasi yang tinggi dan *loss* yang relatif rendah. Dimulai dengan nilai *loss* yang cukup tinggi, model menunjukkan penurunan yang tajam pada awal iterasi, kemudian secara bertahap mendekati nilai yang lebih rendah seiring dengan bertambahnya jumlah iterasi. Meskipun tidak secepat penurunan pada skenario lainnya, penurunan yang lebih halus pada grafik ini tetap menunjukkan bahwa model berhasil belajar dengan kinerja yang baik, mengurangi kesalahan klasifikasi secara bertahap. Dengan nilai *loss* yang semakin mendekati 0,2, model berhasil mencapai performa yang optimal dalam mengklasifikasikan ulasan *game* ke dalam kategori *Gameplay*, *Graphics*, dan *Story/Lore*. Grafik *loss* skenario 5 ditunjukkan pada Gambar 4.5.

Gambar 4.5 Grafik *Loss* Skenario 5

Pada grafik *training loss*, kita dapat mengamati penurunan *loss* yang bertahap sejak awal pelatihan. Pada iterasi awal, nilai *loss* dimulai sekitar **1.1**, kemudian secara perlahan menurun dan mendekati 0,2 setelah sekitar 4000 iterasi. Penurunan yang stabil ini menunjukkan bahwa model berhasil belajar secara bertahap. Meskipun penurunan nilai *loss* tidak setajam pada beberapa skenario sebelumnya, grafik ini tetap menunjukkan bahwa model mampu mengurangi kesalahan prediksi dengan efektif. Hal ini menandakan bahwa meskipun proses pelatihan berlangsung lebih lambat, model berhasil mempelajari pola data dengan baik, menyesuaikan parameter secara optimal untuk meningkatkan akurasi prediksi.

Selanjutnya hasil *confusion matrix* untuk uji coba skenario 5 dapat dilihat pada Tabel 4.20 sebagai berikut.

Tabel 4.20 *Confusion Matrix* Skenario 5

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay</i> (A)	<i>Graphics</i> (B)	<i>Story</i> (C)
<i>Gameplay</i> (A)	190	9	5
<i>Graphics</i> (B)	10	101	4
<i>Story</i> (C)	7	7	72

Confusion matrix menggambarkan hasil prediksi model untuk masing-masing kategori. Berikut adalah detail hasil *confusion matrix*:

1. ***Gameplay***: Model memprediksi 190 sampel dengan benar sebagai *Gameplay*. Namun, terdapat 9 kesalahan yang terklasifikasikan sebagai *Graphics* dan 5 kesalahan sebagai *Story/Lore*.
2. ***Graphics***: Model memprediksi 101 sampel dengan benar sebagai *Graphics*, meskipun ada 10 kesalahan yang terklasifikasikan sebagai *Gameplay* dan 4 kesalahan sebagai *Story/Lore*.
3. ***Story/Lore***: Model memprediksi 72 sampel dengan benar sebagai *Story/Lore*, dengan hanya 7 kesalahan yang terklasifikasikan sebagai *Gameplay* dan 7 kesalahan untuk *Graphics*.

Secara keseluruhan, model berhasil mengklasifikasikan kategori *Gameplay* dengan sangat baik, namun ada sedikit kesalahan dalam mengklasifikasikan *Graphics* sebagai *Gameplay*. Model juga menunjukkan performa yang baik dalam mengidentifikasi *Story/Lore*.

Selanjutnya ialah hasil *presicion*, *recall*, dan *f1-score* untuk masing – masing kelas beserta hasil akurasi model untuk hasil uji coba skenario 5 ditunjukkan pada Tabel 4.21 sebagai berikut.

Tabel 4.21 Hasil Uji Coba Skenario 5

Label / Ground Truth	Precision	Recall	F1-score
<i>Gameplay</i>	0,92	0,93	0,92
<i>Graphics</i>	0,86	0,88	0,87
<i>Story / Lore</i>	0,89	0,84	0,86
<i>Accuracy</i>	0,89		

Berdasarkan *classification report*, hasil evaluasi model adalah sebagai berikut:

1. **Gameplay:** Model memiliki *precision* sebesar 0,92, *recall* 0,93, dan *f1-score* 0,92. Model ini menunjukkan kinerja yang sangat baik dalam mengklasifikasikan *Gameplay* dengan *precision* dan *recall* yang tinggi.
2. **Graphics:** *Precision* untuk *Graphics* adalah 0,86, *recall* 0,88, dan *f1-score* 0,87. Meskipun akurasi model untuk *Graphics* masih baik, ada penurunan *recall* yang mungkin terkait dengan kesalahan dalam mengklasifikasikan *Graphics* sebagai *Gameplay*.
3. **Story/Lore:** Untuk kategori *Story/Lore*, model memiliki *precision* 0,89, *recall* 0,84, dan *f1-score* 0,86, menunjukkan bahwa model memiliki *recall* yang mampu untuk mendeteksi *Story/Lore* dengan performa yang baik.

Secara keseluruhan, model menunjukkan akurasi 0,89, yang menunjukkan bahwa model secara keseluruhan dapat mengklasifikasikan ulasan *game* dengan baik.

