



TUGAS AKHIR - KM184801

**ANALISIS SENTIMEN OPINI MASYARAKAT TERHADAP
BAKAL CALON WALIKOTA SURABAYA 2020
BERDASARKAN *SOCIAL MEDIA MINING* MENGGUNAKAN
ALGORITMA *N-GRAM-MULTICHANNEL CNN***

FERISA TRI PUTRI PRESTASI
NRP. 0611164000031

Dosen Pembimbing:
Prof. Dr. Mohammad Isa Irawan, M.T.

DEPARTEMEN MATEMATIKA
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember
Surabaya 2020



FINAL PROJECT - KM184801

***SENTIMENT ANALYSIS OF PUBLIC OPINION TO FUTURE
CANDIDATES SURABAYA MAYOR 2020 BASED ON SOCIAL
MEDIA MINING USING N-GRAM-MULTICHANNEL CNN
ALGORITHM***

FERISA TRI PUTRI PRESTASI

Student Number. 0611164000031

Supervisors:

Prof. Dr. Mohammad Isa Irawan, M.T.

DEPARTMENT OF MATHEMATICS

Faculty of Science and Data Analytics

Institut Teknologi Sepuluh Nopember

Surabaya 2020

LEMBAR PENGESAHAN

ANALISIS SENTIMEN OPINI MASYARAKAT TERHADAP BAKAL CALON WALIKOTA SURABAYA 2020 BERDASARKAN *SOCIAL MEDIA MINING* MENGGUNAKAN ALGORITMA *N-GRAM- MULTICHANNEL CNN*

SENTIMENT ANALYSIS OF PUBLIC OPINION TO FUTURE CANDIDATES SURABAYA MAYOR 2020 BASED ON SOCIAL MEDIA MINING USING N-GRAM- MULTICHANNEL CNN 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 Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :

FERISA TRI PUTRI PRESTASI
NRP. 0611164000031

Menyetujui,
Dosen Pembimbing,



Prof. Dr. techn. Mohammad Isa Irawan, M.T
NIP. 196312251989031001

Mengetahui,

Kepala Departemen Matematika
FSAD ITS



Subchan, Ph.D

NIP. 197105131997021001
Surabaya, 28 Juli 2020

**ANALISIS SENTIMEN OPINI MASYARAKAT TERHADAP
BAKAL CALON WALIKOTA SURABAYA 2020
BERDASARKAN *SOCIAL MEDIA MINING* MENGGUNAKAN
ALGORITMA *N-GRAM-MULTICHANNEL CNN***

Nama Mahasiswa : FERISA TRI PUTRI PRESTASI
NRP : 0611164000031
Departemen : Matematika ITS
Dosen Pembimbing : Prof. Dr. Mohammad Isa Irawan, M.T.

ABSTRAK

Pilwali Surabaya mendatang menjadi perhatian publik, dimana para bakal calon mulai mengajukan diri dan ramai diperbincangkan di media sosial, diantaranya Facebook dan Twitter. Penting bagi bakal calon untuk mengetahui sentimen opini yang berkembang di media sosial dengan menggunakan implementasi analisa sentimen pada *social media mining* untuk dapat di klasifikasi sentimen positif atau negatif dari opini tersebut. Oleh karena itu, pada penelitian analisa sentimen ini digunakan algoritma *N-Gram Multichannel CNN* untuk mendapatkan analisa sentimen dengan memanfaatkan konsep *Natural Language Processing* yang memungkinkan komputer untuk memproses dan memahami bahasa alami manusia dan memperoleh akurasi model optimal terhadap dataset opini. Hasil diperoleh analisis sentimen pada teks berisikan opini masyarakat dari media sosial terhadap bakal calon Walikota Surabaya 2020 dapat diterapkan dengan baik menggunakan Algoritma *N-Gram-Multichannel CNN* dengan diperoleh akurasi model terhadap data latih sebesar 94.38% dan 96.12% pada data uji.

Kata Kunci : Pilwali Surabaya, Facebook, Twitter, *Social Media Mining*, Analisis Sentimen, *Natural Language Processing*, *N-Gram-Multichannel CNN*

**SENTIMENT ANALYSIS OF PUBLIC OPINION TO FUTURE
CANDIDATES SURABAYA MAYOR 2020 BASED ON SOCIAL
MEDIA MINING USING
N-GRAM-MULTICHANNEL CNN ALGORITHM**

Student Name : FERISA TRI PUTRI PRESTASI
Student Number : 06111640000031
Department : Mathematics ITS
Supervisors : Prof. Dr. Mohammad Isa Irawan, M.T.

ABSTRACT

The next Surabaya elections are a matter of public concern, where prospective candidates begin to propose themselves and are widely discussed on social media, including Facebook and Twitter. It is important for prospective candidates to find out opinion sentiments that develop in social media by using the implementation of sentiment analysis in social media mining to be able to classify positive or negative sentiments from these opinions. Therefore, in this sentiment analysis research the N-Gram Multichannel CNN algorithm is used to get sentiment analysis by utilizing the concept of Natural Language Processing which enables computers to process and understand human natural language and obtain optimal model accuracy of opinion datasets. The results obtained sentiment analysis in the text containing public opinion from social media to the prospective Surabaya Mayor 2020 can be implemented well using the CNN N-Gram-Multichannel Algorithm with the accuracy of the model obtained for training data 94.38% and 96.12% in the test data.

Keywords: The mayor election of Surabaya, Facebook, Twitter, Social Media Mining, Sentiment Analysis, Natural Language Processing, N-Gram-Multichannel CNN

KATA PENGANTAR

Dengan mengucapkan Alhamdulillah segala puji dan syukur penulis panjatkan atas kehadiran Allah SWT atas berkat dan rahmat-Nyalah meskipun ditengah pandemi, namun pengerjaan Tugas Akhir yang berjudul “Analisis Sentimen Opini Masyarakat Terhadap Bakal Calon Walikota Surabaya 2020 Berdasarkan *Social Media Mining* Menggunakan Algoritma *N-Gram-Multichannel CNN*” yang penulis lakukan di rumah hingga seminar via daring tetap dapat terselesaikan dengan baik dan tepat pada waktunya. Tugas Akhir ini adalah bagian penting dari rangkaian studi penulis, yakni sebagai salah satu syarat dalam menyelesaikan pendidikan penulis selama di jenjang sarjana Departemen Matematika ITS.

Perjalanan panjang telah penulis lalui dalam rangka perampungan penulisan skripsi ini. Banyak hambatan yang dihadapi dalam penyusunannya, namun berkat kehendak-Nyalah serta bantuan berupa dukungan moriil dan materiil yang telah diberikan oleh beberapa pihak sehingga penulis berhasil menyelesaikan penyusunan skripsi ini. Oleh karena itu, dengan penuh kerendahan hati, pada kesempatan ini patutlah kiranya penulis mengucapkan terima kasih kepada:

1. Kedua orang tua, papa Santoso Jiwo Leksono dan mama tercinta Gusti Ratu Rizkiah yang senantiasa memberikan kasih sayang dan dukungan moriil maupun materiil kepada penulis sekaligus menjadi penyemangat penulis untuk segera menyelesaikan Tugas Akhir,
2. Bapak Subchan, S.Si., M.Sc., Ph.D selaku Kepala Departemen Matematika ITS yang selalu mengontrol anak didiknya dan memberikan semangat,
3. Bapak Prof. Dr. Mohammad Isa Irawan, M.T. selaku dosen pembimbing yang juga selaku dosen wali penulis dimana beliau sudah meluangkan banyak sekali waktu untuk

membimbing penulis, bersedia meminjamkan kunci lab Ilkom kepada penulis untuk mengerjakan Tugas Akhir dan mengakses jurnal dengan mudah, bersedia menjawab pertanyaan penulis, selalu mengingatkan penulis untuk lebih baik lagi, mengarahkan penulis untuk terus berkembang dan terus belajar mempersiapkan diri untuk kehidupan pasca campus. Penulis mengucapkan banyak terima kasih atas untuk kebaikan beliau,

