Pengukuran Kesamaan Semantik Pasangan Kalimat Sitasi Menggunakan Convolutional Neural Network

Janjan Nurjaman[1], Ridwan Ilyas[2], Fatan Kasyidi[3]

[1]Jurusan Informatika, Universitas Jenderal Achmad Yani, Cimahi 40531
janjanurjaman@gmail.com
[2]Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Bandung-Indonesia
rdwnilyas@gmail.com
[3]Jurusan Informatika, Universitas Jenderal Achmad Yani, Cimahi 40531
fatan.kasyidi@lecture.unjani.ac.id

ABSTRAK

Parafrasa merupakan salah satu istilah dalam linguistik yang berisi kalimat atau frasa untuk menyampaikan makna yang sama menggunakan kata-kata yang berbeda. Parafrasa juga digunakan untuk menguraikan suatu teks dalam bentuk atau susunan kata yang lain agar makna yang tersembunyi dalam teks tersebut dapat dijelaskan, namun untuk mengetahui makna suatu kalimat tidaklah mudah. Oleh karena itu, dibutuhkan model komputasi yang dapat mengukur kesamaan semantik pada pasangan kalimat sitasi. Kalimat sitasi diperoleh dari kumpulan sitasi hasil dari paper karya tulis ilmiah yang sudah dikumpulkan dan dilabeli oleh anotator. Pengukuran dilakukan menggunakan Convolutional Neural Network (CNN) dengan representasi vektor menggunakan Word2vec. Representasi kata yang terbentuk dari dua kalimat sebanyak 10.000 vektor menjadi masukan pada arsitektur CNN. Vektor yang terbentuk menjadi masukan untuk proses pelatihan pada MLP. Hasil pengukuran terdiri dari enam jenis kategori kelas hubungan pasangan kalimat sitasi yaitu Equivalent, Similar, Spesific, No Alignment, Related dan Opposite. Hal tersebut dikarenakan setiap pasangan kalimat memiliki kata yang berbeda namun memiliki makna yang sama. Hasil penelitian menunjukan hasil uji semantik pasangan kalimat sitasi dengan 1600 dataset latih menghasilkan akurasi sebesar 91% dan dengan menggunakan 400 dataset uji menghasilkan akurasi 79% dengan F1-Score 66%.

Kata Kunci
Parafrasa, Word2vec, Convolutional Neural Network, backpropagation, Machine learning

1 PENDAHULUAN

Parafrasa merupakan kalimat atau frasa yang menyampaikan makna yang sama menggunakan kata-kata yang berbeda. Parafrasa juga digunakan untuk menguraikan suatu teks dalam bentuk atau susunan kata yang lain agar makna yang tersembunyi dalam teks tersebut dapat dijelaskan. Meskipun definisi dari parafrasa membutuhkan kesetaraan makna yang tepat, ilmu bahasa sendiri menerima perkiraan makna dan kesetaraan maknanya. Akan tetapi, perkiraan kesetaraan sulit didefinisikan atau pun dikarakterisasi [1].

Parafrasa yang merupakan bagian dari linguistik biasa digunakan dalam mengutip suatu sumber karya tulis ilmiah seperti mengutip pernyataan, menyalin/mengulang pernyataan seseorang atau mencantumkannya didalam suatu karya tulis yang dibuat, namun tetap mengindikasikan bahwa kutipan tersebut merupakan pernyataan orang lain [2].

Sitasi atau *citation* didalam penulisan ilmiah sangat penting, sebagai bahan pustaka pendukung bagi tulisannya. Kegunaan bahan pustaka pendukung antara lain untuk menunjukkan adanya kebijakan dibidang kajiannya, menerangkan suatu teori, pengertian atau definisi, memperlihatkan kepada pembaca apa yang pernah ditemukan oleh ilmuan lain, untuk memperkuat temuannya, untuk memanfaatkan metode sebagai pembanding dan banyak lagi alasan lain yang dapat memperkuat kesahihan penelitian yang dilakukan [2].

