

# Pengaruh Parameter *Word2Vec* terhadap Performa *Deep Learning* pada Klasifikasi Sentimen

Dwi Intan Af'idah<sup>1)</sup>, Dairoh<sup>2</sup>, Sharfina Febbi Handayani<sup>3</sup>, Riszki Wijayatun Pratiwi<sup>4</sup>

<sup>1,2,3,4</sup>Jurusan Teknik Informatika, Politeknik Harapan Bersama, Tegal

<sup>1,2,3,4</sup>Jalan Mataram No 9, Pesurungan Lor, Kota Tegal, 52143, Indonesia

email: <sup>1</sup>dwiintanafidah@poltektegal.ac.id, <sup>2</sup>dairoh@poltektegal.ac.id, <sup>3</sup>sharfina.handayani@poltektegal.ac.id,

<sup>4</sup>riszkipratiwi@poltektegal.ac.id

**Abstract –** The difficulty of sentiment classification on this big data can be overcome using deep learning. Before the deep learning training and testing process is carried out, a word features extraction process is needed. Word2Vec as a word features extraction is often used in sentiment classification pre-training because it can capture the semantic meaning of the text by representing a similar vector for each word that has a close meaning. Word2Vec has three parameters that affect the model learning process namely architecture, evaluation method, and dimensions. This study aims to determine the effect of each Word2Vec parameter on deep learning performance in sentiment classification. The accuracy results of the deep learning model were evaluated to determine the effect of the Word2Vec parameter. The results of this study indicate that the three Word2Vec parameters have an influence on the performance of the deep learning model in sentiment classification. The combination of Word2Vec parameters that produces the highest average accuracy include CBOW (Continuous Bag of Word) architecture, Hierarchical Softmax evaluation method, and a dimension of 100. CBOW produces better performance, because it has slightly better accuracy for words that often appear and in this research dataset there are many words that often appear. Hierarchical Softmax shows better results because it uses a binary tree model which makes words that occur rarely will inherit the vector representation above them. The dimension with a value of 100 produces better accuracy because it is in line with the number of datasets of 10,000 reviews.

**Kata Kunci –** *word2vec, hierarchical softmax, continuous bag of word, dimensoins, sentiment classification*

**Abstrak –** Kesulitan melakukan klasifikasi sentimen pada data yang besar ini dapat diatasi menggunakan deep learning. Sebelum dilakukan proses pelatihan dan pengujian deep learning, diperlukan suatu proses word features extraction. Word2Vec sebagai salah satu metode word features extraction yang sering digunakan dalam pra pelatihan klasifikasi sentimen karena dapat menangkap makna semantik teks dengan merepresentasikan vektor yang mirip untuk setiap kata yang memiliki kedekatan makna. Word2Vec memiliki tiga parameter yang berpengaruh pada proses pembelajaran model yaitu arsitektur, metode evaluasi, dan dimensi. Penelitian ini bertujuan untuk mengetahui pengaruh tiap parameter Word2Vec terhadap performa deep learning dalam klasifikasi sentimen. Hasil akurasi dari model deep learning dievaluasi untuk mengetahui pengaruh dari parameter Word2Vec. Hasil

**\*) penulis korespondensi:** Dwi Intan Af'idah

Email: dwiintanafidah@poltektegal.ac.id

penelitian ini menunjukkan bahwa ketiga parameter Word2Vec memiliki pengaruh terhadap performa model deep learning dalam melakukan klasifikasi sentimen. Kombinasi parameter Word2Vec yang menghasilkan rata-rata akurasi tertinggi antara lain, arsitketur CBOW (Continuous Bag of Word), metode evaluasi Hierarchical Softmax, dan dimensi bernilai 100. CBOW menghasilkan performa yang lebih baik, sebab memiliki akurasi yang sedikit lebih baik untuk kata-kata yang sering muncul dan pada dataset penelitian ini terdapat banyak kata yang sering muncul. Hierarchical Softmax menunjukkan hasil yang lebih baik karena menggunakan model binary tree yang menjadikan kata yang jarang muncul pasti akan mewaris representasi vektor di atasnya. Sementara dimensi bernilai 100 menghasilkan akurasi yang lebih baik dikarenakan selaras dengan jumlah dataset 10.000 ulasan.

**Kata Kunci –** *word2vec, hierarchical softmax, continuous bag of word, dimensi, klasifikasi sentimen*

## I. PENDAHULUAN

Klasifikasi sentimen digunakan untuk melakukan penilaian suatu entitas tekstual yang menunjukkan emosi atau sentimen. Tujuan utama dari klasifikasi sentimen teks adalah mengelompokkan suatu data teks agar ditemukan polaritasnya. Jumlah data yang besar berdampak pada semakin sulitnya melakukan klasifikasi sentimen. Kesulitan melakukan klasifikasi sentimen pada data yang besar ini dapat diatasi menggunakan *deep learning* [1].