4. Bapak Bapak Dr. Dieky Adzkiya, S. Si., M. Si., Bapak Drs. Daryono Budi Utomo, M.Si., dan Bapak Drs. Bandung Arry S.,MI.Komp selaku dosen penguji Tugas Akhir penulis yang telah bersedia memberikan masukan, arahan, dan juga saran-saran yang sangat berguna bagi penulis,
5. Ibu Dr. Dwi Ratna S., MT selaku Sekretaris I Departemen Matematika ITS dan Bapak Dr. Budi Setiyono, MT. selaku Sekretaris Departemen Matematika ITS yang selama masa pengerjaan Tugas Akhir online dirumah ini sudah banyak membantu menjadwalkan dan menanggapi kebutuhan penulis,
6. Seluruh Bapak/Ibu dosen dan seluruh staf Departemen Matematika ITS yang sudah membimbing penulis selama perkuliahan 4 tahun,
7. Mbak Putri, Dek Jaka, dan Dek Rian selaku saudara kandung penulis yang selalu mendukung dan menjadi penyemangat untuk untuk penulis segera menyelesaikan Tugas Akhir ini,
8. Teman-teman Matematika ITS angkatan 2016, khususnya Zhafira Ardelia Irawan yang selalu memberikan bantuan apabila penulis ada kesulitan dalam berkas yudisium, perkuliahan dan sudah menjadi keluarga penulis selama kuliah,
9. Teman-teman rapat dan organisasi penulis, HIMATIKA ITS khususnya untuk SRD (2017-2018), Paduan Suara Mahasiswa

- ITS, ITS Badminton Community, Pemandu Matematika (Matriks),
10. Teman-teman Lab. Ilmu Komputer (Novia, Shafira, Islah, Alvaro, Rani, Ario, Fira, Soma, Fityan, Chandra, Fachri, dll) yang sudah bersama-sama mengerjakan Tugas Akhir di Lab dimana saling membantu dan selalu berbagi pengetahuan kepada penulis,
 11. Teman-teman satu dosen pembimbing dengan penulis (Yoyo, Tiara, dan Bella) yang sudah saling sharing dan membantu penulis,
 12. Dosen-dosen di Lab Ilmu Komputer yang sudah sangat baik memberikan bimbingan kepada penulis, penulis mengucapkan banyak terimakasih,
 13. Teman-teman, kakak-kakak di Komunitas Data Science Indonesia, Region Jawa Timur khususnya yang sudah memberikan banyak arahan dan dukungan kepada penulis,
 14. Semua pihak yang tidak dapat penulis sebutkan namanya satu per-satu, terima kasih sudah hadir dalam kehidupan penulis dan memberikan banyak sekali pelajaran untuk penulis.

Akhir kata, semoga Tugas Akhir ini dapat dijadikan sumber pembelajaran dan bermanfaat untuk semua, dan dapat dijadikan sebagai salah satu karya tulisan yang berguna dan dapat diteruskan untuk kemaslahatan. Penulis sangat terbuka akan diskusi dan kritik serta saran yang membangun.

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	Error! Bookmark not defined.
ABSTRAK	vii
<i>ABSTRACT</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR SIMBOL.....	xxiii
DAFTAR LAMPIRAN	xxv
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	6
1.3 Batasan Masalah.....	6
1.4 Tujuan.....	7
1.5 Manfaat.....	7
1.6 Sistematika Penulisan.....	8
BAB II.....	11
TINJAUAN PUSTAKA.....	11
2.1 Penelitian Terdahulu.....	11
2.2 <i>Social Media Mining</i>	13
2.3 <i>Natural Language Processing (NLP)</i>	14
2.4 Analisis Sentimen.....	14

2.5	<i>Convolutional Neural Network</i>	15
2.6	Metode <i>N-Gram</i>	19
2.7	<i>Word2Vec</i>	20
2.7	Evaluasi Performansi	21
BAB III.....		23
METODOLOGI PENELITIAN		23
3.1	Obyek dan Aspek Penelitian	23
3.2	Peralatan.....	23
3.3	Data Penelitian	24
3.4	Tahapan Penelitian.....	24
3.5	Diagram Alir Penelitian	31
BAB IV.....		33
PERANCANGAN DAN IMPLEMENTASI SISTEM		33
4.1	Analisis Sistem	33
4.1.1	<i>Use Case Diagram</i>	33
4.1.2	<i>Swimlane Diagram</i>	34
4.2	Analisis Kebutuhan Fungsional Sistem	35
4.3	Perancangan Data.....	36
4.3	Perancangan Pra-pemrosesan Data	43
4.3.1	<i>Load Data Opini dalam Python</i>	44
4.3.2	Menghapus Opini yang Berulang	44
4.3.3	Menghapus URL.....	45
4.3.4	Menghapus Tanda Baca, Simbol, dan Angka	47
4.3.5	Menghapus Huruf yang Berulang.....	48
4.3.6	<i>Case Folding</i>	48
4.3.7	Tokenisasi.....	49

4.3.8	<i>Spelling Normalization</i>	50
4.3.9	Filterisasi.....	53
4.4	Pelabelan Data.....	55
4.5	Ekstraksi Fitur.....	57
4.6	Desain Model <i>N-Gram-Multichannel CNN</i>	62
4.7	Implementasi Desain Antarmuka atau <i>Graphic User Interface (GUI)</i>	66
BAB V.....		69
UJI COBA DAN ANALISA PEMBAHASAN.....		69
5.1	Deskripsi Data Uji Coba.....	69
5.2	Proses Uji Coba.....	77
5.2.1	Hasil Uji Coba Impor Data.....	77
5.2.2	Hasil Uji Coba Praproses Data.....	78
5.2.3	Hasil Uji Coba Algoritma.....	79
BAB VI.....		89
KESIMPULAN DAN SARAN.....		89
6.1	Kesimpulan.....	89
6.2	Saran.....	90
DAFTAR PUSTAKA.....		91
LAMPIRAN A. Tabel Perbandingan Penelitian Terdahulu.....		95
LAMPIRAN B. <i>Corpus</i> Pengkoreksian Kata.....		97
LAMPIRAN C. <i>Corpus</i> Stopword.....		102
LAMPIRAN D. Proses Model <i>N-Gram-Multichannel CNN</i>		105
LAMPIRAN E. Visualisasi <i>Running Model</i>		110

DAFTAR GAMBAR

Gambar 2.1 Model arsitektur dengan 2 layer untuk sebuah contoh kalimat [12].....	15
Gambar 2.2 Contoh dari proses konvolusi [15].....	16
Gambar 2.3 Contoh dari proses <i>Pooling</i>	19
Gambar 2.4 Arsitektur Utama <i>Word2Vec</i> [23].....	21
Gambar 3.1 Implementasi Model.....	27
Gambar 3.2 Diagram Alir Metodologi Penelitian.....	31
Gambar 3.3 Perincian Diagram Alur Penelitian.....	32
Gambar 4.1 <i>Use Case Diagram</i> Sistem.....	34
Gambar 4.2 <i>Swimlane Diagram</i> Sistem.....	35
Gambar 4.3 Contoh Data Masukan.....	36
Gambar 4.4 <i>Syntax Crawling</i> Data Twitter.....	41
Gambar 4.5 <i>Syntax Crawling</i> Data Facebook.....	42
Gambar 4.6 <i>Code Load</i> Data Opini.....	44
Gambar 4.7 <i>Flowchart</i> Penghapusan Duplikasi Data Opini	45
Gambar 4.8 <i>Syntax</i> Penghapusan Duplikasi Data Opini.....	45
Gambar 4.9 <i>Syntax</i> Penghapusan URL.....	46
Gambar 4.10 <i>Syntax</i> Penghapusan Tanda Baca, Simbol, dan Angka.....	47
Gambar 4.11 <i>Syntax</i> Penghapusan Huruf Berulang.....	48
Gambar 4.12 <i>Syntax</i> Case Folding.....	49
Gambar 4.13 <i>Syntax</i> Tokenisasi.....	50
Gambar 4.14 <i>Syntax</i> Spelling Normalization.....	51
Gambar 4.15 Potongan <i>corpus</i>	52
Gambar 4.16 <i>Flowchart</i> Spelling Normalization.....	52
Gambar 4.17 <i>Flowchart</i> Filterisasi.....	54
Gambar 4.18 <i>Syntax</i> Filterisasi.....	54
Gambar 4.19 <i>Flowchart</i> Anotasi Label Opini.....	56