Penelitian ini dilakukan berdasarkan pada isu analisis tren suatu karya tulis ilmiah untuk setiap tahun dengan beberapa kelas tertentu [3]. Kalimat sitasi yang dikutip dari bagaimana membaca kalimat sitasi berdasarkan *background*, tujuan, masalah dan maknanya juga berpengaruh terhadap analis tren semantik teks. Sehingga luaran dari perbandingan pasangan kalimat sitasi dapat menghasilkan ukuran tren yang sering terjadi seperti pola pengutipan yang langsung utuh dikutip atau perubahan kata-kata tetapi masih memiliki makna yang sama dengan tetap tanpa mengindikasikan bahwa kutipan tersebut adalah pernyataan orang lain[4].

Sebagai representasi kata menjadi vektor digunakan metode Word2Vec untuk proses perubahan setiap kata didalam konteks kalimat menjadi vektor dengan dimensi tertentu.

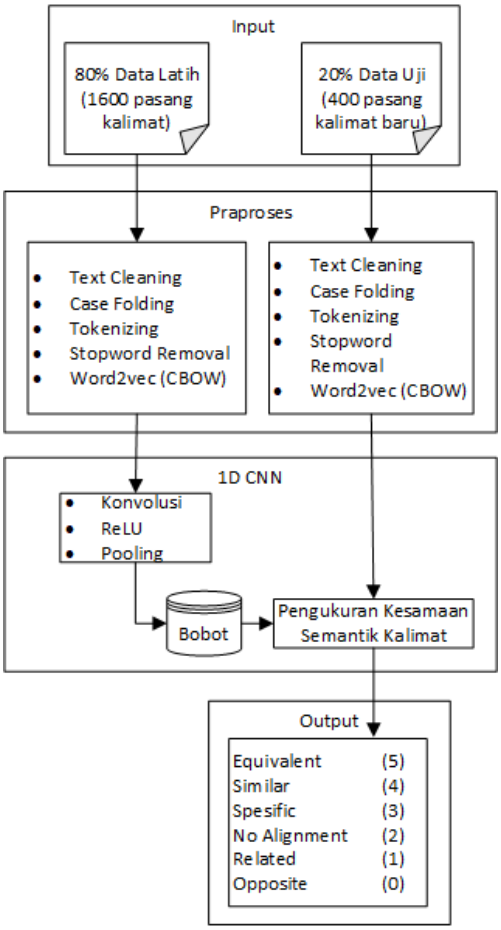
Dalam merepresentasikan suatu kata, Word2vec mengimplementasikan Neural Network untuk menghitung *contextual* dan *semantic similarity* (kesamaan kontekstual dan semantik) dari setiap kata (*inputan*) yang berbentuk *one-hot encoded vectors* [5].

Beberapa penelitian terdahulu menggunakan Convolutional Neural Network dalam mendeteksi kesamaan tekstual semantik bahasa arab mampu meningkatkan akurasi dari 85% menjadi 87% [6]. Pada penelitian lain dalam mengembangkan model kesamaan tekstual semantik kalimat sitasi juga dilakukan terhadap korpus berbahasa indonesia untuk setiap pasangan tipe seperti EQUI, SPE1, SPE2, REL, SIMI, NOALI dan skor 0 hingga 5 dengan model VRep dan UWB [7].

Penelitian ini bertujuan membuat model komputasi yang dapat mengukur kesamaan semantik pasangan kalimat sitasi menggunakan Convolutional Neural Network dengan representasi kata menjadi vektor dengan menggunakan Word2Vec. Bobot awal pelatihan dari *input layer* ke *hidden layer,* dan dari *hidden layer* ke *output layer* pada Backpropagation diperoleh dari hasil bobot *random.* Hasil pengukuran kesamaan semantik pasangan kalimat terdiri dari enam kelas yaitu Equivalent, Similar, Spesific, No Alignment, Related dan Opposite. Hasil ekstraksi kata menjadi vektor oleh Word2Vec menjadi fitur masukan pada proses pengukuran. Pengukuran dilakukan terhadap 2000 dataset kalimat sitasi.