4.4 Pembahasan

Setelah melakukan uji coba pada berbagai skenario yang telah ditentukan, dilakukan pembahasan terhadap hasil evaluasi yang diperoleh dari eksperimen yang telah dilakukan dengan menggunakan berbagai skenario pengujian. Masing-masing

skenario menguji pengaruh dari rasio pembagian data yang berbeda (90:10, 80:20, 70:30, 60:40, dan 50:50) terhadap performa model dalam mengklasifikasikan aspek-aspek dalam ulasan *game*, yang meliputi *Gameplay*, *Graphics*, dan *Story/Lore*. Evaluasi dilakukan dengan menggunakan beberapa metrik, seperti *precision*, *recall*, *F1-score*, dan *Accuracy*, yang bertujuan untuk menilai seberapa baik model dapat mengenali dan mengklasifikasikan setiap label tersebut.

Pada pembahasan ini, hasil-hasil dari tiap skenario akan dianalisis untuk mengidentifikasi kekuatan dan kelemahan model pada masing-masing label dan skenario. Perbandingan antar skenario juga akan dilakukan untuk memahami bagaimana perubahan rasio pembagian data mempengaruhi kinerja model secara keseluruhan. Rata-rata hasil evaluasi tiap skenario di tunjukkan pada Tabel 4.22.

Tabel 4.22 Rata-Rata Hasil Uji Coba Skenario Pengujian

Skenario	Precision	Recall	F1-score	Accuracy
1 (90:10)	0,85	0,88	0,86	0,87
2 (80:20)	0,88	0,88	0,88	0,89
3 (70:30)	0,89	0,88	0,89	0,90
4 (60:40)	0,89	0,91	0,90	0,90
5 (50:50)	0,89	0,88	0,88	0,89

Dalam analisis hasil evaluasi model berdasarkan lima skenario pengujian, terdapat perubahan pada masing-masing metrik yang digunakan.

Pada metrik *precision*, terjadi perubahan dari skenario 1 ke skenario 2, dengan peningkatan sebesar 3,53%. Setelah itu, perubahan pada skenario 2 ke skenario 3 dengan peningkatan sebesar 1.14%. skenario 4 dan skenario 5 dengan tidak ada perubahan pada dua skenario terakhir.

Untuk *recall*, nilainya relatif stabil antara skenario 1 hingga skenario 3, dengan nilai yang tetap pada 0,88. Namun, pada skenario 4 (60:40), *Recall*

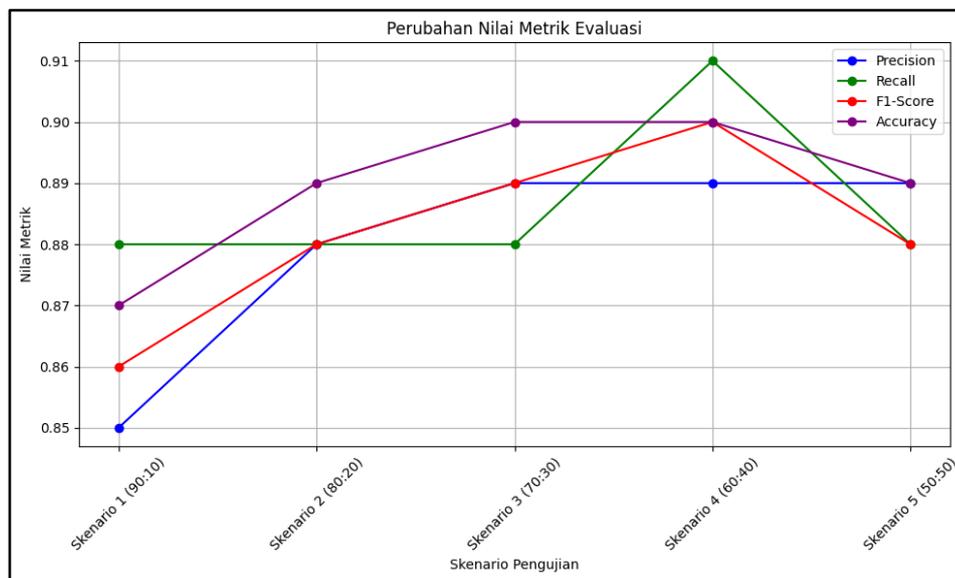
mengalami peningkatan sebesar 3,41%, mencapai 0,91. Meskipun begitu, pada skenario 5 (50:50), terjadi penurunan kecil sebesar 3,29%, kembali ke nilai 0,88.

Pada *F1-score*, yang merupakan gabungan antara *precision* dan *recall*, terjadi perubahan bertahap hingga skenario 4. Terjadi peningkatan dari skenario 1 ke skenario 2 sebesar 2,33%. meningkat lagi sebesar 1,14% dari skenario 2 ke skenario 3, serta peningkatan terakhir sebesar 1,12% dari skenario 3 ke skenario 4 dengan nilai mencapai 0,90, Namun, pada skenario 5, *F1-score* mengalami penurunan sebesar 2,22%, kembali ke nilai 0,88. Peningkatan yang stabil pada skenario awal menunjukkan performa yang semakin baik, namun penurunan di skenario 5.

