

## IMPLEMENTASI METODE *LONGEST COMMON SUBSEQUENCES* UNTUK PERBAIKAN KATA PADA KASUS ANALISIS SENTIMEN OPINI PEMBELAJARAN DARING DI MEDIA SOSIAL TWITTER

Alfisyar Jefry Pranata<sup>\*1</sup>, Yuita Arum Sari<sup>2</sup>, Edy Santoso<sup>3</sup>

<sup>1,2,3</sup>Universitas Brawijaya, Malang

Email: <sup>1</sup> alfisyar@student.ub.ac.id, <sup>2</sup> yuita@ub.ac.id, <sup>3</sup> edy144@ub.ac.id

\*Penulis Korespondensi

(Naskah masuk: 26 September 2021, diterima untuk diterbitkan: 31 Januari 2022)

### Abstrak

*Coronavirus* merupakan salah satu parasit yang menyerang sistem pernapasan manusia. Peningkatan kasus *coronavirus* berlangsung sangat cepat dan menyebar ke berbagai negara. Oleh karena itu, World Health Organization (WHO) menetapkan *Coronavirus* sebagai pandemi. Hal ini mengakibatkan seluruh kegiatan yang sebelumnya tatap muka atau luar jaringan (luring) menjadi dalam jaringan (daring), termasuk kegiatan belajar mengajar. Dengan ditetapkannya pembelajaran secara daring menyebabkan adanya opini yang bersifat pro dan kontra dari berbagai kalangan masyarakat. Opini tersebut akan digunakan dalam penelitian ini dan akan diolah terlebih dahulu dalam tahap *preprocessing*. Metode yang digunakan dalam penelitian ini adalah *Longest Common Subsequences* (LCS) dan *Support Vector Machine* (SVM) dengan data sebesar 500 yang terbagi menjadi 250 data berlabel positif dan 250 data berlabel negatif. Dari 500 data tersebut dibagi menjadi 450 data untuk data latih dan 50 data untuk data uji. Dengan menggunakan metode *Longest Common Subsequences* untuk perbaikan kata dan metode *Support Vector Machine* untuk klasifikasi dengan nilai parameter terbaik yaitu *learning rate* ( $\gamma$ ) = 0,0001, *lambda* ( $\lambda$ ) = 0,1, *complexity* (C) = 0,001, *epsilon* ( $\epsilon$ ) = 0,0001 dan iterasi maksimum = 50 dapat menghasilkan nilai rata-rata hasil evaluasi yaitu *precision* = 0,5653, *recall* = 0,948, *f-measure* = 0,7047 dan *accuracy* = 0,598. Hasil pengujian tersebut menunjukkan bahwa dengan menambahkan metode *Longest Common Subsequences* untuk perbaikan kata dapat meningkatkan tingkat akurasi yang sebelumnya hanya 0,59 menjadi 0,598.

**Kata kunci:** analisis sentimen, pembelajaran daring, longest common subsequences, support vector machine.

## IMPLEMENTATION METHOD OF LONGEST COMMON SUBSEQUENCES FOR WORD IMPROVEMENT IN THE CASE OF OPINION SENTIMENT ANALYSIS OF ONLINE LEARNING IN TWITTER SOCIAL MEDIA

### Abstract

*Coronavirus* is a parasite that attacks the human respiratory system. The increase incases *coronavirus* took place very fast and spread to various countries. Therefore, the World Health Organization (WHO) has designated *Coronavirus* as a pandemic. This results in all activities that were previously face-to-face or offline (offline) becoming online (online), including teaching and learning activities. With the establishment of online learning, there are pro and contra opinions from various circles of society. This opinion will be used in this research and will be processed first in the stage *preprocessing*. The method used in this research is *Longest Common Subsequences* (LCS) and *Support Vector Machine* (SVM) with 500 data divided into 250 data labeled positive and 250 data labeled negative. Of the 500 data is divided into 450 data for training data and 50 data for test data. By using the method *Longest Common Subsequences* for word improvement and the method *Support Vector Machine* for classification with the best parameter values, namely *learning rate* ( $\gamma$ ) = 0.0001, *lambda* ( $\lambda$ ) = 0.1, *complexity* (C) = 0.001, *epsilon* ( $\epsilon$ ) = 0.0001 and the maximum iteration = 50 can produce the average value of the evaluation results, namely *precision* = 0.5653, *recall* = 0.948, *f-measure* = 0.7047 and *accuracy* = 0.598. The test results show that by adding method of *Longest Common Subsequences* for word improvement, it can increase the level of accuracy which was previously only 0.59 to 0.598.

**Keywords:** sentiment analysis, online learning, longest common subsequences, support vector machine.

### 1. PENDAHULUAN

*Coronavirus* merupakan salah satu parasit yang menyerang sistem pernapasan manusia. *Coronavirus*

(CoV) termasuk dalam *Severe Acute Respiratory Syndrome* (SARS) dan *Middle East Respiratory Syndrome* (MERS) yang sebelumnya telah

ditetapkan sebagai ancaman kesehatan manusia (Rothan, et al., 2020). Peningkatan kasus *coronavirus* berlangsung sangat cepat dan menyebar ke berbagai negara. Oleh karena itu, WHO menetapkan *Coronavirus* sebagai pandemi. Di Indonesia, pada tanggal 2 maret 2020, telah mengonfirmasi kasus pertama *coronavirus*. Per tanggal 29 maret 2020, jumlah kasus telah meningkat sebanyak 1.285 kasus di 30 provinsi (Tosepu, et al., 2020).

