ISSN: 2339-2541
JURNAL GAUSSIAN, Volume 14, Nomor 2, Tahun 2025, Halaman 345 - 355
Online di: https://ejournal3.undip.ac.id/index.php/gaussian/

IMPLEMENTASI METODE *CONVOLUTIONAL NEURAL NETWORK* UNTUK KLASIFIKASI SENTIMEN ULASAN PENGGUNA APLIKASI MYPERTAMINA

Arta Marisa Pardede[1*], Mustafid[2], Sugito[3]
[1,2,3] Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro
***e-mail**: artamarisa1@gmail.com

DOI: 10.14710/j.gauss.14.2.345-355

Article Info:
Received: 2024-10-13
Accepted: 2025-10-03
Available Online: 2025-10-14

Keywords:
Classification; *Word2Vec; CNN;* MyPertamina

Abstract: *Sentiment analysis has gained significant attention in recent years, with researchers employing various machine learning and deep learning techniques. While Convolutional Neural Networks (CNN) were initially designed for image processing, their effectiveness in text classification tasks has been established. This study focuses on enhancing sentiment classification performance for customer reviews of MyPertamina by utilizing a CNN model with Word2Vec embeddings. The research involves analyzing user reviews obtained from the Google Play Store for MyPertamina's mobile application. The CNN model, incorporating Word2Vec embeddings, is trained to classify these reviews into positive and negative sentiment categories. Experimental evaluations reveal that the optimal hyperparameters for the Word2Vec model are a window size of 5 and a word embedding dimension of 300. Regarding the CNN model, both the dropout rate and learning rate significantly impact classification performance. The best results are achieved with a learning rate of 0.001 and a dropout rate of 0.3. The findings demonstrate that the CNN model with Word2Vec embeddings achieves an impressive accuracy of 96.14% in classifying customer reviews within the MyPertamina application. This underscores the efficacy of employing this approach to improve sentiment classification for customer feedback.*

1. PENDAHULUAN

MyPertamina adalah Inovasi yang dilakukan pemerintah dalam penerapan e-government yang dinilai dapat menyederhanakan persyaratan administrasi. Aplikasi MyPertamina dirilis untuk memastikan bahwa pembeli BBM bersubsidi jenis pertalite dan solar adalah memang masyarakat yang membutuhkan. MyPertamina cukup menjadi perbincangan masyarakat, terutama di *Google Play*. Analisis sentimen pada *Google Play* dapat menjadi pilihan penulis untuk melihat respon masyarakat terhadap kondisi layanan yang diberikan MyPertamina kepada pelanggan.

Analisis sentimen telah banyak dilakukan dalam beberapa tahun terakhir baik dengan menggunakan model machine learning konvensional maupun metode deep learning meskipun pada awalnya model deep learning seperti CNN dikembangkan untuk pengolahan citra digital ternyata juga mampu mengklasifikasikan teks. (Kim, 2014). Pada model arsitektur CNN dapat memilih jenis dan jumlah layer yang digunakan untuk mempengaruhi kebaikan model. Pada Word2vec adalah algoritma penyisipan kata yang diperkenalkan Oleh Mikolov et al., 2013. Word2vec dapat merepresentasikan sebuah kata ke dalam *N-length* menggunakan neural network dengan 3 *layer* : *input, projection/hidden,* dan *output.* Teks tidak dapat dikenali dan diproses secara langsung oleh komputer. Diperlukan konversi data ulasan dari bentuk teks ke bentuk numerik seperti vector. Diharapkan dari analisis sentimen dengan metode Deep Learning for NLP menggunakan algoritma CNN dan Word2Vec, dapat membantu menilai opini publik terhadap objek aplikasi MyPertamina secara lebih tepat.

2. TINJAUAN PUSTAKA

Text Mining merupakan proses untuk menghasilkan informasi terstruktur dari data teks yang semula tidak terstruktur (M. Bonzanini, 2016). Biasanya, *Text Mining* melibatkan langkah-langkah seperti parsing untuk mengorganisir teks input, termasuk penambahan fitur linguistik yang diperoleh dan penghapusan yang tidak relevan, serta menyimpannya ke dalam basis data. Selanjutnya, dilakukan analisis untuk mengidentifikasi pola-pola dalam data yang telah terstruktur, dan akhirnya dilakukan evaluasi dan interpretasi terhadap hasil yang diperoleh (Kumar & Bhatia, 2013).

