

Klasifikasi Kalimat Ilmiah Menggunakan 1D Convolutional Neural Networks

Yoga Esa Mahendra¹, Ridwan Ilyas², Fatan Kasyidi³

¹Jurusan Informatika, Universitas Jenderal Achmad Yani, Cimahi 40513
E-mail : me.yogaesa@gmail.com

²Sekolah Teknik Elektro dan Informatika, Institut Teknologi, Bandung 40132
E-mail : rdwnilyas@gmail.com

³Jurusan Informatika, Universitas Jenderal Achmad Yani, Cimahi 40513
E-mail : fatan.kasyidi@lecture.unjani.ac.id

ABSTRAK

Kalimat-kalimat yang tuliskan dalam suatu karya ilmiah memiliki banyak sekali ragamnya yang dipengaruhi dari hasil rujukan yang berbeda. Keberagaman kalimat ilmiah tersebut dapat diklasifikasi berdasarkan kelas yang telah didefinisi. Pada penelitian sebelumnya telah menunjukkan bahwa penggunaan arsitektur CNN dapat digunakan untuk klasifikasi kalimat dengan masukan berupa vektor kata. Vektor kata didapatkan melalui hasil praproses dan embedding dengan metode Word2vec. Proses klasifikasi dalam penelitian ini menggunakan 2019 data berupa kalimat tunggal yang diperoleh dari makalah ilmiah komputasi dan telah dilabeli kedalam empat kelas yaitu “Weak”, “Comparison”, “Point”, dan “Neutral”. Penelitian ini telah melakukan simulasi pengujian menggunakan CNN-Multichannel dengan model optimasi Adam yang memiliki learning-rate 0,001 menghasilkan akurasi 74,51% dengan nilai loss 0,82 sedangkan untuk model CNN-Singlechannel memperoleh akurasi sebesar 70,76% dan nilai loss 1,73. Dari serangkaian pengujian menunjukkan bahwa untuk model terbaik dicapai ketika menggunakan model CNN-Multichannel dengan skor F-Measure 45,55%.

Kata Kunci

Natural Language Processing, Klasifikasi, Kalimat Ilmiah, 1D CNN, Word2vec

1. PENDAHULUAN

Dalam setiap tahunnya karya tulis ilmiah akan terus bertambah sesuai dengan semakin banyaknya hasil-hasil penelitian yang ditemukan, kalimat-kalimat ilmiah yang dihasilkan juga akan semakin beragam [1]. Keberagaman tersebut dipengaruhi oleh rujukan yang digunakan oleh peneliti dalam melakukan pengutipan baik dari segi latar belakang, motivasi penelitian, penggunaan metode atau set data yang digunakan, serta perbandingan penelitian yang akan dilakukan untuk menunjukkan hasil yang terbaik.

Sitasi merupakan salah satu penanda yang mencerminkan bagaimana para peneliti merancang penelitian mereka, dan juga ikut serta mempengaruhi pengambilan argumen untuk para peneliti di masa depan [2]. Menuliskan sitasi merupakan bentuk pengakuan terhadap pengarang, ide, pendapat, gagasan atau bahkan teorinya telah kita gunakan.

Kalimat yang terdapat dalam sebuah makalah ilmiah dapat diklasifikasikan kedalam kelas yang terdefinisi. Setiap kelas yang memiliki pola anotasi dan juga kata kunci tersendiri yang membedakan antara kelas satu dengan kelas yang lainnya [3]. Selain untuk

klasifikasi, kalimat ilmiah dapat dianalisis lebih lanjut untuk melihat tren kalimat rujukan [1], *Citation Recommendation* [4], *Plagiarism Detection* [5], *Summarization* [6], dan *Analysis for Scientometrics* [7].

