

Perbandingan Kinerja RNN dan CNN Dalam Klasifikasi Sentimen Ulasan Pengguna Aplikasi di Play Store

Satria Nugraha Saputra^{1,*}, Galet Guntoro Setiaji¹, Max Teja Ajie Cipta Widiyanto²

¹Fakultas Teknologi Informasi dan Komunikasi, Prodi Teknik Informatika, Universitas Semarang, Semarang, Indonesia

²Fakultas Telematika Energi, Prodi Teknik Informatika, Institut Teknologi Perusahaan Listrik Negara, Jakarta, Indonesia

Email: ^{1,*}satrians49@gmail.com, ²gallet@usm.ac.id, ³max@itpln.ac.id

Email Penulis Korespondensi: satrians49@gmail.com

Submitted: 30/11/2024; Accepted: 30/11/2024; Published: 30/11/2024

Abstrak—Masyarakat sering menyampaikan pemikiran dan pendapat mereka tentang berbagai hal, seperti produk, tokoh masyarakat, atau kebijakan pemerintah di dunia internet. Proses untuk menganalisis data ulasan disebut sebagai analisis sentimen. Penelitian ini bertujuan untuk membandingkan performa dua model deep learning, yaitu Recurrent Neural Network (RNN) dan Convolutional Neural Network (CNN) dalam mengklasifikasikan sentimen ulasan pengguna berdasarkan lima kategori data ulasan dari Google Play Store yaitu desain, fotografi, game, media sosial, dan streaming. Pemilihan algoritma yang sesuai menjadi elemen penting untuk mencapai tingkat akurasi yang optimal, mengingat adanya variasi bahasa dan pola ekspresi dalam ulasan. Data yang digunakan dalam penelitian ini terdiri dari 50.000 ulasan dengan distribusi sentimen positif dan negatif yang tidak seimbang. Untuk mengatasi ketidakseimbangan ini, diterapkan teknik oversampling menggunakan Synthetic Minority Oversampling Technique (SMOTE). Proses evaluasi dilakukan dengan mengukur tingkat akurasi dan tingkat loss dari masing-masing model. Hasil penelitian menunjukkan bahwa CNN secara konsisten unggul dibandingkan RNN dalam sebagian besar kategori. Pada kategori desain, CNN mencapai akurasi tertinggi sebesar 85% dengan nilai loss 0,41, sedangkan RNN hanya mencapai akurasi 83% dengan nilai loss 0,53. Sebaliknya, kategori streaming menghasilkan performa terendah, dengan CNN memiliki akurasi 69% dan loss 0,63, sementara RNN hanya mencapai akurasi 67% dengan loss 0,72. Temuan ini menunjukkan bahwa CNN lebih efektif dalam analisis sentimen pada berbagai kategori ulasan pengguna.

Kata Kunci: Sentimen Analisis; Deep Learning; RNN; CNN; Ulasan Pengguna

Abstract—The public frequently shares their thoughts and opinions on various topics, such as products, public figures, or government policies, through online platforms. The process of analyzing review data is referred to as sentiment analysis. This study aims to compare the performance of two deep learning models Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) in classifying user sentiments across five review categories from the Google Play Store: design, photography, gaming, social media, and streaming. Choosing the right algorithm is essential to achieving optimal accuracy, given the variations in language and expression patterns within reviews. The dataset used in this study consists of 50,000 reviews with an imbalanced distribution of positive and negative sentiments. To address this imbalance, oversampling techniques were applied using the Synthetic Minority Oversampling Technique (SMOTE). The evaluation process measured each model's accuracy and loss levels. The results show that CNN consistently outperformed RNN across most categories. For the design category, CNN achieved the highest accuracy of 85% with a loss value of 0.41, compared to RNN, which achieved 83% accuracy and a loss of 0.53. On the other hand, the streaming category showed the lowest performance, with CNN achieving an accuracy of 69% and a loss of 0.63, while RNN achieved 67% accuracy with a loss of 0.72. These findings highlight CNN's superior effectiveness in sentiment analysis across diverse user review categories.

Keywords: Sentiment Analysis; Deep Learning; RNN; CNN; User Reviews

1. PENDAHULUAN

Analisis sentimen merupakan salah satu bidang yang berkembang pesat dalam pemrosesan bahasa alami atau *Natural Language Processing* (NLP). Teknik ini digunakan untuk memahami opini, emosi, atau persepsi yang terkandung dalam teks, sehingga menjadi alat yang penting dalam memahami interaksi manusia dengan teknologi. Dengan kemajuan era digital, data ulasan pengguna dari berbagai *platform* seperti Google Playstore, media sosial, layanan *streaming*, dan *e-commerce* setiap harinya berjumlah sangat banyak. Data ini mencakup berbagai pendapat tentang produk, layanan, atau bahkan konten digital yang sering kali menggambarkan pengalaman langsung pengguna. Selain menjadi representasi opini individu, data ulasan tersebut juga dapat dimanfaatkan untuk mendukung pengambilan keputusan berbasis data yang lebih strategis dalam dunia bisnis, teknologi, dan kebijakan publik[1][2] [3].

Pengelolaan data yang besar tersebut membutuhkan teknologi yang mampu mengolahnya. Data ini sering kali terletak pada volume data yang besar dan tidak terstruktur, mencerminkan bahasa informal, penggunaan emoji, singkatan, hingga konteks lokal atau budaya yang spesifik. Hal ini menjadikan analisis sentimen sebagai tantangan yang kompleks, terutama ketika melibatkan data dalam jumlah besar yang berasal dari berbagai sumber. Misalnya, ulasan tentang aplikasi desain cenderung menggunakan terminologi teknis tertentu, sementara ulasan *game* mungkin lebih bersifat emosional atau subjektif. Perbedaan ini menambah kompleksitas dalam mengembangkan model yang mampu memahami dan memproses data opini secara akurat[4][5][6].

Dengan meningkatnya jumlah pengguna, analisis sentimen menghadapi tantangan yang kompleks. Salah satu tantangan utamanya adalah keberagaman bahasa dan ekspresi dalam data ulasan. Ulasan pengguna sering kali mencerminkan gaya bahasa informal, penggunaan slang, singkatan, atau bahkan emoji yang semuanya dapat

memperumit proses analisis. Selain itu, ulasan juga sering kali memiliki makna yang kontekstual dan implisit, sehingga sulit untuk diklasifikasikan dengan metode tradisional. Dalam upaya mengatasi tantangan ini, berbagai pendekatan telah dikembangkan, mulai dari metode berbasis aturan hingga algoritma *machine learning*. *Deep Learning* adalah salah satu cara untuk menganalisa data yang besar tersebut. *Deep learning* menawarkan kemampuan untuk mengekstrak pola dan fitur yang kompleks dalam data teks tanpa memerlukan ekstraksi fitur secara manual[7][8][9].

Penelitian ini berfokus pada pengolahan data analisis sentimen dari lima kategori ulasan pengguna, yaitu desain, fotografi, game, media sosial, dan streaming, dengan memanfaatkan model *deep learning* seperti CNN (*Convolutional Neural Network*) dan RNN (*Recurrent Neural Network*). Selain itu, pendekatan RNN yang digunakan mencakup varian seperti *bidirectional LSTM* untuk meningkatkan pemahaman konteks dalam data. Proses pengolahan data dilakukan dengan mengintegrasikan teknik-teknik NLP (*Natural Language Processing*) untuk mendukung klasifikasi sentimen secara lebih akurat. Dalam implementasinya, penelitian ini menggunakan Python sebagai *platform* utama dengan dukungan *library* seperti *scikit-learn* dan keras untuk membangun, melatih, dan mengevaluasi model[10].

Meskipun sejumlah penelitian sebelumnya telah membandingkan algoritma pembelajaran mesin untuk analisis sentimen, penelitian pertama mengenai Perbandingan Metode *Decision Tree* dan *Support Vector Machine* untuk Analisis Sentimen pada Instagram Mengenai Kinerja PSSI, klasifikasi dilakukan dengan menggunakan dua algoritma pada data yang telah dipreproses. Hasilnya menunjukkan bahwa algoritma *Decision Tree* memiliki performa dengan akurasi 87,45%, presisi 87,72%, recall 91,74% dan F1-Score 89,69% sedangkan algoritma *Support Vector Machine* dengan akurasi 94,36%, presisi 96,78%, recall 94,30% dan F1-Score 95,53%. Didapatkan bahwa algoritma SVM mendapatkan nilai yang baik dari perbandingan tersebut[11].

Penelitian selanjutnya mengenai perbandingan kinerja algoritma *support vector machine* dan *k-nearest neighbor* terhadap analisis sentimen kebijakan new normal. Hasil penelitian ini menunjukkan bahwa algoritma *k-nearest neighbor* memiliki performa dengan akurasi yang tinggi sebesar 72,96% dibandingkan dengan SVM 71,19%[12].