2 METODE

Tahapan penelitian dimulai dengan perolehan dataset dari sistem informasi pelabelan pasangan kalimat sitasi oleh anotator dengan kategori kelas yang telah didefinisikan yaitu Equivalent, Similar, Spesific, No Alignment, Related dan Opposite. Dataset tersebut didapat dari sejumlah makalah ilmiah komputasi dalam bahasa Inggris. Kemudian, pasangan kalimat yang sudah diberi label akan masuk pada tahap praproses yaitu, *text cleaning, case folding, tokenizing* dan *stopword removal.* Selanjutnya adalah tahap *word embedding* (mengubah korpus kata unik pada dataset pasangan kalimat menjadi nilai vektor) dengan menggunakan Word2vec model CBOW. Langkah berikutnya adalah proses pelatihan menggunakan 1D Convolutional Neural Network (1D CNN) dengan *multi channel*, bobot awal pelatihan merupakan angka *random.* Sistem pengukuran kesamaan semantik pasangan kalimat sitasi dapat dilihat pada gambar 1.



Gambar 1. Sistem Pengukuran Semantik

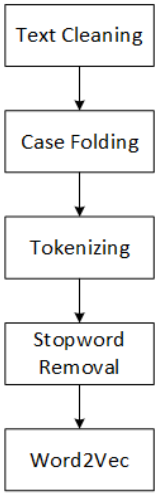
2.1 Perolehan Data

Pada penelitian ini digunakan dataset kalimat sitasi hasil dari pengambilan dari banyak makalah komputasi berbahasa Inggris yang sudah tersedia di dalam korpus. Pelabelan menggunakan enam kategori kelas yaitu Equivalent, Similar, Spesific, No Alignment, Related dan Opposite dengan 2000 dataset, dimana 1600 data latih dan 400 data uji.
Korpus dataset kalimat sitasi tersebut dianotasi ulang oleh anotator sesuai dengan kategori kelas yang sudah ditentukan dengan dengan membandingkan dua kalimat untuk dilihat makna keterhubungannya. Kemudian, setelah dapat dikategorikan, anotator dapat memilih kategori kelas yang bersesuaian dengan pasangan kalimat yang dibandingkan. Sebagai contoh, perbandingan pasangan kalimat "The bird is bathing in the sink" dengan "Birdie is washing itself in the water basin". Pasangan kalimat tersebut menggunakan kata yang berbeda tetapi memiliki makna yang sama atau termasuk ke dalam kelas *equivalent.*

2.2 Praproses

Praproses merupakan tahapan mengolah dataset maupun data uji yang akan digunakan untuk proses ektraksi fitur dan identifikasi. Praproses dilakukan untuk memastikan kalimat yang akan diproses nantinya tidak ada *noise.* Praproses yang dilakukan terdiri dari segmentasi dan ekstraksi kalimat untuk

mendapatkan nilai vektor. Tahapan praproses terdiri dari beberapa tahapan yaitu Text Cleaning, Case Folding, Tokeninzing, *one hot encoding* dan *Embedding* dengan model Word2vec CBOW [8]. Untuk dataset sendiri memiliki proses tambahan diawal yaitu proses pelabelan data. Alir data praproses dapat dilihat pada gambar 2.



Gambar 2. Skema Praproses

2.2.1 Text Cleaning

Keseluruhan data teks dalam korpus dibersihkan dari seluruh atribut seperti angka dan tanda baca. Contohnya, we use the SCFG decoder cdec (dyer et al., 2010) menjadi we use the scfg decoder cdec dyer et al 2010. Angka 2010 tidak dibersihkan, karena menjadi kata unik.