Sebelum dilakukan proses pelatihan dan pengujian *deep learning*, diperlukan proses *word features extraction* lebih dulu dikarenakan *deep learning* tidak bisa secara langsung mengolah data dalam bentuk teks, sehingga data teks perlu direpresentasikan dulu ke dalam bentuk vektor. Selain itu, pemilihan metode *word features extraction* yang tepat dapat meningkatkan akurasi *learning model* yang lebih akurat [2].

Perkembangan *word features extraction* diawali dengan metode *Bag of Words* yang terdiri dari *Countvectorizer*, *N-gram*, dan *term frequency-inverse document frequency* (TF-IDF). *Bag of Words* seringkali disebut sebagai *features extraction* tradisional karena hasil dari representasi kata selanjutnya hanya dapat digunakan untuk pelatihan *statistical machine learning*. *Bag of Word* mentransformasikan teks ke dalam vektor berdasarkan frekuensi kemunculan kata dalam sebuah dokumen. Adapun ketidakmampuan *Bag of Word* dalam menangkap makna semantik dan sintaksis menjadi limitasi dari metode ini [3].

Perkembangan *machine learning* menjadi *deep learning* pada klasifikasi sentimen diiringi juga dengan perkembangan *word features extraction* yang dinamakan *word embedding*. *Word embedding* merupakan metode *unsupervised learning* menggunakan *neural network* [4]. *Word2Vec* sebagai salah satu metode *word embedding* yang dikembangkan oleh Mikolov sering digunakan dalam *pretrained* klasifikasi sentimen karena dapat menangkap makna semantik teks dengan cara merepresentasikan vektor yang mirip untuk setiap kata yang memiliki kedekatan makna [5] [6].

*Word2Vec* memiliki tiga parameter yang berpengaruh pada proses pembelajaran model yaitu arsitektur, metode evaluasi, dan dimensi. Ada dua jenis arsitektur *Word2Vec*, yaitu *Skip-gram* dan *Continuous Bag of Words* (CBOW). Adapun metode evaluasi *Word2Vec* juga terdapat 2 tipe, yaitu *Hierarchical Softmax* dan *Negative Sampling*. Sedangkan dimensi *Word2Vec* dapat diubah-ubah untuk mendapatkan akurasi model yang optimal [7].

Masing-masing jenis dari ketiga parameter yang dimiliki *Word2Vec* memiliki peranan dalam menghasilkan model terbaik pada metode *deep learning*. Karena itu, pengaruh tiap parameter *Word2Vec* terhadap performa *deep learning* dalam klasifikasi sentimen menjadi motivasi untuk mengetahui kinerja dari setiap parameter yang dimiliki *Word2Vec*. Hasil akhirnya akan diperoleh kombinasi parameter *Word2Vec* terbaik.

## II. PENELITIAN YANG TERKAIT

Pada perkembangannya, banyak teknik yang digunakan dalam merepresentasikan teks ke dalam bentuk vektor. Pada tahun 2003 Y. Bengio dkk. mengusulkan model *word embeddings* dimana kata-kata direpresentasikan sebagai vektor padat yang kontinyu. *Word embeddings* diperoleh dengan menggunakan *neural network* dengan arsitektur *feedforward* untuk menemukan kemunculan kata-kata yang muncul di jendela konteks. Model yang diusulkan menggunakan arsitektur neural network menghasilkan metode *Word2Vec*. Untuk melatih model prediksi konteks kata dengan mudah telah diperkenalkan menggunakan arsitektur CBOW dan *Skip-gram*. Model CBOW memprediksi konteks kata-kata dari kata-kata sebelumnya dan model *Skip-gram* memprediksi konteksnya kata tengah dari kata-kata yang ada di kiri atau di kanan dari kata yang diberikan. Metode ini pertama kali dikemukakan oleh T.Mikolov dkk. [7].

Penelitian yang baru terkait *Word2Vec* telah digunakan untuk bahasa Urdu [8]. Metode ini membantu membuat representasi vektor kata dari kata-kata pada bahasa Urdu yang dapat digunakan untuk pra pelatihan *Word2Vec*. Hasil penelitian ini menunjukkan bahwa *Word2Vec* mengungguli kinerja *word embedding* yang lain.