Klasifikasi sentimen adalah bentuk klasifikasi teks di mana sepotong teks harus diklasifikasikan ke dalam salah satu kelas sentimen yang telah ditentukan (Munikar et al., 2019). Sebagian besar sentimen yang ditangkap dalam klasifikasi sentimen bersifat biner, hanya dua polaritas, yaitu positif dan negatif. Analisis yang cermat dapat memberikan hasil yang lebih tepat untuk sistem otomatis yang memprioritaskan penanganan keluhan pelanggan (Rao, 2019).

Preprocessing adalah salah satu teknik pada data *mining* yang di dalamnya terdapat perubahan data mentah menjadi format yang lebih terstruktur dan mudah dipahami (Andika et al., 2019). *Pre-processing* yang akan dilakukan adalah sebagai berikut : *Case folding, Remove Punctuation, Remove Punctuation,* Normalisasi Kata, *Remove Number, Strip White Space*

Pada penelitian ini, pemilihan fitur yang digunakan adalah stopwords removal, stemming, dan tokenizing. Tahap stopwords removal dilakukan untuk menghilangkan term yang irrelevant dengan subjek utama. Tahap stemming dilakukan untuk mengubah kata-kata yang memiliki imbuhan menjadi bentuk dasarnya dapat mengurangi jumlah indeks yang berbeda dalam suatu dokumen. Tahap tokenisasi merupakan proses memecah rangkaian kata dalam kalimat menjadi kata-kata tunggal.

Word2vec adalah algoritma penyisipan kata yang diperkenalkan Oleh Mikolov et al., 2013. *Word2vec* dapat merepresentasikan sebuah kata ke dalam *N-length* menggunakan neural network dengan 3 *layer* : *input, projection/hidden,* dan *output.* Langkah proses pemodelan model *Word2vec* menggunakan model *Skipgram* sebagai berikut:

1. Propagasi maju *Input layer*

Secara matematika dapat dinyatakan sebagai berikut :

$$h = x \times W_{ki}^{IH} \tag{2.1}$$

h merupakan *output* dari perkalian antara *input* kata X dengan matriks bobot W^{IH}dengan ukuran $V \times N$, selanjutnya menghitung nilai $u_{C,J}$ dari perkalian $h$dengan matriks bobot W^{HO} dengan ukuran $N \times V$ dan tidak ada fungsi aktivasi. Berikut ini formula untuk perhitungan nilai $u_{C,J}$:

$$u_{c,j} = h \times W_{ij}^{HO} \tag{2.2}$$

Akan tetapi, untuk *output layer* pada masing-masing *output* kata yang membagi bobot yang sama menghasilkan formula $u_{c,j} = u_j$ sehingga dapat melewatkan *output* masing-masing node kata melalui fungsi *softmax* sebagai berikut:

$$y_{c,j} = \frac{e^{(u_{c,j})}}{\sum_{j=1}^{v} e^{(u_j)}} \tag{2.3}$$

Dimana $y_{c,j}$ merupakan *output* dari node ke-j pada *output* kata ke-c dari *output layer*, $u_{c,j}$ merupakan *input* dari node ke-j pada *output* kata ke-c dari *output layer* dan u_j yang merupakan *input* dari node ke-j

2. Menghitung *loss function*

$$E = -\sum_{c=1}^{C} u_{j_c^*}$$
$$E = -\sum_{c=1}^{c} u_{j_c^*} + C \times \log \sum_{j=1}^{V} e^{(u_j)} \quad (2.4)$$

Dimana E merupakan *loss function*, j_c^* merupakan indeks *output* kata ke-c dan C merupakan banyaknya kata konteks sebelum dan sesudah kata *input*.
Langkah berikutnya adalah meng-*update hidden output layer* dengan bobot W^{IH}
3. Memperbarui bobot *hidden output layer*

$$DLF = \frac{\partial E}{\partial u_{c,j}} = y_{c,j} - t_{c,j} \quad (2.5)$$

Jika node ke-j dari *output* kata ke-c yang sebenarnya sama dengan 1, sebaliknya$t_{c,j} = 0$.$\frac{\partial E}{\partial u_{c,j}}$ merupakan nilai kesalahan dari prediksi node c,j pada *output layer*.
Selanjutnya hitung turunan $-E$ yang terhubung ke output dengan bobot W^{HO} menggunakan *chain rule*.