Untuk *accuracy*, ada perubahan yang terjadi di beberapa skenario pengujian. Peningkatan sebesar 2,30% terjadi dari skenario 1 ke skenario 2. Meningkat lagi dari skenario 2 ke skenario 3 sebesar 1.12% dan tidak ada perubahan antara skenario 3 dan 4. Namun, antara skenario 3 dan skenario 4, *Accuracy* tetap pada nilai 0,90, tanpa perubahan lebih lanjut. Pada skenario 5, terjadi penurunan kecil sebesar 1,11%, kembali menjadi 0,89. Model skenario 4 (60:40) menunjukkan kinerja terbaik dibandingkan dengan model lainnya, karena pembagian data yang memungkinkan model untuk belajar cukup banyak dari data pelatihan, namun tetap memiliki cukup data untuk menguji performanya pada data yang tidak terlihat sebelumnya. Pembagian data yang cukup ini mengurangi risiko *overfitting*, di mana model terlalu terfokus pada data pelatihan dan kurang mampu menangkap pola pada data uji. Dengan pembagian data yang lebih merata, model ini lebih mampu menangkap pola yang lebih umum, yang tercermin pada *precision*,

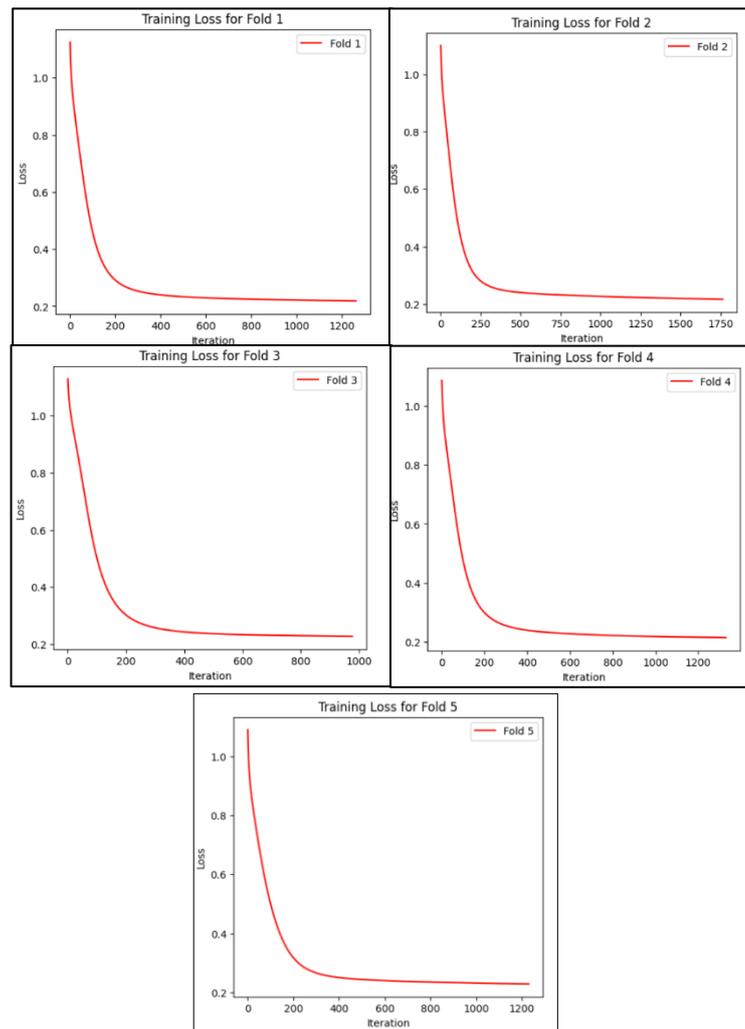
recall, F1-score, dan accuracy yang lebih tinggi. Sebaliknya, model 1 (90:10) memiliki pembagian yang lebih berat pada data pelatihan (90%) dan sangat sedikit untuk pengujian (10%) sehingga cenderung untuk *overfitting*. Model ini cenderung terlalu menghafal data pelatihan dan tidak dapat mengaplikasikan pola-pola tersebut dengan baik pada data uji, yang mengakibatkan performa pada accuracy dan F1-score yang lebih rendah dibanding model lain.

Model skenario 2 (80:20) dan Model skenario 3 (70:30) menunjukkan perbaikan dibandingkan dengan model 1. Pembagian data yang memberikan model kesempatan lebih besar untuk mempelajari data dengan lebih baik, meskipun model 2 masih rentan terhadap *overfitting*. Model 3 dengan pembagian yang lebih optimal menunjukkan peningkatan pada precision dan recall, mengurangi risiko *overfitting*, dan memberikan hasil yang lebih konsisten pada data uji dan pelatihan. Di sisi lain, Model skenario 5 (50:50) memiliki kinerja yang kurang karena pembagian yang terlalu banyak untuk pelatihan, yang menyebabkan model cenderung *underfit* dan gagal menangkap pola dalam data, sehingga menghasilkan precision dan F1-score yang lebih rendah. Pembagian data yang lebih banyak untuk pengujian, meskipun baik untuk menguji performa model, mengurangi jumlah data pelatihan yang tersedia untuk belajar, sehingga model tidak cukup terlatih untuk menangani data dengan baik. Grafik perubahan ditunjukkan pada Gambar 4.6..