2.2.2 Case Folding

Seteleh melewati proses *text cleaning,* data teks dalam korpus diubah menjadi suatu bentuk standar (huruf kecil atau sejenisnya). Contohnya, we use the SCFG decoder cdec dyer et al 2010 menjadi we use the scfg decoder cdec dyer et al., 2010.

2.2.3 Tokenizing

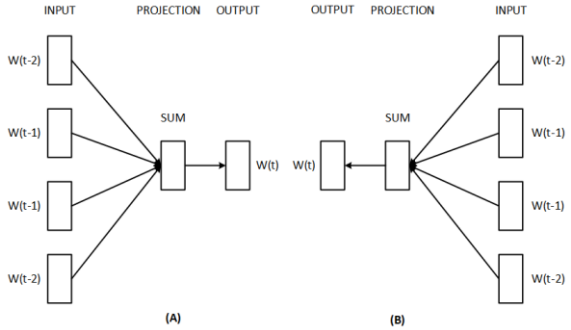
Dari prose pembersihan data, kalimat tersebut akan dipotong menjadi beberapa token/bagian. Contohnya, we use the scfg decoder cdec dyer et al 2010 menjadi ‘we’ ‘use’ ‘the’ ‘scfg’ ‘decoder’ ‘cdec’ ‘dyer’ ‘et’ ‘al’ ‘2010’.

2.2.4 Stopword Removal

Setelah semua proses (*text cleaning, case folding* dan *tokenizing*) dilakukan, kalimat dengan kata yang tidak perlu akan dihilangkan, seperti *the, of* dan sebagainya. Contohnya, ‘we’ ‘use’ ‘the’ ‘scfg’ ‘decoder’ ‘cdec’ ‘dyer’ ‘et’ ‘al’ ‘2010’ menjadi ‘we’ ‘use’ ‘scfg’ ‘decoder’ ‘cdec’ ‘dyer’ ‘et’ ‘al’ ‘2010’

2.3 Word2vec

Word2vec adalah salah satu teknik *word embedding* (mengubah kata menjadi vektor yang terdiri dari kumpulan angka). Kata pada sebuah kalimat bisa merepresentasikan makna kata itu sendiri dan konteks kata (kalimat) merepresentasikan makna secara keseluruhan sebagai hasil dari gabungan setiap kata yang menyusun kalimat tersebut. Cara kerja Word2vec yaitu dengan mengambil korpus data sebagai input yang sudah melalui tahap praproses dan *one hot encoding* (membuat variabel *binary code* sebanyak jumlah teks yang ada dalam kalimat). Kemudian akan menghasilkan nilai vektor dari setiap kata yang ada ada korpus data. Word2vec mempunyai dua jenis model arsitektur untuk merepresentasikan vektor kata yaitu, *Continous bag-of-word* (CBOW) dan *Skip-gram.* Model tersebut dapat dilihat pada gambar 3.



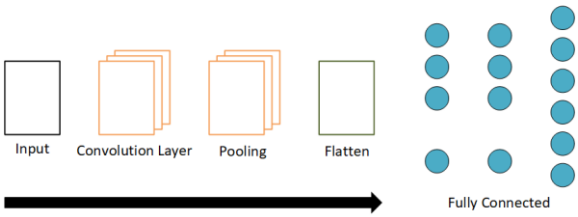
Gambar 3. (A) Model CBOW dan (B) Model *Skip-gram*

w(t) adalah kata target atau kata masukan yang diberikan. Disini ada satu *projection* yang melakukan perkalian titik *(dot product)* antara matriks bobot dan vektor masukan w(t) dan di dalam *projection* tidak ada fungsi aktivasi yang digunakan. Hasil dari *dot product* pada *projection* diteruskan ke lapisan luaran *(output).* Di *output layer* dihitung *dot product* antara *vector output* dari *projection* dengan matriks bobot pada *output layer.*

2.4 Convolutional Neural Network

Convolutional Neural Network adalah salah satu metode *machine learning* dari pengembangan Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi.