$$\frac{\partial E}{\partial W_{ij}^{HO}} = \sum_{c=1}^{c} (y_{c,j} - t_{c,j-}) \times h \quad (2.6)$$

Sekarang telah diketahui nilai gradien yang terhubung dengan bobot *output* matriks W^{HO}. Lalu dapat ditentukan persamaan gradien stokastiknya yaitu:

$$W_{ij}^{HO(baru)} = W_{ij}^{HO(lama)} - \eta \times \frac{\partial E}{\partial W_{ij}^{HO}} \quad (2.7)$$

Dengan $\eta > 0$ merupakan nilai *learning rate*.
4. Memperbarui bobot *input hidden layer*

Setelah pembaruan bobot pada *hidden layer* didapatkan, selanjutnya adalah menghitung turunan dari E yang terhubung ke *hidden layer*.

$$EH = \frac{\partial E}{\partial h_i} = \sum_{j=1}^{V} \frac{\partial E}{\partial u_j} \times \frac{\partial u_j}{\partial h_i}$$
$$EH = \sum_{j=1} \sum_{c=1} (y_{c,j} - t_{c,j})^T \times (W_{ij}^{HO(baru)})^T \quad (2.8)$$

Langkah selanjutnya adalah menghitung turunan dari E yang terhubung dengan bobot *input* W^{1H}

$$Z = x_k \times EH \quad (2.9)$$

Langkah terakhir adalah *update* bobot pada *input layer*.

$$W_{ij}^{IH(baru)} = W_{ij}^{IH(lama)} - \eta \times Z \quad (2.10)$$

Convolutional Neural Network (CNN) adalah jenis khusus dari jaringan saraf yang digunakan untuk memproses data dengan topologi jaringan yang diketahui (Goodfellow et al., 2016; Rokhana et al., 2019). Seperti jaringan saraf pada umumnya, CNN terdiri dari banyak neuron dengan bobot dan bias yang dapat dilatih, yang diorganisasikan dalam beberapa tahap. Masukan (input) dan keluaran (output) yang terdiri dari beberapa array yang

disebut sebagai feature map. Setiap tahap memiliki tiga layer utama, yaitu layer konvolusi, layer fungsi aktivasi, dan layer pooling. CNN memiliki beberapa lapisan, termasuk:

1. *Feature Learning*

Convolutional layer terdiri dari *input* $\boldsymbol{I}$, *filter* kernel $\boldsymbol{K}$ dengan ukuran k_1 dan k_2, dan bias $\boldsymbol{b}$ serta menghasilkan *output* berupa *feature map*. Ukuran *output convolutional layer* dipengaruhi oleh bentuk *input*-nya, ukuran kernel, *padding*, dan *strides* (Dumoulin & Visin, 2016). *Filter* akan digeser keseluruhan bagian dokumen dengan operasi "dot" sehingga menghasilkan *output* yang biasa disebut *feature map*. Persamaan 2.11 menunjukkan formula untuk *convolutional layer* di mana i menunjukkan indeks baris fitur pada data *input*, j menunjukkan indeks kolom fitur pada data *input*, m menunjukkan baris pada kernel konvolusi, dan n menunjukkan kolom pada kernel konvolusi.

$$(\boldsymbol{I} * \boldsymbol{K})_{IJ} = \sum_{m=0}^{U} \sum_{n=0}^{U} \boldsymbol{I}_{(i+m,j+n)} * \boldsymbol{K}_{q}(m,n) + b_q \tag{2.11}$$

Activation function adalah fungsi non-linear yang memungkinkan sebuah jaringan untuk menyelesaikan permasalahan non-trivial (Zufar & Setiyono, 2016). *Activation function* dalam CNN terletak pada perhitungan akhir keluaran *feature map*. ReLU merupakan salah satu fungsi aktivasi pada *Artificial Neural Network* yang saat ini banyak digunakan digunakan. ReLU merupakan *activation function* yang memiliki *range* antara 0 sampai tak terhingga. Persamaan 2.12 menunjukkan formula aktivasi ReLU.

$$f(x) = \max(x, 0) \tag{2.12}$$

Pooling layer adalah komponen penting kedua pada CNN setelah *convolution layer*. Operasi *pooling* dirumuskan sebagai berikut:

$$\hat{c} = \max\{c\} \tag{2.13}$$

c merupakan nilai tiap *feature map*. Hasil dari *pooling layer* merupakan suatu vektor yang terdiri atas nilai maksimum tiap feature map dan vektor tersebut memiliki jumlah elemen sebanyak m. Hal ini dikarenakan banyaknya *filter* berjumlah m pula.