Penelitian sebelumnya mengenalkan metode *annotation schemes* untuk memetakan kalimat kutipan yang dibagi kedalam 12 bagian yang didapatkan melalui analisis linguistik [3]. Penelitian lainnya mencoba mengimplementasikan fitur kutipan *annotation schemes* untuk memfilter referensi ilmiah sebagai langkah awal dalam pembuatan ringkasan artikel ilmiah [8]. Selain dari kalimat ilmiah penelitian lainnya juga mencoba melakukan klasifikasi kalimat retorik untuk membuat suatu rangkuman dari sebuah makalah. Metode yang digunakan adalah SVM dikombinasikan dengan metode Word2vec dan juga Glove. Akurasi tertinggi dari serangkaian pengujian yang dilakukan dicapai oleh Word2vec Model CBOW mendapatkan F-measure 43,44% dengan menggunakan set data sebanyak 20.155 makalah ilmiah [9].

CNN merupakan metode *deep learning* yang paling umum digunakan untuk menganalisis data citra visual. Selain untuk menganalisis data citra, dalam beberapa tahun terakhir metode CNN terbukti mampu menganalisis bahasa alami dan juga menjadi model arsitektur yang umum digunakan untuk klasifikasi kalimat [10][11][12].

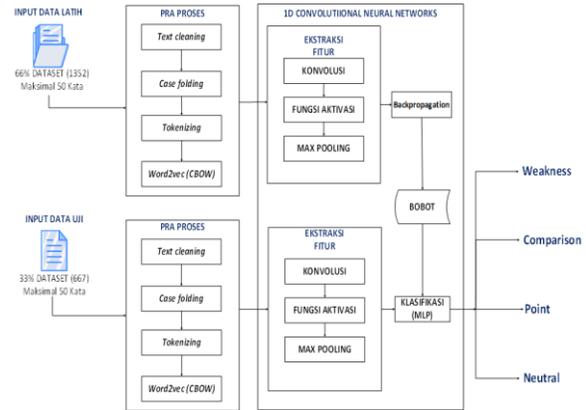
Penelitian sebelumnya mengusulkan metode untuk mengklasifikasi kalimat menggunakan Convolutional Neural Networks. Konfigurasi model yang dipakai adalah dengan ukuran vektor input sebesar 300, satu lapisan konvolusi dan menggunakan tiga kernel yang masing-masing memiliki ukuran filter 3, 4, 5. Pengujian tersebut menggunakan empat model berbeda CNN-rand, CNN-static, CNN-non-static, dan CNN-multichannel memperoleh akurasi tertinggi dengan skor 89,6% dari model CNN-static [11]. Penelitian lain mencoba melakukan klasifikasi dengan menggunakan metode ConvNets dengan karakter sebagai fitur vektor, terbukti efektif dibandingkan dengan kata yang digunakan untuk fitur vektornya. Jumlah karakter terdiri dari 70 jenis yang bersal dari 26 huruf, 10 digit, dan 33 karakter tambahan. Tetapi untuk melakukan hal tersebut membutuhkan set data dengan jumlah yang sangat banyak, dan juga tentunya klasifikasi tersebut terbatas hanya kalimat atau kata yang dibuat dari karakter alphabet [10].

Penelitian ini telah membangun sistem yang dapat mengklasifikasi kalimat ilmiah dari makalah ilmiah komputasi berbahasa inggris secara otomatis. Tahapan yang dilakukan pertama kali adalah pelabelan kalimat, tahapan selanjutnya merubah kalimat kedalam bentuk vektor dengan metode Word2vec model CBOW, lalu masuk kedalam tahapan klasifikasi menggunakan 1D CNN. Hasil klasifikasi terdiri dari 4 kelas yaitu “Weak”, “Comparison”, “Point”, dan “Neutral” kelas tersebut didefinisikan ulang dari kelas Simone Teufel yang berjumlah 12 kelas dengan tujuan untuk mempermudah proses klasifikasi kalimat ilmiah [3].

2. METODE

Tahapan penelitian dimulai dari perolehan dataset dari sejumlah makalah dengan materi komputasi dalam bahasa inggris. Selanjutnya setiap kalimat yang ada pada set data diberi label sesuai dengan yang telah didefinisi, setelah itu masuk pada tahap proposes seperti *text cleaning*, *case folding*, dan *tokenizing* yang dimaksudkan untuk mengurangi data *noise*. Selanjutnya adalah tahap *embedding* merubah korpus kata unik menjadi bentuk vektor kata menggunakan Word2vec model CBOW. Pada langkah berikutnya adalah proses *learning* menggunakan 1D CNN-Multichannel, untuk mendapatkan bobot

pelatihan sebelum melakukan klasifikasi kalimat ilmiah.



