TUGAS AKHIR- KM184801

ANALISIS SENTIMEN TANGGAPAN PELANGGAN OPERATOR TELEKOMUNIKASI DI TWITTER DENGAN ALGORITMA DCNN-SVM

INAYAH EKA FIRDAUSI
NRP 06111540000018

Dosen Pembimbing :
Dr. Imam Mukhlash, S.Si, M.T
Drs. Nurul Hidayat, M.Kom

DEPARTEMEN MATEMATIKA
Fakultas Matematika Komputasi dan Sains Data
Institut Teknologi Sepuluh Nopember
Surabaya 2019

FINAL PROJECT– KM184801

SENTIMENT ANALYSIS OF CUSTOMER RESPONSE OF TELECOMMUNICATION OPERATOR IN TWITTER USING DCNN-SVM ALGORITHM

INAYAH EKA FIRDAUSI
NRP 06111540000018

Supervisors :
Dr. Imam Mukhlash, S.Si, M.T
Drs. Nurul Hidayat, M.Kom

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics, Computating, and Data Science
Sepuluh Nopember Institute of Technology
Surabaya 2019

LEMBAR PENGESAHAN

ANALISIS SENTIMEN TANGGAPAN PELANGGAN OPERATOR TELEKOMUNIKASI DI TWITTER DENGAN ALGORITMA DCNN-SVM

SENTIMENT ANALYSIS OF CUSTOMER RESPONSE OF TELECOMMUNICATION OPERATOR IN TWITTER USING DCNN-SVM ALGORITHM

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Matematika
Pada bidang studi ilmu komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
INAYAH EKA FIRDAUSI
NRP. 06111540000018

Menyetujui,

Dosen Pembimbing II,	Dosen Pembimbing I,
Drs. Nurul Hidayat, M.Kom	Dr. Imam Mukhlash, S.Si, MT
NIP. 19630404 198903 1 002	NIP. 19700831 199403 1 003

Mengetahui,
Kepala Departemen Matematika
FMKSD ITS

Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003

Surabaya, Juli 2019

ANALISIS SENTIMEN TANGGAPAN PELANGGAN OPERATOR TELEKOMUNIKASI DI TWITTER DENGAN ALGORITMA DCNN-SVM

Nama : Inayah Eka Firdausi
NRP : 06111540000018
Departemen : Matematika FMKSD - ITS
Pembimbing : 1. Dr. Imam Mukhlash, S.Si, M.T
2. Drs. Nurul Hidayat, M.Kom

ABSTRAK

Seiring perkembangan zaman, media sosial banyak diminati oleh berbagai kalangan masyarakat karena media sosial memungkinkan penggunanya untuk mengungkapkan pikiran atau perasaan mereka secara bebas. Penting bagi sebuah perusahaan untuk mengetahui tanggapan publik mengenai produk atau layanan yang ditawarkan. Dengan tanggapan publik ini, perusahaan dapat menganalisis kebutuhan pelanggan dan membuat perencanaan produk atau layanan yang lebih memuaskan. Untuk dapat mengetahui sentimen dari tanggapan, maka perlu pengklasifikasian tanggapan. Oleh karena itu, pada penelitian ini digunakan metode *Deep Convolutional Neural Network* (DCNN) sebagai pengekstraksi fitur dan *Support Vector Machine* (SVM) sebagai pengklasifikasiannya. Hasil performansi dari penelitian ini yaitu akurasi data uji sebesar 63%, presisi data uji sebesar 63% dan *recall* data uji sebesar 50%.

Kata Kunci: *Analisis Sentimen, Deep Convolutional Neural Network, Support Vector Machine, Twitter.*

SENTIMENT ANALYSIS OF CUSTOMER RESPONSE OF TELECOMMUNICATION OPERATOR IN TWITTER USING DCNN-SVM ALGORITHM

Name	**: Inayah Eka Firdausi**
NRP	**: 06111540000018**
Department	**: *Mathematics FMCDS - ITS***
Supervisors	**: 1. Dr. Imam Mukhlash, S.Si, M.T**
	2. Drs. Nurul Hidayat, M.Kom

ABSTRACT

Along with the development of the times, social media is in great demand by various circles of society because social media allows users to express their thoughts or feelings freely. It is important for a company to know public responses about the product or service offered. With this public response, companies can analyze customer needs and plan more satisfying products or services. To be able to know the sentiments of responses, it is necessary to classify responses. Therefore, in this study used the Deep Convolutional Neural Network (DCNN) method as a feature extraction and Support Vector Machine (SVM) as its classification. The performance results of this research are 63% for accuracy of test data, 63% for precision of test data and 50% for recall of test data..

Keywords*: Sentiment Analysis, Deep Convolutional Neural Network, Support Vector Machine, Twitter.*

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillaahirobbil'aalamiin, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan limpahan rahmat, taufik dan hidayah – Nya, sehingga penulis dapat menyelesaikan penelitian yang berjudul "**ANALISIS SENTIMEN TANGGAPAN PELANGGAN OPERATOR TELEKOMUNIKASI DI TWITTER DENGAN ALGORITMA DCNN-SVM**" sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika FMKSD Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Penelitian ini dapat terselesaikan dengan baik dan tepat waktu berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terimakasih kepada :

1. Mama dan bapak yang sangat berjasa dalam hidup penulis, Sheril selaku adik penulis serta keluarga besar penulis yang telah banyak mendukung dan memberikan semangat dalam menjalani masa perkuliahan.
2. Bapak Dr. Imam Mukhlash, S.Si, M.T dan Bapak Drs. Nurul Hidayat, M.Kom selaku dosen pembimbing penelitian penulis yang sudah meluangkan banyak waktu untuk membimbing penulis dan terima kasih atas dedikasinya untuk kebaikan penulis.
3. Bapak Prof. Dr. Mohammad Isa Irawan, M.T, Bapak Drs. Soetrisno, MI.Komp. dan Ibu Dra. Wahyu Fistia Doctorina, M.Si selaku dosen penguji yang telah memberikan kritik, saran dan masukan yang membangun dalam menyelesaikan penelitian.

4. Bapak Dr. Didik Khusnul Arif, S.Si, M.Si sebagai dosen wali penulis, yang telah memberikan motivasi, bimbingan dan pengarahan dalam menyelesaikan penelitian ini.
5. Sahabat penulis, Nenek, Mbak Cungcung, Elsa, Cindy, Syuliliq, Nedor, Rama, Athyah, dan Komtiang yang selalu memberikan dukungan dan doa kepada penulis.
6. Teman kelompok Alan Turing, Capt, Faris, Luthfi, Michi, Corry, Evika, Rika dan pendamping kelompok Mbak Tiara yang sudah membantu penulis selama menjadi mahasiswa baru di ITS.
7. Teman–teman mahasiswa Ijul, Dina, Anita, Nirwana, Kakek Rikochan, Sima dan seluruh mahasiswa Matematika 2015 serta keluarga STI-50 DOHMAIn (yang tidak dapat penulis sebut satu persatu) yang telah banyak mendukung dan memberi semangat kepada penulis selama masa perkuliahan.
8. Elfi dan Ardi yang sudah membantu penulis untuk menjadi anotator.
9. Bapak Dr. Imam Mukhlash, S.Si, M.T sebagai Kepala Departemen Matematika FKMSD ITS.
10. Bapak Dr. Didik Khusnul Arif, S. Si., M. Si. selaku Ketua Program Studi S-1 Departemen Matematika ITS dan Bapak Drs. Iis Herisman, M. Si. selaku Sekretaris Program Studi S-1 Departemen Matematika yang selama ini sudah bekerja keras dalam membantu dan menanggapi kebutuhan penulis.
11. Seluruh Bapak/Ibu dosen dan seluruh staff Departemen Matematika ITS yang selama perjalanan kuliah telah memberikan pelajaran berharga kepada penulis baik akademik maupun moral.

12. Semua pihak yang telah memberikan dukungan dan ilmu kepada penulis dalam masa perkuliahan hingga penyelesian penelitian ini.

Penulis menyadari bahwa dalam penelitian ini masih terdapat kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan. Akhirnya penulis berharap semoga penelitian ini dapat bermanfaat bagi banyak pihak.

Wassalamu'alaikum Wr. Wb.

Surabaya, Juli 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE

BAB 1
PENDAHULUAN

Pada bab ini dibahas latar belakang yang mendasari penulisan penelitian. Uraian ini bersifat umum yang menjelaskan hal – hal yang dilakukan pada penyelesaian penelitian. Kemudian dijabarkan dalam rumusan masalah, batasan masalah, tujuan, dan manfaat yang diambil berdasarkan latar belakang penyusunan penelitian ini.

1.1. Latar Belakang

Era *big data* dan *internet of thing* merambah pada hampir semua bidang, salah satu bidang yang paling terpengaruh adalah bidang informasi dan komunikasi, oleh karena itu manusia membutuhkan telepon sebagai alat komunikasi. Menurut Databoks, pengguna telepon seluler di Indonesia mencapai 371, 4 juta pengguna atau 142% dari total populasi sebanyak 262 juta jiwa [1]. Artinya, rata-rata setiap penduduk Indonesia memakai 1-2 telepon seluler karena satu orang terkadang menggunakan 2-3 kartu telepon seluler. Berdasarkan Siaran Pers No. 112/HM/KOMINFO/05/2018, menyatakan bahwa jumlah pelanggan provider di Indonesia adalah sejumlah 254.792.159 pelanggan. Jumlah pengguna telepon seluler yang terus meningkat ini dapat dimanfaatkan oleh perusahaan yang memberikan layanan telekomunikasi untuk memberikan tawaran yang menarik kepada pelanggan seperti paket internet, sms, dan telepon.

Pengguna internet di dunia semakin meningkat setiap tahunnya begitu pula di Indonesia. Menurut hasil survei APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) pada tahun 2017, penetrasi pengguna internet di Indonesia mencapai 143.26 juta [2]. Angka pengguna internet di Indonesia ini

mengalami kenaikan 7.96 % dibandingkan dengan tahun 2016 [2]. Angka ini menunjukkan penetrasi pengguna internet sebesar 54.68% dari total populasi penduduk Indonesia [2]. Salah satu penggunaan internet terbesar adalah untuk media sosial yaitu 87.13% dari jumlah penetrasi pengguna internet [2]. Media sosial yang paling sering dikunjungi di Indonesia adalah youtube 43%, facebook 41%, disusul dengan instagram 38%, lalu twitter 27%, dan google plus 25% [3]. Adanya media sosial telah memberikan wadah bagi pengguna internet untuk mengekspresikan dan berbagi pemikiran serta pendapat mereka tentang topik atau acara yang berbeda. Twitter adalah salah satu media sosial yang digunakan oleh perusahaan-perusahaan salah satunya operator telekomunikasi untuk memonitor reputasi dan merek mereka dengan mengekstrak dan menganalisis sentiment dari *tweet* yang diposting oleh publik tentang mereka, pasar mereka, dan pesaing.

Analisis sentimen adalah suatu *task* untuk mengindetifikasi dan mengelompokkan sentimen dan pendapat yang diungkapkan dalam sebuah teks untuk memahami sikap terhadap sebuah produk tertentu, topik, layanan dan sebagainya. Tugas dasar dalam analisis sentimen adalah mengelompokkan polaritas dari teks yang ada dalam dokumen, kalimat, atau pendapat. Polaritas mempunyai arti apakah teks yang ada dalam dokumen, kalimat, atau pendapat memiliki aspek positif atau negatif . Pada penelitian kali ini mengolah data yang terkait analisis sentimen melalui twitter. Analisis sentimen atas data twitter dan mikro-blog serupa lainnya menghadapi beberapa tantangan baru karena panjang pendeknya *tweet* tidak teratur struktur kontennya. Twitter dengan 280 karakter membuat penggunanya harus menyingkat suatu kata dan menyisipkan suatu slang. Hal ini menunjukkan perlu dilakukan suatu

ekstraksi terhadap hal tersebut. Proses ini dilakukan dengan menggunakan metode *Natural Language Processing* dan metode analisis teks [4]. Penting bagi sebuah perusahaan atau organisasi untuk mengetahui tanggapan publik mengenai produk atau layanan yang mereka tawarkan, dengan tanggapan publik ini, perusahaan dapat menganalisis kebutuhan pelanggan dan membuat perencanaan produk atau layanan yang lebih memuaskan. Disamping itu tidak bisa dipungkiri bahwa tanggapan yang muncul dari publik dapat mempengaruhi citra dari sebuah perusahaan [5]. Akan tetapi, memantau dan mengorganisasi tanggapan dari masyarakat di sosial media juga bukanlah hal yang mudah. Tanggapan yang dimuat jumlahnya terlalu banyak untuk diproses secara manual. Oleh sebab itu, diperlukan sebuah metode atau teknik khusus yang mampu mengkategorikan tanggapan di sosial media tersebut secara otomatis, apakah termasuk positif atau negatif.

Banyak ide telah muncul selama beberapa tahun belakangan tentang teknik *machine learning* untuk permasalahan analisis sentimen. Terdapat banyak metode yang telah digunakan dalam penggalian suatu opini. Penelitian dari Pang dan Lee menggali opini dari ulasan film dengan menggunakan metode *Naïve Bayes* dan *Support Vector Machine* (SVM) serta dengan seleksi fitur berupa *Based on Minimum Cut* [6]. Penelitian tersebut menghasilkan performansi sebesar 86.4% [6]. Pada penelitian Rozi dkk tentang "Opinion Mining On Book Review Using CNN-L2-SVM Algorithm" didapatkan performansi sebesar 83.23% untuk data latih dan sebesar 64.4% untuk data uji [7]. Pada Tahun 2018, Mukhlash dkk juga meneliti tentang "Opinion Mining On Book Review Using CNN-LSTM Algorithm" dan

didapatkan performansi sebesar 99.55% untuk data latih dan sebesar 65.03% untuk data uji [8].

Oleh karena itu, pada penelitian ini digunakan sebuah kombinasi metode DCNN-SVM dalam pengklasifikasian suatu teks. Fungsi aktivasi pada metode DCNN ini yang digunakan adalah SVM. Metode ini mengklasifikasikan suatu opini menjadi dua kelas utama yaitu positif dan negatif. Selain itu didapatkan akurasi dari metode tersebut sebagai bahan pertimbangan dalam penelitian selanjutnya.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah disajikan di atas, permasalahan yang dibahas dalam penelitian ini adalah sebagai berikut:

1. Bagaimana cara melakukan analisis sentimen tanggapan pelanggan operator telekomunikasi di twitter menggunakan algoritma DCNN-SVM?
2. Bagaimana cara menganalisis berapa banyak tweet yang bersifat negatif dan positif di twitter operator telekomunikasi serta mendapatkan kata yang paling banyak muncul dalam *tweet* pelanggan?
3. Bagaimana akurasi algoritma DCNN-SVM?

1.3. Batasan Masalah

Dalam penelitian ini, penulis membatasi permasalahan sebagai berikut:

1. Data yang digunakan bersumber dari twitter dari tanggal 2 April 2019 sampai 20 April 2019.
2. Tanggapan berupa kalimat berbahasa Indonesia.
3. Data diambil dengan kata kunci ‘telkomsel’, ‘indosatcare’, ’myxl’, ’myxlcare’, ’triindonesia’.

1.4. Tujuan

Tujuan dari penelitian ini dibagi menjadi dua, yaitu:

1. Tujuan umum dari penelitian ini adalah mengklasifikasikan tanggapan pelanggan di twitter.
2. Tujuan khusus dari penelitian ini adalah :
    a. Mengembangkan perangkat lunak yang digunakan untuk analisis sentimen tanggapan pelanggan dari operator telekomunikasi di twitter menggunakan algoritma DCNN-SVM.
    b. Manganalisis berapa banyak tweet yang bersifat negatif dan positif di twitter operator telekomunikasi serta mendapatkan kata yang paling banyak muncul.
    c. Mendapatkan akurasi algoritma DCNN-SVM.

1.5. Manfaat

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Dalam bidang akademik dan penelitian, untuk mengetahui dan memahami metode *machine learning* dan *deep learning* pada NLP (*Natural Language Processing*). Serta untuk mengetahui analisis sentimen dari sekumpulan dataset posting dan tanggapan sosial media menggunakan algoritma DCNN-SVM serta mengetahui nilai akurasinya.
2. Bagi perusahaan, sebagai bentuk rekomendasi terhadap kualitas penyedia layanan telekomunikasi serta untuk penelitian awal yang memungkinkan untuk terdapat penelitian-penelitian selanjutnya dan dapat dikembangkan kedalam beberapa hal seperti bisnis maupun rancang bangun aplikasi yang memanfaatkan algoritma DCNN-SVM ini.

1.6. Sistematika Penulisan

Penelitian ini mempunyai susunan dalam penulisan agar tersusun secara sistematis dan memudahkan pembaca untuk mempelajarinya. Berikut sistematika penulisan penelitian ini:

1. BAB I PENDAHULUAN

Bab ini menjelaskan gambaran umum dari penulisan penelitian yang terdiri atas latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan penelitian.

2. BAB II TINJAUAN PUSTAKA

Pada bab ini dijelaskan beberapa teori dasar yang mendukung dalam pengerjaan penelitian ini yang meliputi penelitian terdahulu, landasan teori dari Analisis Sentimen, *Natural Language Processing, Deep Learning, Deep Convolutional Neural Network, Support Vector Machine,* dan Operator Telekomunikasi.

3. BAB III METODE PENELITIAN

Bab ini menjelaskan tentang tahapan-tahapan dan metode yang digunakan disertai penjelasan dalam tiap tahapan yang dilakukan dalam menyelesaikan penelitian.

4. BAB IV PERANCANGAN DAN IMPLEMENTASI

Pada bab ini menjelaskan tentang model dan desain dari sistem yang akan dibentuk. Hal-hal tersebut meliputi, *scraping* data, tahap pra-pemrosesan data, transformasi data dengan *Word2vec,* pembuatan model DCNN, dan SVM, visualisasi data sebagai acuan dalam mengimplementasikan sistem.

5. BAB V UJI COBA DAN EVALUASI SISTEM

Bab ini membahas tentang pengujian sistem yang telah terimplementasi dengan melakukan proses verifikasi dan validasi beserta pengujian kinerja dari sistem yang telah dibuat.

6. BAB VI KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dari penelitian yang diperoleh dari bab uji coba dan evaluasi serta saran untuk pengembangan penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

Pada bab ini diuraikan mengenai penelitian terdahulu dan landasan teori yang meliputi analisis sentimen, *machine learning*, *natural language processing*, *deep learning*, *word2vec*, *deep convolutional neural network*, *support vector machine*, dan operator telekomunikasi.