*Word2Vec* pada [9] penelitian lain digunakan untuk mentransformasikan teks berita berbahasa mandarin ke dalam vektor-vektor. Vektor tersebut selanjutnya dilakukan metode *similarity* untuk menentukan apakah berita tersebut tergolong dalam berita olahraga atau bukan. Hasil percobaan menunjukkan bahwa *Word2vec* dapat mencocokkan terkait kata-kata dalam urutan kesamaan sehingga pembaca dapat memahami berita dengan lebih baik.

Penelitian lainnya [10] menggunakan *Word2Vec* yang dikombinasikan dengan *deep learning* CNN-BiLSTM (*Convolutional Neural Network-Bidirectional Long Short*

*Term memory*) dalam melakukan analisis sentimen. Penelitian ini menggunakan *dataset* yang berasal dari situs Quora. Pada penelitian ini ditunjukkan bahwa *Word2Vec* dapat menempatkan kata-kata dengan arti yang sama di tempat yang dekat. Hal ini dilakukan dengan melatih sejumlah besar korpus untuk mendapatkan vektor kata, masalah homonya dapat diselesaikan.

*Word2Vec* pada penelitian [11] digunakan untuk mengklasifikasi kalimat pendek sentimental. Penelitian ini memberikan hasil yang lebih akurat untuk ekstraksi fitur melalui metode *Word2Vec* dan CNN (*Convolutional Neural Network*) untuk kalimat pendek ulasan film. Peningkatan akurasi ditunjukkan dengan model yang dilatih secara distribusi.

## III. METODE PENELITIAN

### A. *Word2Vec*

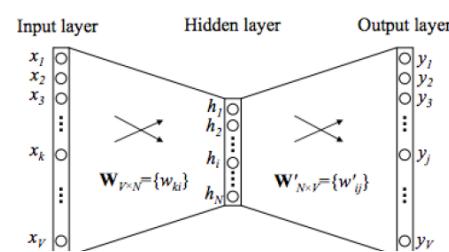
Metode *Word2Vec* diusulkan oleh Mikolov pada tahun 2013 yang mempertimbangkan korpus sebagai *input* dan *output* berupa vektor. *Word2Vec* mentransformasikan setiap kata unik sebagai vektor. Kelebihan *Word2Vec* yaitu dapat merepresentasikan kesamaan kontekstual dari dua kata pada vektor yang dihasilkan [12].

*Word2Vec* merupakan algoritma representasi vektor kata yang mampu mencapai kinerja terbaik dalam NLP (*Natural Language Processing*) dengan cara mengelompokan kata yang serupa memiliki vektor yang sama. *Word2Vec* menghitung representasi kata ke dalam vektor menggunakan *neural network*. Vektor kata yang dihasilkan adalah vektor ruang dimensi yang menangkap makna semantik dari kata [13].

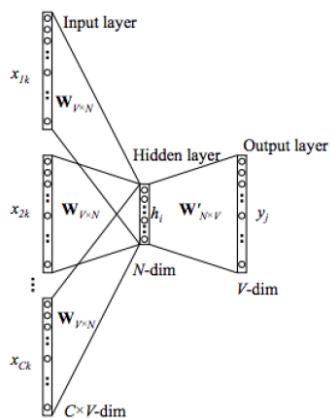
*Word2Vec* memiliki tiga parameter yang berpengaruh pada proses pembelajaran model yaitu arsitektur, metode evaluasi, dan dimensi. Masing-masing jenis dari ketiga parameter yang dimiliki *Word2Vec* memiliki pengaruh terhadap performa akurasi *deep learning*.

#### 1) Arsitektur *Word2Vec*

Arsitektur pada *Word2Vec* terdiri dari dua jenis yaitu, *Continuous Bag of Words* (CBOW) dan *Skip-gram*. CBOW memprediksi kata target berdasarkan konteksnya atau kata-kata di sekitarnya. Jadi, dalam CBOW posisi tertentu kata dalam kata-kata tetangga tidak masalah. (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013c). Model ini menggunakan konteks untuk memprediksi target kata. CBOW membutuhkan waktu pelatihan yang lebih cepat dan menghasilkan akurasi yang sedikit lebih baik untuk *frequent words* [14]. Gambar 1 dan Gambar 1 dan Gambar 2 menjelaskan arsitektur model CBOW.