2. *Classification*

Lapisan ini berguna untuk mengklasifikasikan tiap neuron yang telah diekstraksi fitur pada sebelumnya yang terdiri dari :

Flatten membentuk ulang fitur (*reshape feature map*) menjadi sebuah *vector* agar bisa kita gunakan sebagai *input* dari *fully-connected layer.*

Fully-connected akan menghitung skor kelas. *Layer* FC yaitu lapisan neuron yang terhubung sepenuhnya. Seluruh neuron pada lapisan FC memiliki hubungan dengan seluruh aktivasi pada lapisan sebelumnya.

Persamaan 2.15 menunjukkan fungsi *sigmoid activation* di mana x menunjukkan *input* dan i menunjukkan urutan *input*.

$$f(x_i) = \frac{1}{1 + e^{(-xi)}} \tag{2.15}$$

Dropout adalah sebuah metode di mana beberapa neuron secara acak dipilih dan dinonaktifkan selama proses pelatihan. Pendekatan ini memiliki efek positif terhadap

performa model selama pelatihan dan dapat mengurangi overfitting dalam arsitektur CNN, sehingga meningkatkan kinerja model (Fesseha et al., 2021).

Loss function adalah fungsi yang digunakan untuk mengukur kinerja jaringan saraf dalam memprediksi kelas target dengan membandingkan hasil prediksi dengan target yang sebenarnya (Bircanoğlu, 2017). Persamaan 2.16 menunjukkan rumus binary cross entropy di mana L adalah loss function, y adalah output aktual, dan $\hat{y}$ adalah output prediksi.

$$L = -(y \ln(\hat{y}) + (1-y) \ln(1-\hat{y})) \quad (2.16)$$

Optimizer merupakan algoritma yang digunakan untuk memperbaharui bobot dan bias agar *error* berkurang. Dengan inisialisasi dimana α adalah *learning rate,* ε adalah *epsilon*, $\beta_1, \beta_2 \epsilon (0,1)$ merupakan pangkat dari estimasi momen, $f(w)$ adalah fungsi *stochastic* objektif dengan parameter dengan W_0 merupakan bobot awal, m_0 merupakan momen vektor pertama dan v_o merupakan momen vektor kedua dimana t merupakan inisialisasi t.
Langkah dalam melakukan perhitungan dapat dilihat pada persamaan 2.17 sampai persamaan 2.22, sebagai berikut:

$$g_t = \nabla_w f_t(w_{t-1}) \quad (2.17)$$

$$m_t = \beta_1 . m_{t-1} + (1-\beta_1) . g_t \quad (2.18)$$

$$v_t = \beta_2 . v_{t-1} + (1-\beta_2) . g_{t^2} \quad (2.19)$$

$$\widehat{m_t} = \frac{m_t}{(1-\beta_1^t)} \quad (2.20)$$

$$\widehat{v_t} = \frac{v_t}{(1-\beta_{1)}^t} \quad (2.21)$$

$$w_t = w_{t-1} - \alpha \times \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}+\varepsilon}} \quad (2.22)$$

Dengan t adalah $t = 1,2$, g adalah perubahan parameter, ∇w adalah gradien parameter, $f(w)$ merupakan fungsi *stochastic* objektif dengan parameter, m adalah momen vector pertama dan v adalah momen vector kedua sedangkan $m^{\wedge}$ merupakan perubahan momen vector pertama dan $v^{\wedge}$ merupakan perubahan momen vector kedua. β_1 adalah beta 1, β_2 adalah beta 2, ε adalah epsilon, α adalah *learning rate* dan w adalah bobot.

Penelitian ini menggunakan teknik evaluasi dengan menghitung *accuracy* menggunakan bantuan *confusion matric*. *Accuracy* digunakan untuk mengukur keakuratan model dalam memprediksi label kelas pada tuple. *Confusion matrix* untuk mencari nilai *accuracy* pada klasifikasi data dua kelas.