CNN terdiri dari Convolutional Layer untuk ekstraksi fitur secara otomatis menggunakan kernel konvolusi (convolution filters) dengan ukuran berbeda, ReLu Layer, Pooling Layer, Flatten dan Fully-Connected Layer. CNN juga dapat memahami kalimat dari kata-kata yang telah diubah menjadi nilai vektor untuk setiap kata yang bersesuaian [9]. Jumlah input pada layer konvolusi sebanyak input kata yang digunakan (maksimal kata = 50 dengan besar vektor 100) untuk kalimat satu dan dua yang akan dipasangkan. Jadi, total 10000 untuk dua kalimat. Arsitektur CNN yang dibangun dengan konsep *multi channel* dapat dilihat pada gambar 4.



Gambar 4. Arsitektur CNN

2.4.1 Lapisan Ekstraksi Fitur

Lapiran Ekstraksi Fitur berfungsi untuk melakukan operasi konvolusi pada output dari layer sebelumnya. Untuk melakukan proses konvolusi dapat dilakukan seperti pada persamaan 1.

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

dimana :

FM = *feature map*
l = index layer atau input
m_l = index map pada layer ke-l
i_l = index baris FM layer ke-l
j_l = index kolom FM layer ke-l
k_h = tinggi kernel
r_l = panjang kernel layer ke-l
k_w = panjang kernel
c_l = tinggi kernel layer ke-l
C = kernel konvolusi

2.4.2 Fully Connected Layer

Arsitektur Multilayer Perceptron digunakan untuk melakukan generasi dengan baik pada proses pelatihan. Data latih yang berjumlah 1600 dataset, selanjutnya akan mulai diproses sebagai pelatihan dan 400 dataset sebagai data uji. Parameter arsitektur CNN yang digunakan sebagai data masukan pada sistem pengukuran kesamaan semantik dapat dilihat pada tabel 1.

Tabel 1. Parameter model arsitektur CNN

Lapisan (tipe)	Output Shape
Input layer	Input (50) Output (50)
Embedding	Input (50) Output(50,100)
Conv1d_1:filter[2x64]	Input (50,100) Output(49,64)
Pooling1:maxPooling	Input (49,64) Output(1,64)
Conv1d_2:filter[3x64]	Input (50,100) Output(48,64)
Pooling2:maxPooling	Input (48,64) Output(1,64)
Conv1d_3:filter[4x64]	Input (50,100) Output(47,64)
Pooling3:maxPooling	Input (47,64) Output(1,64)
Concatenate	Input[(1,64), (1,64),(1,64)]
	Output(1,192)
Flatten	Input (1,192) Output(192)
Dense	Input (192) Output(6)

Dalam menentukan banyaknya neuron pada *hidden layer* dapat menggunakan persamaan 2, untuk fungsi Rectified Linear Unit (ReLU) dapat dilihat pada persamaan 3 dan untuk mengurangi dimensi dari *feature map (pooling)* dapat menggunakan persamaan 4.

$$p = \sqrt{(m \times n)} \quad (2)$$

dimana :

m = jumlah neuron input
n = jumlah neuron output

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

dimana :

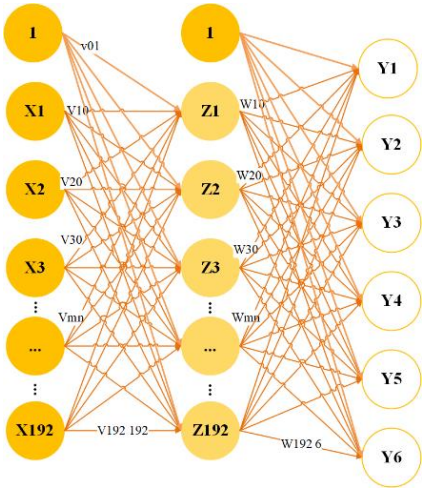
x = input vector

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

dimana :