2.1 Penelitian Terdahulu

Penulisan penelitian ini merujuk pada beberapa penelitian sebelumnya. Salah satunya pada penelitian tentang pembandingan sentimen analisis pada twitter yang dilakukan oleh Abbasi dkk dengan lima bidang dan dua puluh metode mendapatkan nilai akurasi berforma baik antara 65% sampai 71% dan nilai akurasi berperforma rendah di bawah 50% [9]

Pada tahun 2018, Rozi dkk melakukan penelitian tentang "Opinion Mining On Book Review Using CNN-L2-SVM Algorithm" . Pada penelitian tersebut mengambil ulasan buku yang berbahasa inggris. Sedangkan pada penelitian ini penulis menggunakan data *scraping* dari twitter yang berbahasa indonesia. Performansi yang didapatkan dari model untuk menentukan sentimen ulasan sebesar 83.23% untuk data latih dan sebesar 64.4% untuk data uji [7].

Tahun 2018, Mukhlash dkk juga meneliti tentang "Opinion Mining On Book Review Using CNN-LSTM Algorithm". Pada penelitian tersebut mengambil ulasan buku yang berbahasa inggris. Sedangkan pada penelitian ini penulis menggunakan data *scraping* dari twitter yang berbahasa indonesia. Performansi yang didapatkan sebesar 99.55% untuk data latih dan sebesar 65.03% untuk data uji [8].

Afnandika juga melakukan Tugas Akhir tentang analisis sentimen pada sosial media menggunakan algoritma CNN dengan studi kasus operator telekomunikasi pada tahun 2018. Sedangkan pada penelitian ini, penulis menggunakan gabungan DCNN-SVM untuk mengklasifikasikan tanggapan operator telekomunikasi di twitter. Metode terbaik yang digunakan pada penelitian tersebut untuk fitur pemilihan yaitu *word2vec* dan *learning algorithm skipgram*. *Learning algorithm skipgram* memiliki nilai akurasi terbaik untuk semua subtask. Dengan metode CNN didapatkan hasil akurasi hingga 11.07% (tanpa memperhatikan jumlah label) dan 5.09% jika memperhatikan jumlah label [10].

2.2 Landasan Teori

Landasan teori berisi teori-teori yang digunakan dalam pengerjaan penelitian ini. Dalam landasan teori, acuan yang digunakan adalah berdasarkan penelitian lain dan buku.

2.2.1 Analisis Sentimen

Analisis sentimen adalah disiplin yang mengekstraksi perasaan, pendapat, pikiran, dan perilaku orang-orang dari data teks pengguna menggunakan metode *Natural Language Processing* (NLP) [11]. Selain itu, analisis sentimen juga dikenal sebagai *opinion mining*. Analisis sentimen dapat digunakan untuk menemukan pola opini dalam populasi seperti di mana orang lebih bahagia atau apa persepsi publik tentang suatu merek produk atau layanan baru. Ada beberapa metode dalam analisis sentimen, yaitu metode berbasis leksikon, metode berbasis *machine learning*, dan metode *Hybrid* [12]. Metode berbasis *machine learning* dibagi menjadi tiga yaitu *unsupervised learning*, *supervised learning*, dan *semi-supervised learning* [12]. Pada *supervised learning* terdapat

beberapa algoritma klasifikasi seperti SVM, *Naïve Bayes*, dan *Neural Network*.

2.2.2 *Machine Learning*

Machine learning adalah disiplin yang mempelajari dan mengembangkan algoritma untuk belajar, dan membuat prediksi data [11]. *Machine learning* berfokus pada prediksi berdasarkan yang diketahui properti data [11]. Tujuan *machine learning* adalah untuk menggeneralisasi pola yang terdeteksi atau membuat aturan yang tidak diketahui dari contoh yang diberikan [13]. Masalah yang sering terjadi dalam *machine learning* adalah bahwa set latih yang besar diperlukan untuk generalisasi yang baik, namun rangkaian pembelajaran yang besar juga lebih mahal secara komputasi. Beberapa metode paling populer bisa dikategorikan [11]:

1. *Supervised Learning* : dapat digunakan untuk memecahkan masalah seperti klasifikasi, dimana data dilengkapi dengan atribut tambahan yang ingin kami prediksi, misalnya label suatu kelas. Dalam hal ini, pengklasifikasiaan dapat mengaitkan setiap objek input dengan ouput yang diinginkan. Dengan menyimpulkan dari fitur dari objek input, pengklasifikasian kemudian dapat memprediksi label yang diinginkan untuk input baru yang tidak diketahui. Teknik umum termasuk *Naive Bayes*, *Support Vector Machine* dan model yang termasuk dalam keluarga *Neural Networks*, seperti *perceptrons* atau *multi-layer perceptrons*. Input sampel yang digunakan oleh algoritma pembelajaran untuk membangun model matematika disebut data latih, sedangkan input yang tidak terlihat yang ingin kami dapatkan prediksinya disebut data uji. Input dari

algoritma *machine learning* biasanya dalam bentuk vektor dengan masing-masing elemen vektor mewakili fitur input.

2. *Unsupervised Learning* : diterapkan pada masalah di mana data datang tanpa nilai output yang sesuai. Contoh khas dari masalah semacam ini adalah clustering atau pengelompokan. Dalam hal ini, suatu algoritma mencoba menemukan struktur tersembunyi dalam data untuk mengelompokkan item serupa ke dalam kelompok. Contoh dari algoritma pengelompokan yang umum adalah k-*means*.

2.2.3 *Natural Languange Processing*

Natural Language Processing (NLP) adalah disiplin yang terkait dengan studi metode dan teknik untuk analisis otomatis, pemahaman, dan generasi bahasa alami, yaitu bahasa yang ditulis atau diucapkan secara alami oleh manusia [11]. *Natural Language Processing* (NLP) adalah salah satu teknik dalam *Text Mining* [14]. Tantangan dalam *Natural Language Processing* sering kali melibatkan pengenalan suara harus jelas, mengharuskan manusia untuk dapat berbicara kepada komputer dalam bahasa pemrograman yang tepat, tidak ambigu dan sangat terstruktur.

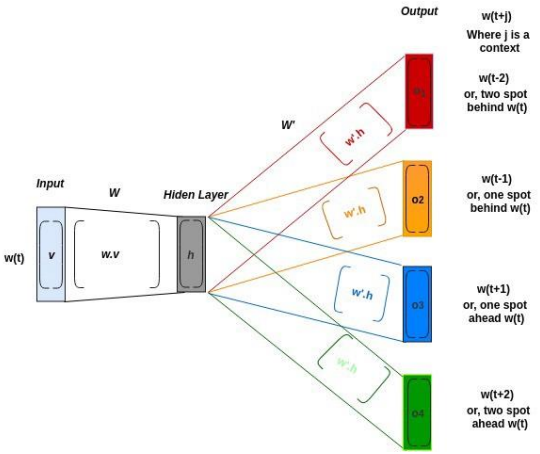
2.2.4 *Deep Learning*

Deep learning merupakan subbidang dari *machine learning* yang berkaitan dengan algoritma yang terinspirasi oleh struktur dan fungsi otak yang disebut jaringan saraf tiruan. *Deep learning* dapat dianggap sebagai cara untuk prediksi analisis secara otomatis. Algoritma *machine learning* tradisional bersifat linier, sedangkan algoritma *deep learning* ditumpuk dalam hierarki yang meningkatkan kompleksitas dan abtraksi.

2.2.5 *Word2vec*

Word2vec merupakan salah satu bentuk metode untuk mempresentasikan kata ke dalam nilai vektor tertentu yang dikembangkan oleh Google. Vektor tersebut dapat membantu algoritma *machine learning* untuk mencapai performansi lebih baik dalam NLP dengan mengelompokkan kata-kata yang mirip atau sama. *Word2vec* sendiri termasuk dalam kategori *neural network* yang menggunakan *hidden layer* dan beberapa *non-linier layer* didalam algoritmanya.

Salah satu arsitektur model dalam *Word2Vec* untuk mempelajari representasi kata-ata terdistribusi yang mencoba meminimalkan kerumitan komputasi adalah *Skip-gram* model. Arsitektur *Skip-gram* Model memprediksi kata saat ini berdasarkan konteksnya, ia mencoba memaksimalkan klasifikasi kata berdasarkan kata lain dalam kalimat yang sama. Dapat digunakan setiap kata saat ini untuk dapat memprediksi kata-kata dalam rentang tertentu sebelum dan sesudah kata sekarang. Model *Skip-gram* dapat dilihat pada Gambar 2.1.



Gambar 2. 1 *Skip-gram*

Tujuan pembelajaran dari model *Skip-gram* adalah untuk menemukan representasi kata yang berguna untuk memprediksikan kata-kata di sekitarnya dalam kalimat atau dokumen. Lebih formal lagi, diberi urutan pembelajaran kata $w_1, w_2, \dots, w_T$, tujuan dari model *Skip-gram* adalah untuk memaksimalkan rata-rata peluang [15].

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log p(w_{t+j}|w_t) \tag{2.1}$$

dengan

c : ukuran dari *window*
w : kata
T : banyak pembelajaran kata
$p(w_{t+j}|w_t)$ = peluang besyarat dari keakurasian kata w_{t+j} bila kata w_t telah terjadi.

Formula *skip-gram* dasar dari $p(w_{t+j}|w_t)$ menggunakan fungsi *softmax* dapat ditunjukkan oleh persamaan 2.2.

$$p(w_{t+j}|w_t) = \frac{\exp\left({v'_{w_{t+j}}}^{T} v_{w_t}\right)}{\sum_{w=1}^{n}\exp({v'_w}^{T} v_{w_t})} \tag{2.2}$$

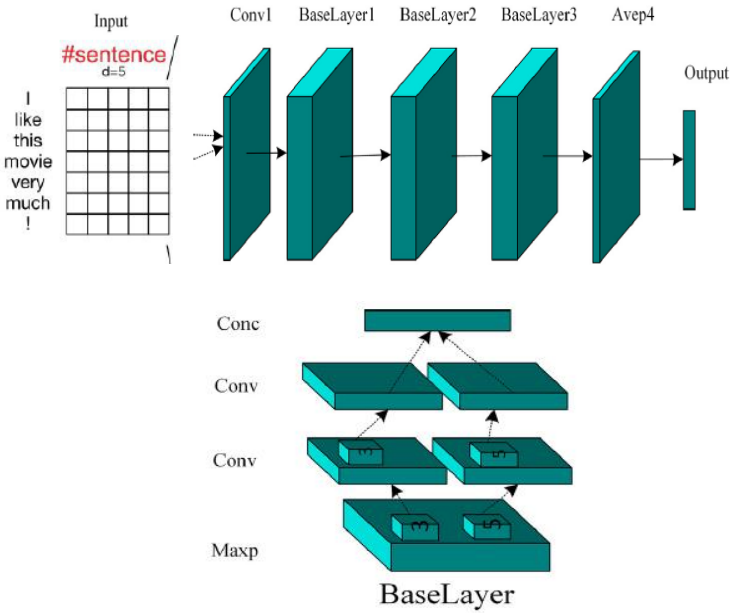
dengan

v_w : input representasi vektor dari *w*
v'_w : output representasi vektor dari *w*
n : jumlah kata dalam *vocabulary*

2.2.6 *Deep Convolution Neural Network*

Deep Convolutional Neural Network (DCNN) pengembangan dari *Convolutional Neural Network* (CNN) sedangkan CNN sendiri, menurut Wayan Suartika adalah pengembangan dari Multilayer Perceptron (MLP) yang

didesain untuk mengolah data dua dimensi [16]. Perbedaan DCNN dan CNN disini adalah jumlah layer [17]. Jadi DCNN merupakan CNN dengan jumlah layer yang lebih banyak [17]. Pada DCNN ini digunakan 12 layer sedangkan pada CNN biasanya hanya 8 layer. CNN pertama kali digunakan pada tahun 1989 oleh Yann LeCun untuk melakukan klasifikasi citra kode zip [16]. CNN pada umumnya digunakan pada klasifikiasi dua dimensi, atau gambar namun CNN juga dapat digunakan klasifikasi teks [18]. Arsitektur DCNN dapat ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Arsitekteur DCNN[7]

1. *Convolution Layer*

Konvolusi adalah lapisan pertama dalam arsitektur jaringan DCCN-SVM yang mengekstraksi fitur data input. Konvolusi menjaga hubungan antara piksel dengan mempelajari fitur data menggunakan kotak kecil data input. Ini adalah operasi matematika yang mengambil dua input seperti matriks gambar dan filter atau kernel.

Secara umum hasil gabungan vektor-vektor kata dari indeks ke- i sampai ke$-i + j$ pada persamaan 2.3.

$$x_i, \quad x_{i+1}, x_{i+2}, \quad \dots, \quad x_{i+j} \tag{2.3}$$

dengan

x_i : vektor kata yang berada pada $\mathbb{R}^{50}$

Untuk mencari sebuah filter untuk menghasilkan suatu nilai fitur maka dapat dirumuskan pada persamaan 2.4.

$$c_i = f(net) \tag{2.4}$$

dengan

c_i : nilai *feature map* pada indeks ke-i
h : ukuran *window* kata
w : filter yang berada pada $\mathbb{R}^{50}$
b : parameter bias

Pada penelitian ini fungsi non linear yang digunakan merupakan fungsi *Rectified Linear Unit* (*ReLu*). Keluaran dari fungsi tersebut memberikan batasan keluaran bernilai positif. Fungsi tersebut dapat dituliskan pada persamaan 2.5.

$$ReLu(x) = \max\{0, x\} \tag{2.5}$$

Sehingga persamaan menjadi seperti berikut:

$$c_i = ReLu(net) \quad (2.6)$$

dengan

$$net = w.x_{i:i+h-1} + b \quad (2.7)$$

Filter *w* di terapkan untuk tiap *window* kata yang mungkin di dalam kalimat *tweet* $\{x_{1:h}, x_{2:h}, \dots, x_{n-h+1:h}\}$ sehingga dihasilkan sebuah *feature map* pada persamaan 2.8.

$$c = [c_1, c_2, \dots, \quad c_{n-h+1}] \quad (2.8)$$

dengan *c* merupakan *feature map.*

2. *Pooling Layer*

Layer selanjutnya adalah *pooling layer*. Fungsi dari *pooling layer* adalah sebagai *down_sampling* yang *non-liner*. Peran algoritma *pooling* antara lain:

a. Dengan mengeliminasi nilai yang tak maksimal, mengurangi komputasi dari layer yang diatasnya.
b. Menyediakan sebuah bentuk dari translasi invarian.

Setelah didapatkan *feature map* yang berasal dari *convolution layer*, maka masing-masing dari *feature map* tersebut dilakukan operasi *pooling* pada *pooling layer*. Pada lapisan ini diambil nilai-nilai yang penting dari feature map dengan mengambil nilai yang paling maksimum di tiap *feature map* [7]. Secara matematis, operasi *pooling* dapat dirumuskan sebagaiberikut:

$$\hat{c} = \max\{c\} \quad (2.9)$$

dengan

$\hat{c}$: nilai maksimum dari feature map c

Dikarenakan terdapat sebanyak m filter, maka hasil dari *pooling layer* merupakan suatu vektor yang terdiri atas nilai maksimum tiap *feature map* dan berjumlah m elemen [9]. Sehingga dapat diperoleh:

$$z = [\hat{c}_1, \hat{c}_2, \hat{c}_3, \dots, \hat{c}_m) \quad (2.10)$$

z merupakan vektor hasil dari *pooling layer* yang akan masuk kelapisan berikutnya.

2.2.7 Support Vector Machine

Support Vector Machine (SVM) secara alamiah didefinisikan untuk klasifikasi biner data numeric. SVM merupakan salah satu metode dalam klasifikasi yang dapat menganalisis data atau mengenali pola. *Support Vector Machine* (SVM) adalah sistem pembelajaran yang pengklasifikasiannya menggunakan ruang hipotesis berupa fungsi-fungsi linear dalam sebuah ruang fitur (*feature space*) berdimensi tinggi, dilatih dengan algoritma pembelajaran yang didasarkan pada teori optimasi dengan mengimplementasikan *learning bias* yang berasal dari teori pembelajaran statistik [19]. Karena SVM merupakan pengklasifikasian maka diperlukan suatu himpunan data latih. Pada proses latihan, SVM membangun sebuah model yang memprediksi apakah data yang diinputkan termasuk dalam jenis kategori data yang terdapat pada SVM tersebut. SVM dapat diaplikasikan pada bidang numerik, termasuk *handwriting digit recognition, object recognition* dan *speaker identification.* SVM bekerja dengan cara mengolah data teks menjadi vektor.

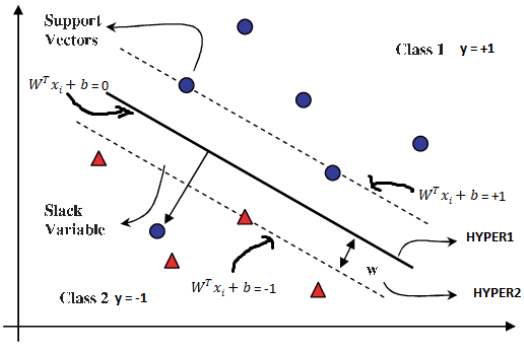
Secara konseptual, proses dalam SVM berusaha menemukan fungsi pemisah (*hyperplane*) terbaik diantara fungsi yang tidak terbatas jumlahnya. *Hyperplane* terbaik antara kedua kelas dapat ditemukan dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya. Adapun data yang berada pada bidang pembatas disebut *support vector*. Arsitektur SVM dapat ditunjukkan pada Gambar 2.3.

Untuk mendapatkan titik maksimal dapat ditunjukkan pada persamaan 2.11.

$$\min \frac{1}{2}|w|^2 \tag{2.11}$$

dengan

w : parameter bobot dari SVM



Gambar 2. 3 SVM

Untuk mendapatkan nilai kelas dapat ditunjukkan pada persamaan 2.12.

$$f(x_i) = W^T x_i + b \tag{2.12}$$

dengan

x : vektor hasil dari pooling layer

W : parameter bobot dari SVM

b : parameter bias dari SVM

Misal y merupakan kelas label dari *tweet*. Jika $f(x_i) \geq 0$, maka nilai y = +1 dan jika $f(x_i) < 0$, maka nilai $y = -1$. Berdasarkan hal tersebut, maka dapat dituliskan sebagai berikut:

$$f(x_i) \geq 0, \quad y = +1 \tag{2.13}$$

$$f(x_i) < 0, \quad y = -1 \tag{2.14}$$

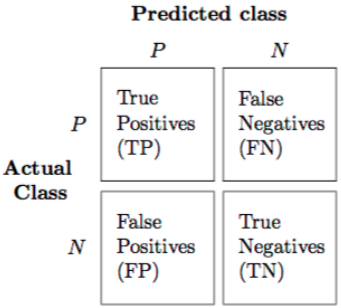
Keterangan:

$f(x_i)$: nilai kelas

y : label *tweet*

2.2.8 Evaluasi Performasi

Evaluasi performasi algoritma klasifikasi *machine learning* menggunakan acuan *Confusion Matrix*. *Confusion Matrix* merepresentasikan prediksi dan kondisi sebenarnya dari data algoritma klasifikasi *machine learning*. Kita bisa menentukan akurasi, presisi dan *recall* berdasarkan *confusion matrix*.