Gambar 4.20	Inisialisasi Parameter Ekstraksi Fitur <i>Word2Vec</i>	60
Gambar 4.21	<i>Syntax</i> Model training Ekstraksi Fitur <i>Word2Vec</i>	61
Gambar 4.22	<i>Array</i> Vektor dari data <i>dummy</i> kalimat opini..	62
Gambar 4.23	<i>Syntax</i> Model <i>N-Gram-Multichannel CNN</i>	64
Gambar 4.24	Desain Model Analisis Sentimen	65
Gambar 4.25	<i>Interface Design</i> Analisis Sentimen	67
Gambar 5.1	<i>Dataframe</i> Opini Nama Bakal Calon Walikota Surabaya 2020.....	70
Gambar 5.2	<i>Dataframe</i> Opini Negatif Nama Bakal Calon Walikota Surabaya 2020.....	71
Gambar 5.3	<i>Dataframe</i> Opini Positif Nama Bakal Calon Walikota Surabaya 2020.....	71
Gambar 5.4	Visualisasi Data Jumlah Opini Nama Bakal Calon Walikota Surabaya 2020.....	72
Gambar 5.5	Wordcloud <i>Visualization Data</i>	73
Gambar 5.6	Potongan data hasil <i>crawling</i> format CSV.....	75
Gambar 5.7	<i>Record Data</i> setiap Minggu.....	76
Gambar 5.8	<i>Dataframe</i> dari Data Tweet.....	78
Gambar 5.9	<i>List</i> dari Data Tweet.....	78
Gambar 5.10	<i>List</i> Hasil Praproses Data.....	79
Gambar 5.11	Hasil Ringkasan Model.....	80
Gambar 5.12	Grafik pergerakan akurasi model pada percobaan pertama.....	82
Gambar 5.13	Grafik pergerakan <i>loss</i> pada percobaan pertama	83
Gambar 5.14	Akurasi pada data latih dan data validasi ke-1..	83
Gambar 5.15	Grafik pergerakan akurasi model pada percobaan kedua.....	86
Gambar 5.16	Grafik pergerakan <i>loss</i> pada percobaan kedua..	87
Gambar 5.17	Akurasi pada data latih dan data validasi ke-2..	87

DAFTAR TABEL

Tabel 3.1	Spesifikasi Perangkat	23
Tabel 4.1	Tabel Data Proses.....	37
Tabel 4.2	Kode API Twitter	39
Tabel 4.3	Kode Token <i>Graph API</i> Facebook.....	40
Tabel 4.4	Data <i>Dummy</i>	43
Tabel 4.5	Hasil Penghapusan URL Data <i>Dummy</i>	46
Tabel 4.6	Hasil Penghapusan Tanda Baca, Simbol, dan Angka	47
Tabel 4.7	Hasil Penghapusan Huruf Berulang	48
Tabel 4.8	Hasil <i>Case Folding</i>	49
Tabel 4.9	Hasil Tokenisasi	50
Tabel 4.10	Hasil <i>Spelling Normalization</i>	53
Tabel 4.11	Hasil Filterisasi Menghilangkan <i>Stopwords</i>	55
Tabel 4.12	Hasil Pelabelan Data <i>Dummy</i> oleh Anotator.....	57
Tabel 5.1	Rincian Jumlah Opini Masing-masing <i>keyword</i>	69
Tabel 5.2	Rincian Jumlah Opini Positif dan Negatif.....	74
Tabel 5.3	Jumlah Data <i>Tweet</i> per-Minggu	75
Tabel 5.4	Data Kalimat Opini	77
Tabel 5.5	Hasil Akurasi Sistem.....	81
Tabel 5.6	Hasil Akurasi Sistem <i>train</i> kedua.....	85

DAFTAR SIMBOL

No	Simbol	Nama	Keterangan
1	\oplus	<i>Join Operator</i>	Merepresentasikan kombinasi hasil dari vektor kata selang indeks tertentu
2	\in	<i>Set Membership</i>	Merepresentasikan elemen atau anggota dari sebuah himpunan
3	\mathbb{R}	Bilangan real	Merepresentasikan bilangan yang bisa dituliskan dalam bentuk desimal. Bilangan real meliputi bilangan rasional dan bilangan irasional.

DAFTAR LAMPIRAN

LAMPIRAN A.	Tabel Perbandingan Penelitian Terdahulu.....	93
LAMPIRAN B.	<i>Corpus</i> Pengkoreksian Kata.....	95
LAMPIRAN C.	<i>Corpus Stopword</i>	100
LAMPIRAN D.	Proses Model <i>N-Gram-Multichannel CNN</i> ..	104
LAMPIRAN E.	Visualisasi <i>Running Model</i>	109

BAB I PENDAHULUAN

Bab I ini diuraikan mengenai latar belakang masalah Tugas Akhir ini, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan laporan Tugas Akhir ini.

1.1 Latar Belakang

Indonesia adalah salah satu negara yang menganut sistem demokrasi¹. Hal ini ditandai dengan diadakannya suatu pemilihan umum terhadap Walikota dan Wakil Walikota salah satunya di kota Surabaya. Pemilihan umum pada suatu negara yang menganut sistem demokrasi diselenggarakan secara periodik. Komisi Pemilihan Umum (KPU) RI telah menetapkan jadwal Pemilihan Wali Kota (Pilwali) Surabaya yang akan datang. Bahkan pemilihan kepala daerah (Pilkada) itu akan dilaksanakan secara serentak di seluruh Indonesia². Seorang tokoh politik yang ingin maju sebagai bakal calon Walikota Surabaya akan melihat atau mempertimbangkan popularitas mereka berdasarkan opini masyarakat. Dahulu masyarakat mengungkapkan opini, kritik, dan sarannya melalui media cetak yang tidak semua orang mempunyai kemampuan menulis dan kesempatan menerbitkan tulisannya. Namun, perkembangan teknologi komunikasi saat ini telah merubah kecenderungan kebiasaan masyarakat dalam

¹Welianto, Ari. 2019. **Sistem Demokrasi di Indonesia**. Kompas.com [Online]. Tersedia: <https://www.kompas.com/skola/read/2019/12/11/201742369/sistem-demokrasi-di-indonesia> (diakses 1 Januari 2020).

²Yohanes, Erwin. 2019. **KPU Tetapkan Jadwal Pemilihan Wali Kota Surabaya pada September 2020**. Merdeka.com. [Online]. Tersedia: <https://www.merdeka.com/politik/kpu-tetapkan-jadwal-pemilihan-wali-kota-surabaya-pada-september-2019.html> (diakses 1 Januari 2020).

mengekspresikan opininya pada jejaring sosial. Salah satu jejaring sosial yang populer di kalangan pengguna internet saat ini adalah *Facebook* dan *Twitter* [1]. Suatu proses untuk merepresentasikan, menganalisis, dan mengekstraksi pola data bersumber dari media sosial disebut *social media mining* [2].