x = input vector

Arsitektur Multilayer Perceptron yang digunakan pada penelitian ini diperlihatkan pada gambar 5 yang terdiri dari *input layer, hidden layer* dan *output layer.* Proses pembelajaran dimulai dari setiap neuron input sebanyak 192 neuron terhadap setiap neuron pada *hidden layer* sebanyak 192 neuron, ynag diteruskan hingga setiap neuron dalam *output layer* sebanyak 6 neuron. Bobot awal diperoleh dari hasil *random.* Arstitektur Multilayer Perceptron dapat dilihat pada gambar 5.



Gambar 5. Arsitektur Multilayer Perceptron

Untuk menghasilkan output, *fully connected layer* ini menggunakan fungsi aktivasi *softmax* dengan menggunakan persamaan 5.

$$\sigma_i(a) = \frac{e^a i}{\sum_j e^a j} \quad (5)$$

dimana :
$\sigma_i(a)$ = nilai output ke-i
$e^a i$ = nilai unit ke-i
Σj = jumlah neuron lapisan output
$e^a j$ = nilai neuron ke-j

Pada proses selanjutnya, pelatihan menggunakan Backpropagation terhadap tiga fase yang akan dilakukan untuk proses identifikasi yaitu *feed forward* dan *backpropagation*. Berikut penjelasan dari ketiga fase tersebut.

2.4.2.1 Feed Forward

Feed Forward atau fase maju merupakan tahap dimana setiap neuron *input xi* akan mengirimkan sinyal masukkan pada *hidden layer.* Masing-masing neuron pada *hidden layer* dikalikan dengan bobot dan dijumlahkan dengan bias. Perhitungannya dilakukan menggunakan Persamaan 6.

$$z\_net\ j = vj0 + \Sigma n\ xi\ vkj \quad (6)$$

dimana :
i = indeks neuron input
j = indeks neuron hidden
v0j = bobot bias dari input layer ke *hidden layer*
n = jumlah neuron pada *input layer*
xi = data nilai masukan yang ke-i
vkj = bobot dari *input layer* ke *hidden layer*

Selanjutnya melakukan perhitungan pada masing-masing neuron *output* yang dikalikan dengan bobot dan dijumlahkan dengan bias menggunakan persamaan 7.

$$y_{net}k = wk0 + \Sigma p\ zj\ ykj \quad (7)$$

Dimana :
j = indeks neuron *hidden layer*
p = jumlah neuron pada *hidden layer*
k = indeks neuron *output*
wk0 = bobot *bias* dari *hidden layer* ke *output layer*
zj = nilai hasil aktivasi dari *hidden layer*
ykj = bobot dari *hidden layer* ke *output layer*

2.4.2.2 Backpropagation

Backpropagation atau fase mundur merupakan tahap masing-masing neuron *output* menerima pola target yang telah dicapai sesuai pola masukkan saat pelatihan. Lakukan perhitungan untuk perubahan bias menggunakan persamaan 8.

$$\Delta wk = \alpha\ x\ \delta k \quad (8)$$

dimana:
α = nilai *learning rate*
δk = komponen kesalahan pada *output layer*

2.5 Pengujian Model Komputasi

Machine learning merupakan penerapan dari disiplin ilmu kecerdasan buatan (*artificial intelligence)* yang menggunakan teknik statistika untuk menghasilkan suatu model otomatis dari sekumpulan data yang bertujuan memberikan komputer kemampuan untuk "belajar". Salah satu metode untuk mengukur performa dari suatu model klasifikasi adalah dengan mencari nilai Accuracy, Precision, Recall dan F1 Score dari suatu model dengan beberapa istilah dasar dalam pencarian nilai tersebut seperti True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Istilah-istilah tersebut biasa dirangkum sebagai suatu matriks yang disebut *confusion matrix* sebagaimana ditunjukkan pada tabel 2.