Penelitian selanjutnya tentang Perbandingan Kinerja *Naive Bayes* dan SVM pada Analisis Sentimen Twitter Ibukota Nusantara, Hasil penelitian menunjukkan bahwa metode SVM secara konsisten memberikan tingkat akurasi sentimen analisis yang lebih tinggi (94%) dibanding dengan *Naive Bayes* (91%). Ini menandakan bahwa SVM dapat menjadi pilihan yang lebih efektif untuk mengukur sentimen terhadap Ibukota Nusantara dalam konteks penelitian ini[13].

Penelitian berikutnya mengenai analisis Komparasi Algoritma *Machine Learning* Untuk Menganalisis Sentimen Ulasan Pada Aplikasi *Digital* Korlantas Polri, Hasil penelitian ini menunjukkan bahwa metode *Random Forest* dan SVM melakukannya dengan paling baik. *Random Forest* memiliki akurasi 90,77%, recall 90,77%, dan nilai F1 tertinggi yaitu 90,79%. SVM memiliki presisi tertinggi dengan 91,14% di antara algoritma lainnya, yang menunjukkan potensi besar kedua algoritma ini menganalisis sentimen ulasan aplikasi *digital* Korlantas Polri[14].

Penelitian berikutnya tentang Perbandingan Algoritma *Machine Learning* untuk Analisis Sentimen Berbasis Aspek pada *Review Female Daily*, Hasil penelitian ini menunjukkan bahwa algoritma SVM dengan *kernel Linear* mendapatkan nilai akurasi terbaik sebesar 67,10% dibanding algoritma lainnya[15].

Penelitian ini bertujuan untuk menganalisis perbedaan kinerja model RNN dan CNN dalam klasifikasi sentimen berbasis teks. Untuk mendukung proses tersebut, penelitian ini juga mengintegrasikan beberapa teknik dari *Natural Language Processing* (NLP). Hasil yang diperoleh diharapkan dapat menjadi acuan bagi pengembang atau peneliti yang ingin menerapkan model *deep learning* dalam pengolahan data sentimen berbentuk teks, sekaligus memberikan wawasan praktis mengenai efektivitas masing-masing model[16].

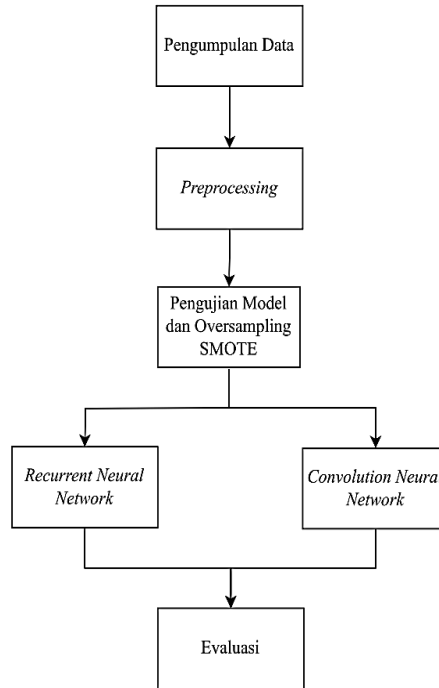
Hasil penelitian ini diharapkan dapat memberikan kontribusi signifikan pada pengembangan studi komputasi cerdas, khususnya dalam penerapan algoritma pembelajaran mesin untuk analisis sentimen berbasis teks. Secara khusus, hasil komparasi performa model *deep learning* seperti CNN dan RNN, serta analisis faktor-faktor yang memengaruhi kinerja masing-masing model, diharapkan menjadi panduan bagi peneliti dan pengembang dalam memilih, mengoptimalkan, dan menerapkan algoritma yang sesuai untuk klasifikasi sentimen pada berbagai domain ulasan pengguna.

Rekomendasi spesifik terkait model yang paling efektif untuk data ulasan dari kategori seperti desain, fotografi, game, media sosial, dan *streaming* juga diharapkan memberikan manfaat praktis bagi industri digital, terutama dalam meningkatkan pengalaman pengguna melalui analisis sentimen yang lebih akurat dan relevan. Secara umum, temuan studi ini memperkaya literatur tentang analisis sentimen berbasis *deep learning* dan membuka peluang untuk penelitian lebih lanjut, baik pada domain serupa maupun pada ulasan produk digital lainnya, seperti aplikasi *mobile*, *platform* media sosial, hingga layanan berbasis *streaming*. Dengan demikian, penelitian ini tidak hanya berkontribusi pada pengembangan teknologi analisis sentimen tetapi juga memberikan landasan kuat untuk kemajuan komputasi cerdas dan transformasi digital di era *digital*.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menggunakan metode eksperimen dengan melakukan pengujian pada beberapa model *deep learning* dan membandingkan hasilnya berdasarkan tinjauan literatur yang telah dilakukan sebelumnya. Proses penelitian ini terdiri dari beberapa tahapan pada Gambar 1 tahapan penelitian berikut:



Gambar 1. Tahapan Penelitian

Berdasarkan gambar 1 menunjukkan beberapa langkah yang harus dilakukan dalam penelitian, sebagai berikut;

1. Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari Google Play Store, mencakup analisis sentimen terhadap ulasan pengguna dalam lima kategori yaitu desain, fotografi, *game*, media sosial, dan *streaming*. Dataset ini terdiri dari dua kolom utama: kolom pertama berisi komentar pengguna, sedangkan kolom kedua menunjukkan klasifikasi sentimen, yaitu positif (bernilai 1) dan negatif (bernilai 0). Total data yang digunakan sebanyak 10.000 baris setiap kategori.

2. Preprocessing

Tahap ini menggunakan beberapa teknik *Natural Language Processing* (NLP) untuk mengolah data teks menjadi bentuk yang dapat diproses oleh model *deep learning*. Teknik yang diterapkan meliputi *count vectorizer*, dan *oversampling*.

3. Pengujian Model

Pada tahap ini, berbagai model *deep learning* diuji untuk menentukan performa terbaik dalam klasifikasi sentimen. *Deep learning* adalah metode khusus dalam pembelajaran mesin yang menggunakan jaringan saraf berlapis-lapis untuk mempelajari pola dalam data secara bertahap dengan meniru cara kerja otak manusia. Proses ini memungkinkan komputer untuk menangani masalah kompleks dan abstraksi yang sulit didefinisikan secara jelas. Konsep *deep learning* pertama kali diperkenalkan oleh Geoffrey Hinton pada tahun 2006 sebagai solusi atas keterbatasan metode pembelajaran mesin konvensional. Salah satunya dalam hal *feature engineering* dengan fitur dapat diekstraksi secara otomatis tanpa memerlukan proses manual. Teknologi ini juga menunjukkan peningkatan performa yang signifikan seiring bertambahnya jumlah data yang digunakan[17]. Model yang diuji meliputi:

a. CNN menggunakan 1D *Convolution* dan *BatchNormalization*

CNN (*Convolutional Neural Network*) adalah jenis jaringan saraf tiruan yang dirancang khusus untuk memproses data dengan struktur berbentuk grid. Contoh penerapannya meliputi data deret waktu, yang dapat dianggap sebagai grid 1D, dan data gambar, yang direpresentasikan sebagai grid piksel 2D. CNN terdiri dari beberapa komponen utama lapisan masukan (*input layer*), lapisan keluaran (*output layer*), dan sejumlah lapisan tersembunyi. Lapisan tersembunyi ini mencakup *convolutional layers*, *pooling layers*, *normalization layers*, *relu layers*, *fully connected layers*, serta *loss layers*. Salah satu penerapan CNN

dengan 1D *Convolution* adalah untuk klasifikasi teks, di mana pola sekuensial dalam data teks dapat diekstraksi dan dianalisis secara efektif[18].

b. RNN menggunakan *Bidirectional LSTM* dan *BatchNormalization*

RNN (*Recurrent Neural Network*) memiliki prinsip serupa dengan CNN, tetapi dirancang khusus untuk memproses data berurutan. RNN bekerja dengan melewati elemen-elemen dalam suatu urutan secara berulang sambil mempertahankan state internal yang menyimpan informasi relevan dari data yang telah diproses sebelumnya. Hal ini memungkinkan RNN untuk menangkap hubungan temporal atau konteks dalam data. Ciri khas dari RNN adalah adanya *loop* internal dalam arsitekturnya, yang memungkinkan model untuk mengingat informasi dari langkah sebelumnya[19].