Tabel 1. Confusion Matrix Mencari Nilai Accuracy

Confusion Matrix		*Predicted*	
		Negative (-)	Positive (+)
Actual	Negative (-)	TN	FP
	Positive (+)	FN	TP

Akurasi digunakan untuk mengukur persentase input pada *test set* yang peng-klasifikasinya sudah diberi label dengan benar. Menurut (Han *dkk*, 2012) nilai akurasi dapat dihitung dengan formulas sebagai berikut.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.23)$$

3. METODE PENELITIAN

Dalam penelitian ini, digunakan jenis data sekunder. Data yang digunakan adalah ulasan atau review yang diambil dari *Google Play Store* menggunakan teknik scraping. Proses

scraping dilakukan menggunakan platform Web Google Colaboratory dan bahasa pemrograman Python pada tanggal 4 Mei 2023. Sumber data diambil dari alamat website MyPertamina di https://play.google.com/store/apps/details?id=com.dafturn.mypertamina. Data yang diambil terdiri dari 5000 ulasan atau review terbaru dalam bahasa Indonesia yang ditulis oleh pengguna aplikasi MyPertamina.

Langkah-langkah dalam analisis data yang dilakukan dalam mengerjakan skripsi :

1. Pengumpulan data, Pengumpulan data adalah tahapan pengumpulan data merupakan proses mencari dan menjelaskan data yang akan digunakan dalam penelitian.
2. Text preprocessing, Data yang terkumpul dari tahap pengumpulan data akan di *pre-processing* untuk proses pelatihan model.
3. Pelatihan model, Pelatihan model bertujuan untuk membentuk model yang digunakan untuk klasifikasi *ulasan* yang mengandung sentiment negative dan sentiment positif menggunakan *Convolutional Neural Network* dengan model *Word2vec* pada *Google Play.*
4. Validasi, Validasi dilakukan untuk memvalidasi hasil akurasi yang telah diperoleh pada tahap pelatihan model.
5. Pengujian model dan evaluasi, tahap berikutnya setelah validasi adalah pengujian model untuk mengetahui kinerja model dalam klasifikasi *sentiment y*ang mengandung sentiment negative dan positif.

4. HASIL DAN PEMBAHASAN

Dalam penelitian ini, dilakukan pengumpulan data dengan menggunakan teknik *scraping* menggunakan bantuan *Google Colaboratory*. Data yang diambil adalah ulasan atau review dari aplikasi MyPertamina di *Google Play* sebanyak 5000 data terbaru. Proses *scraping* dilakukan pada tanggal 4 Mei 2023. Pada penelitian ini menggunakan *software* GUI-Phyton.

Tahapan pre-processing ini dilakukan untuk mengatasi kendala data yang tidak terstruktur. Berikut adalah langkah-langkahnya:

1. Case Folding: Mengubah seluruh teks menjadi huruf kecil agar seragam..
2. Remove URL: Menghapus link URL dalam tweet yang ditandai dengan kata “http://”.
3. Unescape HTML: Menghapus jejak karakter markup seperti <b>str<b>, <audio>str<audio> dan sebagainya.
4. Remove mention: Menghilangkan kata yang mengandung “@” baik berupa mention, reply, maupun retweet.
5. Remove number: Menghapus semua angka yang terdapat dalam dokumen teks.
6. Remove punctuation: Menghapus karakter non alphabet dari dokumen teks yang biasanya merupakan tanda baca.
7. Remove emoticon: Menghapus emoticon yang menggambarkan emosi pengguna terutama jika emoticon tidak didefinisikan.
8. Strip white spaces: Menghapus beberapa karakter yang digantikan dengan spasi sehingga menyebabkan banyak spasi berlebih dalam dokumen.
9. Normalisasi kata: Mengubah kata yang tidak baku menjadi kata baku sesuai dengan Kamus Besar Bahasa Indonesia (KBBI).

Pada tahap stopwords removal digunakan daftar *stopwords* manual sehingga daftar stopwords tersebut disatukan di dalam satu kamus sehingga kamus *stopwords* yang digunakan berisi 1082 kata. Setelah dilakukan tahap *stopwords removal*, *term* yang terbentuk adalah 4236 *term*. Tahap *stemming* menggunakan library Sastrawi pada Python. Tahap *tokenizing* memisahkan kata-kata dalam sebuah kalimat menjadi unit-unit kata tunggal yang

disebut dengan token dengan tujuan untuk mendapatkan potongan kata yang bermakna dan bernilai dalam dokumen teks.