Pada masa yang serba digital seperti sekarang ini, pelanggan dapat menyampaikan opini melalui berbagai media sosial. Media sosial seperti *Blogger*, *Twitter*, *Facebook*, *Reddit*, dan juga situs ulasan secara daring banyak digunakan karena akses yang mudah [3]. Berdasarkan laporan *Hootsuite (We are Social)* merilis update statistik pengguna media sosial aktif di dunia tahun 2020 sebanyak 3,80 miliar³. Selanjutnya di Indonesia *Wearesocial Hootsuite* merilis pada Januari 2020 pengguna media sosial di Indonesia mencapai 160 juta atau sebesar 59% dari total populasi. Jumlah tersebut naik 23% dari survei sebelumnya. Sementara pengguna media sosial Facebook mencapai 130 juta pengguna dan Twitter mencapai 10.65 juta⁴. Di kota Surabaya sendiri masyarakat pengguna media sosial juga terbilang cukup besar yakni 68,2% masyarakat Surabaya pengguna Twitter dan 68,2% pula masyarakat Surabaya pengguna Facebook⁵. *Twitter* adalah sebuah media sosial berbasis microblogging yang memungkinkan pengguna dapat mengirim posting (tulisan) untuk publik atau tweet, mengirimkan posting kepada orang tertentu (mention), dan

³Riyanto, Andi Dwi. 2019. *Data Statistik Digital dan Pengguna Internet di Dunia tahun 2019 Kuartal Kedua (Q2)* [Online]. Tersedia: <https://andi.link/data-statistik-digital-dan-pengguna-internet-di-dunia-tahun-2019-kuartal-kedua-q2/> (diakses 1 Januari 2020).

⁴We Are Social (Hootsuite) 2020. **Digital 2020: Indonesia**. [Online]. Tersedia: <https://datareportal.com/reports/digital-2020-indonesia> (diakses 27 Juni 2020).

⁵Amalia Pranata.2014. **Survei Pengguna Sosial Network Sites di Kota Surabaya**. [Online]. Tersedia: <https://visual.ly/community/Infographics/social-media/surabaya-digital-native-ber-jejaring-sosial-lewat-smartphone> (diakses 22 April 2020).

membaca sebuah pesan teks serta gambar dalam 280 karakter. Saat ini, jumlah posting yang dikirim oleh pengguna adalah mencapai 500 juta posting per hari atau sebanding dengan sekitar 6000 posting setiap detik⁶. Sedangkan *Facebook* menjadi salah satu media sosial yang digemari oleh masyarakat Indonesia yang disebut sebagai negara dengan pengguna sekaligus target audiens iklan Facebook terbesar di dunia, dengan jumlah 130 juta pengguna aktif bulanan *Facebook*⁷ [4].

Tahun 2020 merupakan periode akhir masa jabatan Walikota Surabaya Tri Rismaharini yang sudah menjabat Wali Kota selama dua periode sejak 28 September 2010, maka selanjutnya perlu diadakan pemilihan walikota baru yang diselenggarakan oleh Komisi Pemilihan Umum (KPU) Kota Surabaya yang dilaksanakan pada 23 September 2020⁸. Kemeriahan Pilwali Surabaya 2020 sudah dirasakan di media sosial khususnya *Twitter* dan *Facebook* yang sekarang ini menjadi tempat yang sangat penting untuk calon dan tim suksesnya melakukan kampanye. Hal ini memicu berbagai opini di *Twitter*. Opini publik memiliki peran penting dalam menyukseskan bakal calon dalam pemilu, seperti yang disampaikan oleh Freddy [5], bahwa opini publik dapat memberikan pengaruh kepada bakal calon dalam menentukan sikap. Opini ini dapat dimanfaatkan untuk melihat bagaimana polaritas tokoh politik yang akan maju sebagai bakal calon

⁶Wikipedia Ensiklopedia Bebas. *Twitter*. 2019. [Online]. Tersedia: <https://id.wikipedia.org/w/index.php?title=Twitter&stable=1> (diakses 1 Januari 2020).

⁷Pertiwi, Wahyunanda Kusuma. 2019. “**Facebook Jadi Medsos Paling Digemari di Indonesia**,” [Online] <https://tekno.kompas.com/read/2019/02/05/11080097/facebook-jadi-medsos-paling-digemari-di-indonesia?page=all> (diakses 1 Januari 2020)

⁸Pemilihan Umum Walikota Surabaya 2020,” [Online], Wikipedia, January 29, 2020. https://id.wikipedia.org/wiki/Pemilihan_umum_Wali_Kota_Surabaya_2020 (diakses 1 Februari 2020)

Walikota dan Wakil Walikota Surabaya tahun 2021. Penentuan polaritas positif atau negatifnya suatu opini dapat dilakukan secara manual, tetapi seiring bertambahnya sumber opini yang semakin banyak, tentunya waktu dan usaha yang dibutuhkan untuk mengklasifikasikan polaritas opini tersebut akan semakin banyak [1]. Besarnya polaritas yang ditujukan kepada suatu bakal calon bisa dijadikan sebuah parameter kemenangan atau kekalahan bakal calon tersebut [6]. Pada penelitian ini analisis sentimen dilakukan untuk melihat dan mengambil informasi berupa opini seseorang dalam bahasa Indonesia di Twitter dan Facebook yang ditujukan kepada bakal calon Walikota dan Wakil Walikota Surabaya periode 2021-2024, apakah opini itu masuk kategori opini positif atau negatif. Penelitian ini menggunakan metode *text mining* dan *deep learning* untuk mengambil dan mengklasifikasi polaritas opini dari sumber data.

Penelitian terdahulu mengenai Analisis Sentimen masyarakat terhadap Pilwali Surabaya yang sudah berjalan 4 kali ini tidak ditemukan atau belum pernah dilakukan, namun penelitian terkait mengenai topik Analisis Sentimen pada calon-calon pemimpin di Indonesia, referensi metode serta literatur yang berkaitan dengan penelitian Tugas Akhir ini diantaranya dapat ditemukan pada 2 obyek penelitian berbeda yang dilakukan oleh Ghulam dengan akurasi 77% pada sentimen Calon Gubernur Jatim 2018 menggunakan *Naïve Bayes* [7] yang topiknya sama dengan Sari Widih, dkk serta obyek lain yakni sentimen calon Gubernur DKI Jakarta 2017 dengan nilai rata-rata akurasi mencapai 95%, pernah dilakukan *research* mengenai *Facebook Analysis* dengan objek penelitian pemilihan Presiden oleh Budi Haryanto, dkk pada 2019, ada pula yang meneliti sentimen paslon Pilpres setelah debat dengan rata-rata akurasi 45% menggunakan data *Twitter* [8], selain itu dengan metode SVM oleh Tata, dkk diperoleh akurasi 86% [9], jurnal dari Zulfadzli juga menjelaskan bagaimana sentimen analisis sangat berperan penting untuk diaplikasikan pada obyek kesehatan, politik, dan bisnis [10].

Untuk lebih jelas mengenai referensi penelitian sebelumnya yang berkaitan dengan topik penulis dapat dilihat pada Lampiran A. Berdasarkan beberapa referensi penelitian tersebut belum ada yang menganalisis sentimen opini masyarakat terhadap pasangan calon dengan obyek yang sedang ramai diperbincangkan rakyat Surabaya yakni Pilwali Surabaya 2020. Selain itu, jumlah yang sangat besar data melalui Facebook dan Twitter yang dihasilkan melalui Internet, dan menghasilkan sejumlah data yang besar dan informasi tag atau yang dihasilkan oleh berita online dan Twitter sesuai dengan materi pembelajaran yang baik untuk *Deep Learning System* [11]. Analisis sentimen sendiri bisa dianggap sebagai kombinasi dari *text mining* dan *natural language processing*. Salah satu metode dari *text mining* yang bisa digunakan untuk menyelesaikan masalah Analisis sentiment adalah algoritma *Convolutional Neural Network (CNN)*. Pada umumnya CNN digunakan pada pengolahan citra digital sebagai alat untuk klasifikasi maupun kluster. Dengan inovasi dari Kim pada tahun 2014 yaitu penerapan model CNN pada NLP khususnya dalam klasifikasi kalimat [12].