4. Hasil dan Evaluasi

Tahapan ini berfokus pada menganalisis dan membandingkan performa setiap model berdasarkan metrik seperti akurasi dan *loss*. Hasil perbandingan kemudian dirangkum untuk menarik kesimpulan dan memberikan rekomendasi terkait model terbaik yang dapat digunakan untuk analisis sentimen berbasis teks. Pendekatan ini memberikan gambaran menyeluruh tentang efektivitas model *deep learning* dalam mengolah data ulasan pengguna.

2.2 Oversampling SMOTE

Dalam penelitian ini, salah satu tantangan dalam memproses data ulasan dari lima kategori desain, fotografi, *game*, media sosial, dan *streaming* adalah menghadapi ketidakseimbangan data, yaitu jumlah ulasan positif dan negatif yang tidak merata. Ketidakseimbangan ini dapat menyebabkan model *machine learning* atau *deep learning* menghasilkan prediksi yang cenderung bias terhadap kelas mayoritas.

SMOTE (*Synthetic Minority Oversampling Technique*) adalah metode *oversampling* yang digunakan untuk menangani ketidakseimbangan kelas dalam data yang sering ditemukan pada analisis sentimen. Dalam konteks penelitian ini, SMOTE diterapkan untuk meningkatkan jumlah ulasan pada kelas minoritas positif dan negatif sehingga seimbang dengan kelas mayoritas dalam setiap kategori data, seperti desain, fotografi, *game*, media sosial, dan *streaming*.

Teknik ini bekerja dengan menciptakan data sintetis baru untuk kelas minoritas. Prosesnya melibatkan interpolasi antara titik-titik data kelas minoritas yang ada di ruang fitur, bukan hanya menyalin data. SMOTE memilih sampel acak dari data minoritas, kemudian menghitung nilai interpolasi berdasarkan nilai secara acak, sehingga menciptakan sampel sintetis baru. Dengan demikian, area keputusan untuk kelas minoritas diperluas, dan model memiliki peluang yang lebih baik untuk mempelajari pola pada kelas minoritas[20].

Prinsip dasar SMOTE adalah menambahkan data sintetis pada kelas minoritas hingga jumlahnya setara dengan kelas mayoritas. Proses ini dilakukan untuk memastikan model *deep learning* seperti RNN dan CNN dapat dilatih secara optimal tanpa bias terhadap kelas mayoritas. Penerapan SMOTE pada penelitian ini memastikan bahwa distribusi ulasan dalam setiap kategori menjadi seimbang, sehingga hasil klasifikasi sentimen lebih akurat dan representatif. Proses pembangkitan data dengan skala numerik berbeda dari skala kategorik. Data numerik menghitung jarak kedekatan antar titik data menggunakan jarak *Euclidean*, sementara data kategorik mengandalkan nilai modus untuk menentukan kesamaan atau kemunculan kategori yang paling dominan[21].

Tahapan teknik yang dilakukan:

1. Identifikasi Kelas Minoritas dan Mayoritas

Dari data ulasan dalam klasifikasi sentimen yang dibagi menjadi kategori positif (bernilai 1) dan negatif (bernilai 0), SMOTE mengidentifikasi jumlah data di setiap kelas. Jika jumlah ulasan positif jauh lebih sedikit daripada ulasan negatif, maka ulasan positif dianggap sebagai kelas minoritas.

2. Pembentukan Data Sintetis

SMOTE menggunakan data kelas minoritas yang ada untuk menghasilkan sampel baru berdasarkan hubungan antar titik data terdekat. Teknik ini bekerja dengan menghitung jarak antara titik data minoritas menggunakan metrik *Euclidean* pada ruang fitur, lalu menciptakan sampel baru di antara titik-titik tersebut.

3. Integrasi Data Sintetis

Data sintetis yang dihasilkan ditambahkan ke dataset asli untuk menyeimbangkan jumlah data antara kelas minoritas dan mayoritas.

4. Penggunaan Dataset yang Seimbang

Dataset yang telah *dioversampling* digunakan dalam pelatihan model RNN dan CNN untuk memastikan model tidak bias terhadap kelas mayoritas.

Rumus untuk menyeimbangkan data sintetis dengan SMOTE sebagai berikut:

$$X_{new} = X_a + \lambda \times (X_b - X_a) \quad (1)$$

Untuk mengukur jarak antara kelas minor yang diubahnya berskala kategorik dilakukan dengan rumus *Value Difference Metric* (VDM) sebagai berikut:

$$\Delta(X, Y) = w_x w_y \sum_{i=1}^N \delta(x_i, y_i)^r \quad (2)$$

Untuk mendapatkan nilai jarak antar kategori menggunakan rumus dengan formula:

$$\delta(x_i, y_i) = \sum_{i=1}^N \left| \frac{c_{1i}}{c_1} - \frac{c_{2i}}{c_2} \right|^k \quad (3)$$

3. HASIL DAN PEMBAHASAN

Dalam penelitian ini, proses pengumpulan data berhasil menghasilkan 50.000 data ulasan dari lima kategori, yaitu desain, fotografi, *game*, media sosial, dan *streaming*, yang diperoleh dari Google Play Store setiap kategori memiliki 10.000 data ulasan. Data ini kemudian melalui tahap *preprocessing*, di mana dilakukan pembersihan terhadap *noise*, standarisasi teks, dan persiapan data untuk keperluan analisis sentimen lebih lanjut. Selanjutnya, dataset dibagi secara acak menjadi data latih dan data uji. Data tersebut kemudian diubah menjadi representasi vektor fitur menggunakan *CountVectorizer* dan *Oversampling Smote*. Vektor fitur ini digunakan untuk melatih dan menguji beberapa model *deep learning*, seperti *Neural Network* dengan *MaxPooling1D*, *CNN*, *RNN*, dan *bidirectional LSTM*.

Setiap model berhasil diimplementasikan untuk melakukan klasifikasi sentimen pada ulasan pengguna. Performanya dievaluasi menggunakan metrik akurasi dan *loss*, yang hasilnya dibandingkan untuk menentukan model yang paling efektif dalam menangani analisis sentimen berbasis teks.

3.1 Data Selection

Tabel 1. Selection Data

ReviewId	User Name	User Image	Content	Score	Thumbs Up Count	Review Created Version	At	Reply Content	Replied At	App Version
a4eed7f8-8bf8-4460-b808-9d3c39dc6	Ajiluk	https://play-lh.google...	membantu dalam membuat desain..	5	40	2.288	2024-11-08	None	Nat	2.288
fc7d61b-473f-44d3-a341-	Keyvinbrins	https://play-lh.google..	So far so good but..	3	4	2.288	2024-11-08	Hai, terima kasih atas masukannya.	2024-11-10	2.288
...
4d9170c2-8dcc-4bb7-bd7a-2622ddbc6	Husainihikmal	https://play-lh.google..	Canva tolong perbaikin bugnya !..	1	0	2.285	2024-11-09	Hai, terima kasih atas masukannya	2024-10-24	2.285
418b2a5c-47dd-4833-9a38-	Ulummif	https://play-lh.google..	Aplikasi Canva ini bagus...	4	39	2.285	2024-10-22	None	Nat	2.285

Pada Tabel 1 menunjukkan data sebelum dilakukan web scrapping data. Sebelum melakukan pengambilan data ulasan dari Google Play Store, terlihat bahwa kolom ulasan yang tersedia sangat beragam, seperti yang ditunjukkan pada tabel tersebut. Oleh karena itu, diperlukan kriteria seleksi yang jelas agar data yang diambil relevan dan sesuai dengan kebutuhan analisis sentimen yang akan dilakukan.

Penelitian ini menggunakan data ulasan dari lima kategori utama, yaitu desain, fotografi, *game*, media sosial, dan *streaming*. Data dari masing-masing kategori diidentifikasi melalui nama aplikasi spesifik untuk memastikan kesesuaian dengan topik yang dianalisis. Selanjutnya, kriteria seleksi lainnya meliputi pemilihan ulasan berbahasa Indonesia (*lang='id'*) agar sesuai dengan kebutuhan analisis berbasis teks dalam penelitian ini. Jumlah ulasan yang diambil dari setiap kategori ditentukan sebanyak 10.000 data (*count=10.000*) untuk memastikan dataset yang dihasilkan cukup representatif. Selain itu, data ulasan diurutkan berdasarkan relevansi terhadap masing-masing kategori (*sort=Sort.MOST_RELEVANT*) untuk memastikan hanya data yang benar-benar relevan dengan penelitian.