Gambar 1. Sistem Klasifikasi Kalimat Ilmiah

2.1 Perolehan Data

Perolehan data pada penelitian ini didapatkan dari kalimat ilmiah berbahasa inggris dari makalah komputasi yang sudah tersedia dalam korpus. Kalimat yang diambil merupakan kalimat rujukan atau kalimat yang memiliki sitasi dan minimal terdiri dari tiga kata yaitu subjek, predikat dan objek atau keterangan. Jumlah dataset yang digunakan berjumlah 2019 kalimat yang telah dilabeli kedalam kelas dari Simone Teufel yang telah disederhanakan dari 12 kelas menjadi 4 kelas, terlihat pada Tabel 1.

Tabel 1. Pemetaan kategori kami dengan kategori Teufel

No	Kategori dari Teufel	Kategori Kami	Keterangan
1	Weak	Weak	Kelemahan dari pendekatan yang dikutip/penelitian sebelumnya
2	CoCo-		
3	CoCoR0	Comparison	Menggambarkan perbandingan atau kontras antara karya sendiri dan lainnya
4	CoCoGM		
5	CoCoXY		
6	PBas	Point	Berkfokus pada sentimen positif diekspresikan pada kutipan, atau argumen/pernyataan penelitian sebelumnya digunakan dalam penelitian saat ini. seperti metode, data, algoritma, definisi
7	PUse		
8	PModi		
9	PSim		
10	PMot	Neutral	Deskripsi netral dari karya yang dikutip, atau tidak cukup bukti tekstual untuk kategori di atas, atau fungsi kutipan tidak terdaftar
11	PSup		
12	Neut		

2.2 Praproses

Praproses merupakan tahapan mengolah dataset maupun data uji yang akan digunakan untuk proses ekstraksi fitur dan klasifikasi. Tahapan praproses terdiri dari beberapa tahapan yaitu *text cleaning*, *case folding*, dan *tokenizing*.

2.2.1 Text Cleaning

Text Cleaning merupakan proses penghapusan karakter *non-alphanumeric* untuk mengurangi noise pada dataset.

Masukan: [We calculate our features using the KenLM toolkit (Heafield, 2011).].

Keluaran: [We calculate our features using the KenLM toolkit Heafield].

2.2.2 Case Folding

Case Folding merupakan proses merubah keseluruhan kata dalam korpus menjadi suatu bentuk standar (biasanya huruf kecil) [13].

Masukan: [We calculate our features using the KenLM toolkit Heafield].

Keluaran: [we calculate our features using the kenlm toolkit heafield].

2.2.3 Tokenizing

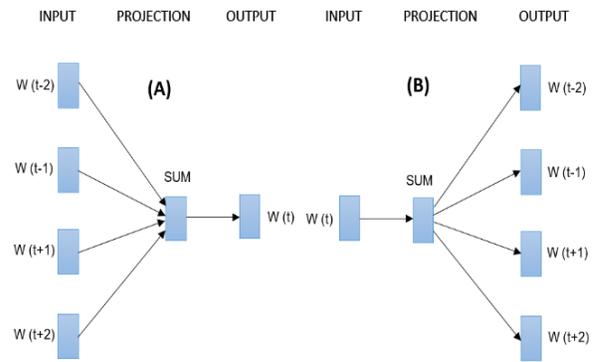
Tokenizing merupakan proses memotong atau memisahkan setiap kalimat menjadi beberapa token/bagian. Sehingga kalimat input menjadi kumpulan kata dalam *list* [13].

Masukan: [we calculate our features using the kenlm toolkit heafield].

Keluaran: ["we", "calculate", "our", "features", "using", "the", "kenlm", "toolkit", "heafield"].