Proses pemberian label pada data dilakukan secara otomatis menggunakan kamus positif dan kamus negatif dengan menghitung skor sentimen. Dalam pelabelan ini, skor sentimen dihitung menggunakan kamus lexicon. Kamus lexicon merupakan sebuah metode klasifikasi yang menggunakan kamus yang berisi kata-kata opini untuk menentukan apakah suatu data teks memiliki sentimen positif atau negatif.

Penelitian ini menggunakan data latih dan data uji, dilakukan perbandingan antara keduanya sebesar 80%:20%. Pembagian presentase tersebut mengartikan bahwa 3817 data *review* dijadikan sebagai data latih dan sisanya 955 data *review* dijadikan sebagai data uji. 955 data yang digunakan sebagai data uji akan diprediksi kelasnya oleh *classifier* dengan menggunakan algoritma *Convolutional Neural Network* menggunakan Word2Vec Word Embeddings berdasarkan data latih yang telah dilakukan pengujian.

Skenario pengujian membahas mengenai data pengujian dan scenario pengujian klasifikasi sentimen ulasan pengguna aplikasi MyPertamina menggunakan *Convolutional Neurak Network* dengan model *Word2vec.* Data yang digunakan dalam pengujian ini terdiri dari ulasan atau review yang diklasifikasikan ke dalam dua kelas, yaitu positif dan negatif. Ulasan pada pengujian ini direpresentasikan menjadi vektor sekuensial.

Representasi vector sekuensial dari suatu data teks merupakan hal yang sangat berpengaruh terhadap hasil klasifikasi. Oleh karena itu, penelitian ini melakukan pengujian terhadap representasi vektor dari teks. Representasi vektor dengan akurasi terbaik digunakan pada proses pelatihan model *Convolutional Neural Network.*

Nilai parameter yang diuji pada nilai *learning rate* yaitu “0,01”, “0,001”, dan “0,0001” sedangkan untuk nilai pada parameter *dropout* adalah “0,2”, “0,3”, dan “0,4”. Nilai-nilai parameter ini dikombinasikan untuk mendapatkan model terbaik. Kinerja dari setiap kombinasi dilihat berdasarkan nilai akurasi. Nilai akurasi dari tiap kombinasi parameter dibandingkan untuk menentukan kombinasi terbaik.

1. Skenario 1: Menentukan Representasi Vektor terbaik untuk model Pelatihan dan Validasi

Skenario 1 adalah skenario yang dilakukan untuk mengetahui seberapa pengaruh representasi vektor dari data teks dalam mengklasifikasi teks yang mengandung kelas positif dan negatif. Representasi vektor yang diuji yaitu representasi vektor kata hasil *Word2vec* dan penyempurnaan representasi vektor kata hasil *Word2vec*. Penyempurnaan representasi vektor dilakukan dengan representasi vektor hasil pelatihan *Word2vec* dilatih kembali pada model pelatihan CNN.

2. Skenario 2: Mencari Nilai *Learning rate* dan *Dropout* Terbaik

Skenario 2 adalah scenario yang digunakan untuk menilai pengaruh dari parameter *learning rate* dan *dropout* terhadap kinerja model CNN dalam mengklasifikasi ulasan yang mengandung kelas positif dan negatif. Nilai parameter *learning rate* yang diuji adalah “0,01”, “0,001”, dan “0,0001” sedangkan untuk nilai parameter *dropout* adalah “0,2”, “0,3” dan “0,4”.

3. Hasil dan Analisa Skenario 1

Skenario 1 melakukan analisa berdasarkan representasi vektor kata yang digunakan dalam pelatihan model *Convolutional Neural Network* untuk mengklasifikasikan tweet yang mengandung kelas positif dan negatif. Representasi vektor yang diuji yaitu representasi vektor kata hasil Word2vec dan penyempurnaan representasi vektor kata hasil Word2vec.

Model yang digunakan untuk mencari representasi vektor terbaik adalah model CNN dengan kernel bigram. Hasil akurasi training dan akurasi validasi terbaik dari penggunaan representasi vektor kata tersebut dapat dilihat pada Tabel 2.