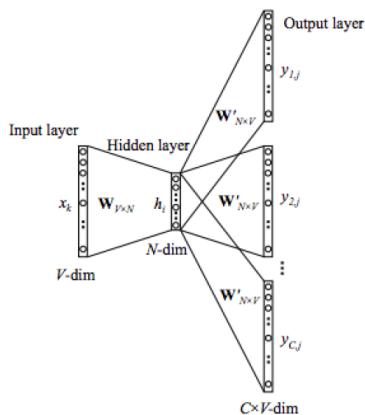


Gambar 1. CBOW Satu Kata Konteks [14]



Gambar 2. CBOW Multipel Kata Konteks [14]

Kebalikan dari CBOW, model *Skip-gram* justru Model ini menggunakan sebuah kata untuk memprediksi target konteks [7]. *Skip-gram* bekerja dengan baik dengan data pelatihan yang jumlahnya sedikit dan dapat merepresentasikan kata-kata yang dianggap langka [14]. Gambar 3 dan Gambar 3 merupakan detail dari arsitektur *Skip-gram*.

Gambar 3. *Skip-gram* Arsitektur [14]

## 2) Metode Evaluasi *Word2Vec*

Model *CBOW* dan *Skip-gram* memerlukan proses komputasi yang sangat besar karena semua bobot dalam *embedding matriks* harus diperbaharui untuk setiap kata target, kata konteks, atau pasangan kata target. Hal ini semakin tidak praktis untuk data dengan jumlah kosakata yang banyak [14]. Mikolov memberikan solusi dari masalah dengan metode evaluasi yaitu *Negative Sampling* dan *Hierarchical Softmax*.

*Negative Sampling* memungkinkan pelatihan untuk hanya memodifikasi sebagian kecil dari bobot, daripada semuanya untuk setiap sampel pelatihan. Sebagian kecil kata negatif atau kata-kata yang tidak diharapkan muncul dalam konteks dari kata target, dipilih dan hanya bobot negatif ini saja bobotnya diperbaharui. Akibatnya hanya sebagian kecil saja matriks bobot yang diperbarui [15].

Sedangkan metode evaluasi *Hierarchical Softmax* menggunakan representasi pohon biner dari lapisan keluaran dengan kata-kata sebagai daunnya dan, untuk setiap simpul,

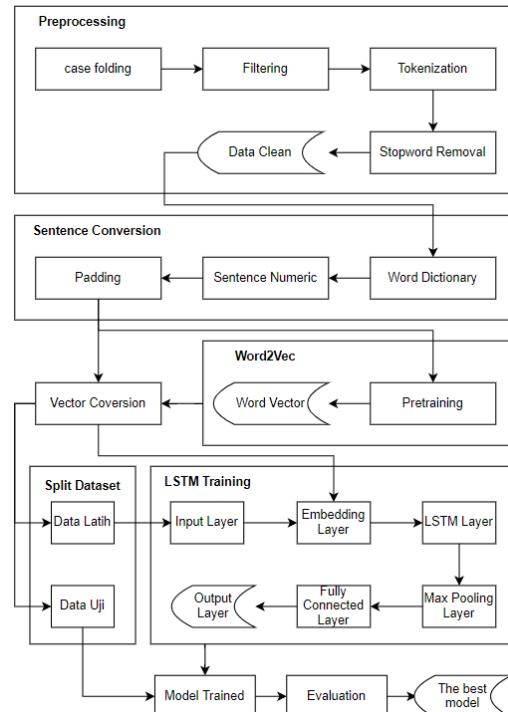
secara eksplisit mewakili probabilitas relatif dari simpul turunannya. Model *binary tree* ini menjadikan kata yang jarang muncul pasti akan mewarisi representasi vektor di atasnya [7].

## 3) Dimensi *Word2Vec*

Dimensi dari *Word2Vec* sangat tergantung pada korpus yang digunakan dalam pelatihan. Besaran dimensi yang rendah tidak akan berhasil menangkap makna semantik dari korpus, sedangkan besaran dimensi yang tinggi dapat menyebabkan waktu pemrosesan lebih lama. Oleh karena itu, penentuan dimensi yang optimal merupakan hal yang penting dalam mentransformasikan kata dalam bentuk vektor. Dimensi optimal dari suatu kosakata itu terjadi dengan memperhatikan dua aspek. Pertama, vektor yang dihasilkan dapat mendefinisikan kata-kata yang berbeda. Kedua, vektor yang dihasilkan dapat merepresentasikan semantik dari kata-kata [16].

## B. Alur Klasifikasi Sentimen

Alur klasifikasi sentimen dijelaskan pada Gambar 4.



Gambar 4. Alur Klasifikasi Sentimen

*Preprocessing* merupakan proses untuk merapikan data dan menghilangkan data yang tidak penting supaya memudahkan proses selanjutnya. *Preprocessing* dimulai dari proses *case folding*, *filtering*, *tokenization*, dan *stopword removal*. *Case folding* dilakukan untuk mengubah semua bentuk huruf ke dalam *lowercase*. *Filtering* merupakan proses untuk menghapus karakter ilegal seperti tanda baca, symbol, dan lain-lain. Sedangkan *tokenization* diperlukan untuk melakukan proses pemenggalan dokumen teks menjadi kata per kata. Proses terakhir *preprocessing* yaitu *stopword removal* yang bertujuan menghapus kata yang tidak penting yang tidak memiliki kontribusi pada klasifikasi sentimen [17].