2.3 Word2vec

Word2vec adalah suatu metode dapat mengolah kata-kata dari korpus kata dalam waktu yang relatif singkat. Word2vec juga merupakan *neural networks* yang digunakan untuk memproses teks sebelum teks tersebut diproses lebih lanjut oleh algoritma *deep-learning* [14]. Cara kerja metode ini yaitu dengan mengambil korpus teks sebagai input dengan melalui tahapan praproses dan *one hot-encoding*, lalu menghasilkan representasi vektor dari setiap kata yang ada pada korpus kata tersebut sebagai output [13]. Word2vec mempunyai dua jenis arsitektur pemodelan yang dapat digunakan untuk merepresentasikan vektor kata, arsitektur tersebut diantaranya yaitu *Continuous bag-of-word* (CBOW) dan *Skip-gram* seperti yang ditunjukkan pada Gambar 2.

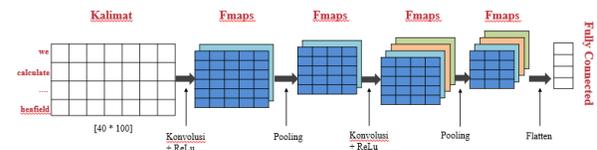


Gambar 2. (A) Model *Continuous bag-of-word* dan (B) Model *Skip-gram*

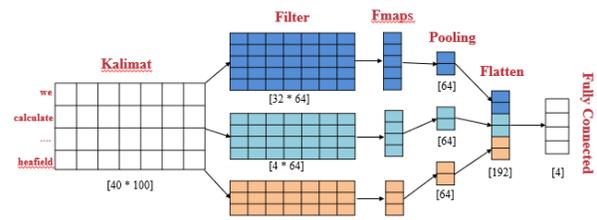
2.4 Convolutional Neural Networks

Convolutional Neural Networks (CNN) merupakan salah satu metode dari *deep learning* dan yang paling umum digunakan untuk menganalisis data berbentuk citra. CNN juga telah banyak digunakan untuk representasi sintaksis dan semantik dari teks dalam berbagai tugas NLP.

CNN terdiri dari dua lapisan utama yaitu lapisan ekstraksi fitur dan lapisan klasifikasi atau. Untuk arsitektur yang digunakan dalam penelitian memiliki dua jenis terlihat seperti pada Gambar 3 dan Gambar 4.



Gambar 3. Arsitektur CNN-Singlechannel



Gambar 4. Arsitektur CNN-Multichannel

Lapisan *input* menampung seluruh korpus kata yang telah diubah kedalam bentuk vektor dengan besaran vektor adalah 100 dan maksimal kata sebanyak 40.

$$X_n = X_1 \oplus X_2 \oplus \dots \oplus X_n \quad (1)$$

Di mana \oplus menunjukkan operator *concatenation*. Dengan X_n adalah vektor kata indeks ke- n dan hasil dari *concatenation* direpresentasikan sebagai matriks n

$\times k$, dimana n adalah banyak kata dan k adalah besaran vektor kata.

$$\text{Softmax}(y_j) = \frac{e^{x^T w_j}}{\sum_{k=1}^K e^{x^T w_k}} \quad (5)$$

2.4.1 Lapisan Ekstraksi Fitur

Lapisan ini memiliki fungsi untuk merubah masukan menjadi fitur berdasarkan data dalam bentuk vektor. Dalam lapisan ini terdiri dari lapisan konvolusi, *pooling*, *concatenate* dan *flatten*. Konvolusi adalah operasi perkalian matriks dot product antara data masukan dengan kernel yang digunakan sebagai filter terhadap data masukan sehingga menghasilkan *featuremap*. Konvolusi dibagi menjadi beberapa *channel* dengan ukuran filter yang berbeda yaitu 3,4,5 pada masing-masing *channel*

$$FM_{(i_l, j_l) h}^{(l, m_l)} = f\left(\sum_{r_l=0}^{k_h} \sum_{c_l}^{k_w} C_{(r_l, c_l)}^{(l, m_l)} * FM_{(r_l + i_{l-1}, c_l + j_{l-1})}^{(l-1)}\right) \quad (2)$$