Untuk bisa melawan COVID-19 pemerintah telah menetapkan beberapa peraturan, seperti selalu memakai masker, selalu cuci tangan, *social distancing* dan *physical distancing* serta larangan untuk berkerumun. Hal ini, akan menjadi hambatan dalam lembaga pendidikan. Oleh karena itu, lembaga pendidikan mengeluarkan surat edaran Kemendikbud Dikti No. 1 tahun 2020 yang memberikan instuksi kepada perguruan tinggi untuk melarang penyelenggaraan pembelajaran tatap muka dan sebagai gantinya akan diselenggarakan pembelajaran secara daring (Firman & Rahman, 2020).

Media sosial merupakan wadah untuk seseorang dapat bertukar informasi satu sama lain dan salah satu sosial media yang paling populer adalah twitter. Twitter merupakan layanan *microblogging real-time*, dimana pengguna mengirimkan pesan singkat dengan maksimal 140 karakter (*tweet*) kepada pengikut pengguna (Zhang, et al., 2016). Dengan adanya twitter, seseorang bisa mengekspresikan pendapat mereka mengenai berbagai macam hal, mulai dari opini, keluhan dan pembelajaran secara daring yang disimpan dalam pesan singkat (*tweet*).

Normalisasi merupakan suatu proses yang digunakan untuk mengubah kata yang dianggap tidak baku menjadi kata yang baku dalam sebuah kalimat (Ivan, et al., 2019). Normalisasi perlu dilakukan karena kebanyakan orang dalam mengekspresikan opininya di sosial media menggunakan kata yang tidak baku dan biasanya dalam sebuah kalimat terdapat kata yang salah ketik (*typo*), sedangkan dari segi sistemnya normalisasi juga perlu dilakukan karena ketika ada kata yang tidak baku maupun *typo*, maka kata tersebut bisa dianggap kata yang berbeda meskipun kata tersebut berarti sama.

Ada berbagai macam metode yang dapat digunakan untuk normalisasi kata seperti metode *Levenshtein Distance* dan *Longest Common Subsequences*. Dari penelitian sebelumnya yang membandingkan antara metode *Oliver*, *Longest Common Subsequences* dan *Levenshtein Distance* dapat ditarik kesimpulan bahwa metode yang menghasilkan kinerja sistem terbaik didapatkan oleh metode *Longest Common Subsequences* (Sembilu, et al., 2018), maka penelitian ini akan menggunakan metode *Longest Common Subsequences* untuk proses normalisasi kata.

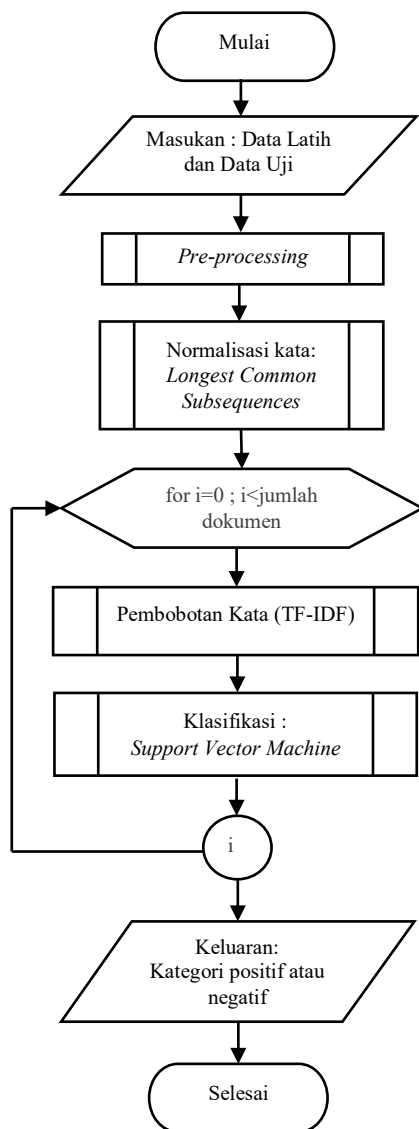
Analisis sentimen merupakan ilmu yang mempelajari tentang suatu opini, penilaian, emosi dalam sebuah entitas yang dapat berupa produk, pelayanan maupun topik tertentu (Suryanto, 2015). Tujuan dilakukannya analisis sentimen adalah untuk mengelompokkan suatu teks ke dalam kelas positif maupun kelas negatif. Dengan ditetapkannya pembelajaran secara daring menyebabkan adanya opini yang bersifat pro dan kontra dari berbagai kalangan masyarakat. Opini tersebut akan menjadi landasan untuk penelitian ini.

Untuk bisa mengelompokkan jenis opini tersebut, perlu dilakukannya klasifikasi teks terlebih dahulu. Algoritma klasifikasi merupakan inti dari teks klasifikasi. Ada berbagai macam algoritma dalam proses klasifikasi, yaitu *K-Nearest Neighbor* (KNN), *Naive Bayes* (NB) dan *Support Vector Machine* (SVM). Dari penelitian sebelumnya, ketika membandingkan kinerja dari ketiga algoritma klasifikasi tersebut dapat disimpulkan bahwa kinerja dari algoritma *Support Vector Machine* lebih baik dari pada algoritma KNN dan *Naive Bayes* (Zheng & Ye, 2009). Oleh karena itu, metode yang akan digunakan dalam penelitian ini adalah metode *Support Vector Machine* (SVM).