3.1.1 Scrapping Data

Penelitian ini menggunakan beberapa *library* Python yang mendukung proses pengambilan dan analisis data secara efektif. Salah satu *library* utama adalah *google_play_scraper*, yang digunakan khusus untuk mengambil informasi dari Google Play Store melalui teknik *web scraping*. Informasi yang diperoleh meliputi ulasan pengguna, peringkat, waktu ulasan, dan nama pengguna, yang kemudian digunakan sebagai bahan analisis sentimen. Selanjutnya, *library pandas* memainkan peran penting dalam manipulasi dan analisis data. Data yang telah dikumpulkan disimpan dalam struktur DataFrame untuk memudahkan pengolahan lebih lanjut. Untuk visualisasi hasil analisis, *library matplotlib.pyplot* digunakan untuk membuat berbagai jenis grafik yang membantu menyajikan data secara lebih mudah dipahami. Selain itu, *library NumPy* digunakan untuk mendukung komputasi numerik yang diperlukan dalam pengolahan data kuantitatif. Dengan memanfaatkan kombinasi *library* Python yang tepat, penelitian dapat melakukan proses pengambilan hingga analisis data secara terstruktur dan komprehensif. Berikut hasil scrapping data pada Tabel 2.

Tabel 2. Hasil Scrapping Data

Content	Score
Canva sangat membantu dalam membuat desain dengan cepat dan mudah! Banyak pilihan template dan elemen yang menarik, cocok untuk segala kebutuhan. Sangat intuitif, bahkan untuk pemula. Aplikasi wajib bagi siapa saja yang ingin membuat konten visual	5
Canva tolong perbaiki bugnya! Setelah membuat akun baru, layarnya menjadi sedikit gelap dan tidak bisa menekan apapun. Ketika saya ingin membeli pro di akun yang sudah saya miliki juga terjadi hal yang sama. Barnya tidak muncul, mohon untuk segera pebaiki!!!!	1
sangat pengalaman yang buruk kecewa!! tidak bisa menempel hasil ketikan salin kita sangatgg kecewa dan membingungkan dan juga berbayar harus berbayar dan menyulitkan bagi kita yang menggunakan gratis sangat tidak cocok!!	2
Aplikasi ini cocok untuk remaja untuk menyunting video dan foto untuk tugas sekolah, tapi kurang responsif dan sering error dan lamban saat dibuka, terkadang seperti itu dan itu membuang-buang waktu saya.	3
aplikasinya oke bagus dan sangat membantu buat pengeditan, tpi kalo buat ngedit d HP ram rendah seperti punyaku ram 4 kok berat banget ya. semoga bisa d upgrade supaya lebih ringan biar ga patah2□□	4

3.2 Preprocessing

Sebelum melakukan pengujian model, terdapat dua tahap awal yang penting, yaitu pengumpulan data dan preprocessing. Data diperoleh dari *platform* Google Play Store, kemudian dibagi menjadi dua bagian, yakni data latih (*training*) dan data uji (*test*). Pada tahap *preprocessing*, data diolah menggunakan beberapa teknik NLP, yaitu:

1. *Count Vectorizer*. Teknik ini memanfaatkan *library scikit-learn* untuk mengubah teks menjadi representasi kalimat vektor. Proses ini melibatkan ekstraksi kata-kata dari setiap kalimat dalam dataset dan pembuatan vocabulary dari kumpulan kata-kata unik yang muncul dalam data. Teknik ini memungkinkan model memahami struktur teks dalam bentuk numerik. Berikut contoh hasil untuk *Count vectorizer* pada gambar 2 berikut.

```

Contoh dokumen

canva membantu desain cepat mudah pilihan template elemen menarik cocok kebutuhan
intuitif pemula aplikasi wajib konten visual

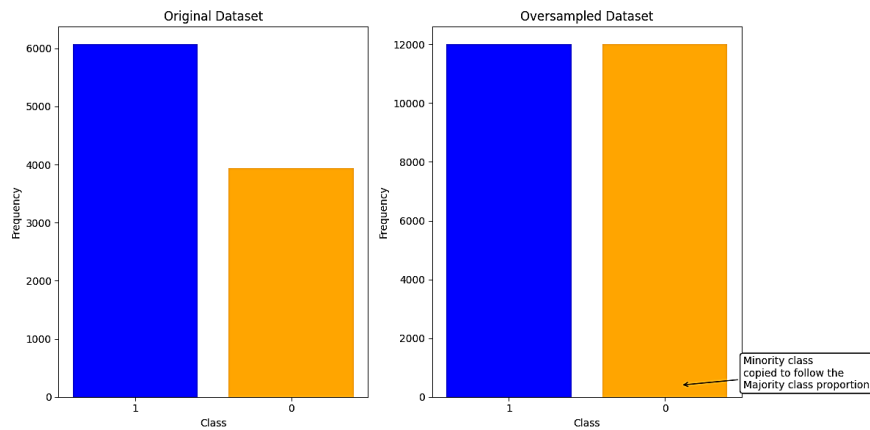
Hasil Text Vectorization
tf.Tensor(
[[ 2  8 10 102 13  88  39 115  77  78 323 2096  74  4 337 266
1336  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
dtype=int64) Vector size : (1, 100)+
    
```

Gambar 2. Contoh Hasil Count Vectorizer

Hasil *count vectorizer* pada gambar 2 merepresentasikan dokumen teks dalam bentuk vektor numerik, di mana setiap elemen dalam vektor menunjukkan jumlah kemunculan kata tertentu dalam dokumen. Proses ini mengubah kata-kata dalam teks, seperti "canva," "membantu," dan lainnya, menjadi nilai numerik berdasarkan frekuensi kemunculannya. Vektor yang dihasilkan memiliki ukuran tetap, yaitu (1, 100), yang

terdiri dari 100 elemen, di mana nilai 0 menunjukkan bahwa kata tertentu tidak ada dalam dokumen. Representasi ini memungkinkan data teks diolah lebih lanjut oleh algoritma pembelajaran mesin, menjadikannya format yang sesuai untuk analisis berbasis komputasi.

2. *Oversampling*, menggunakan SMOTE ini bertujuan untuk menyeimbangkan kelas pada data latih. Teknik ini dilakukan untuk mengatasi ketimpangan distribusi data, di mana sebelumnya diketahui bahwa data pada kelas “sentiment” memiliki jumlah yang tidak seimbang dibandingkan kelas lainnya setelah dilakukan *preprocessing*. Dengan menerapkan SMOTE, data latih menjadi lebih seimbang, sehingga model deep learning dapat dilatih dengan lebih optimal tanpa bias terhadap kelas dominan. Berikut hasil gambar 3 yang menunjukkan distribusi data setelah proses *oversampling*.



Gambar 3. Hasil Setelah Di *Oversampling*

Hasil smote pada gambar 3 menunjukkan perbandingan *dataset* sebelum dan sesudah *oversampling* menggunakan teknik SMOTE. Pada *Original Dataset* (kiri), kelas mayoritas (1) memiliki frekuensi lebih tinggi dibandingkan kelas minoritas (0). Setelah SMOTE diterapkan (kanan), frekuensi kelas minoritas meningkat dengan menambahkan data sintesis sehingga seimbang dengan kelas mayoritas. Teknik ini membantu model untuk belajar secara adil tanpa bias terhadap kelas mayoritas.

3.3 Pengujian Model

Tahapan berikutnya adalah pengujian model. Pada tahap ini, dilakukan evaluasi dan analisis terhadap performa masing-masing model *deep learning* yang telah diterapkan untuk klasifikasi sentimen ulasan pengguna.

3.3.1 Model RNN dengan *Bidirectional LSTM*

Pada struktur model RNN menggunakan layer *Bidirectional LSTM*, yang dirancang untuk menangkap informasi dari urutan teks baik dari arah maju maupun mundur. Berikut adalah hasil dan rincian dari tiap *layer* dalam model ini pada tabel 3 dan 4 sebagai berikut:

Tabel 3. Hasil *Layer Model RNN*

Layer (type)	Output Shape	Param #
<i>Embedding_3 (Embedding)</i>	(None, 100, 100)	500.000
<i>bidirectional_1 (Bidirectional)</i>	(None, 64)	34.048
<i>dropout_6 (Dropout)</i>	(None, 64)	0
<i>dense_6 (Dense)</i>	(None, 16)	1.040
<i>batch_normalization_3 (BatchNormalization)</i>	(None, 16)	64
<i>dropout_7 (Dropout)</i>	(None, 16)	0
<i>dense_7 (Dense)</i>	(None, 2)	34
Total params: 535.186 (2.04 MB)		
Trainable params: 535.154 (2.04 MB)		
Non-trainable params: 32 (128.00 B)		

Tabel 4. Rincian Hasil *Layer Model RNN*

<i>Embedding Layer</i>	Mengonversi kata-kata ke dalam representasi vektor dimensi 100 dan jumlah parameter sebesar 500,000, yang dihitung dari jumlah kata unik (<i>vocabulary</i>) dikalikan dengan dimensi vektor.
<i>Bidirectional Layer</i>	Menggunakan LSTM untuk memahami hubungan sekuensial dalam data dan jumlah parameter adalah 34,048, yang melibatkan bobot untuk arah maju dan mundur.