Gambar 4.6 Grafik Perubahan Nilai Metrik Evaluasi

Penelitian ini juga menerapkan teknik *k-fold cross-validation*, yang bertujuan untuk memastikan keandalan dan akurasi model dalam berbagai kondisi dengan melatih dan menguji model secara berulang menggunakan pembagian data yang berbeda pada setiap iterasi. Metode ini membantu mengurangi *overfitting* dan *varians* dalam estimasi kinerja model dengan memanfaatkan beberapa subset data untuk pelatihan dan pengujian. Dalam penelitian ini, nilai $k = 5$ dipilih, yang berarti data dibagi menjadi lima bagian yang berbeda, di mana model dilatih dan diuji sebanyak lima kali, masing-masing dengan kombinasi data latih dan data uji yang berbeda. Pendekatan ini sangat efektif untuk model klasifikasi yang menggunakan dataset dengan ukuran sekitar 1000 data, seperti yang digunakan dalam penelitian ini (Jain & Kumar, 2024). Dengan menggunakan *k-fold cross-validation*, kita dapat memperoleh estimasi kinerja yang lebih stabil dan terhindar dari ketergantungan pada pembagian data tertentu. Grafik *loss* tiap-tiap *fold* ditampilkan pada Gambar 4.7.



Gambar 4.7 Grafik Loss K-Fold

Pada masing-masing grafik, kita dapat melihat bahwa *training loss* berkurang secara signifikan pada iterasi pertama hingga beberapa iterasi berikutnya. Hal ini menandakan bahwa model belajar dengan cepat pada awalnya. Variasi yang ada antara tiap *fold* dapat menunjukkan bagaimana pembagian data mempengaruhi kemampuan model dalam melakukan generalisasi. Jika ada *fold* yang mengalami penurunan *training loss* yang lebih cepat atau lebih lambat, hal ini bisa disebabkan oleh perbedaan karakteristik data yang ada pada *fold* tersebut, meskipun secara keseluruhan model tetap menunjukkan kemampuan yang baik

dalam mengurangi kesalahan prediksi. Untuk *confusion matrix* model tiap *fold* ditunjukkan pada Tabel 4.23, Tabel 4.24, Tabel 4.25, Tabel 4.26, dan Tabel 4.27.

Tabel 4.23 *Confusion Matrix Fold 1*

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay (A)</i>	<i>Graphics (B)</i>	<i>Story (C)</i>
<i>Gameplay (A)</i>	101	6	5
<i>Graphics (B)</i>	7	44	2
<i>Story (C)</i>	4	0	34

Tabel 4.24 *Confusion Matrix Fold 2*

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay (A)</i>	<i>Graphics (B)</i>	<i>Story (C)</i>
<i>Gameplay (A)</i>	85	5	2
<i>Graphics (B)</i>	2	57	3
<i>Story (C)</i>	3	5	40

Tabel 4.25 *Confusion Matrix Fold 3*

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay (A)</i>	<i>Graphics (B)</i>	<i>Story (C)</i>
<i>Gameplay (A)</i>	89	8	2
<i>Graphics (B)</i>	0	60	1
<i>Story (C)</i>	2	4	36

Tabel 4.26 *Confusion Matrix Fold 4*

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay (A)</i>	<i>Graphics (B)</i>	<i>Story (C)</i>
<i>Gameplay (A)</i>	85	3	2
<i>Graphics (B)</i>	9	61	1
<i>Story (C)</i>	2	2	37

Tabel 4.27 *Confusion Matrix Fold 5*

Kelas Aktual/ <i>Ground Truth</i>	Kelas Prediksi		
	<i>Gameplay (A)</i>	<i>Graphics (B)</i>	<i>Story (C)</i>
<i>Gameplay (A)</i>	95	3	2
<i>Graphics (B)</i>	2	57	3
<i>Story (C)</i>	1	4	35

Pada *fold* pertama, model memiliki kinerja yang sangat baik dalam mengklasifikasikan kelas "*Gameplay (A)*" dan "*Graphics (B)*" dengan jumlah

prediksi yang sangat tepat. Misalnya, untuk kelas "*Gameplay (A)*", 101 data berhasil diklasifikasikan dengan benar, dengan hanya 6 prediksi yang salah ke kelas "*Graphics (B)*" dan 5 ke kelas "*Story (C)*". Kelas "*Graphics (B)*" mengalami sedikit kebingungan, di mana 7 data dari kelas ini diprediksi ke kelas "*Gameplay (A)*", tetapi sisanya (44) berhasil diprediksi dengan benar. Kelas "*Story (C)*" memiliki prediksi yang sangat baik dengan 34 data yang berhasil diklasifikasikan dengan benar, meskipun terdapat beberapa kesalahan kecil (4 data diprediksi sebagai "*Gameplay (A)*" dan 0 sebagai "*Graphics (B)*").