Gambar 2. 4 *Confusion Matrix*

Berdasarkan Gambar 2.4 hasil dari proses klasifikasi data dapat dikategorikan kedalam empat jenis data yaitu :

- TP adalah *True Positive*, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.
- TN adalah *True Negative*, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.
- FN adalah *False Negative*, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.
- FP adalah *False Positive*, yaitu jumlah data positif namun terklasifikasi salah oleh sistem.

Berikut merupakan penjelasan dari masing-masing performansi:

1. Akurasi

Akurasi didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai sebenarnya atau pada penelitian ini adalah label dari hasil pelabelan manual. Akurasi merupakan rasio prediksi benar dengan keseluruhan data. Secara umum perhitungan akurasi dapat dirumuskan sebagai berikut:

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Presisi

Presisi adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Secara umum perhitungan presisi dapat dirumuskan sebagai berikut:

$$Presisi = \frac{TP}{TP + FP}$$

3. *Recall*

Recall adalah tingkat keberhasilan sistem dalam menentukan kembali sebuah informasi. Secara umum perhitungan *recall* dapat dirumuskan sebagai berikut:

$$recall = \frac{TP}{TP + FN}$$

2.3. Profil Operator Telekomunikasi

Operator telekomunikasi adalah perusahaan yang menawarkan layanan telekomunikasi meliputi sms, telepon dan akses internet kepada penggunanya. Teknologi yang digunakan oleh perusahaan ini dibagi menjadi dua jenis yaitu *global system mobile communication* (GSM) dan *code division multipple acces* (CDMA). Jumlah operator telekomunikasi sendiri di indonesia saat ini ada empat yaitu Telkomsel, Indosat Ooredoo (IM3), Hutchison (3) dan XL Axiata (XL dan Axis).

BAB III
METODOLOGI PENELITIAN

3.1 Objek dan Aspek Penelitian

Objek yang digunakan dalam penelitian ini adalah data *scraping* dari twitter dengan *keyword* pada batasan masalah yang ada pada Bab 1. Sedangkan aspek penelitiannya adalah mengklasifikasikan tanggapan pelanggan operator telekomunikasi di twitter dengan algoritma DCNN-SVM.

3.2 Peralatan Penunjang

Penelitian ini menggunakan perangkat keras dan perangkat lunak untuk menunjang proses pengerjaan. Untuk spesifikasi perangkat keras dan perangkat lunak yang digunakan dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Spesifikasi Perangkat

Nama Perangkat	Laptop
Processor	Intel® Core™ i3-4005U Processor @ 1.70GHz
Memory	2 GB DDR-3
Sistem Operasi	Windows 10
Arsitektur Sistem	64-bit *Operating System, x64-based processor*

Sedangkan aplikasi dikembangkan dengan menggunakan beberapa teknologi seperti *code editor, database*, bahasa pemrograman, dan *library* yang disajikan dalam Tabel 3.2.

Tabel 3. 2 Daftar *Library*

Bahasa Pemrograman	Python
Code Editor (IDE)	Jupyter Notebook, Google Colab

Virtual Environment	Anaconda
Library	• Tweepy • NLTK • Gensim • Sklearn • Keras • Sckit-learn • Pandas • Numpy • Matplotlib • CSV

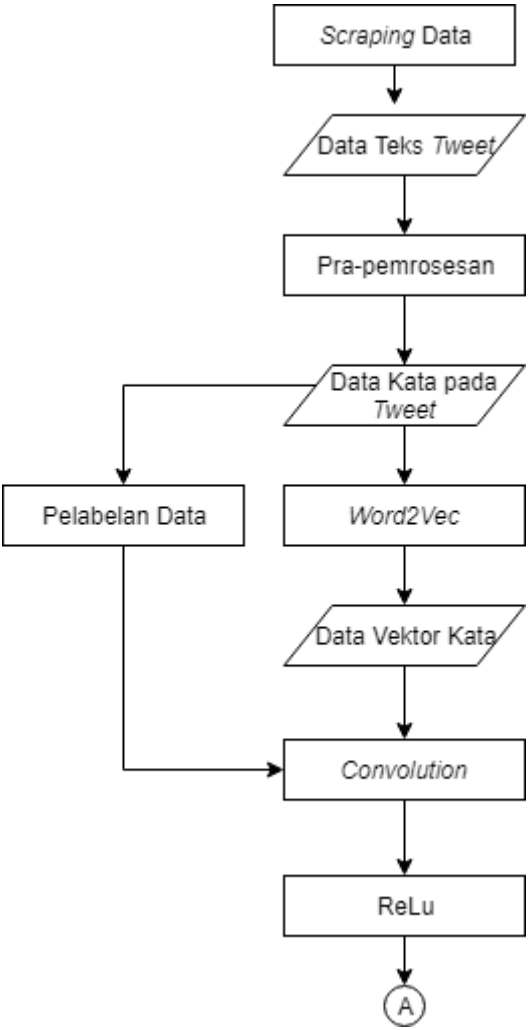
3.3 Data Penelitian

Data yang digunakan dalam sistem klasifikasi tanggapan pelanggan operator telekomunikasi di twitter menggunakan algoritma DCNN-SVM, yaitu:

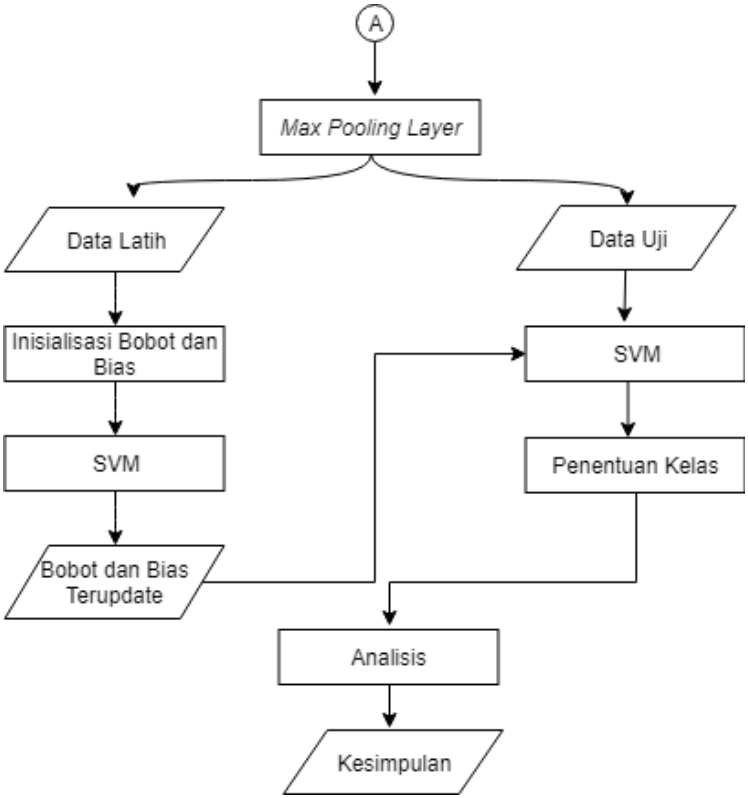
1. Data masukan, yaitu data s*craping* dari twitter dengan *keyword* pada Bab 1.
2. Data proses, yaitu data ketika tahap-tahap pemrosesan data yang sedang dilakukan
3. Data keluaran, yaitu hasil klasifikasi data *tweet* berupa *tweet* negatif atau *tweet* positif.

3.4 Tahapan Pelaksanaan Penelitian

Pada bagian ini akan dijelaskan mengenai metodologi atau tahapan pelaksanaan penelitian yang dapat ditunjukkan pada Gambar 3.1



Gambar 3. 1 Diagram Alir Metodologi Penelitian



Gambar 3. 1 Diagram Alir Metodologi Penelitian (lanjutan)

Keterangan:

= proses

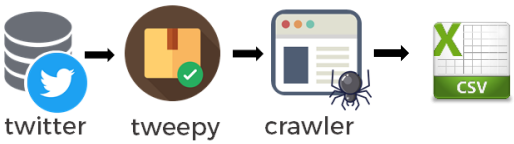
= output

3.5 *Scraping* Data

Scraping adalah teknik dimana program komputer mengekstraksi data dari output yang dapat dibaca manusia yang berasal dari program lain. Pada tahap ini akan dilakukan

ekstraksi data dari twitter menggunakan *Application Programming Interface* (API) twitter yaitu *Search* API.

Pada tahap ini akan dilakukan pengambilan data di twitter dengan kata kunci yang ada pada batasan masalah diatas. *Tweet* yang akan diambil adalah produk dari operator telekomunikasi yaitu kartu prabayar. Kata kunci tersebut akan mengambil semua *tweet* yang mengandung kata kunci tersebut, yaitu seperti *reply*, *mention*, *thread*, *hashtag* atau *retweet* maka dari itu akan dibatasi dan hanya akan mengambil *tweet* pertama. Selanjutnya data ini akan disimpan per hari pada CSV. Proses ini dapat ditunjukkan pada Gambar 3.2.



Gambar 3. 2 Proses *Scraping* Twitter

3.6 Pra-pemrosesan Data

Data hasil *scraping* tersebut masih mentah untuk diproses dengan *machine learning* dan akan diolah pada tahap ini.

Stopwords dapat diartikan sebagai proses untuk menghilangkan karakter, tanda baca, serta kata-kata umum yang tidak memiliki makna atau informasi yang dibutuhkan. *Stopwords* umumnya digunakan dalam pengambilan informasi salah satu contohnya adalah mesin pencari Google. Pengurangan ukuran indeks dalam teks dengan penghilangan beberapa kata kerja, kata sifat, dan kata keterangan lainnya dapat dimasukkan ke dalam daftar *stopwords*. Contoh *stopwords* : ‘dan’, ’atau’ dan sebagainya.

Tokenization adalah pemotongan urutan karakter dan sebuah set dokumen yang diberikan menjadi potongan-

potongan kata atau karakter yang sesuai dengan kebutuhan sistem. Potongan-potongan tersebut dikenal dengan istilah token. Jadi pada *tokenization* ini akan memotong *tweet* menjadi per kata.

Pada tahap ini juga dilakukan pembersihan *tweet* dari URL(*Uniform Resource Locator*), emotikon dan tanda baca serta mengubah semua *tweet* mejadi huruf kecil.

3.7 *Labeling*

Pada tahap *labeling* ini akan menggunakan metode HIT (*Human Intelligence Task)* dimana proses ini harus memiliki tingkat persetujuan lebih besar dari 95% [20]. Setiap HIT akan dilakukan oleh 3 orang. Masing-masing anotator akan memberikan label pada *tweet* dengan tujuan mempertimbangkan asumsi masing-masing anotator. Untuk mendapatkan label akhir dari masing-masing *tweet* apabila dua orang atau lebih memilih label yang sama, maka label akhirnya adalah berdasarkan pemilihan tersebut.

3.8 *Word2vec*

Tahap ini data akan dilakukan ekstraksi fitur. Ekstraksi fitur ini untuk mendapatkan ciri dari suatu data atau kata. Metode ini bertugas untuk merubah sebuah kata menjadi suatu vektor. Vektor yang dihasilkan merupakan konteks dari kata tersebut yang memperhatikan peluang kata yang muncul disekitarnya.

3.9 Implementasi algoritma DCNN-SVM

Selanjutnya output dari *word2vec* akan berupa matriks yang berisi kumpulan vektor-vektor kata dalam satu kalimat. Selanjutnya proses *convolution*, pada tahap ini akan dilakukan operasi *cross product* antara input *layer* dan *kernel / filter*.

Operasi tersebut akan dilakukan secara terus menerus hingga semua nilai pada dimensi vektor terkalkulasi. Selanjutnya adalah proses ReLu, yang memiliki tujuan untuk merubah vektor yang bernilai negatif menjadi 0. *Max Pooling*, membagi output vektor *convolutional layer* ke beberapa *grid* vektor dan mengambil nilai maksimal dari setiap *grid*. Tahap selanjutnya yaitu *Average Pooling*, yaitu menghubungkan lapisan terakhir dari lapisan sebelumnya di setiap kernel menjadi dua dimensi, sehingga dapat melakukan transformasi pada dimensi agar dapat diklasifikasikan secara linear. Pada tahap ini akan diklasifikasi, biasanya DCNN menggunakan fungsi aktivasi *softmax* untuk pengklasifikasiannya, tapi pada penelitian ini fungsi aktivasi *softmax* akan diganti dengan SVM. Jadi Algoritma SVM ini digunakan pada *layer* terakhir pada model jaringan DCNN sehingga sebelum masuk ke dalam jaringan SVM inputan vektor harus melalui proses dalam DCNN. Sebelum proses SVM dilakukan, data akan dibagi dua, yaitu data latih dan data uji. Data latih adalah data yang sudah ada sebelumnya berdasarkan fakta yang sudah terjadi yang digunakan untuk membuat model klasifikasi *machine learning*. Sedangkan data uji adalah data yang sudah belabel/berkelas yang digunakan untuk menghitung akurasi model klasifikasi *machine learning* yang dibentuk.

3.10 Analisis Hasil dan Penarikan Kesimpulan

Pada tahap ini akan dilakukan analisis terhadap hasil akhir pengolahan data yang telah dimasukkan ke dalam program. Dalam analisis hasil ini akan didapatkan hasil yaitu tanggapan pelanggan untuk operator telekomunikasi yang telah diklasifikasi. Dari data yang telah diklasifikasi juga didapatkan data olahan yaitu *tweet* yang telah diberi sentimen positif dan

negatif serta diketahui hasil akurasi dari *machine learning* yang diolah oleh DCNN-SVM. Data klasifikasi ini akan dibuat *word cloud* untuk mendapatkan kata yang sering muncul. *Word Cloud* adalah representasi visual dari data teks, biasanya digunakan untuk menggambarkan metadata kata kunci pada sebuah *website* atau situs, untuk memvisualisasikan suatu bentuk teks secara bebas.

BAB IV
PERANCANGAN DAN IMPLEMENTASI

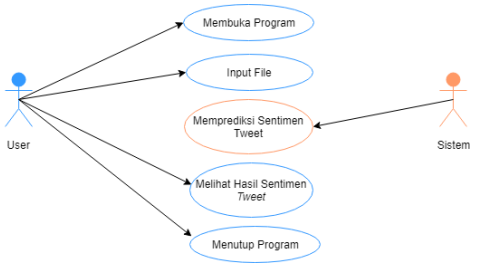
Bab ini akan menjelaskan mengenai rancangan dari desain sistem yang digunakan sebagai acuan untuk implementasi sistem. Perancangan yang akan dibuat berupa rancangan *scraping* data, pra-pemrosesan data, dan pembuatan model DCNN-SVM. Penjelasan tahapan implementasi adalah dengan menggunakan data *dummy* untuk memudahkan interpretasi.

4.1 Analisis Implementasi Sistem

Pada proses implementasi suatu sistem diperlukan suatu analisis terhadap sistem agar sistem bisa bekerja secara optimal. Program yang dibuat berupa perangkat lunak yang digambarkan dalam *use case diagram* dan *swimlane diagram*.

4.1.1 *Use Case Diagram*

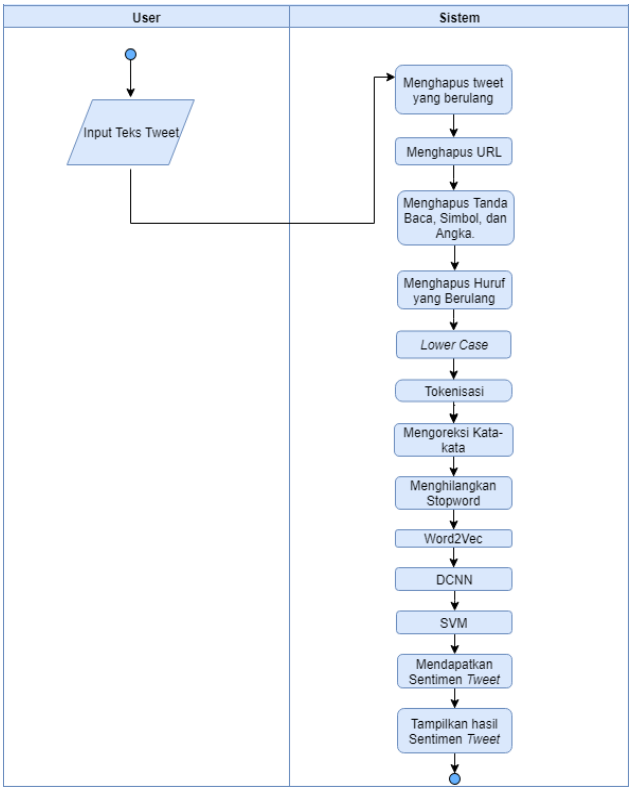
Use Case Diagram adalah gambaran *graphical* dari beberapa atau semua *user*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara rinci tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *usecase*, *user*, dan sistem. Gambar 4.1 merupakan *use case diagram* dari penelitian ini.



Gambar 4. 1 *Use Case Diagram*

4.1.2 *Swimlane Diagram*

Swimlane diagram adalah jenis diagram alur yang menggambarkan siapa melakukan apa dalam suatu proses. Dengan menggunakan metafora lajur dalam kumpulan, *swimlane diagram* memberikan kejelasan dan akuntabilitas dengan menempatkan langkah-langkah proses dalam "swimlanes" horisontal atau vertikal dari karyawan, kelompok kerja atau departemen tertentu. Gambar 4.2 merupakan *swimlane diagram* dari penelitian ini.



Gambar 4. 2 *Swimlane Diagram*

4.3 *Scraping* Data

Tahap awal pada perancangan dimulai dengan mengumpulkan seluruh data yang dibutuhkan dari media sosial twitter untuk tahap pengolahan data selanjutnya. Pengambilan data dilakukan melalui *crawler* yang dirancang untuk mengumpulkan data dari media sosial untuk proses selanjutnya yang akan disimpan ke dalam format *file* CSV.