<i>Dropout Layer</i>	Digunakan untuk mengurangi risiko <i>overfitting</i> dengan mengabaikan unit-unit tertentu selama pelatihan.
<i>Dense Layer</i>	Lapisan <i>fully connected</i> dengan 16 unit, menghasilkan jumlah parameter sebesar 1,040.
<i>Batch Normalization Layer</i>	Menstabilkan distribusi aktivasi dari <i>layer</i> sebelumnya, meningkatkan efisiensi pelatihan
<i>Output Layer</i>	Lapisan terakhir dengan dua unit untuk melakukan klasifikasi biner (sentimen positif atau negatif) dengan jumlah parameter adalah 34.

Model ini memiliki jumlah 535,186 parameter, di mana 535,154 adalah parameter yang dapat dilatih. Struktur ini dirancang untuk menangani data sekuensial dan untuk memahami konteks teks dalam analisis sentimen.

3.3.2 Model CNN dengan 1D Convolution

Pada struktur ini menampilkan model CNN berbasis 1D Convolution, yang digunakan untuk menangkap pola lokal dalam data teks melalui fitur convolution. Berikut adalah hasil dan rincian dari tiap *layer* dalam model ini pada tabel 5 dan 6 sebagai berikut:

Tabel 5. Hasil *Layer* CNN

Layer (type)	Output Shape	Param #
<i>Embedding_4 (Embedding)</i>	(None, 100, 300)	1.500.000
<i>conv1d_2 (Conv1D)</i>	(None, 100, 32)	38.432
<i>max_pooling1d_2 (MaxPooling1D)</i>	(None, 50, 32)	0
<i>dropout_8 (Dropout)</i>	(None, 50, 32)	0
<i>batch_normalization_4 (BatchNormalization)</i>	(None, 50, 32)	128
<i>flatten_2 (Flatten)</i>	(None, 1600)	0
<i>dense_8 (Dense)</i>	(None, 16)	25.616
<i>dropout_9 (Dropout)</i>	(None, 16)	0
<i>dense_9 (Dense)</i>	(None, 2)	34
Total params: 1.564.210 (5.97 MB)		
Trainable params: 1.564.146 (5.97 MB)		
Non-trainable params: 64 (250.00 B)		

Tabel 6. Rincian Hasil *Layer* Model CNN

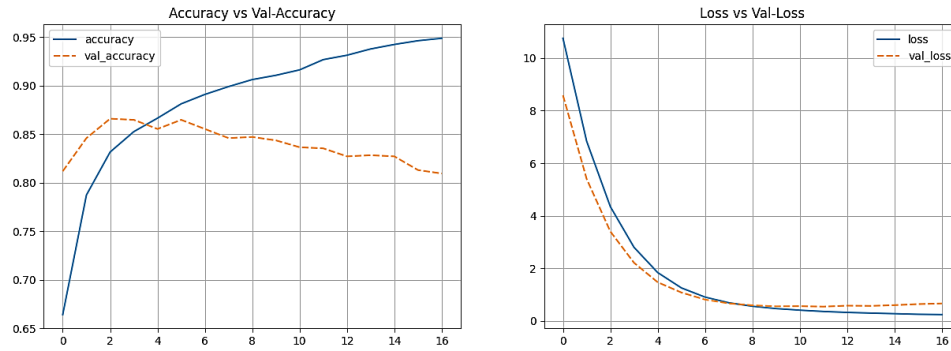
<i>Embedding Layer</i>	Mengonversi kata-kata menjadi representasi vektor dengan dimensi 300 dengan jumlah parameter 1,500,000 dan dihitung dari jumlah kata unik (<i>vocabulary</i>) dikalikan dengan dimensi <i>embedding</i> .
<i>Conv1D Layer</i>	<i>Layer</i> ini menerapkan <i>filter convolutional</i> pada dimensi 1D untuk menangkap fitur lokal dari teks dan jumlah parameter sebesar 38,432, tergantung pada jumlah filter dan ukuran kernel.
<i>MaxPooling1D Layer</i>	Digunakan untuk mengurangi dimensi data, meningkatkan efisiensi komputasi, dan menangkap fitur paling penting dalam jendela <i>pooling</i> .
<i>Dropout layer</i>	Menambahkan regularisasi untuk mencegah <i>overfitting</i> .
<i>Batch Normalization Layer</i>	Digunakan untuk menormalkan <i>output</i> dari <i>layer</i> sebelumnya, mempercepat konvergensi selama pelatihan.
<i>Flatten Layer</i>	Mengubah <i>output</i> dari <i>layer</i> sebelumnya menjadi vektor 1D untuk dihubungkan ke <i>layer dense</i>
<i>Dense Layer</i>	Digunakan untuk melakukan klasifikasi atau regresi pada data <i>layer</i> dengan 16 unit yang memiliki jumlah parameter 25,616.
<i>Output Layer</i>	Lapisan terakhir dengan dua unit untuk klasifikasi sentimen (positif/negatif) dengan jumlah parameter adalah 34.

Model ini memiliki total 1,564,210 parameter, dengan 1,564,146 parameter yang dapat dilatih. CNN efektif untuk menangkap pola penting dalam data teks, seperti kata-kata atau frasa penting.

1. Data kategori Desain

Pada kategori desain mencakup ulasan yang berkaitan dengan aplikasi atau produk yang berfokus pada elemen visual dan estetika, seperti aplikasi pengeditan gambar, ilustrasi, atau desain grafis. Ulasan dalam kategori ini sering kali mencerminkan kebutuhan pengguna terhadap fitur spesifik, kemudahan penggunaan, dan kualitas hasil desain. Analisis sentimen dalam kategori ini berguna untuk memahami kepuasan pengguna terhadap fitur-fitur yang ditawarkan dengan label klasifikasi sentimen positif sebesar 6068 dan sentimen negatif sebesar 3932.

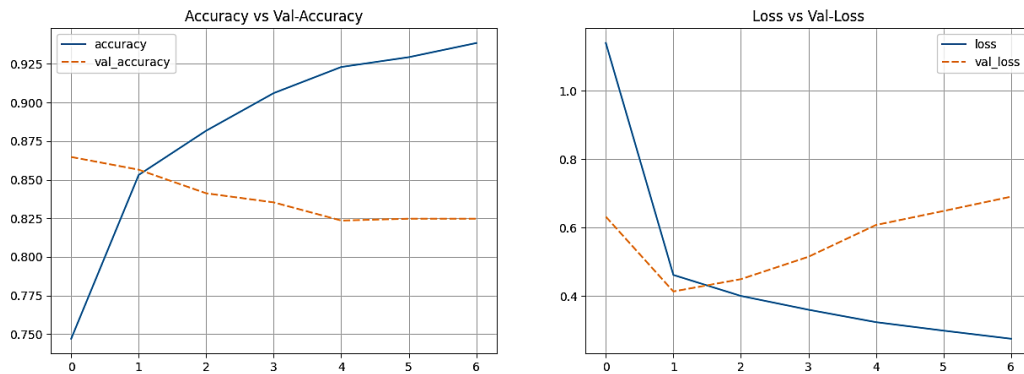
a. Hasil Model *Training* RNN dengan *Bidirectional LSTM*



Gambar 4. Hasil Pengujian Model RNN Kategori Desain

Pada kategori desain, model RNN mencapai akurasi 83% dan nilai *loss* sebesar 0,53. Model ini menunjukkan performa akurasi yang sangat baik dalam kategori desain. *Loss* yang cukup rendah mengindikasikan model mampu mempelajari pola data dengan baik.

b. Hasil model *Training CNN* dengan 1D *Convolution*



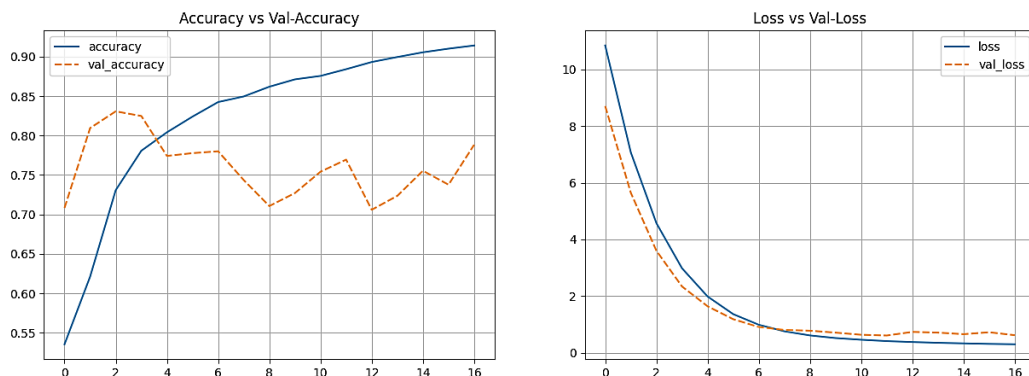
Gambar 5. Hasil Pengujian Model CNN Kategori Desain

Pada kategori desain, model CNN mencapai akurasi 85% dengan nilai *loss* sebesar 0.41. Model ini mengungguli RNN dalam kategori ini, dengan akurasi lebih tinggi dan *loss* lebih rendah. Hal ini menunjukkan bahwa CNN lebih efektif dalam menangkap pola data ulasan desain.