Di mana h menunjukkan indeks *channel*, kemudian k_h adalah tinggi dan k_w adalah panjang kernel, r_l dan c_l adalah panjang dan tinggi kernel indeks ke l , dan C adalah kernel konvolusi. Selanjutnya hasil tersebut diberi fungsi aktivasi ReLU untuk menghindari nilai negative, dengan nilai x adalah nilai yang dihasilkan *featuremaps*

$$\text{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Selanjutnya adalah tahapan *pooling* merupakan proses *down sampling* setiap *featuremap* yang dihasilkan oleh lapisan konvolusi dan diambil nilai maksimum untuk setiap *channel*, dengan nilai x adalah nilai yang dihasilkan *featuremaps*

$$f(x) = \max(0, x) \quad (4)$$

Hasil dari proses *pooling* dari setiap *channel* selanjutnya digabungkan menjadi satu lapisan pada proses *concatenate*, terakhir adalah proses *flatten* untuk mereduksi dimensi menjadi vektor satu dimensi.

2.4.2 Lapisan Klasifikasi

Lapisan klasifikasi berbentuk Multilayer Perceptron (MLP) yang terdiri dari *input layer*, *output layer* dan beberapa *hidden layer*. Setiap lapisan dibangun dari neuron-neuron yang saling terhubung satu sama lain. Untuk proses pembelajaran menggunakan beberapa metode seperti Backpropagation. Pada lapisan ini terdapat *droupout* yang berfungsi untuk mencegah overfitting pada saat proses pembelajaran. Untuk fungsi aktivasi yang umum digunakan dalam lapisan ini adalah Softmax yang mengubah output menjadi nilai probabilitas untuk setiap kelas.

Dengan y_j merupakan *output* indeks ke- j dan K adalah jumlah total *output* yang dihasilkan dalam proses klasifikasi.

3. HASIL PENELITIAN

Di bagian ini kami melakukan serangkaian pengujian mulai dari membandingkan model CNN-Multichannel dan CNN-SingleChannel, membandingkan beberapa model optimasi dan, menguji model klasifikasi sistem dari hasil Akurasi, Recall, Precision dan F-measure.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (6)$$

$$\text{Precision} = \frac{TP}{TP+FP} * 100\% \quad (7)$$

$$\text{Recall} = \frac{TP}{TP+FN} * 100\% \quad (8)$$

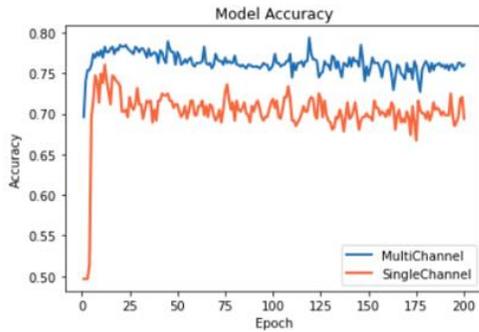
$$\text{F-Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Recall} + \text{Precision}} * 100\% \quad (9)$$

Akurasi merupakan tingkat kedekatan antara nilai prediksi dengan nilai aktual (nilai sebenarnya), Precision adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem, Recall adalah data penghapusan yang berhasil diambil dari data yang relevan dengan *query* atau tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi dan F-Measure atau F1-Score merupakan salah satu perhitungan evaluasi dari hasil mengkombinasikan Recall dan Precision.

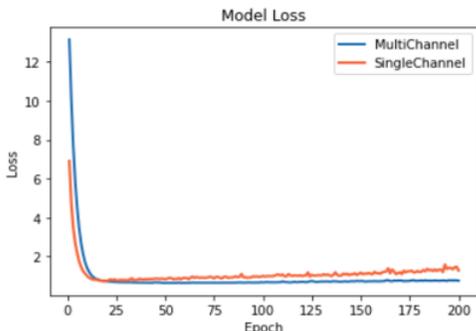
Dataset yang digunakan dalam pengujian ini sebanyak 2019 kalimat tunggal dari satu subjek, satu predikat, dan bisa dilengkapi dengan objek dan keterangan. Kalimat tersebut telah dilabeli dibagi menjadi dua bagian untuk proses pengujian yang terdiri dari data latihan sebanyak 1352 kalimat dan data uji sebanyak 607 kalimat.