Dalam penelitian ini, peneliti berfokus pada normalisasi kata menggunakan metode *Longest Common Subsequences*. Proses pertama yang akan dilakukan adalah melakukan teknik *preprocessing* terlebih dahulu, seperti *cleaning*, *case folding*, *tokenizing*, *filtering* dan *stemming*. Kemudian akan dilakukan proses normalisasi kata menggunakan metode *longest common subsequences*. Setelah didapatkan kata yang sudah ternormalisasi, akan dilakukan proses pembobotan menggunakan TF-IDF, serta proses klasifikasi menggunakan metode *Support Vector Machine*. Dari penelitian yang telah diuraikan tersebut, bisa ditarik kesimpulan bahwa penambahan metode *Longest Common Subsequences* dapat digunakan untuk meningkatkan tingkat akurasi sistem terhadap analisis sentimen opini pembelajaran daring ke dalam kategori kelas positif dan negatif.

## 2. METODE PENELITIAN

Data yang digunakan dalam penelitian ini adalah data *tweet* dari media sosial twitter. Untuk mendapatkan dan mengumpulkan data tersebut menggunakan *library Twint* dengan bahasa *python*. Dari *library* tersebut, didapatkan data *tweet* sebanyak 500 data. 500 data *tweet* yang telah didapatkan akan dilakukan pelabelan jenis kategori positif atau negatif. kemudian data tersebut bisa digunakan sebagai data latih sebanyak 450 data *tweet* dan 50 data *tweet* akan digunakan dalam data uji. Gambaran secara umum mengenai algoritma yang digunakan pada penelitian ini dapat dilihat pada Gambar 1. Gambar 1 menjelaskan mengenai seluruh alur dari implementasi penelitian ini.



Gambar 1. Diagram Alir Implementasi Algoritme

Dari gambar 1, perancangan algoritme akan melakukan langkah-langkah dari implementasi penelitian yang dilakukan secara garis besar. Dalam penelitian ini, langkah pertama yang dilakukan adalah proses *pre-processing* data latih. Proses *pre-processing* terdiri dari *Cleaning*, *Case Folding*, *Tokenisasi*, *Filtering* dan *Stemming*. *Cleaning* merupakan proses penghapusan semua karakter selain huruf yang tidak mempunyai pengaruh terhadap pemrosesan teks, seperti tanda baca. *Case Folding* merupakan proses mengubah huruf yang ada dalam dokumen menjadi *lowercase*. *Tokenisasi* merupakan proses memecah dokumen kalimat menjadi dokumen kata. *Filtering* merupakan proses penghapusan kata yang tidak terlalu penting dalam dokumen. Proses *filtering* dalam penelitian ini menggunakan *stopword Tala*. *Stemming* merupakan proses mengubah bentuk kata menjadi kata yang paling dasar. Langkah selanjutnya adalah proses penerapan metode *Longest Common Subsequences*

(LCS). Kemudian dilakukan proses perhitungan *Term Frequency Inverse Document Frequency* (TF-IDF). Hasil dari perhitungan TF-IDF akan digunakan dalam proses klasifikasi menggunakan metode *Support Vector Machine* untuk mengklasifikasikan data latih kedalam salah satu kelas sentimen.

### 2.1. Text Mining

*Text mining* digunakan untuk menemukan, mengambil dan mengekstrak informasi dari korpus teks agar informasi tersebut bisa digunakan untuk tujuan tertentu (Wang, et al., 2017). *Text mining* menggabungkan teknologi seperti *Natural Language Processing*, *Artificial Intelligence*, *Information Retrieval*, dan *Data Mining* untuk membantu menemukan sebuah pola yang terkandung dalam teks. Tujuan dari *text mining* adalah untuk mengubah data yang tidak terstruktur menjadi lebih terstruktur. Data yang sudah diolah dalam *text mining* bisa digunakan untuk memenuhi kebutuhan informasi dalam berbagai metode klasifikasi dan analisis sentimen.

### 2.2. Analisis Sentimen

Analisis sentimen adalah sebuah ilmu yang mempelajari tentang sikap, penilaian, emosi dan opini orang mengenai suatu kelompok ataupun individu, masalah dan peristiwa (Liu & Zhang, 2012). Analisis sentimen digunakan untuk mengklasifikasikan atau menentukan nilai terhadap suatu permasalahan ke dalam kategori positif atau negatif. Nilai tersebut bisa digunakan sebagai acuan dalam upaya menentukan suatu keputusan.

### 2.3. Text Preprocessing

*Text Preprocessing* merupakan tahapan untuk menyeleksi data menjadi lebih terstruktur. Agar memperoleh data yang terstruktur, perlu dilakukan beberapa tahap *text preprocessing*, seperti *Data Cleaning* (menghapus semua karakter yang tidak mempunyai pengaruh terhadap pemrosesan teks), *Case Folding* (mengubah huruf dalam dokumen teks menjadi *lowercase*), *Tokenizing* (memecah dokumen kalimat menjadi dokumen kata), *Filtering* (penghapusan kata-kata yang tidak penting) dan *Stemming* (mengubah bentuk kata menjadi kata yang paling dasar).