Dilihat dari sisi waktu pengambilan data aturan yang digunakan dari twitter API hanya memperbolehkan mengakses *posting* twitter hingga satu minggu sebelum hari pengambilan data. Jadi, semisal mengambil data yang lebih dari satu minggu sebelumnya maka Twitter API akan mengembalikan nilai null. Namun, pada *library* Tweepy data akan diambil dari data paling baru hingga satu minggu sebelumnya sesuai dengan ketersedian.

Untuk mendapatkan akses untuk mengambil data dari twitter perlu 4 kode rahasia yaitu : *customer key*, *customer secret*, *token acces*, dan *token acces secret* yang didapatkan dari situs dev.twitter.com dengan mendaftar sebagai *developer*.

Berikut merupakan kode yang didapatkan dari dev.twitter.com.

customer key : yalIxYhkLaDyqnkNU3fjKcNgE
customer secret :
iNZGwv4F6MU5OEaW4x4retXus04zZ4TJswmN17vAxfOnyVzjVZ
token acces : 1016593462953558016-
JrnZ1eTAaRc86xW7XMiy8daGsOyAHc
token acces secret :
3QwfKp1TIO99tsw95hZFNY5rSftpZsCui5TTP7MaJhML2

```
auth = tweepy.auth.OAuthHandler(customer key, customer
secret) #autentikasi twitter
auth.set_access_token(token acces, token acces secret)
api = tweepy.API(auth)

csvFile = open('xl21.csv', 'a') #membuat file
csvWriter = csv.writer(csvFile)
for tweet in tweepy.Cursor(api.search, q = "@myxlcare
until:2019-3-21 ", tweet_mode='extended').items():
    print (tweet.full_text)
    csvWriter.writerow([tweet.full_text.encode('utf-
8')]) #memasukkan tweet ke dalam csv
csvFile.close()
```

Kode 4. 1 *Scraping* Data

Kode 4.1 menunjukkan *script* yang digunakan untuk melakukan pengambilan data twitter sesuai dengan *keyword* pencarian yang ada. Pada implementasi program ini akan dilakukan pencarian semua *tweet* yang mengandung *keyword*. Kemudian pada proses ini juga menghasilkan sebuah *file* CSV Twitter yang disimpan untuk keperluan berikutnya. Proses ini akan berulang hingga hasil yang didapatkan dari pengambilan data twitter tidak menemukan *tweet* lagi. Proses ini menggunakan *search API twitter*. Kode diatas juga menjelaskan proses kode sebuah instance yang menghubungkan *client* dengan twitter API menggunakan *API_Key* dan *API_Secret* yang telah didapatkan dari *website* Twitter API.

Contoh kutipan *posting* pelanggan yang diambil dari akun twitter Telkomsel adalah sebagai berikut:

```
Halo @Telkomsel ini sinyal 4G strip 1-2, tapi
kok koneksi internet lama dan sering tidak bisa
digunakan ya?
Kamu @Telkomsel kenapa sih, pasca lebaran kacrut
banget deh jaringannya.... 😒😒😒😒
Pagi @Telkomsel. Kemarin teman saya kehilangan
hp (beserta nomornya). Apabila ingin menggunakan
nomor lama, apa saja yg harus dibawa ke Grapari?
Cukup KK + KTP asli? Sudah telfon CS untuk
diblokir nomornya.
```

Adapun data *dummy* yang berjumlah empat *tweet* atau empat tanggapan dari masing-masing operator telekomunikasi untuk mempermudah dalam interpretasi. Tabel 4.1 merupakan contoh data *dummy*:

Tabel 4. 1 Data *Dummy*

untung sinyal XL lancar di tempat kkn 😁, terima kasih @MyXL
sinyal tri kalo malem cepat sekaleee
TELKOMSEL KENAPA LAGI SIH? LEMOT BANGET DIBUAT INTERNETAN. UDA MAHAL, LEMOT @telkomsel
sinyal indosat jelek amat, tlng perbaiki dong @indosatcare

4.3 Pra-pemrosesan Data

Setelah mendapatkan *tweet* yang masih mentah, maka harus diproses terlebih dahulu agar lebih mudah untuk melakukan klasifikasi sentimennya. Beberapa proses awal sebelum mengklasifikasikan sentimen atau yang biasa di sebut dengan *Normalization Text*. Tahapan tersebut adalah sebagai berikut.

4.3.1 *Load* Data *Tweet* ke Modul Python

Data *tweet* yang disimpan dalam bentuk CSV di-*load* dalam python dan ditampilkan dalam bentuk *list*.

```
with open ('scraping.csv','r') as f:
  text=f.readlines()
print(text)
```

Kode 4. 2 *Load* Data *Tweet*

Kode 4.2 untuk menge*load* data *tweet* dari CSV, Kode 4.2 bisa membaca semua *tweet* dalam CSV, sementara jika menggunakan DataFrame, *tweet* setelah koma tidak terbaca.

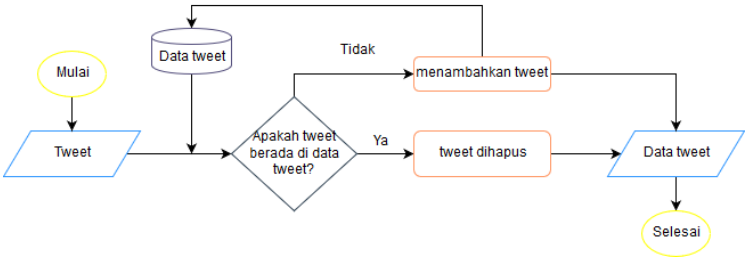
Contoh kutipan *tweet* pelanggan yang diambil dari twitter salah satu akun operator telekomunikasi yaitu Indosat yang di-*load* ke python pada Gambar 4.3.

```
['untung sinyal XL lancar di tempat kkn ??, terima kasih
@MyXL\n', 'sinyal tri kalo malem cepat sekale\n',
'TELKOMSEL KENAPA LAGI SIH? LEMOT BANGET DIBUAT
INTERNETAN. UDA MAHAL, LEMOT @telkomsel\n', 'sinyal
indosat jelek amat, tlng perbaiki dong @indosatcare\n']
```

Gambar 4. 3 Tampilan *Load* Data ke Python

4.3.2 Menghapus *Tweet* yang Berulang

Data yang didapatkan dari *scraping* masih terdapat data yang sama, oleh karena itu harus dihapus salah satunya hingga setiap data yang ada merupakan data yang unik. Proses ini dapat ditunjukkan pada Gambar 4.4.



Gambar 4. 4 Diagram Alir Menghapus *Tweet* Berulang

```
Data_Mentah=set(data_scraping)
Print(Data_Mentah)
```

Kode 4. 3 Menghapus *Tweet* yang Berulang

Kode 4.3 merupakan potongan *syntax* untuk menghapus salah satu dari data yang sama dengan fungsi set.

4.3.3 Menghapus *URL*

Data *tweet* yang sudah ada mengandung banyak token dan karakter asing dan tidak perlu, yang harus dihapus sebelum melakukan proses lebih lanjut seperti tokenisasi atau teknik normalisasi. Ini termasuk mengekstraksi teks yang bermakna dari sumber data seperti data HTML, yang terdiri dari tag HTML yang tidak perlu, atau bahkan data dari umpan XML dan JSON seperti emotikon. Dalam kasus ini menggunakan *beautiful soup* hanya untuk mengambil bentuk teks dari *tweet* yang telah di miliki.

```
pat1 = r'@[A-Za-z0-9_]+'#menghapus mention
pat2 = r'https?://[A-Za-z0-9./]+' #menghapus url
pat3 = r'#[A-Za-z0-9]+' #menghapus hastag
pat4 = r'\[A-Za-z0-9]+' #menghapus format maps yang eror
combined_pat = r'|'.join((pat1, pat2, pat3,pat4)) #pembuatan
kombinasi filter
def url_clean(a): #fungsi menghapus url
    soup = BeautifulSoup(a, 'lxml')
    souped = soup.get_text()
    stripped = re.sub(combined_pat, '', souped)
    try:
        clean = stripped.decode("utf-8-sig").replace
(u"\ufffd", "?")
    except:
        clean = stripped
    return (" ".join(clean)).strip()
```

Kode 4. 4 Menghapus URL

Berikut merupakan hasil dari Kode 4.4 yang menggunakan data dummy pada Tabel 4.2 berikut

Tabel 4. 2 Hasil Menghapus URL

untung sinyal XL lancar di tempat kkn , terima kasih
sinyal tri kalo malem cepat sekaleee
TELKOMSEL KENAPA LAGI SIH? LEMOT BANGET DIBUAT INTERNETAN. UDA MAHAL, LEMOT
sinyal indosat jelek amat, tlng perbaiki dong

4.3.4 Penghapusan Tanda Baca, Simbol, dan Angka

Salah satu hal penting dalam normalisasi teks adalah menghilangkan karakter yang tidak perlu dan karakter khusus/spesial. Termasuk simbol khusus atau bahkan tanda baca yang muncul dalam kalimat. Alasan utama untuk melakukannya adalah karena sering tanda baca atau karakter khusus tidak memiliki banyak arti ketika kita menganalisis teks dan menggunakannya untuk mengekstraksi fitur atau informasi.

```
def tweet_cleaner(a):
  b = re.sub("[^a-zA-Z]", " ", a)
  return b
```

Kode 4. 5 Menghapus Tanda Baca, Simbol, dan Angka

Kode 4.5 menghapus tanda baca, simbol dan angka atau dengan kata lain hanya mengambil kata saja.

Berikut merupakan hasil dari Kode 4.5 yang menggunakan data *dummy* pada Tabel 4.3.

Tabel 4. 3 Hasil Menghapus Tanda Baca, Simbol, dan Angka

untung sinyal XL lancar di tempat kkn terima kasih
sinyal tri kalo malem cepat sekaleee

TELKOMSEL KENAPA LAGI SIH LEMOT BANGET DIBUAT INTERNETAN UDA MAHAL LEMOT
sinyal indosat jelek amat tlng perbaiki dong

4.3.5 Menghapus Huruf yang Berulang

```
def word_double(a):
  c = re.sub(r'(.)\1{2,}', r'\1\1',a)
  return c
```

Kode 4. 6 Menghapus Huruf yang Berulang

Kode 4.6 menjelaskan proses penghapusan huruf yang berulang. Pada proses ini dibatasi untuk setiap huruf hanya boleh berulang dua kali.

Berikut merupakan hasil dari Kode 4.6 yang menggunakan data *dummy* pada Tabel 4.4 berikut

Tabel 4. 4 Hasil Menghapus Huruf yang Berulang

untung sinyal XL lancar di tempat kkn terima kasih
sinyal tri kalo malem cepat sekalee
TELKOMSEL KENAPA LAGI SIH LEMOT BANGET DIBUAT INTERNETAN UDA MAHAL LEMOT
sinyal indosat jelek amat tlng perbaiki dong

4.3.6 *Lowercase*

Dalam pra-pemrosesan data seringkali memodifikasi huruf atau kalimat untuk mempermudah mencocokkan kata atau token tertentu. Biasanya ada dua jenis operasi konversi kasus yang banyak digunakan. Ini adalah konversi huruf kecil dan huruf besar, di mana kata dikonversi sepenuhnya menjadi huruf kecil atau huruf besar. Pada proses ini yang di gunakan adalah perubahan data teks *tweet* menjadi huruf kecil semua.

```
def lowercase(string):
        return string.lower()
```

Kode 4. 7 Mengubah Kata Menjadi Huruf Kecil

Berikut merupakan hasil dari Kode 4.7 yang menggunakan data *dummy* pada Tabel 4.5.

Tabel 4. 5 Hasil Mengubah Huruf Menjadi *Lowercase*

untung sinyal xl lancar di tempat kkn terima kasih
sinyal tri kalo malem cepat sekalee
telkomsel kenapa lagi sih lemot banget dibuat internetan uda mahal lemot
sinyal indosat jelek amat tlng perbaiki dong

4.3.7 Tokenisasi

Pada tahap tokenisasi dilakukan suatu proses pemecahan tiap kalimat *tweet* menjadi bentuk kata-kata yang terpisah satu sama lain atau disebut juga dengan token. Tokenisasi kata sangat penting dalam banyak proses, terutama dalam membersihkan dan menormalkan teks di mana operasi seperti *stemming* dan *lemmatization* bekerja pada setiap kata berdasarkan pada masing-masing kata.

```
from nltk.tokenize import word_tokenize
def token(test_result):
  tokenized_sents = [word_tokenize(i) for i in test_result]
  for i in tokenized_sents:
      print (i)
  return i
```

Kode 4. 8 Tokenisasi

Berikut merupakan hasil dari Kode 4.8 yang menggunakan data *dummy* pada Tabel 4.6.

Tabel 4. 6 Hasil Tokenisasi

['untung', 'sinyal', 'xl, 'lancar', 'di', 'tempat', 'kkn', 'terima', 'kasih']
['sinyal', 'tri', 'kalo', 'malem', 'cepat', 'sekalee']
['telkomsel', 'kenapa', 'lagi', 'sih', 'lemot', 'banget', 'dibuat', 'internetan', 'uda', 'mahal', 'lemot']
['sinyal', 'indosat', 'jelek', 'amat', 'tlng', 'perbaiki', 'dong']

4.3.8 Mengoreksi Kata

Data yang diambil adalah data *tweet* yang merupakan bahasa yang bukan bahasa baku, proses mengkoreksi kata-kata sangatlah penting. Perbedaan satu huruf saja dalam suatu kata program yang dibuat bisa menganggap kata tersebut merupakan kata yang berbeda. Kata-kata yang salah mencakup kata-kata yang memiliki kesalahan ejaan serta kata-kata yang disingkat. Pada penelitian ini mempunyai korpus sendiri yang dibuat secara manual dalam *file* teks yang akan diubah menjadi tipe data *dictionary*. Tujuan utama dari proses ini adalah untuk menyatukan berbagai bentuk kata-kata ke dalam bentuk yang benar sehingga kita tidak akan kehilangan informasi penting dari token yang berbeda dalam teks. Bagian ini membahas tentang karakter yang diulang serta mengoreksi ejaan. Meskipun keterbatasan pengoreksian kata karena terlalu randomnya kata-kata yang digunakan setiap orangnya.

```
d = {}
with open("dictionary.txt") as text:
    for line in text:
        if line.strip():
            key, val = line.split(None, 1)
            d[key]=val.split()
```

Kode 4. 9 Membaca Korpus sebagai *Dictionary*

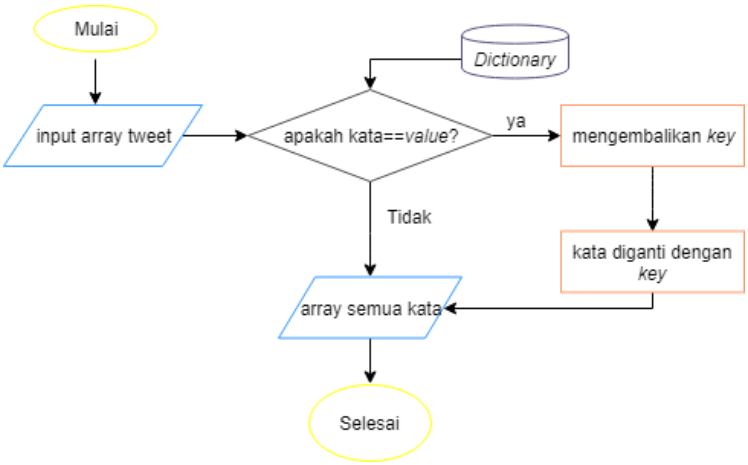
Kode 4.9 merupakan kode untuk membaca korpus sebagai *Dictionary*. *Dictionary* adalah stuktur data yang bentuknya seperti kamus. Ada kata *key* dan ada *value*. Kata *key* harus unik, sedangkan *value* boleh diisi dengan apa saja.

Berikut merupakan potongan dari korpus tersebut . Untuk korpus yang lengkap bisa dilihat pada Lampiran 1.

```
{'pelanggan': ['customer', 'kastemer', 'costumer',
'pelanggannya', 'planggan', 'plnggn', 'plnggan',
'pelnggan'],
internet': ['inet', 'intrnt', 'internetnya', 'intrnet',
'internt'],
'jaringan': ['jaringam', 'jrngn', 'jatingan', 'jaringa',
'jaringannya', 'jarngan', 'jaringanmu']}
```

Gambar 4. 5 Potongan *Dictionary*

Gambar 4.5 menunjukkan potongan *dictionary* dimana *key* menunjukkan kata yang benar sedangkan *value* berisikan *list* kata-kata yang kurang tepat dalam menuliskan kata pada *key*.



Gambar 4. 6 Diagram Alir Mengoreksi Kata

Gambar 4.6 merupakan proses mengoreksi kata yang salah ejaan dan menyamakan kata yang mempunyai arti yang sama.

```
def mencaritypo(kata):
    for key in d:
        list1=d.get(key)
        if kata in list1:
            return key
    return kata
```

Kode 4. 10 Mencari Kata yang Salah Ejaan

Kode 4.10 mencari kata yang salah ejaan dalam *tweet*, dengan cara jika kata yang ada dalam *dictionary* maka akan mengembalikan *key* dari data berupa kata yang benar.

```
def replacetypo(tupel):
    temp_data=[]
    for kalimat in tupel:
        temp_kalimat=[]
        for kata in kalimat:
            lit=kata.replace(kata, mencaritypo(kata))
            temp_kalimat.append(lit)
        temp_data.append(temp_kalimat)
    return temp_data
```

Kode 4. 11 Mengganti Kata yang Salah Ejaan

Kode 4.11 merupakan kode untuk mengganti kata yang salah ejaan dan menyamakan kata yang mempunyai arti yang sama. Berikut merupakan hasil dari kode mengoreksi kata-kata yang menggunakan data *dummy* pada Tabel 4.7.