3.1 Pengujian Multichannel dan Singlechannel

Kedua model yang diujikan menggunakan konfigurasi yang sama dengan model optimasi Adam, *learning-rate* 0,001, dan *batch size* 64. Untuk variasi percobaan pengujian menggunakan epoch 50, 100, dan 200. Model multichannel menggunakan tiga ukuran filter yang berbeda untuk setiap *channel* 3,4,5. Hasil perbandingan akurasi ditunjukkan oleh Gambar 4 dan nilai loss ditunjukkan oleh Gambar 5.



Gambar 5. Perbandingan akurasi Multichannel dan Singlechannel



Gambar 6. Perbandingan loss Multichannel dan Singlechannel

Hasil terbaik dicapai oleh model Multichannel dengan akurasi sebesar 74,51% sedangkan model Singlechannel mendapatkan akurasi sebesar 70,76%, untuk nilai *loss* model Multichannel cenderung lebih stabil dengan nilai *loss* sebesar 0,82 dibandingkan Singlechannel yang terlihat tampak adanya kenaikan dengan hasil nilai *loss* sebesar 1,30 walaupun peningkatan tersebut tidak terlalu signifikan. Perbandingan pengujian model Multichannel dan Singlechannel dengan variasi *epoch* 50, 100, dan 200 dapat dilihat pada Tabel 2.

Tabel 2. Perbandingan akurasi dan loss setiap epoch dari model Multichannel dan Singlechannel

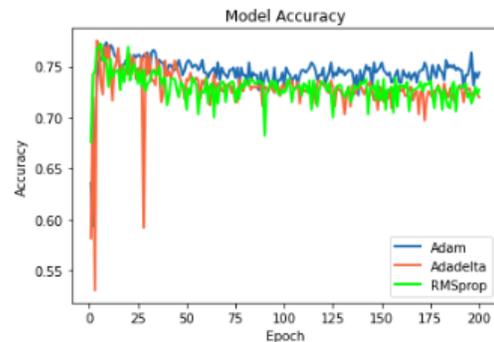
Model	Epoch	Data Latih		Data Uji	
		Acc (%)	Loss	Acc (%)	Loss
Multi channel	50	84,48	0,46	74,36	0,77
	100	87,32	0,51	75,26	1,03
	200	89,73	0,26	74,51	0,82
Single channel	50	85,25	0,45	69,73	0,90
	100	88,63	0,25	70,47	1,03
	200	89,29	0,20	70,76	1,30

Hasil yang diperoleh pada Tabel 2, menunjukkan bahwa model Multichannel mendapatkan hasil yang lebih baik dalam serangkaian pengujian variasi *epoch* yang

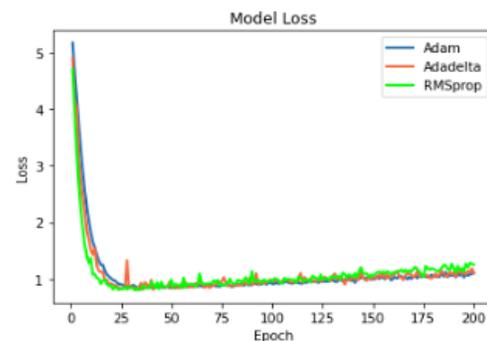
diujikan, dikarenakan model ini membagi proses konvolusi kedalam beberapa *channel* dengan filter yang berbeda, proses tersebut menyebabkan model ini dapat membaca kalimat dengan ukuran *n-gram* yang berbeda (kelompok kata).

3.2 Pengujian Model Optimasi

Pengujian ini bermaksud membandingkan beberapa model optimasi diantaranya Adaptive Moment Optimization (Adam), Adaptive Learning Rate (Adadelata), dan Root Mean Square Propagation (RMSprop). Untuk konfigurasi sama seperti pengujian sebelumnya dan menggunakan model CNN-Multichannel. Hasil perbandingan akurasi ditunjukkan oleh Gambar 6 dan nilai loss ditunjukkan oleh Gambar 7.