Tabel 2. Hasil Analisa Akurasi Skenario 1

Representasi Vektor	Akurasi	
	Training	Validasi
Representasi Vektor Kata Word2vec	0.9970	0.9694
Penyempurnaan Representasi Vektor Kata Word2vec	0.9970	0.9694

Representasi Vektor Kata Word2vec

	precision	recall	f1-score	support
0	0.9801	0.9740	0.9787	353
1	0.9732	0.9790	0.9775	333
accuracy			0.9781	686
macro avg	0.9781	0.9782	0.9781	686
weighted avg	0.9781	0.9781	0.9781	686

Penyempurnaan Representasi Vektor Kata Word2vec

	precision	recall	f1-score	support
0	0.9829	0.9773	0.9787	353
1	0.9760	0.9820	0.9776	333
accuracy			0.9781	686
macro avg	0.9780	0.9782	0.9781	686
weighted avg	0.9782	0.9781	0.9781	686

Karena untuk percobaan Word2Vec baseline maupun penyempurnaan Word2Vec diperoleh hasil akurasi yang sama, maka dapat dilihat dari metrik akurasi yang lain seperti Precision, Recall, dan F1-Score. Dimana Recall dan F1-Score pada penyempurnaan Word2Vec lebih unggul pada kelas positif serta Recall pada penyempurnaan Word2Vec lebih unggul pada kelas negatif. Maka untuk percobaan eksperimen learning rate dan dropout rate menggunakan Word2Vec dari hasil penyempurnaan Word2Vec.

Dari skenario penelitian ini representasi vektor kata yang akan digunakan untuk pelatihan model CNN dengan mengkonkatenasikan bigram, trigram dan fourgram adalah representasi vektor kata hasil Word2vec karena menghasilkan akurasi validasi yang lebih baik dibandingkan dengan menggunakan penyempurnaan representasi vektor kata hasil Word2vec. Representasi vektor kata Word2vec yang digunakan langsung tanpa penyempurnaan dapat mengurangi komputasi pada proses pelatihan model CNN.

4. Hasil dan Analisa Skenario 2

Model CNN yang digunakan pada skenario pengujian 2 adalah model CNN yang akan mengkonkatenasikan hasil max pooling dengan nilai kernel size 2 (bigram), nilai kernel size

3 (trigram), dan nilai kernel size 4 (fourgram). Kernel size merupakan ukuran yang dipakai untuk mengukur similarity (kesamaan). Digabungkannya hasil max pooling dari bigram, trigram dan fourgram berpengaruh terhadap nilai akurasinya yang menjadi semakin baik dikarenakan mampu menangkap fitur yang lebih kecil dan kompleks serta jumlah informasi yang diekstrak menjadi lebih banyak. Skenario pengujian 2 melakukan analisa berdasarkan kombinasi parameter antara learning rate dan dropout yang terbaik dengan konsep K-Fold Cross Validation dimana nilai K yang digunakan adalah 5. Pengaruh dari kombinasi parameter ini dihitung untuk mendapatkan nilai akurasi. Tabel 12 menunjukkan kombinasi parameter yang digunakan beserta akurasi training dan akurasi validasi rata-rata pada setiap kombinasi parameter.

Tabel 3. Hasil Analisa Akurasi Skenario 2

Kombinasi Parameter	Rata-rata Akurasi	
	Training	Validasi
***Learning rate* = 0,0001 *; Dropout* = 0,2**	0.9899	0.9598
***Learning rate* = 0,001 *; Dropout* = 0,2**	0.9832	0.9605
***Learning rate* = 0,01 *; Dropout* = 0,2**	0.9827	0.9597
***Learning rate* = 0,0001 *; Dropout* = 0,3**	0.9712	0.9588
***Learning rate* = 0,001 *; Dropout* = 0,3**	0.9755	0.9614
***Learning rate* = 0,01 *; Dropout* = 0,3**	0.9713	0.9566
***Learning rate* = 0,0001 *; Dropout* = 0,4**	0.9690	0.9527
***Learning rate* = 0,001 *; Dropout* = 0,4**	0.9700	0.9594
***Learning rate* = 0,01 *; Dropout* = 0,4**	0.9742	0.9535

Akurasi training rata-rata terbaik yaitu pada kombinasi parameter dengan nilai learning rate 0,0001 dan nilai dropout 0,2 yaitu sebesar 98,99%. Akurasi validasi rata-rata terbaik terdapat pada beberapa kombinasi parameter learning rate 0,001 dan nilai dropout 0,3 yaitu sebesar 96,14%.

Kombinasi nilai parameter learning rate dan dropout pada penelitian ini untuk klasifikasi ulasan positif dan negatif pada data testing berturut-turut adalah 0,001 dan 0,3 karena melihat dari nilai akurasi validasi rata-rata yang terbaik. Dropout bernilai besar maka unit yang dibuang lebih banyak sehingga menyebabkan lebih banyak makna semantik yang terbuang. .