Pada *fold* kedua, model masih menunjukkan kinerja yang sangat baik dalam klasifikasi, tetapi dengan sedikit penurunan dibandingkan *fold* pertama. Untuk kelas "*Gameplay (A)*", 85 data berhasil diklasifikasikan dengan benar, sementara 5 diprediksi sebagai "*Graphics (B)*" dan 2 sebagai "*Story (C)*". Kelas "*Graphics (B)*" memiliki performa yang baik, dengan 57 prediksi yang benar dan hanya 2 kesalahan ke kelas "*Gameplay (A)*" dan 3 kesalahan ke kelas "*Story (C)*". Untuk kelas "*Story (C)*", prediksi yang benar cukup baik, dengan 40 data yang diklasifikasikan dengan benar, meskipun ada sedikit kebingungan dengan beberapa data diprediksi sebagai "*Gameplay (A)*" (2 data) dan "*Graphics (B)*" (3 data).

Pada *fold* ketiga, performa model sedikit meningkat. Kelas "*Gameplay (A)*" memiliki 89 prediksi yang benar, dengan 8 kesalahan ke kelas "*Graphics (B)*" dan 2 kesalahan ke kelas "*Story (C)*". Kelas "*Graphics (B)*" mengalami prediksi yang sangat tepat, dengan 60 data yang diklasifikasikan dengan benar. Kelas "*Story (C)*" juga menunjukkan hasil yang baik dengan 36 data yang berhasil diklasifikasikan

dengan benar, sementara 2 kesalahan ke kelas "*Gameplay* (A)" dan 4 kesalahan ke kelas "*Graphics* (B)" masih terjadi.

Pada *fold* keempat, performa model cukup konsisten. Kelas "*Gameplay* (A)" berhasil diklasifikasikan dengan benar sebanyak 85 data, meskipun ada beberapa kesalahan prediksi ke kelas "*Graphics* (B)" (9 data) dan sedikit ke kelas "*Story* (C)" (2 data). Kelas "*Graphics* (B)" memiliki 61 data yang diklasifikasikan dengan benar, dengan hanya 9 kesalahan ke kelas "*Gameplay* (A)" dan 1 kesalahan ke kelas "*Story* (C)". Kelas "*Story* (C)" masih menunjukkan kinerja yang baik, dengan 37 data yang diklasifikasikan dengan benar, meskipun 2 data diprediksi ke kelas lain.

Pada *fold* kelima, model menunjukkan kinerja yang sangat baik di semua kelas. Kelas "*Gameplay* (A)" menunjukkan hasil terbaik dengan 95 prediksi yang benar, dengan hanya 3 kesalahan ke kelas "*Graphics* (B)" dan 2 kesalahan ke kelas "*Story* (C)". Kelas "*Graphics* (B)" memiliki 57 data yang berhasil diklasifikasikan dengan benar, dengan kesalahan yang sangat sedikit (2 ke kelas "*Gameplay* (A)" dan 3 ke kelas "*Story* (C)"). Kelas "*Story* (C)" memiliki 35 data yang diklasifikasikan dengan benar, dengan hanya sedikit kebingungan terhadap kelas lain. Untuk hasil rata-rata evaluasi model tiap *fold* ditunjukkan pada tabel 4.28.

Tabel 4.28 Rata-Rata Evaluasi Model Tiap *Fold*

<i>Fold</i>	Precision	Recall	F1-score	Accuracy
1	0,8825	0,8818	0,8817	0,8818
2	0,9025	0,9010	0,9011	0,9010
3	0,9229	0,9158	0,9164	0,9158
4	0,9071	0,9059	0,9056	0,9059
5	0,9265	0,9257	0,9260	0,9257

Berdasarkan hasil evaluasi model menggunakan *k-fold cross-validation* dengan $k = 5$, kita dapat melihat bahwa model menunjukkan peningkatan yang

konsisten di setiap *fold*. Secara umum, *precision*, *recall*, *F1-score*, dan *Accuracy* mengalami peningkatan dari *fold* 1 hingga *fold* 5. *Fold* 3 memiliki nilai tertinggi untuk sebagian besar metrik, dengan *Precision* 0,9229, *Recall* 0,9158, *F1-score* 0,9164, dan *Accuracy* 0,9158. Meskipun terdapat variasi antara setiap *fold*, model tetap menunjukkan kinerja yang relatif stabil, dengan *Precision*, *Recall*, *F1-score*, dan *Accuracy* berada dalam rentang yang cukup sempit, yaitu antara 0,8818 hingga 0,9265. Variasi yang sempit antara setiap *fold* menunjukkan bahwa model memiliki konsistensi yang baik dalam performanya. Artinya, model tidak terlalu terpengaruh oleh pembagian data pada tiap *fold*, yang menandakan bahwa model dapat beradaptasi dengan baik dengan baik terhadap data yang berbeda-beda. Hal ini juga menunjukkan bahwa model tidak mengalami *overfitting* pada data latih atau *underfitting* pada data uji, melainkan mampu mempertahankan kinerja yang stabil meskipun pada setiap *fold* data yang digunakan berbeda.