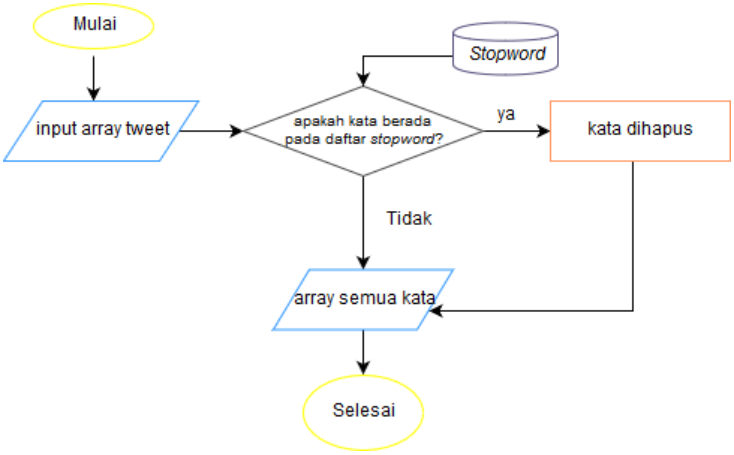
Tabel 4. 7 Hasil dari Proses Mengoreksi Kata-kata

['untung', 'sinyal', 'xl', 'lancar', 'di', 'tempat', 'kkn', 'terima', 'kasih']
['sinyal', 'tri', 'kalau', 'malam', 'cepat', 'sekali']
['telkomsel', 'kenapa', 'lagi', 'sih', 'lemot', 'sekali', 'dibuat', 'internetan', 'sudah', 'mahal', 'lemot']

```
['sinyal', 'indosat', 'jelek', 'sekali', 'tolong',
'perbaiki', 'dong']
```

4.3.9 Menghilangkan *Stopwords*

Stopwords adalah kata-kata yang memiliki signifikansi sedikit atau tidak sama sekali. Kata-kata tersebut biasanya dihapus dari teks selama pemrosesan untuk mempertahankan kata-kata yang memiliki signifikansi dan konteks yang penting. *Stopwords* biasanya kata-kata yang paling sering muncul jika mengumpulkan kumpulan teks berdasarkan token tunggal dan memeriksa frekuensinya. Kata-kata seperti ‘sebuah’, ‘ini’, ‘itu’, dan sebagainya adalah kata-kata penghenti. Tidak ada daftar *stopwords* universal atau lengkap. Proses ini dapat ditunjukkan oleh diagram alir pada Gambar 4.7. Untuk daftar *stopwords* bisa dilihat pada Lampiran 2.



Gambar 4. 7 Diagram Alir Menghilangkan *Stopwords*

```
def stopwords(tupel):
    temp_data=[]
    for kalimat in tupel:
        temp_kalimat=[]
        for kata in kalimat:
            lit=str.remove(kata)
            if(lit!=''):
              temp_kalimat.append(lit)
        temp_data.append(temp_kalimat)
    return temp_data
```

Kode 4. 12 Menghilangkan *Stopwords*

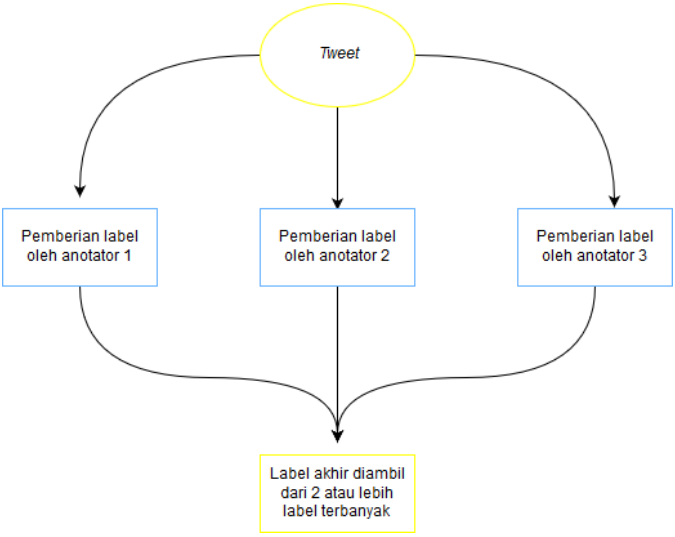
Kode 4.12 merupakan kode untuk menghilangkan *stopwords*. Berikut merupakan hasil dari kode mengoreksi kata-kata yang menggunakan data *dummy* pada Tabel 4.8 berikut

Tabel 4. 8 Hasil dari Proses Menghilangkan *Stopwords*

['untung', 'sinyal', 'xl, 'lancar', 'tempat', 'kkn', 'terima', 'kasih']
['sinyal', 'tri', 'kalau', 'malam', 'cepat', 'sekali']
['telkomsel', 'kenapa', 'lagi', 'lemot', 'sekali', 'dibuat', 'internetan', 'sudah', 'mahal', 'lemot']
['sinyal', 'indosat', 'jelek', 'sekali', 'tolong', 'perbaiki']

4.4 Pelabelan Data

Dataset yang berasal dari akun 4 operator telekomunikasi dan sudah melalui proses pembersihan pada tahap pra-pemrosesan data akan dilakukan proses pelabelan. Jumlah anotator yang akan memberikan label *tweet* berjumlah tiga orang. Jumlah anotator yang lebih dari satu orang memiliki tujuan untuk menghindari subjektifitas seseorang terhadap *tweet* tertentu sehingga sentimen *tweet* didapat dari perspektif dari satu orang lebih. Label negatif bernilai -1 dan label positif bernilai 1. Alur untuk menentukan label akhir dari sebuah *tweet* dapat ditunjukkan pada Gambar 4.8.



Gambar 4. 8 Alur Anotasi Label *Tweet*

Tabel 4. 9 Contoh Hasil pelabelan data *Dummy*.

Tweet	**Label**
Untung sinyal xl lancar tempat kkn terima kasih	Positif
Sinyal tri kaluar malam cepat sekali	Positif
telkomsel kenapa lagi lemot sekali dibuat internetan sudah mahal lemot	Negatif
sinyal indosat jelek sekali tolong perbaiki	Negatif

Tabel 4.9 merupakan contoh hasil akhir pelabelan dari data dummy.

4.5 *Word2Vec*

Proses pembuatan model *Word2Vec* ini menggunakan *library gensim*. Proses ini menggunakan algoritma *Skip-gram*.

```
size = 50
min_count = 2
workers = 2
window = 7
subsampling = 1e-3
```

Kode 4. 13 Inisialisasi Parameter Model *Word2Vec*

Kode 4.13 merupakan inisialisasi parameter untuk model Word2vec. Berikut merupakan penjelasan dari parameter yang digunakan untuk pembelajaran model *Word2Vec* :

1. Size : Dimensi dari vektor kata
2. Window : Jarak maksimum antara kata saat ini dengan prediksi dalam sebuah kalimat
3. Min_count : Parameter yang menentukan bahwa sebuah kata akan di pembelajaran apabila kata tersebut muncul minimal dalam jumlah yang ditentukan.
4. Workers : banyak utas pekerja ini untuk melatih model.
5. Subsamplings : Parameter ini merupakan salah satu parameter penting karena akan menentukan kandidat-kandidat yang dihasilkan dari proses prediksi menggunakan model yang telah terbentuk.

```
with open('data.csv','r') as r:
  p = r.readlines()#read data dari csv

tokenized_sents = token(p) #memanggil fungsi token
model = Word2Vec(tokenized_sent, workers, size,
min_count, window, sample = subsampling)

vocab_w2v = 'word2vec_model'
model.save(vocab_w2v) #menyimpan model
```

Kode 4. 14 Pembelajaran Model *Word2Vec*

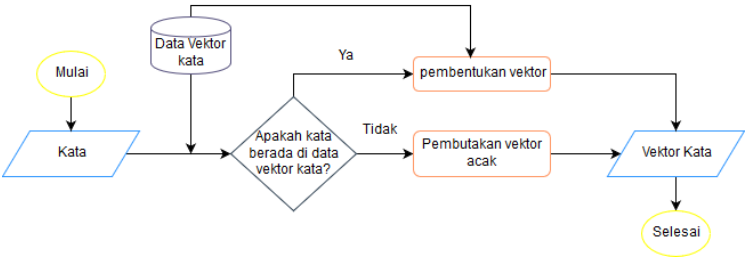
Kode 4.14 menjelaskan proses pembelajaran yang dilakukan pada algoritma *Word2Vec*. Setiap data akan dibaca per kata, maka data perlu dipisahkan terlebih dahulu proses ini disebut tokenisasi. Kemudian data akan masuk dalam suatu *list* dan dilakukan perhitungan *vocab* yang dididapatkan dari data latih model. Kemudian setiap kata akan diberikan nilai vektor dengan besar dimensi yang telah ditentukan. Terakhir *vocab* disimpan.

Berikut merupakan hasil dari Kode 4.14 yang menggunakan data *dummy* pada Gambar 4.9.

```
[array([-1.0849777 ,  0.44347578,  0.80632055, -0.55019647, -0.89073646,
        -0.5691194 ,  0.68800455,  0.00701755, -0.85650563, -0.46055126,
         0.18716167,  0.53544414,  0.7234436 , -0.06164705,  0.9083985 ,
        -0.55947983,  0.6505707 , -0.67439   , -0.00725378,  0.3742339 ,
        -1.1343596 ,  0.825441  ,  0.63944167,  0.61623824,  0.47513062,
        -0.59367377, -1.3853798 , -0.8101975 , -1.1934137 ,  0.58253807,
         0.30739346, -0.5505491 ,  0.42771137, -0.5833536 , -1.2702295 ,
         0.48490003,  0.82249767, -1.04592   ,  0.15339357,  0.8791019 ,
         0.7108408 , -0.7092029 , -0.20609556, -1.1325492 ,  1.3931099 ,
         0.07140531, -0.67285675,  0.171023  , -0.871494  ,  0.36598042],
       dtype=float32),
 array([-0.842254  ,  0.44600752,  0.7203129 , -0.4622567 , -0.7195468 ,
        -0.4112209 ,  0.57736   ,  0.00283157, -0.69192004, -0.38131595,
         0.16164033,  0.40420276,  0.62352157, -0.02892836,  0.7378063 ,
        -0.45656508,  0.585728  , -0.5832903 , -0.01000546,  0.30845052,
        -0.8743219 ,  0.6494313 ,  0.5531789 ,  0.4816269 ,  0.4138243 ,
        -0.46144497, -1.1111007 , -0.6611182 , -0.94393384,  0.5129219 ,
         0.30357203, -0.43761444,  0.3510695 , -0.48913866, -1.0611347 ,
         0.38596603,  0.6752844 , -0.8924879 ,  0.05711827,  0.74338275,
         0.5761465 , -0.57135415, -0.18794881, -0.9138407 ,  1.2059667 ,
         0.05792369, -0.5762689 ,  0.08086512, -0.7773318 ,  0.30315274],
```

Gambar 4. 9 Contoh Vektor Kata

Pada Gambar 4.9 merupakan hasil dari vektor kata ‘jaringan’ dan ‘lemot’.



Gambar 4. 10 Mekanisme Pembentukan Vektor Kata

Gambar 4.10 merupakan alur *Word2Vec* untuk masukan ke proses DCNN.

```
def vectorize(dataset, maxlen):
  vectorized_data=[]
  expected=[]
  for kalimat in dataset:
    sample_vecs=[]
    for kata in kalimat:
      if kata in word_vectors:
        sample_vecs.append(word_vectors[kata])
      else:
        sample_vecs.append(np.random.uniform(-
0.0,0.0,50))
    if len(sample_vecs) < maxlen:
      additional_elems = maxlen- len(sample_vecs)
      for _ in range (additional_elems):

sample_vecs.append(np.random.uniform(0.0,0.0,50))
    sample_vecs=np.array(sample_vecs)
    vectorized_data.append(sample_vecs)
  vectorized_data = np.array(vectorized_data)
  return vectorized_data
```

Kode 4. 15 Kode Mengubah Ukuran *Word2Vec*

Kode 4.15 mengubah ukuran *Word2Vec* menjadi ukuran maksimal vektor terpanjang dalam dataset. Jika panjang suatu

vektor kurang dari panjang vektor terpanjang, maka akan ditambah dengan nilai 0.

Berikut merupakan hasil dari Kode 4.15 yang menggunakan data *dummy* pada Gambar 4.11 berikut

```
array([[[-0.05652067,  0.02900808,  0.06096465, ...,  0.01255472,
         -0.06668233,  0.01957355],
        [-0.66971731,  0.44884259,  0.81386125, ..., -0.02181145,
         -0.82270002,  0.25308403],
        [-0.98692727,  0.50085753,  0.90295893, ...,  0.09754463,
         -0.91000766,  0.36050016],
        ...,
        [ 0.        ,  0.        ,  0.        , ...,  0.        ,
          0.        ,  0.        ],
        [ 0.        ,  0.        ,  0.        , ...,  0.        ,
          0.        ,  0.        ],
        [ 0.        ,  0.        ,  0.        , ...,  0.        ,
          0.        ,  0.        ]],
```

Gambar 4. 11 Contoh Hasil Menyamakan Panjang Vektor

4.6 Desain Model DCNN-SVM

Algoritma dari DCNN-SVM dapat ditunjukkan oleh Kode 4.16 berikut.

$window = \{h_1, h_2, h_3, \dots, h_n\}$
$z = \emptyset$
$data = \{X_1, X_2, \dots, X_m\}$
$for\ each\ h\ in\ window$:
    $w \leftarrow initializeFilter(h,m)$
    $for\ X\ in\ data$:
        $s \leftarrow size(X)$
        $c = \emptyset$
        $for\ i\ in\ 0{:}s-n+1$
            $x \leftarrow concantenate(X^{i:i+h-1})$
            $temp \leftarrow nonLinear(w^T x+b)$
            $c \leftarrow c \cup temp$
        end
        $\hat{c} \leftarrow \max(c)$
        $z \leftarrow z \cup \hat{c}$
    end
end

```
w ← initializeWeight(size(z))
skor ← svm(w^T z+b)
if skor > 0
        return "positive"
else
        return "negative"
```

Kode 4. 16 Algoritma DCNN-SVM

Proses pembuatan model DCNN ini menggunakan *library Keras.*

```
maxlen = max_len
batch_size = 10
embedding_dims = 50
filters = 100
kernel_size = 5
hidden_dims = 100
epoch =30
```

Kode 4. 17 Inisialisasi Parameter Model DCNN

Kode 4.17 merupakan inisialisasi parameter untuk model DCNN. Berikut merupakan penjelasan dari parameter yang digunakan untuk pembelajaran model DCNN :

1. Maxlen : kalimat terpanjang dalam dataset
2. Batch_size : Jumlah mini *batch* data yang digunakan selama pembelajaran model.
3. Embedding_dims : Ukuran dimensi vektor kata dari *word embedding*
4. Filters : mendefinisikan berapa banyak jendela geser yang akan dijalankan menurut data. fitur yang berbeda dapat dideteksi di lapisan ini. juga disebut fitur detektor
5. Kernel_size : Jumlah dan ukuran dari *filte*r kata yang digunakan

6. Epoch : Jumlah iterasi yang dilakukan selama pembelajaran model

```
split_point = int(len(tokenized_sent)*.8)
x_train = vectorize_data[:split_point]
y_train = df.label[:split_point]
x_test = vectorize_data[split_point:]
y_test = df.label[split_point:]
```

Kode 4. 18 *Split* Data Menjadi Data Latih dan Data Uji

Kode 4.18 membagi data menjadi 80% data latih dan 20% data uji. X untuk data *tweet* dan y untuk label.

```
Y_train=pd.get_dummies(y_train).values
Y_test=pd.get_dummies(y_test).values
```

Kode 4. 19 *Reshape* Label Data

Kode 4.19 mengubah data label dari 1 dimensi menjadi 2 dimensi. Berikut merupakan contoh hasil mengubah bentuk data pada Gambar 4.12.

```
array([[1, 0],
       [1, 0],
       [0, 1],
       [0, 1] ], dtype=uint8)
```

Gambar 4. 12 Contoh Hasil *Reshape* Label Data

Gambar 4.12 contoh hasil *reshape* label data, label -1 (negatif) akan menjadi [1, 0] dan label 1 (positif) akan menjadi [0, 1].

Arsitektur untuk membangun model DCNN dapat ditunjukkan pada Kode 4.20.

```
model = Sequential()
model.add(Conv1D(filters,
kernel_size,padding='valid',activation='relu',
strides =1,input_shape=(maxlen,embedding_dims)))
model.add(MaxPooling1D(pool_size=2, strides=1))

model.add(Conv1D(filters,
kernel_size,padding='valid',activation='relu',
strides=1,input_shape=(maxlen,embedding_dims)))
model.add(Conv1D(filters,
kernel_size,padding='valid',activation='relu',
strides =1,input_shape=(maxlen,embedding_dims)))
model.add(MaxPooling1D(pool_size=2, strides=1))

model.add(Conv1D(filters,
kernel_size,padding='valid',activation='relu',
strides=1,input_shape=(maxlen,embedding_dims)))
model.add(Conv1D(filters,
kernel_size,padding='valid',activation='relu',
strides =1,input_shape=(maxlen,embedding_dims)))
model.add(MaxPooling1D(pool_size=2, strides=1))

model.add(Flatten())
model.add(Dense(hidden_dims))
model.add(Dropout(0.3))
model.add(Dense(2, activation = 'sigmoid'))
```

Kode 4. 20 Model DCNN

Sedangkan arsitektur untuk membangun model SVM dapat ditunjukkan pada Kode 4.21.

```
model_svm =
Model(model.input,model.layers[5].output)
model_svm.compile(loss='categorical_crossentropy',op
      timizer=adam,metrics=['accuracy'])
x_svm_train = model_svm.predict(x_train)
clf = svm.LinearSVC()
clf.fit(x_svm_train,y_train)
x_svm_test = model_svm.predict(x_test)
predict_svm_train=clf.predict(x_svm_train)
predict_svm_test=clf.predict(x_svm_test)
```

Kode 4. 21 Model SVM

4.8 Implementasi *Graphic User Interface* (GUI)

Untuk memudahkan pengguna dalam mengoperasikan sistem, maka dibuat GUI sebagai jembatan antara aktor dengan sistem. Berikut tampilan GUI untuk program penelitian ini.

Gambar 4. 13 Tampilan GUI

Pada Gambar 4.13 aktor dapat mengisi *tweet* pada kotak yang telah disediakan setelah itu menekan tombol *check* untuk mengetahui sentimen dari *tweet* tersebut.

BAB V
PENGUJIAN DAN ANALISIS

Pada bab ini akan menjelaskan mengenai pengujian proses dan pengujian hasil.

5.1 Pengujian Proses

Menguji seluruh proses analisis sentimen dari *scraping* data, pra-pemrosesan data dan pelabelan.