Proses selanjutnya adalah *sentence conversion*. *Padding* merupakan tujuan utama dari *sentence conversion*. *Padding*

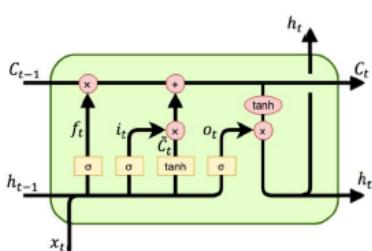
membuat data *input* mempunyai panjang yang sama dengan cara menambahkan kata “<pad>”. Hal ini menjadi penting karena *neural network* yang digunakan pada proses selanjutnya memerlukan masukan data dengan panjang yang sama. Adapun proses selanjutnya yaitu *Word2Vec* yang berguna untuk mentransformasikan teks ke dalam vektor [18].

Tahapan selanjutnya dilakukan proses *split dataset* menjadi data *training* dan data *testing*. Setelah itu, proses *training* dan *testing* model LSTM (*Long Short Term Memory*) dilakukan. Array hasil *preprocessing* teks merupakan masukan dari bagian *input layer*. Setiap kata kemudian masuk ke dalam *embedding layer* untuk ditransformasikan menjadi data vektor yang terdapat di *vector conversion*. Array hasil *embedding layer* *Word2Vec* lalu diproses oleh *LSTM layer*. Output LSTM selanjutnya menuju *pooling layer*. Hasil dari *pooling layer* menuju ke *flatten layer*, setelah itu melalui *dense layer* dengan fungsi aktivasi *sigmoid* yang kemudian akan mengeluarkan *output* dengan ukuran 1x1.

Setelah *training* dan *testing* model LSTM (*Long Short Term Memory*), dilakukan proses *evaluation*. Metode *Confusion matrix* digunakan untuk mengukur hasil kinerja model. Kemudian hasil dari setiap model dibandingkan sehingga didapatkan model terbaik.

### C. Deep Learning Long Short Term Memory

LSTM memiliki tiga jenis gates yaitu, *forget gate* ( $f_t$ ), *input gate* ( $i_t$ ), dan *output gate* ( $o_t$ ) seperti pada Gambar 5, *Forget gate* berfungsi untuk memutuskan informasi yang dihapus dari *cell*. *Input gate* berfungsi untuk memutuskan nilai dari input untuk diperbaharui pada *state memory*. *Output gate* berfungsi untuk memutuskan apakah yang dihasilkan *output* sesuai dengan *input* dan memori pada *cell* [19].



Gambar 5. Arsitektur LSTM [19]

*Long Short Term Memory* (LSTM) dikembangkan dari model *RNN Recurrent Neural network* (RNN) oleh Hochreiter dan Schmidhuber untuk mengatasi masalah difusi gradien neuron [20]. Pada bidang klasifikasi sentimen, LSTM memiliki kelebihan mampu memahami ketergantungan semantik jarak jauh sehingga cocok digunakan untuk data teks yang berdimensi cukup panjang [21].

## IV. HASIL DAN PEMBAHASAN

### A. Hasil

Hasil akhir dari penelitian ini adalah diperoleh kombinasi parameter *Word2Vec* terbaik. Adapun tiga parameter *Word2Vec* yang dapat mempengaruhi akurasi pembelajaran model yaitu arsitektur, metode evaluasi, dan dimensi. Dua jenis arsitektur *Word2Vec*, yaitu *Skip-gram* dan *Continuous*

*Bag of Words* (*CBOW*), sementara metode evaluasi *Word2Vec* juga terdapat 2 tipe, yaitu *Hierarchical Softmax* dan *Negative Sampling*. Adapun dimensi *Word2Vec* yang menjadi parameter pada penelitian ini yaitu: 300, 200, dan 100. Seluruh kombinasi parameter *Word2Vec* dapat dilihat pada Tabel 1.