2. Data kategori Fotografi

Pada kategori fotografi berfokus pada aplikasi yang digunakan untuk pengeditan foto, pengelolaan galeri, atau platform berbagi foto. Ulasan dalam kategori ini biasanya terkait dengan kemampuan aplikasi dalam mengelola warna, resolusi, atau filter khusus. Sentimen pengguna terhadap kategori ini dapat membantu pengembang meningkatkan fitur teknis atau estetika dengan label klasifikasi sentimen positif sebesar 4320 dan sentimen negatif sebesar 5680.

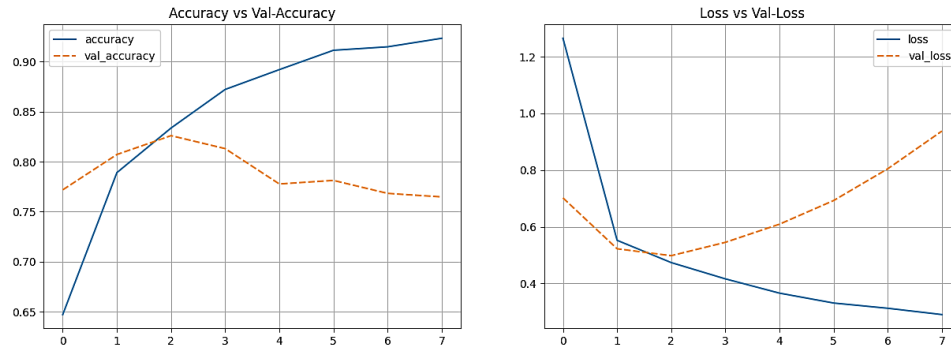
a. Hasil Model *Training RNN* dengan *Bidirectional LSTM*



Gambar 6. Hasil Pengujian Model RNN Kategori Fotografi

Pada kategori fotografi, model RNN mencapai akurasi 76% dan nilai *loss* sebesar 0,61. Model ini memberikan performa akurasi yang cukup baik, tetapi *loss*-nya relatif lebih tinggi dibandingkan kategori sebelumnya, yang mungkin menunjukkan kesulitan model dalam menangkap pola ulasan fotografi.

b. Hasil model *Training CNN* dengan 1D *Convolution*



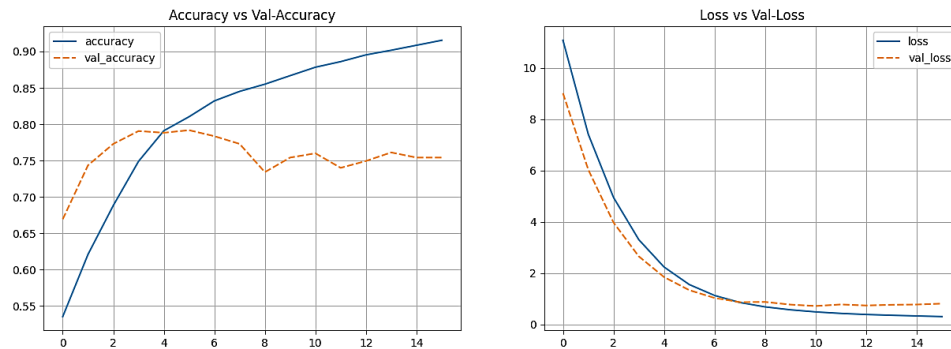
Gambar 7. Hasil Pengujian Model CNN Kategori Fotografi

Pada kategori fotografi, model CNN mencapai akurasi 82% dan nilai *loss* sebesar 0,49. Model ini memberikan performa lebih baik daripada RNN, dengan akurasi lebih tinggi dan *loss* lebih rendah, menunjukkan kemampuan CNN dalam menangani data ulasan fotografi lebih efektif.

3. Data kategori *Game*

Pada kategori ini mencakup ulasan dari aplikasi permainan yang biasanya mengandung ekspresi emosi yang kuat dari pengguna. Ulasan pada kategori *game* sering kali mengomentari mekanisme permainan, grafis, pengalaman pengguna, dan tingkat kesulitan. Pola sentimen di sini sangat bervariasi karena bergantung pada pengalaman subjektif pemain dengan label klasifikasi sentimen positif sebesar 4650 dan sentimen negatif sebesar 5350.

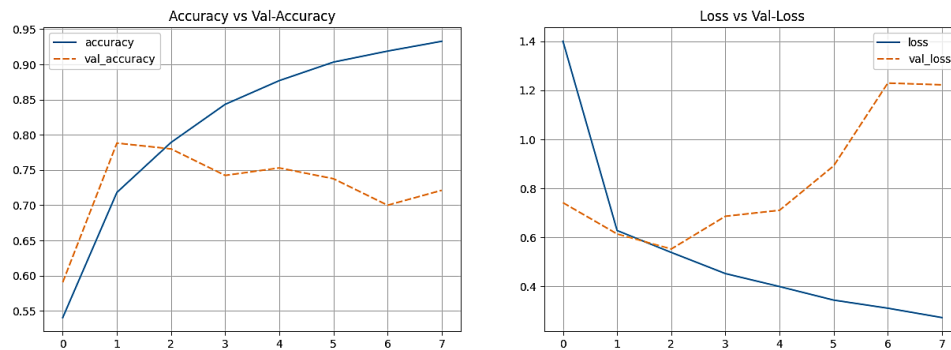
a. Hasil model *Training RNN* dengan *Bidirectional LSTM*



Gambar 8. Hasil Pengujian Model RNN Kategori Game

Pada kategori game, model RNN mencapai akurasi 75% dan nilai *loss* sebesar 0,71. Model ini menunjukkan performa yang cukup baik, meskipun *loss* yang lebih tinggi dapat mengindikasikan bahwa model kesulitan dalam menangkap pola sentimen dalam ulasan *game*.

b. Hasil model *Training CNN* dengan *1D Convolution*



Gambar 9. Hasil Pengujian Model CNN Kategori Game

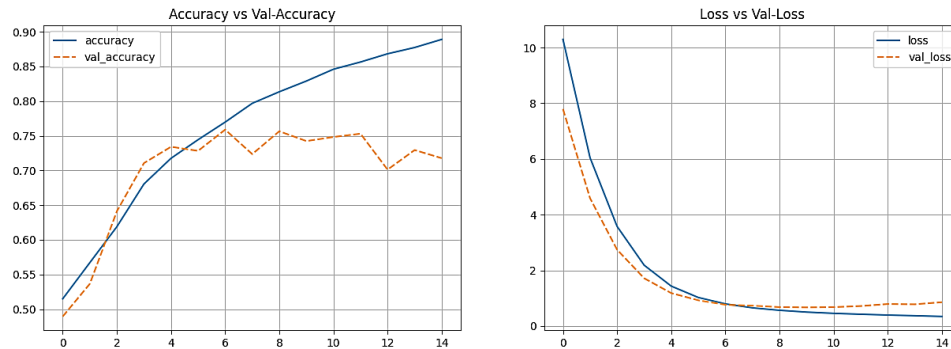
Pada kategori game, model CNN mencapai akurasi 77% dan nilai *loss* sebesar 0,55. memberikan hasil yang lebih baik daripada RNN dengan akurasi lebih tinggi dan *loss* lebih rendah, menandakan kemampuan CNN dalam menangani data ulasan *game* lebih efektif.

4. Data kategori Media Sosial

Pada kategori media sosial melibatkan ulasan dari aplikasi yang dirancang untuk komunikasi, berbagi konten, atau interaksi sosial. Sentimen pengguna dalam ulasan ini biasanya mencakup komentar tentang fitur privasi,

kenyamanan penggunaan, atau efektivitas konektivitas sosial. Ulasan dalam kategori ini sering kali pendek namun sangat subjektif dengan label klasifikasi sentimen positif sebesar 2783 dan sentimen negatif sebesar 7217.