5.1.1 Hasil Data *Scraping*

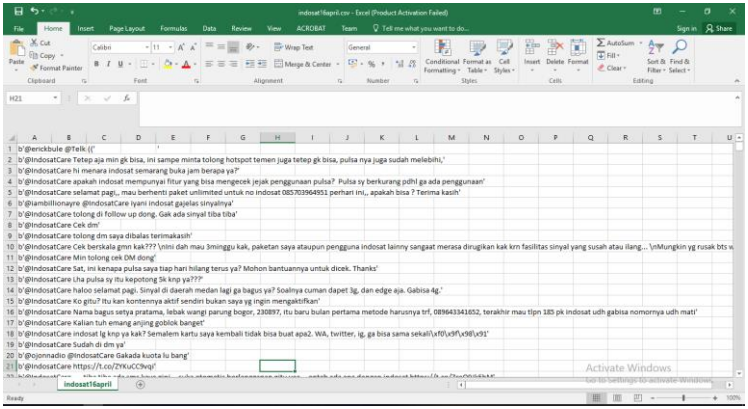
Proses *scraping* data diambil dari akun milik 4 operator telekomunikasi di Indonesia yang telah dijabarkan dalam batasan masalah pada bab 1 yaitu Indosat, Telkomsel, Tri, dan XL. Proses *scraping* ini dilakukan dari tanggal 2 April 2019-20 April 2019 dan mendapatkan hasil *tweet* sebanyak 27.739. Berikut merupakan rincian jumlah *tweet* hasil *scraping* tiap akun operator telekomunikasi pada Tabel 5.1.

Tabel 5. 1 Rincian Jumlah Tiap *Tweet*

Operator Telekomunikasi	Jumlah *Tweet*
Indosat	10.868
Telkomsel	7.783
Tri	2.141
XL	6.947

Berdasarkan Tabel 5.1 dapat ditunjukkan bahwa jumlah *tweet* terbanyak dari hasil *scraping* adalah Indosat dengan jumlah *tweet* 10.868, kedua Telkomsel dengan jumlah *tweet* 7.783 lalu disusul XL dengan jumlah *tweet* 6.947 dan yang terakhir adalah Tri dengan jumlah *tweet* 2.141.

Potongan hasil *scraping* data yang diambil dari twitter salah satu akun operator telekomunikasi yaitu XL yang disimpan dalam format CSV pada Gambar 5.1 berikut.



Gambar 5. 1 Potongan Hasil *Scraping* Data format CSV

5.1.2 Hasil Pra-pemrosesan Data

Setelah melakukan pra-pemrosesan data, data yang dimiliki sekarang sebanyak 20.852. Banyak data yang terduplikasi dan telah melalui proses pembersihan *tweet* sehingga data mengalami pengurangan yang cukup sigfinikan sebesar 6.887. Berikut merupakan rincian jumlah *tweet* tiap akun operator telekomunikasi setelah melalui pra-pemrosesan data pada Tabel 5.2.

Tabel 5. 2 Rincian Hasil Pra-pemrosesan Data

Operator Telekomunikasi	Jumlah *Tweet*
Indosat	10.084
Telkomsel	4.271
Tri	2.024
XL	4.473

Berdasarkan Tabel 5.2 dapat ditunjukkan bahwa jumlah *tweet* terbanyak dari hasil pra-pemrosesan data adalah Indosat dengan jumlah *tweet* 10.084 dengan pengurangan *tweet* sebanyak 784, kedua Telkomsel dengan jumlah *tweet* 4.271 dengan pengurangan *tweet* sebanyak 3.512 lalu disusul XL dengan jumlah *tweet* 4.473 dengan pengurangan *tweet* sebanyak 2.474 dan yang terakhir adalah Tri dengan jumlah *tweet* 2.024 dengan pengurangan *tweet* sebanyak 117. Jadi pengurangan *tweet* terbesar adalah dari data *tweet* Telkomsel.

Berikut potongan hasil proses pra-pemrosesan data pada Table 5.3.

Tabel 5. 3 Potongan Hasil Pra-pemrosesan Data

Aktivitas / Kondisi	**Hasil**
Tweet Awal	XL masiiiihh aja lemooootttt, Sampe kpn sih min? udah 2 jam lohh @myXLCare ☹ https://t.co/9C9ttvLaZ9
Menghapus URL	XL masiiiihh aja lemooootttt, Sampe kpn sih min? udah 2 jam lohh ☹
Menghapus Tanda Baca , Angka dan Simbol	XL masiiiihh aja lemooootttt Sampe kpn sih min udah jam lohh
Menghapus Huruf yang Berulang	XL masiihh aja lemoott Sampe kpn sih min udah jam lohh
Lowercase	xl masiihh aja lemoott sampe kpn sih min udah jam lohh
Tokenisasi	[‘xl’, ‘masiihh’, ‘aja’, ‘lemoott’, ‘sampe’, ‘kpn’, ‘sih’, ‘min’, ‘udah’, ‘jam’, ‘lohh’]

Mengoreksi Kata-Kata	[‘xl’, ‘masih’, ‘saja’, ‘lemot’, ‘sampai’, ‘kapan’, ‘sih’, ‘min’, ‘sudah’, ‘jam’, ‘lohh’]
Menghilangkan *Stopwords*	[‘xl’, ‘masih’, ‘lemot’, ‘sampai’, ‘kapan’, ‘sudah’, ‘jam’,]

Berdasarkan Tabel 5.3 didapatkan data dalam bentuk token setelah proses menghapus URL, tanda baca, angka simbol dan huruf yang berulang, mengubah menjadi *lowercase* dan tokenisasi, mengoreksi kata-kata yang salah dan menghilangkan *stopwords*. Setelah pra-pemrosesan data ini, dataset siap untuk digunakan untuk proses *word2vec*.

Tabel 5. 4 Kata yang Sering Muncul pada Indosat

Kata	**Jumlah**
Tidak	5869
Pulsa	2097
Indosat	1949
Sinyal	1786
Paket	1783
Nomor	1400
Tolong	1256
Jaringan	1237
Internet	1137
DM	892

Berdasarkan Tabel 5.4 dapat diketahui bahwa kata yang paling banyak muncul adalah ‘tidak’. Hal ini menunjukkan bahwa kata ‘pulsa’, ‘sinyal’, ‘paket’, ‘nomor’, ‘jaringan’, dan ‘internet’ tidak ada atau bermasalah.

Tabel 5. 5 Kata yang Sering Muncul pada XL

Kata	Jumlah
Tidak	2447
XL	1802
Sinyal	800
Nomor	716
Pulsa	621
Paket	565
Jaringan	518
Kuota	462
Tolong	448
DM	420

Berdasarkan Tabel 5.5 dapat diketahui bahwa kata yang paling banyak muncul adalah ‘tidak’. Hal ini menunjukkan bahwa kata ‘pulsa’, ‘sinyal’, ‘paket’, ‘nomor’, ‘jaringan’, dan ‘kuota’ tidak ada atau bermasalah.

Tabel 5. 6 Kata yang Sering Muncul pada Tri

Kata	Jumlah
Tidak	940
Sinyal	397
Tri	364
Paket	295
Kuota	229
Nomor	226
Jaringan	215
Pulsa	204
Tolong	171
Pakai	162

Berdasarkan Tabel 5.6 dapat diketahui bahwa kata yang paling banyak muncul adalah ‘tidak’. Hal ini menunjukkan bahwa kata ‘pulsa’, ‘sinyal’, ‘paket’, ‘nomor’, ‘jaringan’, dan ‘kuota’ tidak ada atau bermasalah.

Tabel 5. 7 Kata yang Sering Muncul pada Telkomsel

Kata	Jumlah
Tidak	1928
Telkomsel	707
Paket	615
Sinyal	542
Kuota	492
Nomor	449
Internet	434
Jaringan	425
Pulsa	359
Kartu	322

Berdasarkan Tabel 5.7 dapat diketahui bahwa kata yang paling banyak muncul adalah ‘tidak’. Hal ini menunjukkan bahwa kata ‘internet’, ‘sinyal’, ‘paket’, ‘nomor’, ‘jaringan’, dan ‘kuota’ tidak ada atau bermasalah.

5.1.3 Hasil Pelabelan Data

Proses pelabelan data dilakukan oleh 3 orang yang berbeda kemudian untuk mendapatkan label final yang akan digunakan untuk proses selanjutnya didapat dari 2 label atau lebih yang dipilih oleh masing-masing anotator. Berikut merupakan potongan hasil dari pelabelan data.

Tabel 5. 8 Proses Pelabelan Data

Tweet	Anotator 1	Anotator 2	Anotator 3	Label Akhir
Aku pengguna setia XL sudah 7 tahun, dulu enak-enak saja, sekarang gatau kenapa berubah, makin lemot. Tolong lah sinyalnya dibenerin @myXLCare	Negatif	Negatif	Positif	Negatif

Berdasarkan Tabel 5.8 ditunjukkan bahwa anotator 1 dan anotator 2 memberikan label negatif sedangkan anotator 3 memberikan label positif sehingga didapatkan label akhir negatif.

Berikut merupakan hasil akhir distribusi label yang didapatkan dari proses pelabelan dari 3 orang yang berbeda.

Tabel 5. 9 Jumlah Distribusi Data Berdasarkan Label

Label	Jumlah *Tweet*
Negatif	20.017
Positif	835

Berdasarkan Tabel 5.9 ditunjukkan bahwa jumlah tweet negatif sebanyak 20.017 sedangkan jumlah tweet positif sebanyak 835. Hal ini menunjukkan bahwa data tidak seimbang

karena cenderung ke label negatif karena pelanggan kebanyakan mengeluh dibandingkan pelanggan yang memuji.

Berikut merupakan hasil akhir distribusi data berdasarkan topik yang didapatkan dari proses pelabelan dari 3 orang yang berbeda.

Tabel 5. 10 Jumlah Distribusi *Tweet* Berdasarkan Topik

Topik	Label	Jumlah *Tweet*
Indosat	Negatif	9.841
Telkomsel	Negatif	3.995
Tri	Negatif	1.826
XL	Negatif	4.355
Indosat	Positif	243
Telkomsel	Positif	276
Tri	Positif	198
XL	Positif	118

Berdasarkan Tabel 5.10 data tidak seimbang baik secara topik maupun secara pelabelan karena secara topik jumlah *tweet* Indosat lebih banyak dan jumlah *tweet* Tri lebih sedikit. Hal ini diakibatkan dari keterbatasan yang dimiliki dari proses *scraping* data. Jumlah pelanggan yang mengeluh di Indosat juga lebih banyak dibandingkan dengan Telkomsel, Tri dan XL.

Berdasarkan hasil uji proses di atas dapat disimpulkan bahwa sistem berjalan dengan benar.

5.2 Pengujian Hasil

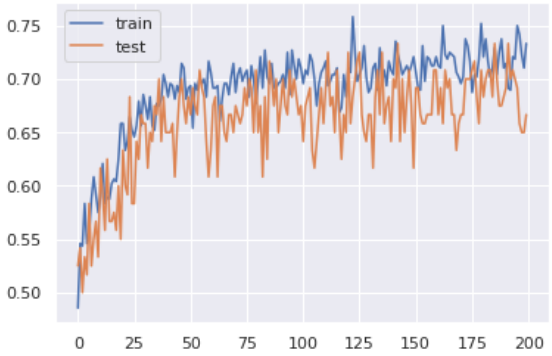
Setelah dilakukan proses pembelajaran maka didapatkan model yang telah menyesuaikan data yang dimasukkan. Model tersebut perlu diuji keakurasiannya sehingga dapat disimpulkan bahwa model yang dihasilkan merupakan model yang sesuai atau tidak.

Pengujian/validasi dilakukan dengan menggunakan data uji yang telah dikelompokkan sebelumnya. Data latih dan data uji merupakan data yang saling independen sehingga dapat dilihat kemampuan model untuk menangani data yang baru. Berdasarkan hasil implementasi yang dilakukan didapatkan keakurasian tiap iterasi untuk tiap jenis data.

Pada penelitian ini mengambil contoh kombinasi data positif 50% dan data negatif 50%. Berikut merupakan perbandingan akurasi data uji dan data latih tiap-tiap operator telekomunikasi dengan kombinasi data positif 50% dan data negatif 50% untuk membuktikan bahwa sistem bekerja dengan baik.

a. Telkomsel

Pada Gambar 5.2 menunjukkan perbandingan akurasi data latih dan data uji pada iterasi 200. Akurasi data fluktuatif. Akurasi model yang dihasilkan pada data latih kurang baik karena hanya dengan 200 kali iterasi akurasi hanya mencapai 73%.



Gambar 5. 2 Perbandingan Akurasi DCNN Telkomsel

Sedangkan akurasi model yang dihasilkan pada data uji memiliki hasil yang lebih rendah dibandingkan dengan data latih. Pada Gambar 5.2 menggambarkan akurasi dari data uji selama 200 kali iterasi. Akurasi cukup stabil. Akurasi model dengan data uji menghasilkan akurasi sebesar 67%. Sehingga didapatkan hasil performansinya pada Tabel 5.11 berikut.

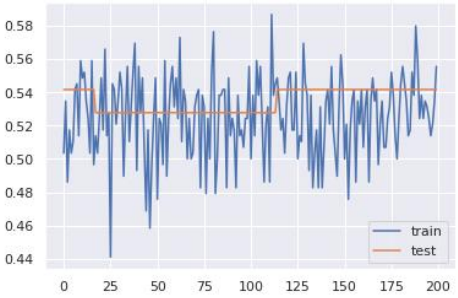
Tabel 5. 11 Hasil Performansi DCNN-SVM Telkomsel

	Presisi	***Recall***	**Akurasi**
Data Latih	59%	59%	59%
Data Uji	54%	54%	54%

Berdasarkan Tabel 5.11 didapat hasil performansi dari data latih dan data uji. Presisi, *recall*, dan akurasi untuk data latih sebesar 59%. Sedangkan presisi, *recall*, dan akurasi untuk data uji sebesar 54%.

b. Tri

Pada Gambar 5.3 menunjukkan perbandingan akurasi data latih dan data uji pada iterasi 200. Akurasi data fluktuatif. Akurasi model yang dihasilkan pada data latih kurang baik karena dengan 200 kali iterasi akurasi tertinggi dapat mencapai 55.9%.



Gambar 5. 3 Perbandingan Akurasi DCNN Tri

Sedangkan akurasi model yang dihasilkan pada data uji memiliki hasil yang lebih rendah dibandingkan dengan data latih. Pada Gambar 5.3 menggambarkan akurasi dari data uji selama 200 kali iterasi. Akurasi model dengan data uji menghasilkan akurasi sebesar 54%. Sehingga didapatkan hasil performansinya pada Tabel 5.12 berikut.

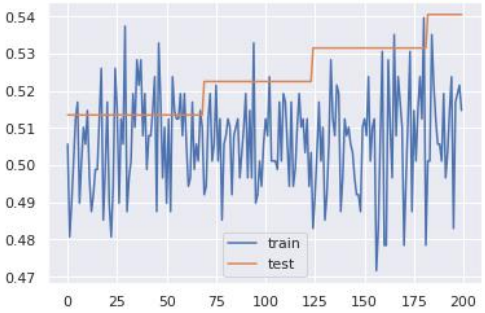
Tabel 5. 12 Hasil Performansi DCNN-SVM Tri

	Presisi	***Recall***	**Akurasi**
Data Latih	62%	61%	61%
Data Uji	56%	56%	56%

Berdasarkan Tabel 5.12 didapat hasil performansi dari data latih dan data uji. R*ecall* dan akurasi untuk data latih sebesar 61% dan presisi untuk data latih sebesar 62%. Sedangkan akurasi, presisi dan *recall* untuk data uji sebesar 56%.

c. INDOSAT

Pada Gambar 5.4 menunjukkan perbandingan akurasi data latih dan data uji pada iterasi 200. Akurasi data fluktuatif. Akurasi model yang dihasilkan pada data latih kurang baik karena dengan 200 kali iterasi akurasi dapat mencapai 51.5%.



Gambar 5. 4 Perbandingan Akurasi DCNN INDOSAT

Sedangkan akurasi model yang dihasilkan pada data uji memiliki hasil yang lebih rendah dibandingkan dengan data latih. Pada Gambar 5.4 menggambarkan akurasi dari data uji selama 200 kali iterasi. Akurasi data naik. Akurasi model dengan data uji menghasilkan akurasi sebesar 54%. Sehingga didapatkan hasil performansinya pada Tabel 5.13 berikut.

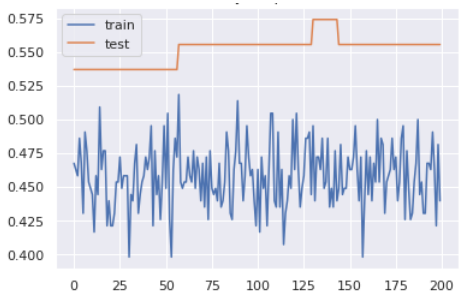
Tabel 5. 13 Hasil Performansi DCNN-SVM Indosat

	Presisi	***Recall***	**Akurasi**
Data Latih	58%	58%	58%
Data Uji	53%	53%	53%

Berdasarkan Tabel 5.13 didapat hasil performansi dari data latih dan data uji. Presisi, *recall*, dan akurasi untuk data latih sebesar 58%. Sedangkan presisi, *recall*, dan akurasi untuk data uji sebesar 53%.

d. XL

Pada Gambar 5.5 menunjukkan perbandingan akurasi data latih dan data uji DCNN pada iterasi 200. Akurasi data fluktuatif. Akurasi model yang dihasilkan pada data latih cukup baik karena hanya dengan 200 kali iterasi akurasi dapat mencapai 50.46%.



Gambar 5. 5 Perbandingan Akurasi DCNN XL

Sedangkan akurasi model yang dihasilkan pada data uji memiliki hasil yang lebih tinggi dibandingkan dengan data latih. Pada Gambar 5.5 menggambarkan akurasi dari data uji selama 200 kali iterasi. Akurasi data naik. Akurasi model dengan data uji menghasilkan akurasi sebesar 52.78%. Sehingga didapatkan hasil performansinya pada Tabel 5.14 berikut.

Tabel 5. 14 Hasil Performansi DCNN-SVM XL

	Presisi	***Recall***	**Akurasi**
Data Latih	61%	58%	58%
Data Uji	59%	54%	54%

Berdasarkan Tabel 5.14 didapat hasil performansi dari data latih dan data uji. Akurasi dan *recall* untuk data latih sebesar 58% dan presisi sebesar 61%. Sedangkan untuk data uji mendapatkan presisi dan *recall* sebesar 54% dan presisi sebesar 59%.

Data setiap operator telekomunikasi akan dibagi menjadi lima kombinasi yaitu 30% data positif dan 70% data negatif, 40% data positif dan 60% data negatif, 50% data positif dan 50% data negatif, 60% data positif dan 40% data negatif dan 70% data positif dan 30% data negatif.

Berikut merupakan ringkasan hasil dari beberapa kasus pembagian data positif dan negatif.