Tabel 1. Kombinasi Parameter *Word2Vec*

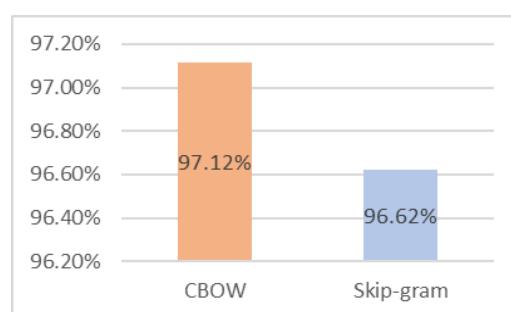
Identitas	Arsitektur	Metode Evaluasi	Dimensi
W2V 01	CBOW	Negative Sampling	300
W2V 02	CBOW	Negative Sampling	200
W2V 03	CBOW	Negative Sampling	100
W2V 04	CBOW	Hierarchical Softmax	300
W2V 05	CBOW	Hierarchical Softmax	200
W2V 06	CBOW	Hierarchical Softmax	100
W2V 07	<i>Skip-gram</i>	Negative Sampling	300
W2V 08	<i>Skip-gram</i>	Negative Sampling	200
W2V 09	<i>Skip-gram</i>	Negative Sampling	100
W2V 10	<i>Skip-gram</i>	Hierarchical Softmax	300
W2V 11	<i>Skip-gram</i>	Hierarchical Softmax	200
W2V 12	<i>Skip-gram</i>	Hierarchical Softmax	100

Seluruh kombinasi parameter *Word2Vec* selanjutnya diproses pada *LSTM layer* dengan parameter antara lain: (i) *dropout* (0,2; 0,5; dan 0,7), (ii) *learning rate* (0,001; 0,0001; dan 0,00001), (iii) *Adam optimizer*, (iv) *sigmoid* pada fungsi aktivasi *ouput*. Karena itu, kombinasi *Word2Vec* dan LSTM membentuk 108 eksperimen yang terdiri dari arsitektur CBOW dan *Skip-gram* masing-masing sebanyak 54, metode evaluasi *Negative Sampling* dan *Hierarchical Softmax* masing-masing sebanyak 54, serta dimensi 300, 200, dan 100 masing-masing sebanyak 36.

Pengaruh parameter *Word2Vec* terhadap performa LSTM pada klasifikasi sentimen dijelaskan sebagai berikut:

### 1) Pengaruh Arsitektur *Word2Vec* terhadap Akurasi Model

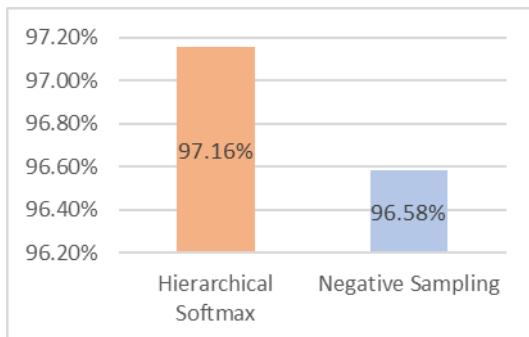
Gambar 6. menunjukkan bahwa rata-rata akurasi model yang lebih tinggi menggunakan arsitektur CBOW, dibandingkan menggunakan arsitektur *Skip-gram*. Rata-rata akurasi CBOW sebesar 97,12%, sedangkan rata-rata akurasi *Skip-gram* sebesar 96,62%.



Gambar 6. Rata-rata Akurasi Berdasarkan Arsitektur *Word2Vec*

- 2) Pengaruh Metode Evaluasi *Word2Vec* terhadap Akurasi Model

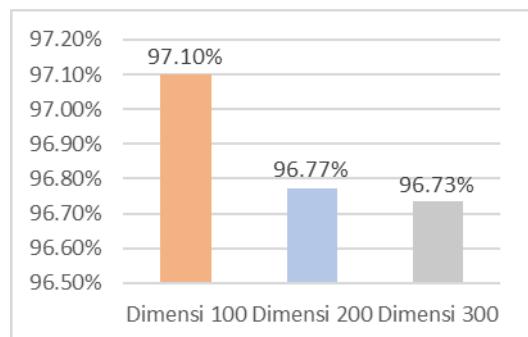
Gambar 7. menunjukkan pengaruh metode evaluasi *Word2Vec*. *Hierarchical Softmax* memiliki rata-rata akurasi sebesar 97,16% yang lebih tinggi dibandingkan *Negative Sampling* yang memiliki rata-rata akurasi sebesar 96,58%.



Gambar 7. Rata-rata Akurasi Berdasarkan Metode Evaluasi *Word2Vec*

- 3) Pengaruh Dimensi *Word2Vec* terhadap Akurasi Model

Gambar 8. menunjukkan bahwa dimensi sebesar 100 memberikan pengaruh yang lebih baik dibandingkan dimensi sebesar 200 dan 300. Adapun rata-rata akurasi dimensi 100 sebesar 97,10%, sedangkan akurasi dimensi 200 dan 300 berturut-turut sebesar 96,77% dan 96,73%.