### 2.4. Normalisasi

Normalisasi merupakan proses pengubahan teks atau kata yang tidak baku menjadi baku. Normalisasi pada teks bisa dilakukan pada pengubahan kata singkatan, akronim, simbol tertentu dan karakter khusus, sehingga kata tersebut tidak mempunyai arti yang ambigu (Sanjaya, 2019). Contoh normalisasi kata terdapat pada Tabel 1.

Tabel 1. Tabel Contoh Normalisasi

no	tanpa normalisasi	dengan normalisasi
1	Sdgkan	Sedangkan
2	Thdp	Terhadap
3	Tlg	Tolong
4	Krn	Karena
5	Sperti	Seperti

## 2.5. Kateglo

Kateglo merupakan sebuah layanan website (<http://kateglo.com>) yang memuat definisi, antonim, sinonim, lema bentukan dan glosarium tentang suatu frasa atau kata (Saniyah, 2019). Pada website kateglo mempunyai entri kamus sebanyak 72253, entri glosarium sebanyak 191200, entri peribahasa sebanyak 2012 dan entri singkatan dan akronim sebanyak 3423. Dengan memasukkan kata kunci yang dibutuhkan, kateglo dapat mengenali kata tersebut dari berbagai macam kamus, glosarium, peribahasa dan singkatan. Kateglo juga dapat digunakan untuk aplikasi dengan dengan API nya yang mempunyai keluaran berformat XML atau JSON.

## 2.6. Longest Common Subsequences

*Longest Common Subsequences* merupakan algoritma untuk menemukan urutan umum terpanjang dari 2 buah *string* (Hunt & Szymanski, 1977). Dalam algoritma LCS menggunakan *Subsequence* dari sebuah *string* yang berarti *string* tersebut mempunyai karakter yang urutan kemunculannya sama. Sebagai contoh, terdapat 2 buah *string*, yaitu *string* 1 dan *string* 2. *String* 1 berisi ABEFG sedangkan *string* 2 berisi BCEFH. Dari 2 buah *string* tersebut, bisa didapatkan *common subsequence*, yaitu BEF. Penyelesaian algoritma LCS bisa dilakukan dengan ilustrasi berikut:

Misalnya  $X = \langle x_1, x_2, x_3, \dots, x_i \rangle$  dan  $Y = \langle y_1, y_2, y_3, \dots, y_j \rangle$  dari kedua *string* tersebut didapatkan  $Z = \langle z_1, z_2, z_3, \dots, z_k \rangle$ .

1.  $x_i = y_j$  maka hasil dari LCS adalah  $x_{i-1}$  dan  $y_{j-1}$ .
2. Jika  $x_i \neq y_j$  maka  $z_k \neq x_i$  dan hasil dari LCS adalah  $x_{i-1}$  dan  $Y$ .
3. Jika  $x_i \neq y_j$  maka  $z_k \neq y_j$  dan hasil dari LCS adalah  $X$  dan  $y_{j-1}$ .

## 2.6. Pembobotan Kata

Pembobotan kata digunakan untuk memberikan nilai atau bobot pada masing-masing *term* atau kata pada suatu indeks/kata. Kata yang sudah diberi nilai atau bobot akan digunakan untuk proses klasifikasi (Rofiqoh, et al., 2017). Pada pembobotan kata ini terdapat beberapa tahapan, yaitu *term frequency*, *document frequency*, *inverse document frequency* dan *term frequency inverse document frequency*.

## 2.6. Sequential Support Vector Machine

*Sequential training* pada *Support Vector Machine* digunakan untuk mencari dan menemukan garis *hyperlane* terbaik dalam memisahkan dua kelas yang berbeda (Vijayakumar & Wu, 1999). Berikut adalah tahapan dari metode *sequential training*:

1. Inisialisasi parameter SVM yaitu nilai  $\lambda$  (*lambda*),  $C$  (*complexity*),  $\gamma$  (*gamma*),  $\epsilon$  (*epsilon*), maksimum iterasi, dan nilai  $\alpha_0=0$ .
2. Melakukan perhitungan matriks Hessian sesuai dengan Persamaan (1).

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda)^2 \quad (1)$$

3. Kemudian akan mencari nilai *support vector*, dengan menggunakan Persamaan (2), Persamaan (3) dan Persamaan (4).

$$E_{ij} = \sum_{i=0}^n \alpha_i D_{ij} \quad (2)$$

$$\delta \alpha_i = \min(\max(\gamma(1 - E_i), -\alpha_i), C - \alpha_i) \quad (3)$$

$$\alpha_i = \alpha_i + \delta \alpha_i \quad (4)$$

4. Melakukan perhitungan secara berulang pada tahap 4 sampai batas iterasi maksimum atau ketika nilai  $\delta \alpha_i$  (perubahan nilai  $\alpha_i$ ) kurang dari nilai epsilon.
5. Tahap selanjutnya adalah menghitung nilai bias dengan Persamaan (5).

$$b = -\frac{1}{2} (\sum_{i=0}^n \alpha_i K(x_i, x_i^-) + \sum_{i=0}^n \alpha_i K(x_i, x_i^+)) \quad (5)$$