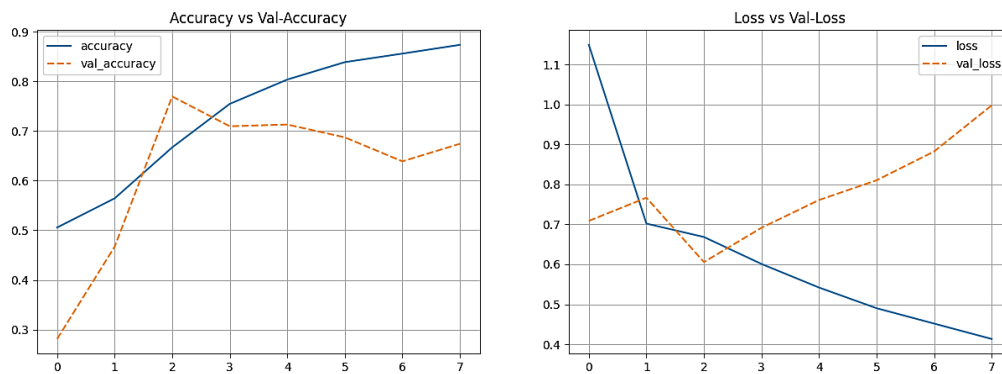
a. Hasil model *Training RNN dengan Bidirectional LSTM*



Gambar 10. Hasil Pengujian Model RNN Kategori Media Sosial

Pada kategori media sosial, model RNN mencapai akurasi 74% dan nilai *loss* sebesar 0,67. Model ini memberikan akurasi yang cukup baik, tetapi *loss* yang tinggi menunjukkan adanya tantangan dalam memahami pola ulasan media sosial yang mungkin lebih kompleks dan variatif.

b. Hasil model *Training CNN dengan 1D Convolution*



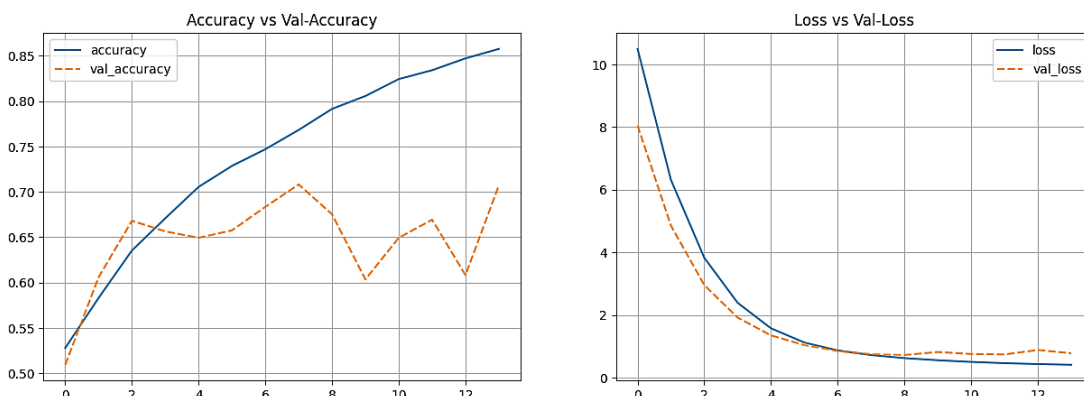
Gambar 11. Hasil Pengujian Model CNN Kategori Media Sosial

Pada kategori media sosial, model CNN mencapai akurasi 76% dan nilai *loss* sebesar 0,60. Model ini sedikit lebih baik dibandingkan RNN dalam kategori ini, dengan akurasi lebih tinggi dan *loss* lebih rendah, menunjukkan kemampuan CNN dalam menangani data ulasan media sosial lebih efektif.

5. Data kategori *Streaming*

Pada kategori streaming mencakup ulasan terkait layanan *streaming* video atau musik, seperti Netflix, Spotify, atau YouTube. Ulasan dalam kategori ini sering berfokus pada kualitas layanan, variasi konten, dan stabilitas koneksi. Pola bahasanya sering kali lebih santai dan beragam dengan label klasifikasi sentimen positif sebesar 2961 dan sentimen negatif sebesar 7039.

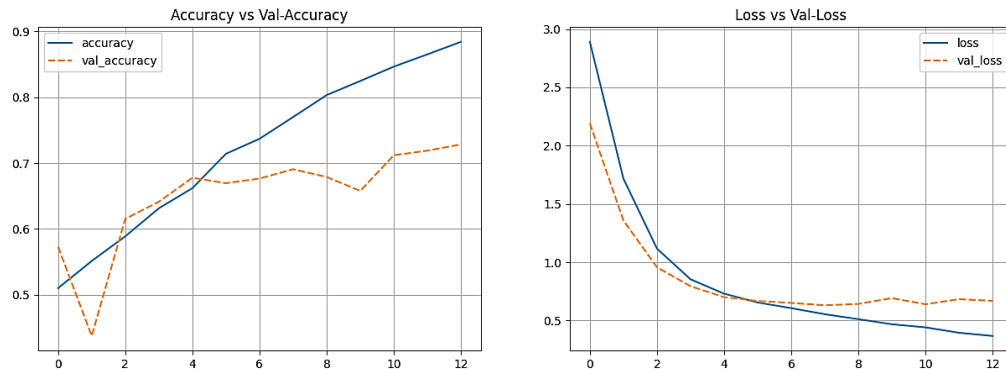
a. Hasil model *Training RNN dengan Bidirectional LSTM*



Gambar 12. Hasil Pengujian Model RNN Kategori Streaming

Pada kategori streaming, model RNN mencapai akurasi 67% dan nilai loss sebesar 0,72. Model ini memberikan performa terendah di antara semua kategori. Akurasi yang rendah dan loss yang tinggi menunjukkan kesulitan besar dalam mempelajari pola sentimen ulasan *streaming*.

b. Hasil model *Training CNN* dengan 1D *Convolution*



Gambar 13. Hasil Pengujian Model RNN Kategori *Streaming*

Pada kategori *streaming*, model CNN mencapai akurasi 69% dan nilai loss sebesar 0,63. Model ini menunjukkan sedikit lebih baik daripada RNN, dengan akurasi lebih tinggi dan loss lebih rendah. Namun, performanya tetap tidak optimal dibandingkan kategori lain.

Hasil pengujian dari percobaan model *deep learning* yang dilakukan telah dirangkum secara sistematis dalam tabel 7 berikut.

Tabel 7. Hasil Pengujian Model

Jenis Data	Jumlah Data	Sentiment		RNN		CNN	
		Negatif (0)	Positif (1)	Akurasi	Loss	Akurasi	Loss
Desain	10.000	3932	6068	0,83%	0,53%	0,85%	0,41%
Fotografi	10.000	5680	4320	0,76%	0,61%	0,82%	0,49%
Game	10.000	5350	4650	0,75%	0,71%	0,77%	0,55%
Media Sosial	10.000	7217	2783	0,74%	0,67%	0,76%	0,60%
Streaming	10.000	7039	2961	0,67%	0,72%	0,69%	0,63%

Berdasarkan hasil pengujian model deep learning menggunakan RNN (*Recurrent Neural Network*) dan CNN (*Convolutional Neural Network*) terhadap lima kategori data ulasan Desain, Fotografi, *Game*, Media Sosial, dan *Streaming*, dapat dihasilkan bahwa model CNN secara konsisten menunjukkan performa yang lebih unggul dibandingkan RNN dalam hal akurasi. Bisa dilihat pada kategori Data Desain, CNN mencapai akurasi tertinggi sebesar 85% dengan *loss* terendah 41%, dibandingkan dengan RNN yang menghasilkan akurasi 83% dan *loss* 53%. Hal ini menunjukkan bahwa CNN lebih efektif dalam menangkap pola dan representasi fitur pada data ulasan. Namun, performa RNN tetap stabil di seluruh kategori, meskipun cenderung kurang kompetitif dibandingkan CNN.

Setiap kategori data memiliki variasi performa. Data Desain mencatat performa terbaik untuk kedua model, sedangkan kategori Data Streaming menunjukkan performa terendah, dengan akurasi CNN sebesar 69% dan RNN sebesar 67%. Hal ini mengindikasikan bahwa ulasan pada kategori tertentu, seperti Streaming, memiliki kompleksitas atau tingkat noise yang lebih tinggi, sehingga lebih sulit untuk diproses oleh kedua model. Selain itu, ketidakseimbangan data antara sentimen positif dan negatif juga memengaruhi hasil *loss*.