Gambar 8. Rata-rata Akurasi Berdasarkan Dimensi *Word2Vec*

## B. Pembahasan

Arsitektur *Word2Vec* memiliki pengaruh terhadap performa model *deep learning*. Hal ini ditunjukan dengan arsitektur *Word2Vec* yang berbeda menghasilkan hasil rataprata akurasi model yang berbeda. Pada hasil penelitian ini arsitektur CBOW menunjukkan pengaruh yang lebih baik terhadap akurasi model dibandingkan arsitektur *Skip-gram*. CBOW memiliki akurasi yang sedikit lebih baik untuk kata-kata yang sering muncul (mikolov). Pada *dataset* ulasan yang digunakan penelitian ini terdapat banyak kata yang sering muncul, sehingga CBOW memiliki pengaruh yang lebih baik dibandingkan *Skip-gram*.

Metode evaluasi *Word2Vec* menjadi parameter yang perlu diketahui pengaruhnya terhadap hasil akurasi *deep learning*.

Metode evaluasi ada dua yaitu, *Hierarchical Softmax* dan *Negative Sampling*. Pada hasil penelitian ini menunjukkan bahwa *Hierarchical Softmax* berpengaruh lebih baik dibandingkan *Negative Sampling*. *Hierarchical Softmax* menggunakan model *binary tree* yang menjadikan kata yang jarang muncul pasti akan mewarisi representasi vektor di atasnya. Karena itu, dengan jumlah *dataset* penelitian ini yang cukup besar, maka lebih cocok menggunakan metode evaluasi *Hierarchical Softmax* yang menjadikan kata jarang muncul seolah diabaikan oleh *Word2Vec*.

Parameter *Word2Vec* lainnya yang diteliti pengaruhnya terhadap akurasi *deep learning* adalah dimensi. Dimensi yang terlalu kecil dan dimensi yang terlalu besar dapat mengurangi nilai akurasi. Hasil dari penelitian ini menunjukkan bahwa dimensi 100 merupakan dimensi yang paling optimal dalam pengaruhnya terhadap akurasi model. Sedangkan dimensi yang semakin besar menjadikan performa model semakin menurun. Hal ini terlihat sesuai hasil penelitian rata-rata akurasi dimensi 100, 200, dan 300 secara berturut-turut sebesar 97,10%, 96,77% dan 96,73%. Oleh karena itu, dimensi 100 selaras dengan jumlah *dataset* 10.000 ulasan. Sementara dimensi dengan ukuran yang lebih besar dapat menurunkan performa model.

## V. KESIMPULAN

Kesimpulan dari penelitian ini menunjukkan bahwa ketiga parameter *Word2Vec* memiliki pengaruh terhadap performa model *deep learning* dalam melakukan klasifikasi sentimen. Kombinasi parameter *Word2Vec* yang menghasilkan rata-rata akurasi tertinggi antara lain: arsitektur CBOW, metode evaluasi *Hierarchical Softmax*, dan dimensi bernilai 100. Akan tetapi, perbedaan performa antar jenis parameter menggambarkan bahwa ketiga parameter *Word2Vec* mempunyai performa yang kompetitif sehingga penggunaannya bergantung pada *dataset* dan permasalahan yang diselesaikan.

*CBOW* menghasilkan performa yang lebih baik, sebab memiliki akurasi yang sedikit lebih baik untuk kata-kata yang sering muncul dan pada *dataset* penelitian ini terdapat banyak kata yang sering muncul. *Hierarchical Softmax* menunjukkan hasil yang lebih baik karena menggunakan model *binary tree* yang menjadikan kata yang jarang muncul pasti akan mewarisi representasi vektor di atasnya. Dengan jumlah *dataset* penelitian ini yang cukup besar, maka lebih cocok menggunakan metode evaluasi *Hierarchical Softmax* yang menjadikan kata jarang muncul seolah diabaikan oleh *Word2Vec*. Adapun dimensi bernilai 100 menghasilkan akurasi yang lebih baik dikarenakan selaras dengan jumlah *dataset* 10.000 ulasan. Sementara dimensi dengan ukuran yang lebih besar dapat menurunkan performa model.

Penelitian selanjutnya disarankan menggunakan model *word embeddings* selain *Word2Vec*, seperti *FastText* dan *Glove*. Penggunaan *word embeddings* lainnya dapat dilakukan untuk mengetahui perbandingan performa ketiga *word embeddings* dalam pengaruhnya terhadap akurasi model *deep learning*.

## UCAPAN TERIMA KASIH

Terima kasih penulis sampaikan kepada pihak program studi Teknik Infomatika Politeknik Harapan Bersama Tegal

yang mendukung penulis untuk menyelesaikan penelitian ini.