Gambar 7. Perbandingan akurasi setiap model optimasi yang diujikan



Gambar 8. Perbandingan loss setiap model optimasi yang diujikan

Dapat dilihat pada Gambar 6 pembelajaran menggunakan model optimasi Adam lebih stabil dalam melakukan klasifikasi jika dibandingkan dengan model optimasi Adadelata dan RMSprop, akan tetapi untuk ketiga model optimasi tersebut masih fluktuatif dilihat dari grafik akurasi, oleh sebab itu dapat disimpulkan bahwa ketiga model optimasi yang diujikan masih belum mencapai titik konvergen. Sedangkan *loss* yang didapat dari ketiga model optimasi tidak berbeda jauh jika ditinjau dari Gambar

7. Hasil akurasi dan loss untuk setiap pengujian dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan akurasi dan loss untuk setiap model optimasi

Model Optimasi	Epoch	Data Latih		Data Uji	
		Acc (%)	Loss	Acc (%)	Loss
Adam	100	85,56	0,48	73,61	0,95
	150	84,84	0,51	71,51	1,03
	200	85,72	0,51	74,51	1,16
RMSprop	100	87,80	0,33	72,86	0,93
	150	87,43	0,32	73,46	1,15
	200	86,72	0,31	72,71	1,25
Adadelata	100	86,76	0,39	73,46	0,97
	150	87,35	0,39	71,36	1,03
	200	87,65	0,36	71,96	1,10

Dari ketiga model optimasi yang diujikan Adam memperoleh skor tertinggi dengan akurasi data uji 74,51% dan akurasi data latih sebesar 87,80%. Adam dapat menghitung estimasi dari gradien momen pertama dan kedua untuk mengupdate *learning-rate* untuk setiap bobot neuron sehingga mendapatkan hasil yang optimal.

3.3 Pengujian Model Klasifikasi

Pengujian model klasifikasi adalah tahapan akhir dari serangkaian pengujian untuk melihat seberapa baik sistem klasifikasi yang dibuat. Konfigurasi yang digunakan diambil dari hasil terbaik yang diperoleh pada pengujian sebelumnya. Proses pengujian ini ditinjau dari hasil Confusion Matrix dari data uji pada Tabel 4.

Tabel 4. Confusion Matrix

		Predicted Label			
		Point	Neutral	Comparison	Weak
True Label	Point	261	82	2	1
	Neutral	44	227	11	2
	Comparison	7	13	7	1
	Weak	0	9	0	0

Dari hasil Confusion Matrix yang ditunjukkan Tabel 4, terlihat kelas terbaik yang berhasil diprediksi dengan tepat adalah kelas "Point" dengan total data yang berhasil terprediksi oleh sistem adalah 261 dari total keseluruhan 346, sedangkan kelas "Weak" tidak ada satupun yang berhasil terprediksi dengan benar dari total data keseluruhan adalah 9, seluruh data kelas "Weak" terprediksi oleh sistem menjadi data kelas "Neutral". Hal tersebut disebabkan oleh sebaran data pada setiap kelas terpaut cukup jauh khususnya untuk kelas "Weak", yang menjadikan sistem kekurangan pola variasi atau ciri dari kalimat yang masuk kedalam data dari Kelas "Weak".

Tabel 5. Hasil Precision, Recall, dan F-Measure

	Precision (%)	Recall (%)	F-Measure (%)
Weak	0%	0%	0%
Comparison	35%	25%	29%
Neutral	69%	80%	74%
Point	84%	75%	79%
Rata-Rata	47%	45%	45.5 %

Hasil kelas "Weak" untuk nilai dari Precision, Recall maupun F-Measure mendapat 0% akurasi dikarenakan sistem tidak berhasil memprediksi pada saat melakukan pengujian menggunakan data uji. Hasil akhir dari pengujian Model Klasifikasi mendapatkan nilai Precision sebesar 47%, Recall sebesar 45%, dan F-Measure mendapatkan akurasi keseluruhan sebesar 45,5%.

4. DISKUSI

Dalam penelitian ini masih terdapat beberapa kelas yang belum terprediksi dengan baik oleh sistem, hal tersebut disebabkan karena kurangnya pola variasi atau ciri khusus suatu kelas. Penelitian selanjutnya mungkin dapat mencoba melakukan klasifikasi dengan mempertimbangkan kalimat sebelum dan sesudahnya, yang dimaksudkan agar makna atau pola kalimat dapat lebih terperinci dan membuat hasil pembelajaran klasifikasi kalimat menjadi lebih baik. Selain itu dapat mencoba melakukan klasifikasi dari kalimat dengan bahasa berbeda (multibahasa), atau dapat juga mencoba mengembangkan dari segi konfigurasi model pembelajaran dan model *embedding* yang berbeda.