Tabel 5. 15 Hasil Akurasi Data Positif 30% dan Data Negatif 70%

		positif	negatif	Akurasi DCNN	Akurasi DCNN-SVM
		30%	70%		
Telkomsel	Jumlah	300	700		
	Data Latih	241	559	64%	70%
	Data Uji	59	141	71%	70%
Indosat	Jumlah	276	644		
	Data Latih	216	520	59.24%	71%
	Data Uji	60	124	63.59%	67%
Tri	Jumlah	180	420		
	Data Latih	140	340	60,42%	71%
	Data Uji	40	80	55%	67%
XL	Jumlah	135	315		
	Data Latih	103	257	41%	71%
	Data Uji	32	58	40%	64%

Berdasarkan Tabel 5.15 dapat diketahui bahwa hasil akurasi data latih dan data uji DCNN tertinggi adalah Telkomsel dengan nilai 64% dan 71%. Sementara hasil akurasi data latih DCNN-SVM tertinggi adalah XL, Indosat dan Tri dengan niai 71% dan hasil data uji DCNN-SVM tertinggi adalah Telkomsel dengan nilai 70%.

Tabel 5. 16 Hasil Akurasi Data Positif 40% dan Data Negatif 60%

		positif	negatif	Akurasi DCNN	Akurasi DCNN-SVM
		40%	60%		
Telkomsel	Jumlah	300	450		
	Data Latih	246	354	59,17%	59%
	Data Uji	54	96	63,33%	63%
Indosat	Jumlah	276	414		
	Data Latih	222	330	56.88%	60%
	Data Uji	54	84	47.10%	62%
Tri	Jumlah	180	270		
	Data Latih	144	216	50,83%	60%
	Data Uji	36	54	37,78%	61%
XL	Jumlah	134	201		
	Data Latih	105	163	52,99%	60%
	Data Uji	29	38	46,27%	57%

Berdasarkan Tabel 5.16 dapat diketahui bahwa hasil akurasi data latih dan data uji DCNN tertinggi adalah Telkomsel dengan nilai 59.17% dan 63.33%. Sementara hasil akurasi data latih DCNN-SVM tertinggi adalah XL, Indosat dan Tri dengan niai 60% dan hasil data uji DCNN-SVM tertinggi adalah Telkomsel dengan nilai 63%.

Tabel 5. 17 Hasil Akurasi Data Positif 50% dan Data Negatif 50%

		positif	negatif	Akurasi DCNN	Akurasi DCNN-SVM
		50%	50%		
Telkomsel	Jumlah	300	300		
	Data Latih	240	240	73%	59%
	Data Uji	60	60	67%	54%
Indosat	Jumlah	276	276		
	Data Latih	220	221	51.5%	58%
	Data Uji	56	55	54%	53%
Tri	Jumlah	180	180		
	Data Latih	146	142	55.9%	61%
	Data Uji	34	38	54%	56%
XL	Jumlah	135	135		
	Data Latih	110	106	50,46%	58%
	Data Uji	25	29	52.78%	54%

Berdasarkan Tabel 5.17 dapat diketahui bahwa hasil akurasi data latih DCNN tertinggi adalah Tri dengan nilai 58.68% dan data uji DCNN tertinggi adalah XL dengan nilai 53.70%. Sementara hasil akurasi data latih DCNN-SVM tertinggi adalah Tri dengan niai 57% dan hasil data uji DCNN-SVM tertinggi adalah Indosat dengan nilai 61%.

Tabel 5. 18 Hasil Akurasi Data Positif 60% dan Data Negatif 40%

		positif	negatif	Akurasi DCNN	Akurasi DCNN-SVM
		60%	40%		
Telkomsel	Jumlah	300	200		
	Data Latih	241	159	43%	61%
	Data Uji	59	41	40%	58%
Indosat	Jumlah	276	184		
	Data Latih	225	143	54.35%	62%
	Data Uji	51	41	45.65%	55%
Tri	Jumlah	180	120		
	Data Latih	142	98	60,42%	68%
	Data Uji	38	22	55%	70%
XL	Jumlah	135	90		
	Data Latih	101	79	62,22%	56%
	Data Uji	34	11	62,22%	76%

Berdasarkan Tabel 5.18 dapat diketahui bahwa hasil akurasi data latih dan data uji DCNN tertinggi adalah Telkomsel dengan nilai 59.17% dan 63.33%. Sementara hasil akurasi data latih DCNN-SVM tertinggi adalah XL, Indosat dan Tri dengan niai 60% dan hasil data uji DCNN-SVM tertinggi adalah Telkomsel dengan nilai 63%.

Tabel 5. 19 Hasil Akurasi Data Positif 70% dan Data Negatif 30%

		positif	negatif	Akurasi DCNN	Akurasi DCNN-SVM
		70%	30%		
Telkomsel	Jumlah	294	126		
	Data Latih	235	101	70.54%	70%
	Data Uji	59	25	70.24%	70%
Indosat	Jumlah	273	117		
	Data Latih	216	96	53.85%	70%
	Data Uji	57	21	50%	71%
Tri	Jumlah	175	75		
	Data Latih	138	62	52.5%	69%
	Data Uji	37	13	56%	74%
XL	Jumlah	133	57		
	Data Latih	102	50	63.16%	68%
	Data Uji	31	7	57.89%	79%

Berdasarkan Tabel 5.19 dapat diketahui bahwa hasil akurasi data latih dan data uji DCNN tertinggi adalah Telkomsel

dengan nilai 70.54% dan 70.24%. Sementara hasil akurasi data latih DCNN-SVM tertinggi adalah Telkomsel dan Indosat dengan niai 70% dan hasil data uji DCNN-SVM tertinggi adalah XL dengan nilai 79%.

Berikut merupakan rata-rata akurasi dari semua kombinasi untuk semua operator telekomunikasi pada Tabel 5.20.

Tabel 5. 20 Rata-rata Akurasi Semua Kombinasi Semua Operator

		Akurasi					
		30%:70%	40%:60%	50%:50%	60%:40%	70%:30%	Rata-rata
Telkomsel	Data Latih	70%	59%	59%	61%	70%	64%
	Data Uji	70%	63%	54%	58%	70%	63%
Indosat	Data Latih	71%	60%	58%	62%	70%	64%
	Data Uji	67%	62%	53%	55%	71%	62%
Tri	Data Latih	71%	60%	61%	68%	69%	66%
	Data Uji	67%	61%	56%	70%	74%	66%
XL	Data Latih	71%	60%	56%	58%	68%	63%
	Data Uji	64%	57%	56%	54%	79%	62%
		Rata-rata Data Latih					64%
		Rata-rata Data Uji					63%

Berdasarkan Tabel 5.20 dapat ditunjukkan bahwa rata-rata akurasi data latih paling tinggi adalah Tri dan rata-rata akurasi data uji paling tinggi adalah Tri.

Tabel 5. 21 Rata-rata *Recall* Semua Kombinasi Semua Operator

		Recall					
		30%:70%	40%:60%	50%:50%	60%:40%	70%:30%	Rata-rata
Telkomsel	Data Latih	70%	59%	59%	60%	70%	64%
	Data Uji	70%	64%	54%	59%	70%	63%
Indosat	Data Latih	71%	60%	58%	53%	69%	62%
	Data Uji	67%	61%	53%	45%	73%	60%
Tri	Data Latih	71%	60%	61%	60%	69%	64%
	Data Uji	67%	60%	56%	65%	74%	64%
XL	Data Latih	50%	61%	58%	56%	67%	58%
	Data Uji	50%	57%	54%	76%	82%	64%
		Rata-rata Data Latih					62%
		Rata-rata Data Uji					63%

Berdasarkan Tabel 5.21 dapat ditunjukkan bahwa rata-rata *recall* data latih paling tinggi adalah Telkomsel dan Tri dan rata-rata *recall* data uji paling tinggi adalah Tri dan XL.

Tabel 5. 22 Rata-rata Presisi Semua Kombinasi Semua Operator

		Presisi					
		30%:70%	40%:60%	50%:50%	60%:40%	70%:30%	Rata-rata
Telkomsel	Data Latih	49%	35%	59%	36%	49%	46%
	Data Uji	50%	41%	54%	35%	49%	46%
Indosat	Data Latih	50%	36%	58%	52%	48%	49%
	Data Uji	45%	37%	53%	42%	53%	46%
Tri	Data Latih	50%	36%	62%	76%	48%	54%
	Data Uji	44%	36%	56%	77%	55%	54%
XL	Data Latih	51%	37%	61%	31%	45%	45%
	Data Uji	42%	32%	59%	76%	67%	55%
		Rata-rata Data Latih					48%
		Rata-rata Data Uji					50%

Berdasarkan Tabel 5.22 dapat ditunjukkan bahwa rata-rata presisi data latih paling tinggi adalah Tri dan rata-rata presisi data uji paling tinggi adalah XL.

Berikut merupakan rata-rata dari setiap performansi data latih dan data uji pada Tabel 5.23.

Tabel 5. 23 Rata-rata Keseluruhan Performansi

	Akurasi	Presisi	Recall
Rata-rata Data Latih	64%	62%	48%
Rata-rata Data Uji	63%	63%	50%

Berdasarkan Tabel 5.23 dapat ditunjukkan bahwa rata-rata akurasi data latih sebesar 64%, rata-rata presisi data latih sebesar 62% dan rata-rata *recall* data latih sebesar 48%. Rata-rata akurasi data uji sebesar 63%, rata-rata presisi data uji sebesar 63% dan rata-rata *recall* data uji sebesar 50%.

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil dari uji coba yang dilakukan maka dapat disimpulkan antara lain:

1. Cara melakukan analisis sentimen tanggapan pelanggan operator telekomunikasi di twitter yaitu dengan cara *scraping* data, pra-pemrosesan data, pelabelan data, *word2vec*, dan pembentukan model klasifikasi menggunakan algoritma DCNN-SVM.
2. Hasil pelabelan didapatkan 20.017 dataset negatif dan 835 dataset positif. Kata yang sering banyak muncul pada dataset Telkomsel adalah 'tidak', 'telkomsel', dan 'paket'. Pada dataset Indosat adalah 'tidak', 'pulsa' dan 'indosat'. Pada dataset Tri adalah 'tidak', 'sinyal', dan 'tri'. Pada dataset XL adalah 'tidak', 'xl', dan 'sinyal'.
3. Hasil performansi dari penelitian ini yaitu akurasi data uji sebesar 63%, presisi data uji sebesar 63% dan *recall* data uji sebesar 50%.
4. Pelanggan operator telekomunikasi yang paling banyak mengeluh adalah Indosat.

6.2 Saran

Saran yang diberikan untuk perbaikan pada penelitian selanjutnya antara lain :

1. *Tweet* yang tidak mengandung sentimen seharusnya langsung masuk ke proses *Word2Vec* tanpa harus memaksakan masuk dalam proses pelabelan.

2. *Keyword* yang digunakan untuk *scraping* data kurang, sehingga dataset positif yang didapatkan sedikit.
3. Menambahkan metode untuk menghindari *overfitting.*

DAFTAR PUSTAKA

[1] Databoks. 2017. “Pengguna Ponsel Indonesia Mencapai 142% dari Populasi” [Online]. Available: https://databoks.katadata.co.id/datapublish/2017/08/29/pengguna-ponsel-indonesia-mencapai-142-dari-populasi. [Diakses 25 January 2019].

[2] A. P. J. I. Indonesia. 2017. “Penetrasi & Perilaku Pengguna Internet Indonesia - survey 2017”. [Online].

[3] Databoks. 2017. [Online]. Available : https://databoks.katadata.co.id/datapublish/2018/02/01/media-sosial-apa-yang-paling-sering-digunakan-masyarakat-indonesia. [Diakses 2019 February 19].

[4] F. Ali, K.-S. Kwak dan Y.-G. Kim. 2016. “Opinion Mining Based on Fuzzy Domain Ontology and Support Vector Machine : A Proposal to Automate Online Review Classification ” applied soft computing, Vol. 47, pp. 235-250.

[5] D. Y. Praptiwi. 2018. “Analisis Sentimen Online Review Pengguna E-Commerce Menggunakan Metode Support Vector Machine dan Maximum Entropy” Yogyakarta: Universitas Islam Indonesia.

[6] V. Singh dan S. K. Dubey. 2014. “Opinion Mining and Analysis : A Literature Review”. IEEE.

[7] Rozi, M.F., Mukhlash, I., Kimura, M. 2018. “Opinion Mining On Book Review Using CNN-L2-SVM Algorithm”. Vol.974. Journal of Physics: Conference Series. Pg. 012004.

[8] Mukhlash, I., Zamrudillah Arham, A., Rozi, F., Kimura, M., Adzkiya, D. 2018.. Vol.8. IJMLC Pg.437-441.

[9] A. Ahmed, H. Ammar, dan D. Milan. 2015. “Benchmarking Twitter Sentiment Analysis Tools”. Virginia: University of Virginia.

[10] Afnandika, Adrian. 2018. “Analisis Sentimen Teks Bahasa Indonesia pada Media Sosial Menggunakan Algoritma Convolutional Neural Network (Studi Kasus : Operator Telekomunikasi). Tugas Akhir. Surabaya: ITS.

[11] M. Bonzanini. 2016. “Mastering Sosial Media Mining with Python”. Birmingham: Packt.

[12] A. C. Pandey, M. Saraswat dan D. S. Rajpoot. 2017. “Twitter sentiment analysis using hybrid cuckoo search method” elsevier, Vol. 53, pp. 764-779.

[13] P. Dangeti. 2017. "Statistic for Machine Learning", Birmingham: Pact.

[14] S. Ananiadou dan J. McNaught. 2006. "Text Mining for Biology and Biomedicine". Boston and London: Artech House.

[15] M. Tomas dkk. 2017. “Distributed Representations of Words and Phrasesand their Compositionality”.

[16] A. Y. Wijaya, W. I. Suartika dan R. Solaiman. 2016. “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101”. Surabaya: ITS.

[17] B. Lin, X. Wei dan Z. Junjie. 2018. “Automatic Recognition and Classification of Multi-Channel Microseismic Waveform Based on DCNN and SVM ”. Elsevier, Vol. 123, pp. 111-120.

[18] Y. Kim. 2014“Convolutional Neural Networks for Sentence Classification”. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751. Qatar.

[19] F. Christianini dan S. T. Jhon. 2000. "An introduction to support vector machines and other kernel-based learning methods", Cambridge: Cambridge University Press.

[20] P. Nakov, A. Ritter, S. Rosenthal, V. Stoyanov, dan F. Sebastiani. 2016 “SemEval-2016 Task 4 : Sentiment Analysis in Twitter, ” pp. 1-18.

Lampiran 1

Corpus Mengoreksi Kata

- 'tidak': ['gak', 'gk', 'tdk', 'gx', 'ga', 'nggak', 'enggak', 'g', 'engga', 'ngga', 'tyda', 'tydac', 'tydak', 'ngak', 'ngk', 'kagak', 'nggk']
- 'lihat': ['lihay', 'liht', 'lhat', 'lht'], 'sedikit': ['dikit', 'sdkt', 'sdikit', 'sdkit']
- 'sebelum': ['sblm', 'sbelum', 'sblum', 'seblum', 'sebelm']
- 'dalam': ['dlm', 'dlam', 'dalm']
- 'paket': ['pkt', 'pket', 'pakt']
- 'hujan': ['ujan', 'hjan', 'hujn', 'hjn']
- 'lambat': ['lmbat', 'lambt', 'slow']
- 'habis': ['hbis', 'hbs', 'abis']
- 'lagu': ['lgu']
- 'emosi': ['esmosi']
- 'hampir': ['hmpr', 'hmpir']
- 'tidur': ['tdr', 'tdur']
- 'aplikasi': ['apps', 'aplks', 'aplkasi', 'aplksi', 'app']
- 'dong': ['donk']
- 'nomor': ['nomer', 'nmer', 'nmr', 'nomr', 'no']
- 'daripada': ['drpd', 'dripada', 'drpda', 'drpada']
- 'pulsa': ['plsa', 'pls']
- 'kangen': ['kgn']
- 'hilang': ['ilang', 'hlang', 'hlng', 'hlg']
- 'internet': ['inet', 'intrnt']
- 'bong': ['boong', 'bhng', 'bhong']
- 'sekarang': ['skr', 'skarang', 'skrng', 'skrg']
- 'tanggal': ['tgl', 'tggl', 'tnggal', 'tanggl']
- 'hey': ['hai', 'hy', 'heyy', 'hayy', 'haii', 'haii', 'hi']
- 'pembaruan': ['penbaruan', 'pmbaruan']

- 'ohh': ['owh', 'oh', 'och', 'ouch', 'ooh', 'oohh']
- 'kita': ['qt', 'qta']
- 'pernah': ['prnah', 'prnh']
- 'favorite': ['fav', 'favorit']
- 'nonton': ['nnton']
- 'provider': ['providr', 'provder', 'provdr']
- 'padahal': ['pdhl', 'pdhal', 'pdahal']
- 'pada': ['pda']
- 'telah': ['tlah', 'tlh']
- 'penuh': ['full', 'pnuh']
- 'makin': ['mkin']
- 'punya': ['pnya', 'pny']
- 'bawah': ['bwh', 'bwah']
- 'asli': ['aseli']
- 'telkomsel': ['tsel', 'tlkomsel']
- 'pelanggan': ['customer', 'kastemer', 'costumer']
- 'makan': ['mkn', 'mkan']
- 'juga': ['jg']
- 'kapan': ['kpan', 'kpn']
- 'dapat': ['dpt', 'dapet', 'dapt', 'dpet', 'dpat']
- 'tapi': ['tp', 'tpi']
- 'dengan': ['dgn', 'dngan', 'dengn', 'dg']
- 'untuk': ['untk', 'utk', 'tuk']
- 'dari': ['dr']
- 'kamu': ['lo', 'kmu', 'km', 'lu']
- 'terus': ['trus', 'trs']
- 'tahu': ['tau']
- 'tah': ['tahh', 'taah', 'taahh']
- 'begini': ['bgini', 'gini', 'ginii', 'giinii', 'giini']
- 'cacat': ['cacad']
- 'gaes': ['gaess', 'gaeess', 'gaees', 'gais', 'guys', 'guy', 'ges', 'gaiiss', 'gaiis', 'gaiss']
- 'yah': ['yahh', 'yaahh', 'yaah', 'yak']]
- 'aktif': ['aktiv', 'aktf']