Kim mengusulkan konsep baru dalam penggunaan CNN pada pengolahan teks yang menunjukkan bahwa CNN merupakan metode yang unggul dalam pengolahan teks. Terbukti dengan adanya penelitian tugas akhir yang dilakukan oleh M. Fakhru Rozi didapatkan performansi sebesar 83.23% untuk data latih dan sebesar 64.4% untuk data uji [13] yang menggunakan metode CNN-L2-SVM untuk analisis sentimen pada ulasan buku. Terdapat akurasi data uji sebesar 63%, presisi data uji sebesar 63% dan recall data uji sebesar 50% menggunakan DCNN-SVM yang dilakukan Inayah Eka untuk menganalisa sentiment operator pelanggan [14], serta penelitian yang dilakukan oleh Imam Mukhlash dkk [15] yaitu analisis sentimen pada ulasan buku dengan menggunakan metode CNN-LSTM didapatkan performansi 99.55% untuk data latih dan sebesar 65.03% untuk data uji. Oleh karena itu, pada penelitian ini digunakan sebuah

kombinasi Algoritma *N-Gram-Multichannel CNN* dalam pengklasifikasian suatu teks. Metode ini mengklasifikasikan suatu opini menjadi dua kelas utama yaitu positif dan negatif dimana akan dibangun suatu sistem dengan metode analisis sentimen dengan obyek yaitu opini masyarakat terhadap bakal calon Walikota Surabaya pada media sosial Facebook dan Twitter dimana menghasilkan akurasi paling optimal yang dapat digunakan sebagai bahan pertimbangan dalam penelitian selanjutnya.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut di atas, adapun rumusan masalah yang dapat diambil pada penelitian Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana menerapkan analisis sentimen pada teks berisikan opini masyarakat terhadap bakal calon Walikota Surabaya 2020 berdasarkan *Social Media Mining* menggunakan Algoritma *N-Gram-Multichannel CNN*?
2. Bagaimana hasil akurasi model terhadap data latih dan data uji dataset opini masyarakat berdasarkan *Social Media Mining* terhadap bakal calon Walikota Surabaya 2020 dalam menganalisis sentimen menggunakan Algoritma *N-Gram-Multichannel CNN*?

1.3 Batasan Masalah

Adapun batasan masalah yang digunakan dalam penelitian Tugas Akhir ini adalah sebagai berikut:

1. Data yang digunakan adalah data tweet dan postingan status facebook dengan kata kunci 'pilwali Surabaya', 'whisnu sakti buana', 'dyah katarina', 'gamal albinsaid', 'eri cahyadi', 'machfud arifin', 'muhammad soleh', 'machfud arifin', 'achmad zakaria', 'sigit sosiantomo', 'achmad suyanto',

‘ahmad jabir’, ‘reni astuti’, ‘sigit sosiantomo’, ‘ahmad dhani’, ‘eri cahyadi’, ‘fandi eko utomo’, ‘untung suropati’, ‘arif afandi’, ‘anwar sadad’, ‘toni ‘tamatompol’ berjumlah 2000 tweet dari tanggal 1 Februari 2020 sampai dengan tanggal 20 April 2020.

2. Data postingan Facebook dan *tweet* dari Twitter yang digunakan merupakan kalimat teks dalam bahasa Indonesia.
3. Bahasa pemrograman yang digunakan adalah Python.

1.4 Tujuan

Tujuan dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. Menerapkan analisis sentimen pada teks berisikan opini masyarakat terhadap bakal calon Walikota Surabaya 2020 berdasarkan *Social Media Mining* menggunakan Algoritma *N-Gram-Multichannel CNN*.
2. Mendapatkan hasil akurasi model terhadap data latih dan data uji dataset opini masyarakat berdasarkan *Social Media Mining* terhadap bakal calon Walikota Surabaya 2020 dalam menganalisis sentimen menggunakan Algoritma *N-Gram-Multichannel CNN*.

1.5 Manfaat

Adapun manfaat yang dapat diambil dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. Dalam bidang akademik, diharapkan dapat mengimplementasikan *deep learning* dalam NLP (*Natural Language Processing*) dan mendapatkan analisis sentimen.
2. Mengetahui analisis sentimen dari sekumpulan dataset posting dan tanggapan sosial media menggunakan Algoritma *N-Gram-Multichannel CNN* serta mengetahui nilai akurasi.

3. Sebagai sumber informasi bakal calon terpilih untuk mengetahui sentimen masyarakat yang nantinya akan berpartisipasi dalam Pilwali Surabaya
4. Sebagai literatur penunjang bagi penelitian selanjutnya.

1.6 Sistematika Penulisan

Adapun sistematika penulisan pada laporan Tugas Akhir ini adalah sebagai berikut:

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.

BAB II. TINJAUAN PUSTAKA

Pada bab ini dijelaskan beberapa teori dasar yang mendukung dalam pengerjaan penelitian ini yang meliputi penelitian terdahulu, landasan teori dari *Social Media Mining*, Analisis Sentimen, *Natural Language Processing*, *Deep Learning*, *Deep Convolutional Neural Network*, *N-Gram* dan Evaluasi Performansi.

BAB III. METODOLOGI PENELITIAN

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

BAB IV. PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini dijelaskan perancangan dan implementasi sistem yang dikerjakan pada penelitian Tugas Akhir ini dimana menjelaskan tentang model dan desain dari sistem yang akan dibentuk. Hal-hal tersebut meliputi, *crawling data*, tahap pra-pemrosesan data, data transformasi dengan *Word2Vec*, pembuatan model *N-Gram*-

Multichannel CNN, serta visualisasi data sebagai acuan dalam mengimplementasikan model.

BAB V. UJI COBA DAN ANALISA PEMBAHASAN

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

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.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

Pada bab ini dipaparkan penelitian terdahulu yang berkaitan dengan penelitian Tugas Akhir ini, referensi, serta berbagai literatur yang digunakan dalam penelitian Tugas Akhir ini.

2.1 Penelitian Terdahulu

Beberapa penelitian yang terkait dengan topik yang akan dibahas dalam penelitian tugas akhir ini yaitu penelitian yang dilakukan oleh Widodo Budiharto dan Meiliana dengan mengambil data opini Twitter dari bulan Maret 2018 sampai dengan bulan Juli 2018 dengan menggunakan data *training* sejumlah 250 tweet dan 100 tweet data *testing*. Algoritma yang digunakan diinisiasi sendiri dengan menggunakan Bahasa pemrograman R dan *textblob* untuk menentukan sentiment polaritas opini. Pada penelitian ini *keyword* pencarian data tweet yaitu yang bertagar calon presiden RI, yang awalnya fokus ke penghitungan tweet masing-masing calon presiden kemudian baru mencari Analisa sentimen yang hasil prediksinya sesuai dengan prediksi dari empat lembaga survey Indonesia antara lain Indikator, Cyrus Network, LitbangKompas, dan Poltracking dengan rata-rata akurasi prediksinya 45% [16]. Kemudian ada pula penelitian yang dipublikasikan melalui jurnal *IOP Conferences* oleh Ariesta Lestari dan Devi Karolita dengan melakukan prediksi model menggunakan metode CART (Classification and Regression Tree) diperoleh hasil akurasinya sebesar 90% [17]. Ghulam Asrofi dengan dua obyek penelitian berbeda yang dilakukan oleh Ghulam dengan akurasi 77% pada sentimen Calon Gubernur Jatim 2018 menggunakan Naïve Bayes [7] yang topiknya sama dengan Sari Widih, dkk serta obyek lain yakni sentimen calon Gubernur DKI Jakarta 2017 dengan nilai