Dalam menentukan aspek ulasan *game* dari pengguna, terdapat beberapa karakteristik yang mendefinisikan aspek-aspek utama *game* tersebut. Aspek *gameplay* mencakup elemen-elemen seperti mekanik permainan, tingkat kesulitan, dan pengalaman yang diberikan kepada pemain. Aspek grafik menggambarkan kualitas visual, desain karakter, dan pengalaman visual secara keseluruhan. Sementara itu, aspek cerita (*Story/lore*) mencakup kedalaman alur cerita, karakter, dan hubungan cerita dengan *gameplay* itu sendiri. Masing-masing aspek ini memiliki karakteristik yang menggambarkan kualitas dan pengalaman yang sejalan dengan nilai-nilai Islam yang mengajarkan kesempurnaan, keseimbangan, dan kedalaman. Sejalan dengan hal ini, Al-Qur'an menjelaskan karakteristik-

karakteristik orang-orang beriman yang memiliki kualitas sebanding dengan karakteristik yang ada pada setiap aspek dalam *game*. Sebagaimana tercantum dalam Q.S Al-Baqarah (2:2-3):

ذَٰلِكَ الْكِتَابُ لَا رَيْبَ فِيهِ هُدًى لِّلْمُتَّقِينَ ﴿٢﴾ الَّذِينَ يُؤْمِنُونَ بِالْغَيْبِ وَيُقِيمُونَ الصَّلَاةَ وَمِمَّا رَزَقْنَاهُمْ يُنفِقُونَ ﴿٣﴾

"(Yaitu) mereka yang mendengarkan perkataan lalu mengikuti apa yang paling baik di antaranya. Mereka itulah orang-orang yang telah diberi petunjuk oleh Allah dan mereka itulah ululalbab (orang-orang yang mempunyai akal sehat)."(Q.S Az-Zumar: 18)"

Menurut Tafsir Wajiz Orang-orang yang bertakwa itu adalah mereka yang beriman kepada hal-hal yang gaib, yang tidak tampak dan tidak dapat dijangkau oleh akal dan indra mereka, seperti Allah, malaikat, surga, neraka, dan lainnya yang diberitakan oleh Allah dan Rasul-Nya. Pada saat yang sama, sebagai bukti keimanan itu, mereka beribadah kepada Allah dengan melaksanakan salat, secara sempurna berdasarkan tuntunan Allah dan Rasul-Nya, khusyuk serta memperhatikan waktu-waktunya, dan mereka juga menginfakkan di jalan kebaikan sebagian rezeki berupa harta, ilmu, kesehatan, kekuasaan, dan hal-hal lainnya yang bermanfaat yang Kami berikan kepada mereka, semata-mata sebagai bentuk ketaatan kepada Allah dan mencari keridaan-Nya, Sejalan dengan konsep pada QS. Al-Baqarah ayat 2-4 ini, orang-orang yang beriman dan orang-orang yang bertaqwa. memiliki ciri cirinya masing-masing. Begitu juga dengan aspek *game* pada ulasan yang memiliki karakteristik dan ciri-cirinya masing-masing.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Klasifikasi aspek utama pada ulasan *game* menggunakan *neural network* dilakukan berdasarkan 5 skenario pengujian pembagian antara data *training* dan *testing*. Model dievaluasi menggunakan *confusion matrix* dan dilakukan perhitungan *precision*, *recall*, *f1-score* untuk memberikan gambaran performa model dalam mengklasifikasikan data. Kemudian model dengan performa terbaik divalidasi menggunakan *k-fold cross validation* dengan nilai $k = 5$ untuk memvalidasi bagaimana berperilaku terhadap data baru.

Model 4 dengan rasio perbandingan data *training* dan *testing* 60:40, *learning rate* 0,01, dan toleransi error $1e-5$ menunjukkan performa yang lebih baik diantara model skenario pengujian lainnya. Model 4 mencapai rata-rata presentase *precision* 89%, *recall* 91%, *f1-score* 90%, dan akurasi 90%. Validasi model menggunakan *k-fold cross validation* mendapatkan hasil rata-rata presentase akurasi yang hampir sama yaitu 90%.

5.2 Saran

Penelitian klasifikasi aspek utama dalam ulasan *game* memiliki potensi untuk peningkatan lebih lanjut, Terdapat beberapa saran yang dapat dipertimbangkan untuk diterapkan pada penelitian yang akan mendatang:

1. Melakukan pengumpulan data yang lebih banyak dan menggunakan dataset yang seimbang untuk setiap kelasnya. Data yang kurang banyak dan tidak

seimbang akan membuat model cenderung hanya mempelajari pola pada kelas yang dominan, hal ini akan membuat model terjadi bias yang besar dan berdampak performa yang buruk pada model. Dengan seimbangnya jumlah data yang disetiap kelas, tentu akan mengurangi bias pada model.