- 'selamat': ['slmt', 'slamat', 'met', 'slamet', 'selamet']
- 'yang': ['yg']
- 'bayar': ['byr']
- 'barang': ['brg']
- 'tanya': ['ty', 'tnya']
- 'mau': ['mw', 'mo']
- 'oleh': ['olh']
- 'kemungkinan': ['kemungjinan']
- 'mungkin': ['mngkin', 'mngkn', 'mgkn']
- 'datang': ['dtg', 'dateng', 'dtang', 'datng']
- 'telepon': ['tlp', 'tlepon', 'telpon', 'tlpon', 'telfn', 'telvn', 'telvon', 'tlpn', 'telp']
- 'kembali': ['kembalu', 'kmbali', 'kmbli']
- 'mohon': ['mhn', 'mhon', 'mohn']
- 'anak': ['ank']
- 'jawaban': ['jwaban', 'jwbn', 'jwb', 'jawab']
- 'daftar': ['dftar', 'dftr']
- 'oke': ['ok', 'okelah']
- 'ada': ['ad']
- 'yuk': ['kuy']
- 'standar': ['standard', 'stndar']
- 'lebih': ['lbh', 'lrbih', 'lbih']
- 'kartu': ['krtu', 'krt']
- 'setiap': ['stiap']
- 'harus': ['hrs']
- 'warna': ['wa4na', 'wrna']
- 'terimakasih': ['thanks', 'tanks', 'thx', 'thank', 'tnks', 'makasih', 'mksh', 'trimakasih', 'tq', 'tnks', 'tx', 'mkasih', 'mksih', 'tks', 'timakaci', 'trims']
- 'tetap': ['tetep', 'ttp']
- 'mas': ['bang', 'kak', 'kakak', 'om', 'kaka', 'ms']
- 'jadi': ['jd', 'jdi']
- 'begitu': ['gitu', 'gtu', 'gt']
- 'semua': ['smua']
- 'baterai': ['batre']

- 'lama': ['lm', 'lma', 'luamaa', 'lamaa']
- 'sama': ['sm', 'ama', 'sma']
- 'pengiriman': ['pngiriman', 'delivery']
- 'agak': ['rada', 'agk']
- 'teman': ['temen', 'tmn', 'tmen']
- 'belanja': ['blnj', 'blnja']
- 'kualitas': ['qualitas', 'kwatitas']
- 'lagi': ['lg', 'lgi']
- 'bonus': ['bnus', 'bns']
- 'karena': ['krn', 'karna', 'krna', 'grgr', 'gara']
- 'sudah': ['udah', 'udh', 'sdh', 'sdah', 'uda', 'dah', 'wes']
- 'direkomendasikan': ['recommended', 'rekomended']
- 'bagus': ['good', 'nice', 'bgus', 'bgs']
- 'tersedia': ['ready']
- 'kesal': ['kesel', 'ksl', 'kzl']
- 'hidup': ['idup', 'hdp', 'hdup']
- 'label': ['lebel', 'lbel']
- 'komplain': ['complain']
- 'produk': ['prodk', 'peoduk']
- 'beberapa': ['bbrp', 'bbrapa']
- 'berapa': ['brp', 'brapa']
- 'dan': ['n', 'dn']
- 'pesanan': ['order', 'pesana', 'psanan']
- 'besar': ['gede', 'gde', 'bsar']
- 'banyak': ['byk', 'bnyak', 'bnyk']
- 'apakah': ['apkh', 'apakh']
- 'bisa': ['bsa', 'bs']
- 'terima': ['trima', 'trm']
- 'saya': ['sya', 'sy', 'gue', 'aku', 'ak', 'gw', 'ane', 'ku', 'gua', 'aq', 'aing']
- 'banget': ['bngt', 'bgt', 'bnget']
- 'tolong': ['tlng', 'tlong', 'tlg', 'please', 'pliss', 'plis']
- 'cepat': ['fast', 'cpt', 'cpat', 'cepet']
- 'respon': ['respond', 'rspon', 'respn', 'response', 'respone']

- 'mantap': ['mantul', 'mntp', 'mantp', 'mntap', 'mantaap', 'mantaff', 'mantaf']
- 'kemarin': ['kmarin', 'kemrin', 'kmrn', 'kemaren']
- 'selalu': ['slalu', 'sll']
- 'mereka': ['mrk', 'mreka']
- 'sedih': ['sdih', 'syedihh', 'syediihh']
- 'keren': ['kren']
- 'kirim': ['krim', 'girim', 'kirm']
- 'belum': ['blum', 'blm', 'belom', 'blom', 'lom']
- 'pelayanan': ['playanan', 'service', 'servis', 'plynn']
- 'wah': ['wa']
- 'nya': ['ny']
- 'gimana': ['gmn', 'gmna', 'gmana']
- 'kecewa': ['kcw', 'kcewa']
- 'balas': ['bls', 'bales']
- 'bukan': ['bkn', 'bkan']
- 'jaringan': ['jaringam', 'jrngn', 'jatingan']
- 'jangan': ['jgn', 'jngn', 'jngan']
- 'kenapa': ['knp', 'knpa', 'knapa', 'napa', 'nape']
- 'jika': ['kalau', 'klo', 'kalo', 'kl', 'klau']
- 'deh': ['dech'], 'disini': ['dsni', 'dsini']
- 'eror': ['error']
- 'mbak': ['mba', 'mb', 'mbsk']
- 'duh': ['duuh', 'duhh', 'duuhh']
- 'woy': ['woi', 'wooy', 'woyy', 'wooyy']
- 'sebagai': ['sbgai', 'sbagai', 'sbg']
- 'main': ['maen']
- 'seperti': ['kayak', 'kya', 'kyk']
- 'memang': ['emang', 'emg', 'emng']
- 'saja': ['aja', 'sja', 'aj']
- 'sampai': ['nyampe', 'smpai', 'nyampai', 'nyampek', 'sampe']
- 'iya': ['ya', 'iy', 'y', 'iye', 'iyak', 'iyyak', 'iyyakk', 'iyo', 'yo', 'ya',

'yaa', 'yyaa', 'yah', 'yahh', 'yaahh', 'yaah']

- 'pakai': ['pkai', 'pake', 'pke', 'pakek']
- 'menit': ['mnt', 'mnit', 'ment']
- 'pengaturan': ['setting', 'seting']
- 'jelek': ['jelel', 'jele', 'jelk', 'jlek']

Lampiran 2

Corpus Stopwords

iya, maya, hi, diatur, lah, yang, dengan, nya, pagi, kak, min, genks, gaes, a, ada, adalah, adanya, adapun, agak, agaknya, agar, akan, akankah, akhir, akhiri, akhirnya, aku, akulah, amat, amatlah, andalah, antar, antara, antaranya, apa, apaan, apabila, apakah, apalagi, apatah, arti, artinya, asal, asalkan, atas, atau, ataukah, ataupun, awal, awalnya, b, bagai, bagaikan, bagaimana, bagaimanakah, bagaimanapun, bagainamakah, bagi, bagian, bahkan, bahwa, bahwasannya, bahwasanya, baik, baiklah, bakal, bakalan, balik, banyak, bapak, baru, bawah, beberapa, begini, beginian, beginikah, beginilah, begitu, begitukah, begitulah, begitupun, belakang, belakangan, belum, belumlah, benar, benarkah, benarlah, berada, berakhir, berakhirlah, berakhirnya, berapa, berapakah, berapalah, berapapun, berarti, berawal, berbagai, berdatangan, beri, berikan, berikut, berikutnya, berjumlah, berkali-kali, berkata, berkehendak, berkeinginan, berkenaan, berlainan, berlalu, berlangsung, berlebihan, bermacam, bermacam-macam, bermaksud, bermula, bersama, bersama-sama, bersiap, bersiap-siap, bertanya, bertanya-tanya, berturut, berturut-turut, bertutur, berujar, berupa, besar, betul, betulkah, biasa, biasanya, bila, bilakah, bisa, bisakah, boleh, bolehkah, bolehlah, buat, bukan, bukankah, bukanlah, bukannya, bulan, bung, c, cara, caranya, cukup, cukupkah, cukuplah, cuma, d, dahulu, dalam, dan, dapat, dari, daripada, datang, dekat, demi, demikian, demikianlah, dengan, depan, di, dia, diakhiri, diakhirinya, dialah, diantara, diantaranya, diberi, diberikan, diberikannya, dibuat, dibuatnya, didapat, didatangkan, digunakan, diibaratkan, diibaratkannya, diingat, diingatkan, diinginkan, dijawab, dijelaskan, dijelaskannya, dikarenakan, dikatakan, dikatakannya, dikerjakan, diketahui, diketahuinya, dikira, dilakukan, dilalui, dilihat, dimaksud, dimaksudkan, dimaksudkannya, dimaksudnya, diminta, dimintai, dimisalkan, dimulai, dimulailah, dimulainya, dimungkinkan, dini, dipastikan, diperbuat, diperbuatnya, dipergunakan, diperkirakan, diperlihatkan, diperlukan, diperlukannya, dipersoalkan, dipertanyakan, dipunyai, diri, dirinya, disampaikan, disebut, disebutkan, disebutkannya, disini, disinilah, ditambahkan, ditandaskan, ditanya, ditanyai, ditanyakan, ditegaskan, ditujukan, ditunjuk, ditunjuki, ditunjukkan, ditunjukkannya, ditunjuknya, dituturkan, dituturkannya, diucapkan, diucapkannya, diungkapkan,

dong, dua, dulu, e, empat, enak, enggak, enggaknya, entah, entahlah, f, g, guna, gunakan, h, hadap, hai, hal, halo, hallo, hampir, hanya, hanyalah, hari, harus, haruslah, harusnya, helo, hello, hendak, hendaklah, hendaknya, hingga, i, ia, ialah, ibarat, ibaratkan, ibaratnya, ibu, ikut, ingat, ingat-ingat, ingin, inginkah, inginkan, ini, inikah, inilah, itu, itukah, itulah, j, jadi, jadilah, jadinya, jangan, jangankan, janganlah, jauh, jawab, jawaban, jawabnya, jelas, jelaskan, jelaslah, jelasnya, jika, jikalau, juga, jumlah, jumlahnya, justru, k, kadar, kala, kalau, kalaulah, kalaupun, kali, kalian, kami, kamilah, kamu, kamulah, kan, kapan, kapankah, kapanpun, karena, karenanya, kasus, kata, katakan, katakanlah, katanya, ke, keadaan, kebetulan, kecil, kedua, keduanya, keinginan, kelamaan, kelihatan, kelihatannya, kelima, keluar, kembali, kemudian, kemungkinan, kemungkinannya, kena, kenapa, kepada, kepadanya, kerja, kesampaian, keseluruhan, keseluruhannya, keterlaluan, ketika, khusus, khususnya, kini, kinilah, kira, kira-kira, kiranya, kita, kitalah, kok, kurang, l, lagi, lagian, lah, lain, lainnya, laku, lalu, lama, lamanya, langsung, lanjut, lanjutnya, lebih, lewat, lihat, lima, luar, m, macam, maka, makanya, makin, maksud, malah, malahan, mampu, mampukah, mana, manakala, manalagi, masa, masalah, masalahnya, masih, masihkah, masing, masing-masing, masuk, mata, mau, maupun, melainkan, melakukan, melalui, melihat, melihatnya, memang, memastikan, memberi, memberikan, membuat, memerlukan, memihak, meminta, memintakan, memisalkan, memperbuat, mempergunakan, memperkirakan, memperlihatkan, mempersiapkan, mempersoalkan, mempertanyakan, mempunyai, memulai, memungkinkan, menaiki, menambahkan, menandaskan, menanti, menanti-nanti, menantikan, menanya, menanyai, menanyakan, mendapat, mendapatkan, mendatang, mendatangi, mendatangkan, menegaskan, mengakhiri, mengapa, mengatakan, mengatakannya, mengenai, mengerjakan, mengetahui, menggunakan, menghendaki, mengibaratkan, mengibaratkannya, mengingat, mengingatkan, menginginkan, mengira, mengucapkan, mengucapkannya, mengungkapkan, menjadi, menjawab, menjelaskan, menuju, menunjuk, menunjuki, menunjukkan, menunjuknya, menurut, menuturkan, menyampaikan, menyangkut, menyatakan, menyebutkan, menyeluruh, menyiapkan, merasa, mereka, merekalah, sih, merupakan, meski, meskipun, meyakini, meyakinkan, minta, mirip, misal, misalkan, misalnya, mohon, mula, mulai, mulailah, mulanya, mungkin, mungkinkah, n,

nah, naik, namun, nanti, nantinya, nya, nyaris, nyata, nyatanya, o, oleh, olehnya, orang, p, pada, padahal, padanya, pak, paling, panjang, pantas, para, pasti, pastilah, penting, pentingnya, per, percuma, perlu, perlukah, perlunya, pernah, persoalan, pertama, pertama-tama, pertanyaan, pertanyakan, pihak, pihaknya, pukul, pula, pun, punya, q, r, rasa, rasanya, rupa, rupanya, s, saat, saatnya, saja, sajalah, salam, saling, sama, sama-sama, sambil, sampai, sampai-sampai, sampaikan, sana, sangat, sangatlah, sangkut, satu, saya, sayalah, se, sebab, sebabnya, sebagai, sebagaimana, sebagainya, sebagian, sebaik, sebaik-baiknya, sebaiknya, sebaliknya, sebanyak, sebegini, sebegitu, sebelum, sebelumnya, sebenarnya, seberapa, sebesar, sebetulnya, sebisanya, sebuah, sebut, sebutlah, sebutnya, secara, secukupnya, sedang, sedangkan, sedemikian, sedikit, sedikitnya, seenaknya, segala, segalanya, segera, seharusnya, sehingga, seingat, sejak, sejauh, sejenak, sejumlah, sekadar, sekadarnya, sekali, sekali-kali, sekalian, sekaligus, sekalipun, sekarang, sekaranglah, sekecil, seketika, sekiranya, sekitar, sekitarnya, sekurang-kurangnya, sekurangnya, sela, selain, selaku, selalu, selama, selama-lamanya, selamanya, selanjutnya, seluruh, seluruhnya, semacam, semakin, semampu, semampunya, semasa, semasih, semata, semata-mata, semaunya, sementara, semisal, semisalnya, sempat, semua, semuanya, semula, sendiri, sendirian, sendirinya, seolah, seolah-olah, seorang, sepanjang, sepantasnya, sepantasnyalah, seperlunya, seperti, sepertinya, sepihak, sering, seringnya, serta, serupa, sesaat, sesama, sesampai, sesegera, sesekali, seseorang, sesuatu, sesuatunya, sesudah, sesudahnya, setelah, setempat, setengah, seterusnya, setiap, setiba, setibanya, terimakasih, setidak-tidaknya, setidaknya, setinggi, seusai, sewaktu, siap, siapa, siapakah, siapapun, sini, sinilah, soal, soalnya, suatu, sudah, sudahkah, sudahlah, supaya, t, tadi, tadinya, tahu, tak, nih, tambah, tambahnya, tampak, tampaknya, tandas, tandasnya, tanpa, tanya, tanyakan, tanyanya, tapi, tegas, tegasnya, telah, tempat, tentang, tentu, tentulah, tentunya, tepat, terakhir, terasa, terbanyak, terdahulu, terdapat, terdiri, terhadap, terhadapnya, teringat, teringat-ingat, terjadi, terjadilah, terjadinya, terkira, terlalu, terlebih, terlihat, termasuk, ternyata, tersampaikan, tersebut, tersebutlah, tertentu, tertuju, terus, terutama, tetap, tetapi, tiap, tiba, tiba-tiba, ifttt, myxlcare, tidakkah, tidaklah, tiga, toh, tuju, tunjuk, turut, tutur, tuturnya, u, ucap, ucapnya, ujar, ujarnya, umumnya, ungkap, ungkapnya, untuk, usah, usai, v, w, waduh, wah, wahai, waktunya, walau, walaupun, wong, x,

y, yy, yaitu, yakin, yakni, yang, z, yaa, tuh, kah, ni, ko, loh, yah, si, deh, an, lho, mah, cc, eh, hehe, koq, woy, dll, tsb, ta, lur, yahh, aye, dah, da, tu, kol, pol, ah, hey, yth, cuss, bae, cuy, yap, rb, gb, mas, cs, gaes, rp, puk, mbak, tah, duh, aa, bb, dd, ee, ff, gg, hh, ii, jj, ll, mm, nn, oo, pp, qq, rr, ss, tt, uu, vv, ww, xx, xa, xb, xc, xd, xe, xf, xg, xh, xi, xj, xk, xm, xn, xo, xp, xq, xr, xs, xt, xu, xv, xw, xy, xzzz, hm, hmm, hhmm.

Lampiran 3

Matriks *Confusion*

		Label Prediksi		
		-1	1	*support*
Label *Tweet*	-1	119	102	221
	1	94	126	220

Perhitungan akurasi:

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} = \frac{119 + 126}{119 + 126 + 94 + 102}$$
$$= 0.56$$

Perhitungan *recall*:

$$recall = \frac{TP}{TP + FN}$$

Label -1	$\frac{119}{119 + 102} = 0.54$
Label 1	$\frac{126}{126 + 94} = 0.57$

$$Weighted\ Avg\ Recall\ = \frac{221}{441} x\ 0.54 + \frac{220}{441} x\ 0.57 = 0.56$$

Perhitungan presisi:

$$presisi = \frac{TP}{TP + FP}$$

Label -1	$\frac{119}{119+94} = 0.56$
Label 1	$\frac{126}{126+102} = 0.55$

$$Weighted\ Avg\ Precision\ = \frac{221}{441} x\ 0.56 + \frac{220}{441} x\ 0.55 = 0.56$$

BIODATA PENULIS

Inayah Eka Firdausi atau biasa dipanggil Inayah lahir di Jember, tanggal 8 Februari 1997. Pendidikan formal yang pernah ditempuh yaitu TK Dharma Wanita, SDN Lembengan 1, SMP Negeri 1 Kalisat dan SMA Negeri 2 Jember. Sekarang penulis menempuh pendidikan S1 di Departemen Matematika Fakultas Matematika, Komputasi dan Sains Data Institut Teknologi Sepuluh Nopember Surabaya dengan bidang minat Ilmu Komputer. Selama kuliah, penulis aktif di organisasi di tingkat Institut yaitu BEM ITS dan tingkat jurusan yakni HIMATIKA ITS. Pada tahun 2016-2017 penulis menjadi anggota aktif HIMATIKA ITS sebagai *staff External Affair* dan *staff* Adkesma BEM ITS. Pada tahun 2017-2018 penulis hanya aktif di HIMATIKA ITS dengan mengemban amanah sebagai *Second Vice of* HIMATIKA ITS. Demikian biodata tentang penulis. Jika ingin memberikan saran, kritik, dan diskusi mengenai penelitian tugas akhir ini, dapat dikirimkan melalui email inayahekaf@gmail.com. Terimakasih.