rata-rata akurasi mencapai 95%. Selanjutnya penelitian yang dilakukan M. Fakhrur Rozi didapatkan fakta bahwa pada penelitian ini dapat dilihat bahwa keakuratan model dengan data pelatihan sangat baik. Baru 20 kali iterasi, akurasinya dapat mencapai 83,23% dan akan meningkat untuk iterasi berikutnya. Di samping itu, untuk mendapatkan hasil itu, dibutuhkan waktu lama sekitar 76 menit. Sementara itu, akurasi model dengan data pengujian memiliki hasil yang lebih rendah daripada data pelatihan. Itu terjadi karena tidak ada pengawasan hasil untuk klasifikasi dalam pengujian data. penelitian ini menggambarkan akurasi data pelatihan dalam 20 kali iterasi. Kita dapat melihat bahwa akurasi data pengujian berfluktuasi jika kita bandingkan dengan hasilnya data pelatihan. Keakuratan model dengan data pengujian menghasilkan hasil akhir 61,94%. Kita dapat melihat bahwa keakuratan data pengujian menurun dari 66% hingga 60-61% pada iterasi terakhir. Jika kita lihat hasil pengujian dan fase pelatihan yang memiliki grafik bertentangan. Ini berarti bahwa model memiliki *overfitting*. *Overfitting* adalah peristiwa di mana model terlalu bagus dalam data pelatihan tetapi terlalu buruk dalam menguji data. Masalah ini umum dalam klasifikasi dengan jaringan saraf tiruan. [13] yang menggunakan metode CNN-L2-SVM untuk analisis sentimen pada ulasan buku. Terdapat akurasi data uji sebesar 63%, presisi data uji sebesar 63% dan recall data uji sebesar 50% menggunakan DCNN-SVM yang dilakukan Inayah Eka untuk menganalisa sentimen operator pelanggan [14], serta penelitian yang dilakukan oleh Imam Mukhlash dkk yaitu analisis sentimen pada ulasan buku dengan menggunakan metode CNN-LSTM didapatkan performansi 99.55% untuk data latih dan sebesar 65.03% untuk data uji. Perbandingan antara pergerakan data latih dan data uji terlihat sangat jauh berbeda. Ini menunjukkan model *overfitting* [15]. Selanjutnya terdapat penelitian yang dilakukan oleh Digna Tata, dkk yang melakukan

Analisis sentimen terhadap calon presiden 2019 dengan metode Support Vector Machine dengan mengambil dataset tweet sebanyak 20.000 tweet menghasilkan akurasi sebesar 86% [9]. Beralih dari twitter, terdapat penelitian yang dilakukan oleh Budi Haryanto, dkk dimana saat itu yang menjadi obyek penelitian adalah dua paslon Jokowi-Maruf dan Prabowo-Sandi dengan obyek data dari Facebook menghasilkan Jokowi-Maruf mendominasi sentimen positif sebesar 56.76% dan 43.24% sentimen negatif, sedangkan paslon Prabowo-Sandi mendapatkan perolehan 24.24% sentimen positif dan 75.79% sentimen negatif [4]. Adapun perbandingan pada masing-masing penelitian terdahulu secara ringkas dapat diperhatikan pada Lampiran A.

2.2 Social Media Mining

Seperti yang didefinisikan oleh Kaplan dan Haenlein [18], media sosial adalah "kelompok aplikasi berbasis internet yang dibangun berdasarkan ideologis dan teknologi dasar-dasar Web 2.0, dan itu memungkinkan adanya pembuatan dan pertukaran konten yang dibuat pengguna." Ada banyak kategori media sosial, *social networking* (Facebook atau LinkedIn), *microblogging* (Twitter), berbagi foto (Flickr, Photobucket, atau Picasa), agregasi berita (pembaca Google, StumbleUpon, atau Feedburner), video berbagi (YouTube, MetaCafe), *livecasting* (Ustream atau Justin.TV), virtual dunia (Kaneva), *social gaming* (World of Warcraft), *social search* (Google, Bing, atau Ask.com), dan *instant messaging* (Google Talk, Skype, atau Yahoo! Messenger). *Social Media Mining* adalah proses merepresentasikan, menganalisis, dan mengekstraksi pola yang mengandung makna berarti dari data di media sosial, yang dihasilkan dari interaksi sosial. *Social Media Mining* menghadapi tantangan besar seperti paradoks data besar, mendapatkan sampel yang cukup, *fallacy removal noise*, dan dilema evaluasi. *Social Media Mining* memperkenalkan konsep dasar dan algoritma

utama yang cocok untuk menyelidiki data dari media sosial dengan jumlah yang besar. Konsep ini membahas teori dan metodologi dari berbagai disiplin ilmu seperti ilmu komputer, *data mining*, *machine learning*, *social network analysis*, *network science*, sosiologi, etnografi, statistik, optimisasi, dan matematika. *Social Media Mining* mewakili dunia virtual media sosial dalam suatu cara yang dapat dihitung, dapat diukur, dan dirancang model yang dapat membantu kita memahami interaksinya [2].

2.3 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah bidang khusus dalam ilmu komputer dan kecerdasan buatan yang berakar pada komputasi linguistik. Hal ini terutama berkaitan dengan perancangan dan pembangunan aplikasi dan sistem yang memungkinkan interaksi antara mesin dan bahasa alami yang dikembangkan oleh manusia. Teknik NLP memungkinkan komputer untuk memproses dan memahami bahasa alami manusia dan menggunakannya lebih lanjut untuk diambil manfaatnya. Dalam prosesnya NLP memiliki berbagai tahapan untuk pengolahannya. Beberapa diantaranya adalah tokenisasi, filtrasi, *stemming*, dan normalisasi [19].

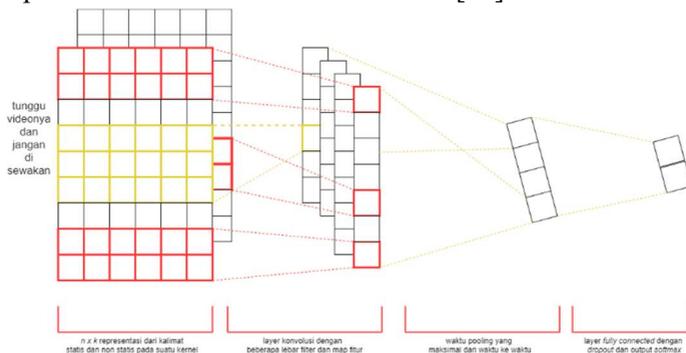
2.4 Analisis Sentimen

Analisis sentimen adalah disiplin ilmu yang mengekstraksi perasaan, pendapat, pikiran, dan perilaku orang-orang dari data teks pengguna menggunakan metode *Natural Language Processing* (NLP) [20]. 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 [21]. Metode berbasis

machine learning dibagi menjadi tiga yaitu *unsupervised learning*, *supervised learning*, dan *semi-supervised learning* [21]. Pada *supervised learning* terdapat beberapa algoritma klasifikasi seperti SVM, *Naïve Bayes*, dan *Neural Network*.

2.5 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang di desain untuk memproses data dua dimensi. CNN masuk dalam tipe *Deep Neural Network* disebabkan oleh jaringan dengan intensitas yang tinggi dan diaplikasikan pada banyak citra data. Algoritma CNN mempunyai kemiripan dengan MLP, tetapi setiap neuron di CNN ditampilkan dalam bentuk dua dimensi, tidak seperti di MLP setiap neuron dalam bentuk satu dimensi [15].



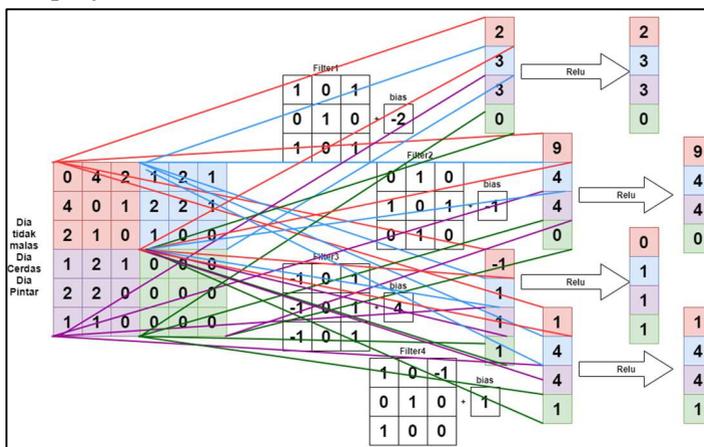
Gambar 2.1 Model arsitektur dengan 2 layer untuk sebuah contoh kalimat [12]

Ada dua layer utama dalam struktur algoritma CNN: layer konvolusi dan layer *pooling*. Dalam analisis sentimen, digunakan dua layer utama yang diilustrasikan pada Gambar 2.1.