2. Melakukan eksperimen arsitektur *neural network* yang lebih kompleks, contohnya dengan menambah hidden layer dan node pada *neural network*. Hal ini dilakukan dengan harapan model dapat mempelajari pola kompleks dari data.

DAFTAR PUSTAKA

- Ali, K., Shaikh, Z. A., Khan, A. A., & Laghari, A. A. (2022). Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer. *Neuroscience Informatics*, 2(4), 100034. <https://doi.org/10,1016/j.neuri.2021.100034>
- Alonso-Fernández, C., Calvo-Morata, A., Freire, M., Martínez-Ortiz, I., & Fernández-Manjón, B. (2019). Applications of data science to *game* learning analytics data: A systematic literature review. *Computers & Education*, 141, 103612. <https://doi.org/10,1016/j.compedu.2019.103612>
- Amorim, P., Sousa Santos, B., Dias, P., Silva, S., & Martins, H. (2020). Serious *Games* for Stroke Telerehabilitation of Upper Limb - A Review for Future Research. *International Journal of Telerehabilitation*, 12(2), 65–76. <https://doi.org/10,5195/ijt.2020,6326>
- Barnes, S., & Prescott, J. (2018). Empirical Evidence for the Outcomes of Therapeutic Video *Games* for Adolescents With Anxiety Disorders: Systematic Review. *JMIR Serious Games*, 6(1), e3. <https://doi.org/10,2196/games.9530>
- Cheah, I., Shimul, A. S., & Phau, I. (2022). Motivations of playing digital *games*: A review and research agenda. *Psychology & Marketing*, 39(5), 937–950. <https://doi.org/10,1002/mar.21631>
- Ciuchita, R., Heller, J., Köcher, S., Köcher, S., Leclercq, T., Sidaoui, K., & Stead, S. (2023). It is Really Not a *Game*: An Integrative Review of Gamification for Service Research. *Journal of Service Research*, 26(1), 3–20. <https://doi.org/10,1177/10946705221076272>
- Damaševičius, R., Maskeliūnas, R., & Blažauskas, T. (2023). Serious *Games* and Gamification in Healthcare: A Meta-Review. *Information*, 14(2), 105. <https://doi.org/10,3390/info14020105>
- Fabricatore Carlo. (2007). *GAMEPLAY AND GAME MECHANICS DESIGN : A KEY TO QUALITY IN VIDEOGAMES. ENLACES (MINEDUC Chile) - OECD Expert Meeting on Videogames and Education*, 29–31. <https://eprints.hud.ac.uk/id/eprint/20927>
- Fauzianto, R. A., & Supatman. (2023). Analisis Sentimen Opini Masyarakat Terhadap Tech Winter Pada Twitter Menggunakan Natural Language Processing. *Jurnal Syntax Admiration*, 3(9), 1577–1585. <https://doi.org/10,46799/jsa.v3i9.909>
- Fukaya, K., Daylamani-Zad, D., & Agius, H. (2025). *Generative Tools for Graphical Assets: Empirical Guidelines based on Game Designers' and Developers' Preferences*. <http://arxiv.org/abs/2503.02703>
- Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class*

- Classification: an Overview*. <https://arxiv.org/abs/2008.05756>
- Gyllenback, K. B. (2009). *Narrative Bridging - a specification of a modelling method for game design*. <http://arxiv.org/abs/0905.3070>
- Hashemi, S. K., Mirtaheri, S. L., & Greco, S. (2023). Fraud Detection in Banking Data by Machine Learning Techniques. *IEEE Access*, *11*, 3034–3043. <https://doi.org/10.1109/ACCESS.2022.3232287>
- Ilham, M. R., & Laksito, A. D. (2023). Comparative Analysis of Using Word Embedding in Deep Learning for Text Classification. *Jurnal Riset Informatika*, *5*(2), 195–202. <https://doi.org/10.34288/jri.v5i2.507>
- Jain, E., & Kumar, R. (2024). Automated Strawberry Leaf Disease Detection Using ResNet-50 with K-Fold Cross-Validation: A Novel Approach. *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, 1636–1641. <https://api.semanticscholar.org/CorpusID:276186317>
- Kouratoras, M. (2020). Existential dramaturgy and video games: a formalistic approach to Telltale's 'The Walking Dead' as Existentialist *gameplay*. *International Journal of Performance Arts and Digital Media*, *16*, 341–360, <https://api.semanticscholar.org/CorpusID:263295417>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional *Neural networks*. In F. Pereira, C. J. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 25). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- Li, Y., Wang, C., & Liu, J. (2020). A Systematic Review of Literature on User Behavior in Video Game Live Streaming. *International Journal of Environmental Research and Public Health*, *17*(9), 3328. <https://doi.org/10.3390/ijerph17093328>
- Marshall, D., Coyle, D., Wilson, S., & Callaghan, M. (2013). Games, Gameplay, and BCI: The State of the Art. *IEEE Transactions on Computational Intelligence and AI in Games*, *5*(2), 82–99. <https://doi.org/10.1109/TCIAIG.2013.2263555>
- Merinda Lestandy, Abdurrahim Abdurrahim, & Lailis Syafa'ah. (2021). Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent *Neural network* dan Naïve Bayes. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(4), 802–808. <https://doi.org/10.29207/resti.v5i4.3308>
- Mika Parwita, I. M., & Siahaan, D. (2019). Classification of Mobile Application Reviews using Word Embedding and Convolutional *Neural network*. *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, *10*(1), 1. <https://doi.org/10.24843/lkjiti.2019.v10.i01.p01>
- Naseer, A., Mudawi, N. Al, Abdelhaq, M., Alonazi, M., Alazeb, A., Algarni, A., &