6. Tahap selanjutnya adalah menghitung nilai kemiripan dengan Persamaan (6).

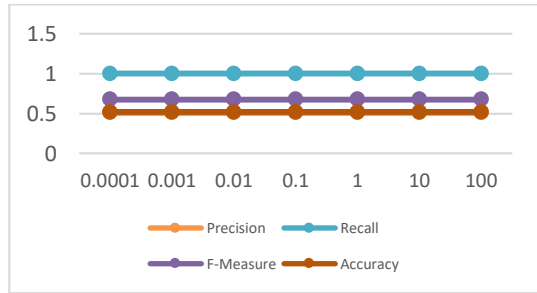
$$f(x) = \sum_{i=0}^n \alpha_i y_i K(x, x_i) + b \quad (6)$$

## 3. PENGUJIAN DAN ANALISIS

Penelitian ini akan menggunakan 2 macam skenario pengujian dan analisis. Pengujian dan analisis yang pertama akan dilakukan terhadap setiap parameter yang digunakan dalam metode *Support Vector Machine* yaitu *Learning Rate* ( $\gamma$ ), *Lambda* ( $\lambda$ ), *Complexity* ( $C$ ), *Epsilon* ( $\epsilon$ ) dan Iterasi Maksimum. Pengujian parameter ini bertujuan untuk mendapatkan nilai terbaik dari setiap parameter yang digunakan agar sistem bisa menghasilkan nilai akurasi yang tinggi. Pengujian dan analisis kedua adalah pengujian pengaruh perbaikan kata menggunakan metode *Longest Common Subsequences*. Pengujian ini bertujuan untuk menguji pengaruh perbaikan kata *Longest Common Subsequences* terhadap nilai akurasi yang dihasilkan sistem.

**3.1. Pengujian Parameter Learning Rate ( $\gamma$ )**

Pengujian parameter *Learning rate* ( $\gamma$ ) bertujuan untuk mendapatkan nilai dari parameter *learning rate* ( $\gamma$ ) yang terbaik. Pengujian ini menggunakan nilai parameter *Lambda* ( $\lambda$ ) = 0,1, *Complexity* (*C*) = 0,0001, *Epsilon* ( $\epsilon$ ) = 0,0001 dan Iterasi Maksimum = 50. Pengujian parameter *learning rate* ( $\gamma$ ) diujikan pada nilai 0,0001, 0,001, 0,01, 0,1, 1, 10 dan 100. Hasil pengujian parameter *Learning rate* ( $\gamma$ ) terdapat pada Gambar 2.

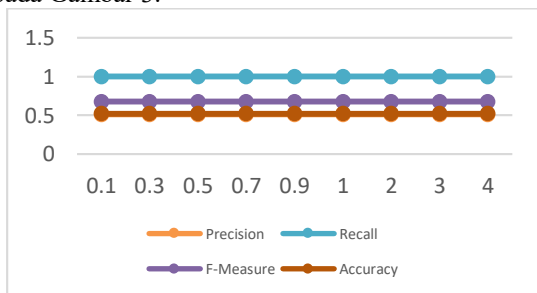


Gambar 2. Diagram Pengujian Parameter Learning Rate

Gambar 2 menunjukkan ketika diuji dengan parameter *learning rate*, hasil dari nilai *precision*, *recall*, *f-measure* dan *accuracy* bernilai sama pada tiap nilai *learning rate* yang diujikan yaitu *precision* = 0,51, *recall* = 1, *f-measure* = 0,676 dan *accuracy* = 0,52. Dapat disimpulkan bahwa parameter *learning rate* tidak berpengaruh terhadap nilai akurasi yang dihasilkan. Nilai dari parameter *learning rate* sendiri digunakan pada saat menghitung perubahan nilai *alpha* ( $\delta\alpha_i$ ). Nilai ini juga digunakan untuk menghentikan iterasi yang sedang berjalan ketika kondisi dari nilai terbesar perubahan nilai *alpha* ( $\delta\alpha_i$ ) kurang dari nilai *epsilon*. Nilai parameter *learning rate* yang akan digunakan dalam pengujian selanjutnya adalah 0,0001.

**3.2. Pengujian Parameter Lambda ( $\lambda$ )**

Pengujian parameter *Lambda* ( $\lambda$ ) bertujuan untuk mendapatkan nilai dari parameter *Lambda* ( $\lambda$ ) yang terbaik. Pengujian ini menggunakan nilai parameter *learning rate* ( $\gamma$ ) = 0,0001, *Complexity* (*C*) = 0,0001, *Epsilon* ( $\epsilon$ ) = 0,0001 dan Iterasi Maksimum = 50. Pengujian parameter *Lambda* ( $\lambda$ ) diujikan pada nilai 0,1, 0,3, 0,5, 0,7, 0,9, 1, 2, 3 dan 4. Hasil pengujian parameter *Lambda* ( $\lambda$ ) terdapat pada Gambar 3.