1. Convolutional Layer

Layer pertama pada arsitektur jaringan CNN menunjukkan sebuah proses konvolusi pada semua vector dari kata yang terdapat pada opini. Sebuah proses

konvolusi dilakukan untuk mendapatkan peta fitur dari data masukan awal. Lapisan konvolusi terdiri dari neuron yang diatur sedemikian rupa sehingga membentuk filter dengan panjang dan tinggi tertentu. Misalnya, akan digunakan korpus "Dia tidak malas. Dia cerdas. Dia pintar" yang dapat dilihat pada Gambar 2.2. Data input adalah 6×6 dan memiliki 4 filter. Keempat filter ini akan dialihkan ke semua bagian input data dan kemudian dilakukan operasi *dot product* antara input dan nilai filter pada setiap shift sehingga hasil operasi akan dioperasikan menggunakan fungsi aktivasi. Output dari fungsi aktivasi adalah peta fitur. Tidak ada aturan spesifik di filter dan pemilihan nilai bias. Nilai-nilai ini digunakan dalam contoh, adalah yang umumnya digunakan dalam penelitian sebelumnya. Perhatikan penjelasan dari Gambar 2.2 berikut:



Gambar 2.2 Contoh dari proses konvolusi [15]

Asumsikan frasa pada opini yang memiliki panjang n vektor kata memiliki dimensi 50 sebagai input dari model. Untuk contoh, misalkan $x_i \in \mathbb{R}^{50}$ dimana x_i

merepresentasikan vektor kata dari indeks ke- i . Ketika vektor kata digabungkan, maka ulasan bisa direpresentasikan sebagai berikut:

$$x_{i:n} = x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n \quad (1)$$

Dimana \oplus adalah *join operator*. Untuk Join Operator dapat dilihat lebih jelas pada Lampiran D. Secara umum $x_{i:i+j}$ merepresentasikan kombinasi hasil dari vektor kata dari indeks ke- i dan indeks ke- j sebagai urutan berikut:

$$x_i, x_{i+1}, x_{i+2}, \dots, x_{i+j} \quad (2)$$

Operasi konvolusi menggunakan filter atau parameter yang dimisalkan dengan w menjadi $w \in \mathbb{R}^{50h}$ dimana h merepresentasikan ukuran *sliding window*. Kemudian gunakan persamaan di bawah untuk mendapatkan nilai fitur:

$$c_i = f(\text{net}) \quad (3)$$

$$\text{net} = w \cdot x_{i:i+h-1} + b \quad (4)$$

Dimana $b \in \mathbb{R}$ adalah parameter bias dan $c_i \in \mathbb{R}$ adalah nilai fitur pada indeks ke- i dan f adalah sebuah fungsi aktivasi. Dalam kasus ini digunakan *Rectified Linear Unit* (ReLU) sebagai fungsi nonlinear. Fungsi ini menghasilkan batas atas positif, yang dapat ditulis sebagai berikut:

$$\text{ReLU}(x) = \max(0, x) \quad (5)$$

Sehingga persamaan menjadi

$$c_i = \text{ReLU}(\text{net}) \quad (6)$$

$$\text{net} = w \cdot x_{i:i+h-1} + b \quad (7)$$

Filter w digunakan untuk setiap jendela kata yang mungkin dari opini $\{x_{1:h}, x_{2:h}, x_{3:h}, \dots, x_{n-h+1:n}\}$ hingga kita mendapatkan peta fitur sebagai berikut:

$$c = [c_1, c_2, c_3, \dots, x_{n-h+1}] \quad (8)$$

Dimana $c \in \mathbb{R}^{n-h+1}$. Untuk mendapatkan hasil yang maksimal, akan digunakan lebih dari satu filter dan lebih dari satu kata jendela.

Keterangan:

- x_i = vektor kata indeks ke- i
- x_j = vektor kata indeks ke- j
- \mathbb{R}^{50} = input model berupa vektor pada bidang atau ruang dimensi 50
- $x_{i:n}$ = gabungan vektor kata
- w = filter atau parameter
- h = ukuran *sliding window*
- c = nilai peta fitur/ *feature map*
- b = parameter bias
- f = fungsi aktivasi

2. *Pooling Layer*

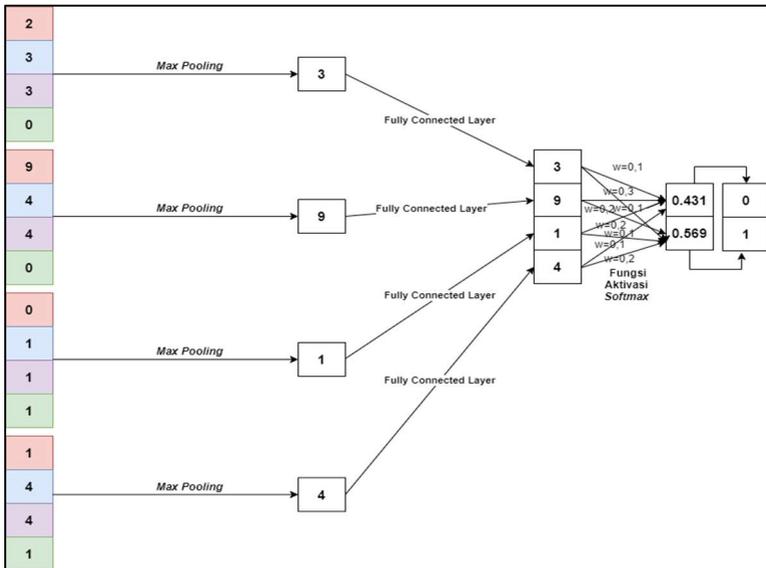
Pada Layer ini, akan dilakukan operasi penggabungan pada masing-masing peta fitur yang telah dihasilkan dari layer konvolusi. Ini akan menangkap nilai maksimal setiap peta fitur untuk memilih nilai yang dirasa penting untuk peta fitur. Berikut merupakan *pooling operation formula*:

$$\hat{c} = \max\{c\} \quad (9)$$

Dimana \hat{c} adalah nilai maksimal dari peta fitur c yang sesuai ke filter. Karena ada m filter, hasil dari *pooling layer* adalah vector dari nilai maksimal untuk setiap peta fitur. Vektor tersebut berisi m entri. Dengan demikian, diperoleh

$$z = [\hat{c}_1, \hat{c}_2, \hat{c}_3, \dots, \hat{c}_m] \quad (10)$$

Dimana z adalah jumlahan vector dari *pooling layer* yang akan digunakan pada step *subsequent*. Proses *pooling* hingga klasifikasi diilustrasikan pada Gambar 2.3 berikut:



Gambar 2.3 Contoh dari proses *Pooling*

2.6 Metode *N-Gram*

N-gram merupakan potongan sejumlah n kata dari sebuah kalimat. N-gram merupakan metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode ini digunakan untuk mengambil potongan-potongan kata sejumlah n dari sebuah kalimat yang secara kontinuitas dari teks sumber hingga akhir dokumen. N-gram dibedakan berdasarkan jumlah potongan karakter sebesar n . Untuk membantu dalam mengambil potongan-potongan kata berupa karakter huruf tersebut, maka dilakukan *padding* dengan blank diawal dan diakhir suatu kata