5. KESIMPULAN

Penelitian ini mengembangkan klasifikasi kalimat ilmiah menggunakan 1D Convolutional Neural Networks model Multichannel dan Singlechannel dengan dikombinasikan metode embedding dari Word2vec. Hasil penelitian ini menunjukkan bahwa klasifikasi kalimat ilmiah menggunakan CNN-Multichannel menghasilkan akurasi sebesar 74,51% sedangkan untuk CNN-Singlechannel memperoleh akurasi sebesar 70,76%. Dalam serangkaian pengujian yang dilakukan menghasilkan konfigurasi terbaik dari model CNN-Multichannel dengan optimasi Adam, *learning-rate* 0,001, *batch size* 64, dan besaran vektor setiap kata sebanyak 100. Untuk F-Measure mendapatkan skor 45,55%.

DAFTAR PUSTAKA

- [1] D. Jurgens, R. Hoover, and D. McFarland, "Measuring the Evolution of a Scientific Field through Citation Frames," vol. 6, pp. 391–406, 2018.
- [2] D. Jurgens, S. Kumar, R. Hoover, D. McFarland, and D. Jurafsky, "Citation Classification for Behavioral Analysis of a Scientific Field," *ArXiv*, vol. abs/1609.0, 2016.
- [3] S. Teufel and D. Tidhar, "Automatic classification of citation function," *Computational Linguistics*, no. July, pp. 103–110, 2006.
- [4] H. Liu, X. Kong, X. Bai, W. Wang, T. M. Bekele, and F. Xia, "Context-Based Collaborative Filtering for Citation Recommendation," *IEEE Access*, vol. 3, pp. 1695–1703, 2015.
- [5] B. Gipp, N. Meuschke, and C. Breitingner, "Citation-based plagiarism detection: Practicability on a large-scale scientific corpus," *Journal of the Association for Information Science and Technology*, vol. 65, no. 8, pp. 1527–1540, 2014.
- [6] A. Cohan and N. Goharian, "Scientific article summarization using citation-context and article's discourse structure," *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pp. 390–400, 2015.
- [7] A. Arbor and A. Arbor, "NLP Driven Citation Analysis for Scientometrics," *Natural Language Engineering*, vol. 23, no. 1, pp. 93–130, 2017.
- [8] G. H. Rachman, M. L. Khodra, and D. H. Widyantoro, "Classification of citation sentence for filtering scientific references," *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2019*, pp. 347–352, 2019.
- [9] G. H. Rachman, M. L. Khodra, and D. H. Widyantoro, "Word embedding for rhetorical sentence categorization on scientific articles," *Journal of ICT Research and Applications*, vol. 12, no. 2, pp. 168–184, 2018.
- [10] X. Zhang, J. Zhao, and Y. Lecun, "Character-level convolutional networks for text classification," *Advances in Neural Information Processing Systems*, vol. 2015-Janua, pp. 649–657, 2015.
- [11] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Aug. 2014.
- [12] H. Xu, A. Kotov, and A. I. Carcone, "Text Classification with Topic-based Word Embedding and Convolutional Neural Networks," 2016.
- [13] M. . Nabila Nanda Widyastuti, Arif Bijaksana, Ir., M.Tech., Ph.D, Indra Lukmana Sardi, S.T., "Analisis Word2vec untuk Perhitungan Kesamaan Semantik antar Kata," *e-Proceeding of Engineering*, vol. Vol.5, No., no. 3, pp. 7603–7612, 2018.
- [14] R. D. Sharma, S. Tripathi, S. K. Sahu, S. Mittal, and A. Anand, "Predicting Online Doctor Ratings from User Reviews Using Convolutional Neural Networks," *International Journal of Machine Learning and Computing*, vol. 6, no. 2, pp. 149–154, 2016.