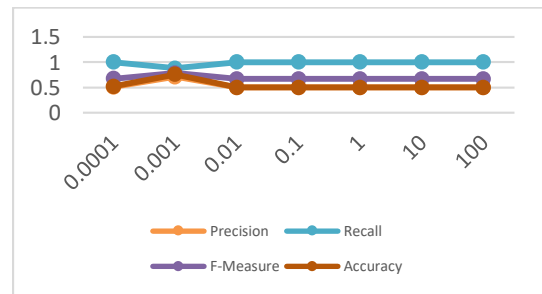


Gambar 3. Diagram Pengujian Parameter Lambda

Gambar 3 menunjukkan ketika diuji dengan parameter *lambda*, hasil dari nilai *precision*, *recall*, *f-measure* dan *accuracy* bernilai sama pada tiap nilai *lambda* yang di ujikan yaitu *precision* = 0,51, *recall* = 1, *f-measure* = 0,676 dan *accuracy* = 0,52. Dapat disimpulkan bahwa parameter *lambda* tidak berpengaruh terhadap nilai akurasi yang dihasilkan. Nilai dari parameter *lambda* sendiri digunakan pada saat menghitung matriks *Hessian*. Nilai parameter *lambda* yang akan digunakan dalam pengujian selanjutnya adalah 0,1.

**3.3. Pengujian Parameter Complexity (C)**

Pengujian parameter *Complexity* (*C*) bertujuan untuk mendapatkan nilai dari parameter *Complexity* (*C*) yang terbaik. Pengujian ini menggunakan nilai parameter *learning rate* ( $\gamma$ ) = 0,0001, *Lambda* ( $\lambda$ ) = 0,1, *Epsilon* ( $\epsilon$ ) = 0,0001 dan Iterasi Maksimum = 50. Pengujian parameter *Complexity* (*C*) diujikan pada nilai 0,0001, 0,001, 0,01, 0,1, 1, 10 dan 100. Hasil pengujian parameter *Complexity* (*C*) terdapat pada Gambar 4.



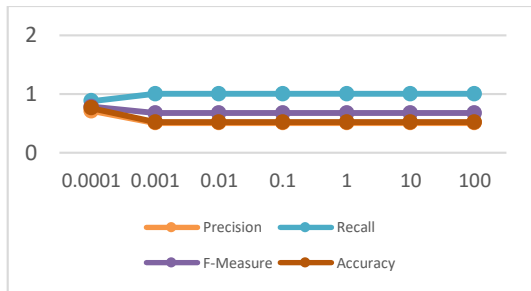
Gambar 4. Diagram Pengujian Parameter Complexity

Gambar 4 menunjukkan ketika diuji dengan parameter *complexity*, hasil terbaik dari nilai *precision*, *recall*, *f-measure* dan *accuracy* pada setiap nilai *complexity* yang diujikan yaitu *precision* = 0,71, *recall* = 0,88, *f-measure* = 0,786 dan *accuracy* = 0,76. Ketika nilai parameter *complexity* diperbesar, akurasi yang dihasilkan juga bertambah besar. Namun jika parameter *complexity* diperbesar lagi, akan terjadi penurunan nilai akurasi. Nilai dari parameter *complexity* sendiri digunakan pada saat menghitung perubahan nilai *alpha* ( $\delta\alpha_i$ ) dari kondisi konvergen sistem. Ketika setelah nilai *complexity* sebesar 0,001 terus diperbesar, maka nilai perubahan *alphanya* tidak bisa mencapai kondisi konvergen. Nilai parameter *complexity* yang akan digunakan dalam pengujian selanjutnya adalah 0,001.

**3.4. Pengujian Parameter Epsilon ( $\epsilon$ )**

Pengujian parameter *Epsilon* ( $\epsilon$ ) bertujuan untuk mendapatkan nilai dari parameter *Epsilon* ( $\epsilon$ ) yang terbaik. Pengujian ini menggunakan nilai parameter *learning rate* ( $\gamma$ ) = 0,0001, *Lambda* ( $\lambda$ ) = 0,1, *Complexity* (*C*) = 0,001 dan Iterasi Maksimum = 50. Pengujian parameter *Epsilon* ( $\epsilon$ ) diujikan pada nilai 0,0001, 0,001, 0,01, 0,1, 1, 10 dan 100. Hasil

pengujian parameter *Epsilon* ( $\epsilon$ ) terdapat pada Gambar 5.

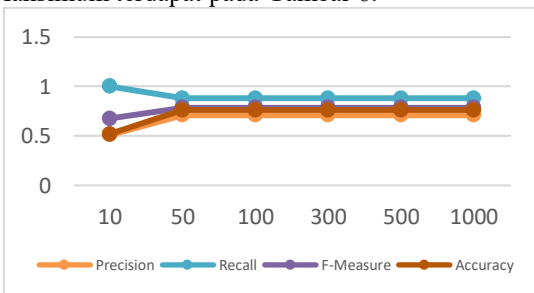


Gambar 5. Diagram Pengujian Parameter Epsilon

Gambar 5 menunjukkan ketika diuji dengan parameter *epsilon*, hasil terbaik dari nilai *precision*, *recall*, *f-measure* dan *accuracy* pada setiap nilai *complexity* yang diujikan yaitu *precision* = 0,71, *recall* = 0,88, *f-measure* = 0,786 dan *accuracy* = 0,76. Nilai dari parameter *epsilon* sendiri digunakan pada saat menghitung perubahan nilai  $\alpha_i$ . Semakin besar nilai *epsilon* maka akan mengakibatkan akurasi yang dihasilkan semakin kecil, karena jika nilai  $\alpha$  lebih kecil dari nilai *epsilon* maka sistem akan terjadi konvergensi dan iterasi akan berhenti sebelum menemukan nilai  $\alpha$  yang optimal. Nilai parameter *epsilon* yang akan digunakan dalam pengujian selanjutnya adalah 0,0001.