4. KESIMPULAN

Penelitian ini secara komprehensif mengevaluasi dan membandingkan performa dua model deep learning, yaitu *Recurrent Neural Network* (RNN) dan *Convolutional Neural Network* (CNN) dalam klasifikasi sentimen berdasarkan lima kategori data ulasan, yaitu desain, fotografi, game, media sosial, dan streaming. Hasil pengujian menunjukkan bahwa CNN secara konsisten memiliki akurasi lebih tinggi dibandingkan RNN, dengan performa terbaik pada kategori desain, mencapai akurasi 85% dan *loss* 0,41%. Sebaliknya, kategori streaming menunjukkan performa terendah pada kedua model, dengan akurasi CNN sebesar 69% dan RNN sebesar 67%. Metode *oversampling* seperti SMOTE yang digunakan dalam penelitian ini berhasil membantu mengurangi dampak ketidakseimbangan data, meskipun belum sepenuhnya mengatasi masalah tersebut. Proses *preprocessing* yang melibatkan teknik *Natural Language Processing* (NLP), seperti *count vectorizer*, juga menjadi elemen penting dalam menghasilkan representasi fitur teks yang optimal untuk pemrosesan lebih lanjut oleh model. Hasil penelitian ini tidak hanya menunjukkan efektivitas CNN dalam menangani kompleksitas pola

pada sebagian besar kategori data, tetapi juga menggarisbawahi pentingnya pemilihan algoritma yang disesuaikan dengan karakteristik *dataset*. Temuan ini memberikan implikasi praktis yang luas bagi pengembangan sistem analisis sentimen berbasis deep learning khususnya dalam ranah ulasan produk *digital*, *platform* media sosial, dan layanan *streaming*. Untuk meningkatkan hasil pada penelitian selanjutnya, diharapkan untuk menerapkan strategi pengolahan data tambahan, seperti *augmentasi* data atau *hyperparameter tuning* lebih lanjut.

REFERENCES

- [1] D. P. Santoso and W. Wibowo, "Analisis Sentimen Ulasan Aplikasi Buzzbreak Menggunakan Metode Naïve Bayes Classifier pada Situs Google Play Store," *J. Sains dan Seni ITS*, vol. 11, no. 2, 2022, doi: 10.12962/j23373520.v11i2.72534.
- [2] I. Aida Sapitri and M. Fikry, "Pengklasifikasian Sentimen Ulasan Aplikasi Whatsapp Pada Google Play Store Menggunakan Support Vector Machine," *J. TEKINKOM*, vol. 6, no. 1, pp. 1–7, 2023, doi: 10.37600/tekinkom.v6i1.773.
- [3] E. Sera, Hazriani, Mirfan, and Yuyun, "Analisis Sentimen Ulasan Produk di E-Commerce Bukalapak Menggunakan Natural Language Processing," *Pros. SISFOTEK*, pp. 237–243, 2023, [Online]. Available: <http://www.seminar.iaii.or.id/index.php/SISFOTEK/article/view/406%0Ahttp://www.seminar.iaii.or.id/index.php/SISFOTEK/article/download/406/338>
- [4] I. M. Karo Karo, J. A. Karo Karo, Y. Yuniyanto, H. Hariyanto, M. Falah, and M. Ginting, "Analisis Sentimen Ulasan Aplikasi Info BMKG di Google Play Menggunakan TF-IDF dan Support Vector Machine," *J. Inf. Syst. Res.*, vol. 4, no. 4, pp. 1423–1430, 2023, doi: 10.47065/josh.v4i4.3943.
- [5] R. Maulana, A. Voutama, and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store menggunakan Algoritma NBC," *J. Teknol. Terpadu*, vol. 9, no. 1, pp. 42–48, 2023, doi: 10.54914/jtt.v9i1.609.
- [6] M. Diki Hendriyanto, A. A. Ridha, and U. Enri, "Analisis Sentimen Ulasan Aplikasi Mola Pada Google Play Store Menggunakan Algoritma Support Vector Machine Sentiment Analysis of Mola Application Reviews on Google Play Store Using Support Vector Machine Algorithm," *J. Inf. Technol. Comput. Sci.*, vol. 5, no. 1, pp. 1–7, 2022.
- [7] A. Nurian, "Analisis Sentimen Ulasan Pengguna Aplikasi Google Play Menggunakan Naïve Bayes," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3s1, pp. 829–835, 2023, doi: 10.23960/jitet.v11i3s1.3348.
- [8] T. Tinaliah and T. Elizabeth, "Analisis Sentimen Ulasan Aplikasi PrimaKu Menggunakan Metode Support Vector Machine," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 4, pp. 3436–3442, 2022, doi: 10.35957/jatisi.v9i4.3586.
- [9] I. Akbar and M. Faisal, "Perbandingan Analisis Sentimen PLN Mobile: Machine Learning vs. Deep Learning," *J. Inf. Technol. Comput. Sci.*, vol. 8, no. 1, pp. 01–10, 2024.
- [10] N. M. Farhan and B. Setiaji, "Indonesian Journal of Computer Science," *Indones. J. Comput. Sci.*, vol. 12, no. 2, pp. 284–301, 2023, [Online]. Available: <http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135>
- [11] A. Hepatitis *et al.*, "Perbandingan Metode Decision Tree dan Support Vector Machine untuk Analisis Sentimen pada Instagram Mengenai Kinerja PSSI," 2020.
- [12] D. Muhidin and A. Wibowo, "Perbandingan Kinerja Algoritma Support Vector Machine dan K-Nearest Neighbor Terhadap Analisis Sentimen Kebijakan New Normal," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.)*, vol. 5, no. 2, p. 153, 2020, doi: 10.30998/string.v5i2.6715.
- [13] A. Supian, B. Tri Revaldo, N. Marhadi, L. Efrizoni, and R. Rahmaddeni, "Perbandingan Kinerja Naïve Bayes Dan Svm Pada Analisis Sentimen Twitter Ibukota Nusantara," *J. Ilm. Inform.*, vol. 12, no. 01, pp. 15–21, 2024, doi: 10.33884/jif.v12i01.8721.
- [14] F. O. Dayera, Musa Bundaris Palungan, "Komparasi Algoritma Machine Learning Untuk Menganalisis Sentimen Ulasan Pada Aplikasi Digital Korlantas Polri," *G-Tech J. Teknol. Terap.*, vol. 8, no. 1, pp. 186–195, 2024, [Online]. Available: <https://ejournal.uniramalang.ac.id/index.php/g-tech/article/view/1823/1229>
- [15] M. H. Wicaksono, M. D. Purbolaksono, and S. Al Faraby, "Perbandingan Algoritma Machine Learning untuk Analisis Sentimen Berbasis Aspek pada Review Female Daily," *eProceedings Eng.*, vol. 10, no. 3, pp. 3591–3600, 2023.
- [16] R. A. E. V. T. Sapanji, D. Hamdani, and P. Harahap, "Sentiment Analysis of the Top 5 E-commerce Platforms in Indonesia using Text Mining and Natural Language Processing (NLP)," *J. Appl. Informatics Comput.*, vol. 7, no. 2, pp. 202–211, 2023, doi: 10.30871/jaic.v7i2.6517.
- [17] M. Scientifict and J. Volume, "Perbandingan Algoritma Decision Tree dan Deep Learning dalam Prediksi Masalah Kesehatan berdasarkan Kebiasaan Gaya Hidup," 2024.
- [18] E. Setia Budi, A. Nofriyaldi Chan, P. Priscillia Alda, and M. Arif Fauzi Idris, "RESOLUSI : Rekeyasa Teknik Informatika dan Informasi Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra Menggunakan Algoritma Convolutional Neural Network," *Media Online*, vol. 4, no. 5, p. 509, 2024, [Online]. Available: <https://djournal.com/resolusi>

- [19] N. Lubis, M. Z. Siambaton, and R. Aulia, "Implementasi Algoritma Deep Learning pada Aplikasi Speech to Text Online dengan Metode Recurrent Neural Network (RNN)," *sudo J. Tek. Inform.*, vol. 3, no. 3, pp. 113–126, 2024, doi: 10.56211/sudo.v3i3.583.
- [20] A. R. Prananda, E. L. Frannita, E. Pramitaningrum, A. Hidayat, W. B. Setiawan, and N. Purwaningsih, "Klasifikasikan Jenis Cacat Kulit Menggunakan SMOTE-GoogLeNet," *JITU J. Inform. Technol. Commun.*, vol. 8, no. 1, pp. 21–30, 2024, doi: 10.36596/jitu.v8i1.1341.
- [21] R. A. Nurdian, Mujib Ridwan, and Ahmad Yusuf, "Komparasi Metode SMOTE dan ADASYN dalam Meningkatkan Performa Klasifikasi Herregistrasi Mahasiswa Baru," *J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 1, pp. 24–32, 2022, doi: 10.28932/jutisi.v8i1.4004.