Tabel 2. Confusion matrix

		Predicted Class	
		yes	*no*
Actual Class	*yes*	*TP*	*FN*
	no	*FP*	*TN*
	Total	*P′*	*N′*

2.5.1 Accuracy

Accuracy merupakan tingkat kedekatan antara nilai prediksi dengan nilai aktual (nilai sebenarnya). Perhitungannya dapat menggunakan persamaan 9.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} * 100\% \quad (9)$$

dimana :
TP = data positif yang diprediksi benar
TN = data negatif yang dipresiksi benar
FN = data positif namun dipresiksi sebagai data negatif
FP = data negatif namun diprediksi sebagai data positif

2.5.2 Precision

Precision atau *positive predictive value* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Untuk mengetahui nilai *precision* dapat menggunakan persamaan 10.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} * 100\% \quad (10)$$

dimana :
TP = data positif yang diprediksi benar
FP = data negatif namun diprediksi sebagai data positif

2.5.3 Recall

Recall adalah data penghapusan yang berhasil diambil dari data yang relevan dengan *query* atau tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Untuk menghitung nilai *recall* dapat menggunakan persamaan 11.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} * 100\% \quad (11)$$

dimana :
TP = data positif yang diprediksi benar
FN = data positif namun dipresiksi sebagai data negatif

2.5.4 F-Measure

F-Measure atau F1-Score merupakan salah satu perhitungan evalusasi dari hasil mengkombinasikan *recall* dan *precision*. Untuk menghitung nilai F1-Score dapat menggunakan persamaan 12.

$$\text{F-Measure}=\frac{2*\text{Precision}*\text{Recall}}{\text{Recall}+\text{Precision}}*100\% \qquad (12)$$

3 HASIL DAN DISKUSI

Data latih yang digunakan pada penelitian ini sebanyak 1600 dataset yang sudah dilabeli oleh anotator melalui sistem informasi parafrasa. Dari proses ekstraksi dengan word2vec, didapatkan nilai vektor untuk setiap kata yang menyusun suatu kalimat sitasi.

Pelatihan data latih dan data uji menggunakan Backpropagation dengan bobot secara random. Pelatihan dilakukan dengan parameter pelatihan yang sama dengan komparasi tiga model optimasi SGD, Adam dan Adadelta dengan 100 epoch dan *learning rate* 0.001. hasil pelatihan dapat dilihat pada tabel 3.

Tabel 3. Komparasi tabel menggunakan model optimasi

Model Optimasi	Akurasi (%)	
	Data Latih	Data Uji
SGD	91	79
Adam	90	78
Adadelta	90	78

Hasil analisis pada tabel 2 menunjukan bahwa akurasi tertinggi diperoleh dengan model optimasi SGD dengan akutasi 91% untuk data latih dan 79% untuk data uji. Hal tersebut karena SGD menggunakan satu atau beberapa bagian saja dari data training yang dipilih secara acak.
Confusion matrix juga dilakukan untuk mengukur kinerja suatu metode klasifikasi. Hasil *confusion matrix* dapat dilihat pada tabel 4 dengan kelas EQUI (Equivalent), SIMI (Similar), SPE (Spesific), NOALI (No Alignment), REL (Related) dan OPPO (Opposite).

Tabel 4. Confusion matrix data uji

P/A	EQUI	SIMI	SPE	NOALI	REL	OPPO
EQUI	129	32	13	0	0	0
SIMI	17	119	5	0	0	0
SPE	11	12	56	0	0	0
NOALI	0	4	1	5	0	0
REL	0	0	0	0	1	0
OPPO	0	1	0	0	0	0

Pada tabel 4 menunjukan kinerja metode klasifikasi yang cukup baik, dengan meratanya pengujian terhadap kategori kelas yang ada. Namun, terdapat beberapa kelas yang sedikit teridentifikasi karena sesuai dengan jumlah kelas pada saat pelabelan yang tidak merata. Kelas Equivalent, Similar dan Spesific menjadi kelas yang memiliki pelabelan paling dominan.