## DAFTAR PUSTAKA

- [1] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 8, no. 4, 2018, doi: 10.1002/widm.1253.
- [2] Q. Huang, R. Chen, X. Zheng, and Z. Dong, "Deep sentiment representation based on CNN and LSTM," Proc. - 2017 Int. Conf. Green Informatics, ICGI 2017, pp. 30–33, 2017, doi: 10.1109/ICGI.2017.45.
- [3] A. Kedia and M. Rasu, Hands-On - Python Natural Language Processing, 2020.
- [4] A. Nurdin, B. Anggo Seno Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks," J. Tekno Kompak, vol. 14, no. 2, p. 74, 2020, doi: 10.33365/jtk.v14i2.732.
- [5] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "Word2vec model analysis for semantic similarities in English words," Procedia Comput. Sci., vol. 157, pp. 160–167, 2019, doi: 10.1016/j.procs.2019.08.153.
- [6] D. I. Af'ida, R. Kusumaningrum, and B. Surarso, "Long short term memory convolutional neural network for Indonesian sentiment analysis towards touristic destination reviews," Proc. - 2020 Int. Semin. Appl. Technol. Inf. Commun. IT Challenges Sustain. Scalability, Secur. Age Digit. Disruption, iSemantic 2020, pp. 630–637, 2020, doi: 10.1109/iSemantic50169.2020.9234210.
- [7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc., pp. 1–12, 2013.
- [8] S. H. Kumhar, M. M. Kirmani, J. Sheetlani, and M. Hassan, "Word Embedding Generation for Urdu Language using Word2vec model," Mater. Today Proc., no. xxxx, 2021, doi: 10.1016/j.matpr.2020.11.766.
- [9] C. Zhang, X. Wang, S. Yu, and Y. Wang, "Research on Keyword Extraction of Word2vec Model in Chinese Corpus," Proc. - 17th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2018, pp. 339–343, 2018, doi: 10.1109/ICIS.2018.8466534.
- [10] W. Yue and L. Li, "Sentiment Analysis using Word2vec-CNN-BiLSTM Classification," pp. 3–7.
- [11] A. K. Sharma, S. Chaurasia, and D. K. Srivastava, "Sentimental Short Sentences Classification by Using CNN Deep Learning Model with Fine Tuned Word2Vec," Procedia Comput. Sci., vol. 167, no. 2019, pp. 1139–1147, 2020, doi: 10.1016/j.procs.2020.03.416.
- [12] Y. Zhu, E. Yan, and F. Wang, "Semantic relatedness and similarity of biomedical terms: Examining the effects of recency, size, and section of biomedical publications on the performance of word2vec," BMC Med. Inform. Decis. Mak., vol. 17, no. 1, pp. 1–8, 2017, doi: 10.1186/s12911-017-0498-1.
- [13] R. P. Nawangsari, R. Kusumaningrum, and A. Wibowo, "Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study," Procedia Comput. Sci., vol. 157, pp. 360–366, 2019, doi: 10.1016/j.procs.2019.08.178.
- [14] X. Rong, "word2vec Parameter Learning Explained," pp. 1–21, 2014, [Online]. Available: <http://arxiv.org/abs/1411.2738>.
- [15] Y. Goldberg and O. Levy, "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method," no. 2, pp. 1–5, 2014, [Online]. Available: <http://arxiv.org/abs/1402.3722>.
- [16] S. Das, S. Ghosh, S. Bhattacharya, R. Varma, and D. Bhandari, "Critical Dimension of Word2Vec," Proc. 2nd Int. Conf. Innov. Electron. Signal Process. Commun. IESC 2019, pp. 202–206, 2019, doi: 10.1109/IESPC.2019.8902427.
- [17] S. Symeonidis, D. Effrosynidis, and A. Arampatzis, "A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis," Expert Syst. Appl., vol. 110, pp. 298–310, 2018, doi: 10.1016/j.eswa.2018.06.022.
- [18] M. Giménez, J. Palanca, and V. Botti, "Semantic-based padding in convolutional neural networks for improving the performance in natural language processing. A case of study in sentiment analysis," Neurocomputing, vol. 378, pp. 315–323, 2020, doi: 10.1016/j.neucom.2019.08.096.
- [19] K. Smagulova and A. Pappachen, "A survey on LSTM memristive neural network architectures and applications," vol. 2324, pp. 2313–2324, 2019.
- [20] J. Xu, D. Chen, X. Qiu, and X. Huang, "Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification," 2016.
- [21] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative Study of CNN and RNN for Natural Language Processing," 2017.