### 3.5. Pengujian Parameter Iterasi Maksimum

Pengujian parameter Iterasi Maksimum bertujuan untuk mendapatkan nilai dari Iterasi Maksimum yang terbaik. Pengujian ini menggunakan nilai parameter *learning rate* ( $\gamma$ ) = 0,0001, *Lambda* ( $\lambda$ ) = 0,1, *Complexity* (*C*) = 0,001 dan *Epsilon* ( $\epsilon$ ) = 0,0001. Pengujian parameter Iterasi Maksimum diujikan pada nilai 10, 50, 100, 300, 500 dan 1000. Hasil pengujian parameter Iterasi Maksimum terdapat pada Gambar 6.



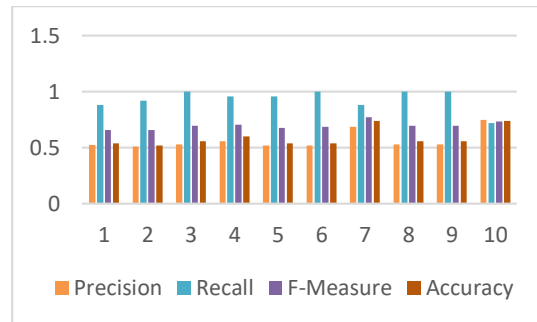
Gambar 6. Diagram Pengujian Parameter Iterasi Maksimum

Gambar 6 menunjukkan ketika diuji dengan parameter iterasi maksimum, hasil terbaik dari nilai *precision*, *recall*, *f-measure* dan *accuracy* pada setiap nilai iterasi maksimum yang diujikan yaitu *precision* = 0,71, *recall* = 0,88, *f-measure* = 0,786 dan *accuracy* = 0,76. Nilai dari parameter iterasi maksimum sendiri digunakan pada saat menghitung nilai  $\alpha_i$ . Ketika nilai iterasi maksimum

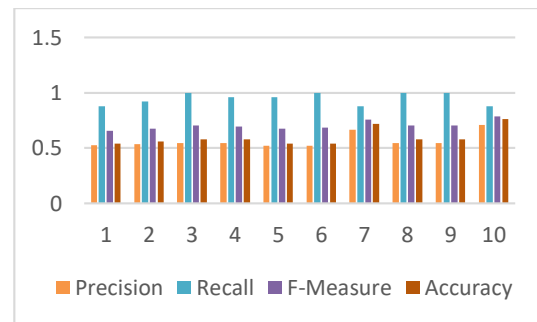
bernilai rendah, maka dapat mengakibatkan nilai  $\alpha$  yang dihasilkan belum mencapai konvergensi yang optimal. Ketika iterasi maksimum bernilai tinggi, maka nilai  $\alpha$  yang dihasilkan bisa mencapai nilai konvergen yang optimal. Nilai parameter iterasi maksimum yang akan digunakan dalam pengujian selanjutnya adalah 50.

### 3.6. Pengujian Parameter Longest Common Subsequences

Pengujian pengaruh *Longest Common Subsequences* ini akan diujikan dengan data yang telah dibagi menjadi 10 *fold*. Pengujian ini menggunakan nilai parameter *learning rate* ( $\gamma$ ) = 0,0001, *Lambda* ( $\lambda$ ) = 0,1, *Complexity* (*C*) = 0,001, *Epsilon* ( $\epsilon$ ) = 0,0001 dan Iterasi Maksimum = 50. Hasil pengujian tanpa *Longest Common Subsequences* terdapat pada Gambar 7 dan pengujian dengan *Longest Common Subsequences* terdapat pada Gambar 8.



Gambar 7. Diagram Pengujian Tanpa Longest Common Subsequences



Gambar 8. Diagram Pengujian Dengan Longest Common Subsequences

Dari pengujian yang telah dilakukan, dapat dirata-ratakan nilai akurasi dari 10 *fold* yang telah diuji pada masing-masing pengujian. Rata-rata dari nilai akurasi tanpa menggunakan perbaikan kata *Longest Common Subsequences* (Gambar 7) adalah sebesar 59% sedangkan rata-rata dari nilai akurasi menggunakan *Longest Common Subsequences* (Gambar 8) adalah sebesar 59,8%.

Perbaikan kata ini dilakukan pada semua kata yang dianggap tidak baku oleh sistem. Ketika ada kata yang tidak baku (*typo* maupun singkatan namun mempunyai arti kata yang sama) dapat dijadikan satu kata yang baku sesuai dengan kamus.

Penambahan perbaikan kata menggunakan *Longest Common Subsequences* telah terbukti mempunyai pengaruh terhadap akurasi sistem yang telah dibuat. Akurasi yang dihasilkan lebih tinggi dari pada tanpa menggunakan perbaikan kata *Longest Common Subsequences*. Dari kedua macam pengujian, terdapat selisih sebesar 0,8%.

Ketika dilakukan penambahan perbaikan kata menggunakan *Longest Common Subsequences*, terdapat kesalahan dalam memperbaiki kata yang tidak baku menjadi baku sesuai dengan kamus. Faktor yang memicu kesalahan tersebut diakibatkan oleh kata yang mempunyai *subsequences* bernilai sama dalam perhitungan *Longest Common Subsequences*. Contohnya dalam dokumen terdapat kata “skrg”, ketika diperbaiki dengan *Longest Common Subsequences* menjadi “sekring”, padahal kata baku yang seharusnya adalah “sekarang”. Contoh lainnya adalah terdapat kata “bngt” ketika diperbaiki dengan *Longest Common Subsequences* menjadi “bangat”, padahal kata baku yang seharusnya adalah “banget”. Hal ini diakibatkan oleh kata “sekring” dan “bangat” berada pada urutan awal dalam kamus dan kata tersebut juga mempunyai *subsequences* bernilai sama dengan kata yang tidak baku.