Parameter learning rate memiliki dampak pada tingkat akurasi pada data training dan data validasi. Berdasarkan Tabel 3, dapat diamati bahwa learning rate 0,001 menghasilkan rata-rata akurasi validasi yang lebih tinggi. Hal ini menunjukkan bahwa dengan learning rate 0,001, model dapat mempelajari bobot secara lebih optimal.

5. Model Evaluation

Pada tahap ini, evaluasi dilakukan untuk menguji model yang telah dibuat dengan mengevaluasi hasil klasifikasi yang telah dilakukan. Pada proses klasifikasi dengan metode *Convolutional Neural Network* menggunakan jumlah epoch 10, didapatkan confusion matrix seperti berikut ini.

Tabel 4. Confusion Matrix *CNN*

Prediksi	Aktual	
	Negatif	Positif
Negatif	348	5
Positif	7	326

Dari *confusion matrix* diatas maka bisa dihitung nilai matriks evaluasi sebagai berikut :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{348 + 326}{348 + 326 + 7 + 5} = 0{,}9825$$

$$Recall = \frac{TP}{TP + FN} = \frac{326}{326 + 5} = 0{,}9848$$

$$Precision = \frac{TP}{TP + FP} = \frac{326}{326 + 7} = 0{,}9789$$

Nilai akurasi klasifikasi model CNN menggunakan data *testing* adalah 0,9825 atau 98,25% artinya metode CNN yang dikembangkan dapat mengenali data sentimen aplikasi MyPertamina dengan tepat sebesar 98,25%.

5. KESIMPULAN

Kesimpulan dari penelitian yang dilakukan untuk menganalisis sentimen pengguna aplikasi MyPertamina mengenai opini publik berdasarkan *reviews* pada *Google Play* menggunakan *Convolutional Neural Network* dengan model *Word2vec* adalah nilai parameter terbaik yang terpilih untuk model *Word2Vec* adalah ukuran *window* sebesar 5, dimensi *word embedding* sebesar 300 sedangkan untuk CNN adalah pada model klasifikasi reviews, learning rate dan dropout memiliki pengaruh terhadap kinerja model. Nilai learning rate 0,001 dan nilai dropout 0,3 memberikan kinerja terbaik. Dalam klasifikasi reviews pada aplikasi MyPertamina menggunakan metode *Convolutional Neural* Network (CNN) dengan model Word2vec, hasil terbaik yang berhasil dicapai adalah akurasi sebesar 96,14%.

DAFTAR PUSTAKA

Andika, L. A., Amalia, P., & Azizah, N. (2019). *Analisis Sentimen Masyarakat terhadap Hasil Quick Count Pemilihan Presiden Indonesia 2019 pada Media Sosial Twitter Menggunakan Metode Naive Bayes Classifier*. *2*(1), 34–41.

Fesseha, A., Xiong, S., Emiru, E. D., Diallo, M., Dahou, A. (2021). Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya. *Information (Switzerland)*, *12*(2), 1–17. https://doi.org/10.3390/info12020052

Goodfellow, Y. Bengio, A. Courville, Deep Learning. Cambridge: The MIT Press, 2016.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1746–1751. https://doi.org/10.3115/v1/d14-1181

Kumar, L., Bhatia, P. K. (2013). *Available Online at www.jgrcs.info TEXT MINING : CONCEPTS , PROCESS AND APPLICATIONS*. *4*(3), 36–39.

Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.

Munikar, M., Shakya, S., Shrestha, A. (2019). Fine-grained Sentiment Classification using BERT. *International Conference on Artificial Intelligence for Transforming Business and Society, AITB 2019*, *November*. https://doi.org/10.1109/AITB48515.2019.8947435

Qeis, M. I. (2015). Aplikasi Wordcloud sebagai Alat Bantu Analisis Wacana. *International Conference on Language, Culture, and Society (ICLCS) - LIPI*, *November 2015*.

https://www.researchgate.net/publication/316736417
Rao, P. (2019). Fine-grained Sentiment Analysis in Python (Part1). https://towardsdatascience.com/fine-grained-sentiment-analysis-in-pythonpart-1-2697bb111ed4.
Zufar, M., & Setiyono, B. (2016). Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time. *JURNAL SAINS DAN SENI ITS, 5*, A-74.