Dari tabel 4, dapat dihasilkan nilai Precision, Recall dan F1-Score untuk setiap kategori kelas yang sudah dilabeli oleh anotator. Nilai tersebut dapat dilihat pada tabel 5.

Tabel 5. Precision, Recall dan F1-Score

Kelas	Precision(%)	Recall(%)	F1-Score(%)
Equivalent	81	82	81
Similar	76	82	79
Spesific	77	71	74
No Alignment	100	50	67
Related	100	100	100
Opposite	0	0	0

Dari tabel 5, untuk keseluruhan kelas diperoleh nilai Precision sebanyak 71%, Recall sebanyak 63% dan F1-Score sebanyak 66%, hal ini menunjukan sistem yang dibangun sudah mencapai performa terbaik yaitu dengan F1-Score sebanyak 66% terhadap kelas yang ditetapkan.

4 KESIMPULAN

Pada penelitian ini telah diimplementasikan model komputasi untuk mengukur kesamaan semantik pasangan kalimat untuk mengetahui skala hasil pengukuruan untuk pasangan kalimat yang dibandingkan, sehingga dapat membantu masyarakat sebagai analisis tren karya tulis ilmiah dikutip pada saat melakukan penelitian.

Hasil pengujian untuk setiap kategori kelas dengan menggunakan optimasi SGD mendapat akurasi 91% untuk data latih dan 79% untuk data uji dan 66% untuk nilai F1-Score seluruh kelas. Hal tersebut dipengaruhi oleh pelabelan data latih yang tidak merata untuk setiap kelas. Sehingga akurasi untuk kelas equivalent mendapat akurasi tertinggi karena kelas tersebut paling banyak terlabeli oleh anotator.
Saran untuk penelitian selanjutnya dapat dilakukan perbaikan dengan menyeimbangkan jumlah kelas yang dilabeli sehingga tidak terjadi ketimpangan jumlah kelas untuk setiap kategori atau dengan mengurangi jumlah kelas menjadi tiga yaitu, Equivalent, Similar dan Spesific.

DAFTAR PUSTAKA

[1] P. Ionescu and Cahill, “String Kernels for Native Language Identification,” *Dissertation Abstracts International, B: Sciences and*

Engineering, vol. 70, no. 8, p. 4943, 2016.

[2] S. Sophia, "Petunjuk sitasi serta cantuman daftar pustaka bahan pustaka online," *Seri Pengembangan Perpustakaan Pertanian*, no. 25, p. 2, 2002.

[3] D. Jurgens, S. Kumar, R. Hoover, D. McFarland, and D. Jurafsky, "Measuring the Evolution of a Scientific Field through Citation Frames," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 391–406, 2018.

[4] P. I. Nakov, A. S. Schwartz, and M. Hearst, "Citances: Citation sentences for semantic analysis of bioscience text," *Proceedings of the SIGIR'04 workshop on Search and Discovery in Bioinformatics*, 2004.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," pp. 1–12, 2013.

[6] A. Mahmoud and M. Zrigui, "Sentence Embedding and Convolutional Neural Network for Semantic Textual Similarity Detection in Arabic Language," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9263–9274, 2019.

[7] R. C. Rajagukguk and M. Leylia Khodra, "Interpretable Semantic Textual Similarity for Indonesian Sentence," *ICAICTA 2018 - 5th International Conference on Advanced Informatics: Concepts Theory and Applications*, pp. 147–152, 2018.

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *In Advances in neural information processing systems*, pp. 3111–3119, 2013.

[9] I. Gallo, S. Nawaz, and A. Calefati, "Semantic Text Encoding for Text Classification Using Convolutional Neural Networks," *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 5, pp. 16–21, 2018.