#### 4. KESIMPULAN DAN SARAN

Dari penelitian yang sudah dilakukan, dapat diambil kesimpulan yaitu dengan menggunakan metode *Support Vector Machine* untuk klasifikasi analisis sentimen opini pembelajaran daring dengan nilai parameter terbaik yaitu *Learning rate* ( $\eta$ ) = 0,0001, *Lambda* ( $\lambda$ ) = 0,1, *Complexity* ( $C$ ) = 0,001, *Epsilon* ( $\epsilon$ ) = 0,0001 dan Iterasi Maksimum = 50 dapat menghasilkan nilai rata-rata dari *Precision* = 0,567, *Recall* = 0,932, *F-Measure* = 0,697 dan *Accuracy* = 0,59.

Sedangkan jika ditambahkan perbaikan kata menggunakan *Longest Common Subsequences* di dapatkan nilai rata-rata evaluasi dari *Precision* = 0,5653, *Recall* = 0,948, *F-Measure* = 0,7047 dan *Accuracy* = 0,598. Penambahan perbaikan kata menggunakan *Longest Common Subsequences* telah terbukti mempunyai pengaruh terhadap akurasi sistem yang telah dibuat. Akurasi yang dihasilkan lebih tinggi dari pada tanpa menggunakan perbaikan kata *Longest Common Subsequences* dengan selisih sebesar 0,008.

Ada beberapa saran untuk bisa diimplementasikan pada penelitian ini, yaitu dapat menambahkan korpus bahasa gaul (*slang Language*) yang sering digunakan dalam bahasa sehari-hari agar sistem dapat memperbaiki kata lebih optimal, dapat menambahkan deteksi bahasa asing agar kata yang menggunakan bahasa asing tidak akan ikut diperbaiki, dapat menggunakan kamus KBBI yang sebelumnya menggunakan kamus kateglo dan dapat menggabungkan algoritma lain agar sistem yang

dibuat bisa menghasilkan hasil evaluasi yang lebih baik.

#### DAFTAR PUSTAKA

- FIRMAN & RAHMAN, S. R., 2020. Pembelajaran Online di Tengah Pandemi Covid-19. *Indonesian Journal of Educational Science (IJES)*, Volume 2, pp. 81-89.
- HUNT, J. W. & SZYMANSKI, T. G., 1977. A Fast Algorithm for Computing Longest Common Subsequences. *Journal of the ACM (JACM)*, Volume 20.
- IVAN, SARI, Y. A. & ADIKARA, P. P., 2019. Klasifikasi Hate Speech Berbahasa Indonesia di Twitter Menggunakan Naive Bayes dan Seleksi Fitur Information Gain dengan Normalisasi Kata. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 3, pp. 4914-4922.
- LIU, B. & ZHANG, L., 2012. A Survey of Opinion Mining and Sentiment Analysis. *University of Illinois at Chicago*, pp. 415-522.
- ROFIQOH, U., PERDANA, S. R. & Fauzi, A. M., 2017. Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexicon Based Features. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 1, pp. 1725-1732.
- ROTHAN, H. A., N, S. & BYRAREDDY, N., 2020. The Epidemiology and Pathogenesis of Coronavirus Disease (COVID-19) Outbreak. *Journal of Autoimmunity*.
- SANIYAH, Z., 2019. Normalisasi Mikroteks Berbentuk Singkatan pada Teks Twitter Berbahasa Indonesia Menggunakan Algoritma Longest Common Subsequences. *Repositori Institusi USU*.
- SEMBILU, N., SAMOPA, F. & ER, M., 2018. Perbandingan Algoritma Kemiripan Teks untuk Perbaikan dan Saran Penulisan Frasa dalam Bahasa Alami. *Open Access Journal Information Systems*, Volume 08, pp. 19-32.
- SURYANTO, A., 2015. Implementasi Metode Improved Fuzzy C-Means pada Analisis Sentimen Review Aplikasi Mobile Bahasa Indonesia.
- TOSEPU, R. et al., 2020. Correlation Between Weather and Covid-19 Pandemic in Jakarta, Indonesia. *Elsevier Public Health Emergency Collection*.
- VIJAYAKUMAR, S. & WU, S., 1999. *Sequential Support Vector Classifier and Regression*. Genoa, Proc. International Conference on Soft Computing (SOCO'99).
- WANG, P. et al., 2017. Classification of Proactive Personality: Text Mining Based on Weibo

Text and Short-answer Questions Text.  
*IEEE Access*, Volume XX.

ZHANG, Y., RUAN, X., WANG, H. & WANG, H.,  
2016. Twitter Trends Manipulation: A First  
Look Inside. *IEEE Xplore*, pp. 1-14.

ZHENG, W. & YE, Q., 2009. Sentiment  
Classification of Chinese Traveler Reviews  
by Support Vector Machine Algorithm.  
*IEEE Xplore*, pp. 335-338.