- Jalal, A. (2024). CNN-Based Object Detection via Segmentation Capabilities in Outdoor Natural Scenes. *IEEE Access*, *12*, 84984–85000, <https://doi.org/10.1109/ACCESS.2024.3413848>
- Pambudi, A., & Suprpto, S. (2021). Effect of Sentence Length in Sentiment Analysis Using Support Vector Machine and Convolutional *Neural network* Method. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *15*(1), 21. <https://doi.org/10.22146/ijccs.61627>
- Putra, I. M. S., Putu Jhonarendra, & Ni Kadek Dwi Rusjyanthi. (2021). Deteksi Kesamaan Teks Jawaban pada Sistem Test Essay Online dengan Pendekatan *Neural network*. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(6), 1070–1082. <https://doi.org/10.29207/resti.v5i6.3544>
- Sari, W. K., Rini, D. P., Malik, R. F., & Azhar, I. S. B. (2017). Klasifikasi Teks Multilabel pada Artikel Berita Menggunakan Long Short-Term Memory dengan Word2Vec. *Masa Berlaku Mulai*, *1*(3), 276–285.
- Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. <https://doi.org/10.1016/j.ipm.2009.03.002>
- Spjut, J., Madhusudan, A., Watson, B., Boudaoud, B., & Kim, J. (2022). The Esports Frontier: Rendering for Competitive Games. In *SIGGRAPH Frontiers 2022: ACM Computer Graphics and Interactive Techniques, Frontiers workshop, August, 2022, Vancouver, Canada* (Vol. 1, Issue 1). Association for Computing Machinery. <https://doi.org/10.48550/arXiv.2208.11774>
- Syaharuddin, S., Fatmawati, F., & Suprajitno, H. (2022). The Formula Study in Determining the Best Number of Neurons in *Neural network* Backpropagation Architecture with Three Hidden Layers. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *6*(3), 397–402. <https://doi.org/10.29207/resti.v6i3.4049>
- Tao, G., Garrett, B., Taverner, T., Cordingley, E., & Sun, C. (2021). Immersive virtual reality health games: a narrative review of game design. *Journal of NeuroEngineering and Rehabilitation*, *18*(1), 31. <https://doi.org/10.1186/s12984-020-00801-3>
- Tavares, P., Teixeira, P. M., Pereira, L., & Cruz-Cunha, M. M. (2011). *Video Games as Aggregating Mediums and Resulting Products of Several Visual Communication Languages*. <https://api.semanticscholar.org/CorpusID:63168736>
- Tlili, A., Denden, M., Duan, A., Padilla-Zea, N., Huang, R., Sun, T., & Burgos, D. (2022). Game-Based Learning for Learners With Disabilities—What Is Next? A Systematic Literature Review From the Activity Theory Perspective. *Frontiers in Psychology*, *12*. <https://doi.org/10.3389/fpsyg.2021.814691>
- Umar Ibrahim, A., Ozsoz, M., Serte, S., Al-Turjman, F., & Habeeb Kolapo, S. (2022). Convolutional *neural network* for diagnosis of viral pneumonia and

COVID-19 alike diseases. *Expert Systems*, 39(10).
<https://doi.org/10.1111/exsy.12705>

Wang, T., Liu, L., Liu, N., Zhang, H., Zhang, L., & Feng, S. (2020). A multi-label text classification method via dynamic semantic representation model and deep neural network. *Applied Intelligence*, 50(8), 2339–2351.
<https://doi.org/10.1007/s10489-020-01680-w>

Wang, X., Cao, Y.-F., Xiong, J., Chen, S., Li, W., Zhang, J., & Li, Q. (2025). *ClueCart: Supporting Game Story Interpretation and Narrative Inference from Fragmented Clues*. <http://arxiv.org/abs/2503.06098>

Zagal, J. P., & Tomuro, N. (2010). The aesthetics of *gameplay*. *Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments*, 9–16. <https://doi.org/10.1145/1930488.1930492>

Zayeni, D., Raynaud, J.-P., & Revet, A. (2020). Therapeutic and Preventive Use of Video Games in Child and Adolescent Psychiatry: A Systematic Review. *Frontiers in Psychiatry*, 11. <https://doi.org/10.3389/fpsy.2020.00036>

Zeng, J., Parks, S., & Shang, J. (2020). To learn scientifically, effectively, and enjoyably: A review of educational games. *Human Behavior and Emerging Technologies*, 2(2), 186–195. <https://doi.org/10.1002/hbe2.188>