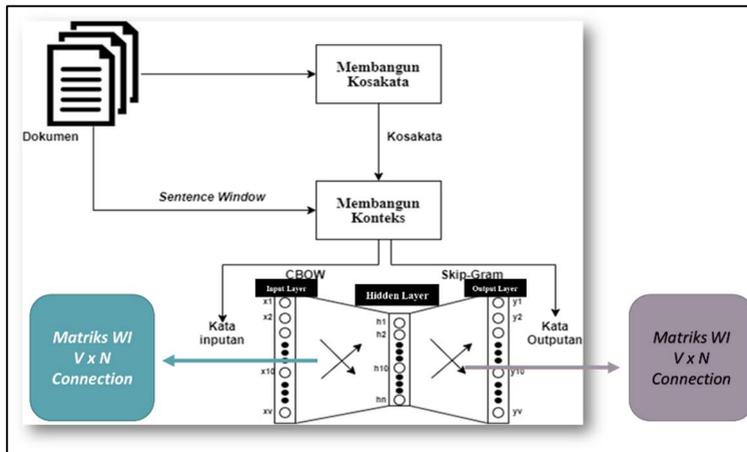
[22]. Berikut ilustrasi penggunaan N-Gram pada kalimat “aku suka banget produk *cleanser* ini”:

1. Unigram : ‘aku’, ‘suka’, ‘banget’, ‘produk’, ‘*cleanser*’, ‘ini’.
2. Bigram : ‘aku suka’, ‘suka banget’, ‘banget produk’, ‘produk *cleanser*’, ‘*cleanser* ini’.
3. Kombinasi : ‘aku’, ‘suka’, ‘banget’, ‘produk’, ‘*cleanser*’, ‘ini’, ‘aku suka’, ‘suka banget’, ‘banget produk’, ‘produk *cleanser*’, ‘*cleanser* ini’.

2.7 *Word2Vec*

Word2Vec merupakan sebuah metode untuk membuat suatu bentuk representasi kata dalam suatu ruang vektor dengan mengelompokkan kata-kata yang mirip atau sama. Basis atau komponen utama untuk menghasilkan nilai vektor di *Word2Vec* adalah jaringan saraf tiruan (*artificial neural network*) yang dibangun dari arsitektur CBOW dan Skip-gram. Sebelum *Word2Vec* dapat mewakili nilai vektor untuk setiap kata, *Word2Vec* pertama-tama akan membuat model distribusi kata selama pelatihan menggunakan data [23].

Membangun model fitur *Word2Vec*, ada tiga proses yang terlibat, yaitu, pembangun kosakata, pembangun konteks, dan *neural network* [23]. Ilustrasi gambar arsitektur *Word2Vec* dapat dilihat pada Gambar 2.4 berikut:



Gambar 2.4 Arsitektur Utama *Word2Vec* [23]

2.7 Evaluasi Performansi

Evaluasi performansi algoritma klasifikasi *deep learning* menggunakan acuan kurva belajar adalah plot yang menunjukkan waktu atau pengalaman (*experience*) pada sumbu x dan pembelajaran atau peningkatan pada sumbu y. Kurva pembelajaran (*Learning Curve*) dianggap sebagai alat yang efektif untuk memantau kinerja sistem dengan *learning task*. LC menyediakan representasi matematis dari proses pembelajaran yang terjadi saat pengulangan tugas terjadi.

Kurva pembelajaran banyak digunakan dalam *machine learning* untuk algoritma yang belajar (mengoptimalkan parameter internal mereka) secara bertahap, seperti *deep learning neural network*. Metrik yang digunakan untuk mengevaluasi pembelajaran dapat dimaksimalkan, artinya skor yang lebih baik (angka yang lebih besar) menunjukkan lebih banyak pembelajaran. Contohnya adalah akurasi klasifikasi. Lebih umum menggunakan skor yang diminimalkan, seperti *loss* atau *error* di mana skor yang lebih baik (angka lebih kecil) menunjukkan lebih

banyak pembelajaran dan nilai 0,0 menunjukkan bahwa dataset training dipelajari dengan sempurna dan tidak ada error yang dibuat.

Selama training model *machine learning*, kondisi model saat ini pada setiap langkah algoritma *training* dapat dievaluasi. Hal ini dapat dievaluasi pada set data *training* untuk memberikan gambaran seberapa baik model tersebut “belajar.” Model ini juga dapat dievaluasi pada data validasi yang mengambil bagian dari set data training. Evaluasi pada dataset validasi memberikan gambaran tentang seberapa baik model tersebut untuk bisa “digeneralisasi.”

- *Train Learning Curve*: Kurva pembelajaran dihitung dari dataset training yang memberikan gambaran seberapa baik model tersebut dipelajari.
- *Validation Learning Curve*: Kurva pembelajaran dihitung dari set data validasi yang memberikan gambaran seberapa baik model tersebut digeneralisasi.

BAB III METODOLOGI PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang digunakan dalam pengerjaan Tugas Akhir ini. Selain itu juga akan dijelaskan detail pekerjaan serta diagram blok dalam penelitian Tugas Akhir ini.

3.1 Obyek dan Aspek Penelitian

Objek yang digunakan dalam penelitian ini adalah data *crawling* dari twitter dan facebook dengan keyword pada batasan masalah yang ada pada Bab 1. Sedangkan aspek penelitiannya adalah mengklasifikasikan sentimen tanggapan atau opini masyarakat mengenai bakal calon Walikota Surabaya dari media sosial Facebook dan Twitter dengan algoritma *N-Gram-Multichannel CNN*.

3.2 Peralatan

Adapun peralatan baik perangkat keras maupun perangkat lunak yang digunakan ditunjukkan oleh Tabel 3.1 berikut.

Tabel 3.1 Spesifikasi Perangkat

Perangkat keras	<ol style="list-style-type: none">1. Laptop ASUS 15,6", SSHD Hard Disk 1TB 5400 rpm + 8GB SSD cache2. Prosesor Intel® Core™ CPU @2.30GHz3. Grafis Intel UHD Graphics 630 dan Nvidia GeForce GTX 1050 Ti VRAM 4GB GDDR54. Memori RAM 8GB DDR4 2666MHz
Perangkat lunak	<ol style="list-style-type: none">1. Sistem Operasi Windows 10 Pro 64 bit2. Anaconda Navigator dengan aplikasi Jupyter Notebook 6.0.1 dan Spyder dengan bahasa pemrograman Python

	<ol style="list-style-type: none"> 3. Power Designer 4. Microsoft Office (Word, Excel)
--	--

3.3 Data Penelitian

Data yang digunakan dalam sistem mengklasifikasikan tanggapan atau opini masyarakat mengenai bakal calon Walikota Surabaya dari media sosial Facebook dan Twitter dengan algoritma *N-Gram-Multichannel CNN*, yaitu:

1. Data masukan, yaitu data *crawling* dari twitter dan facebook 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 opini berupa opini negatif atau opini positif.

3.4 Tahapan Penelitian

Adapun langkah-langkah yang digunakan pada penelitian Tugas Akhir ini adalah sebagai berikut:

1. Studi literatur

Studi literatur dilakukan dengan mengumpulkan bahan referensi dan rujukan tentang dasar dasar teori yang digunakan dalam analisis sentimen, *social media mining*, metode *N-Gram-Multichannel CNN* sebagai pendukung dalam kegiatan pengerjaan tugas akhir ini.

2. Perancangan Data dan Perangkat Lunak

Pada tahap ini dilakukan perancangan perangkat lunak yang akan digunakan dalam implementasi yang meliputi:

2.1 Perancangan Pengambilan Data Masukan

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

(input) berbentuk kalimat opini dengan melakukan ekstraksi data dari sosial media yaitu Facebook dan Twitter. Untuk Twitter dan Facebook proses *crawling* dilakukan dengan memanfaatkan API Twitter dan API Facebook. *Customer key*, *secret key*, *token*, dan sebagainya akan digunakan untuk mengarahkan proses pencarian di Twitter. Selanjutnya untuk data twitter akan dilakukan tahap penghapusan URL dan duplikasi data.

2.2 Perancangan Proses *Sosial Media Mining* dan Perancangan Matriks Masukan

Perancangan proses dan fungsi-fungsi *social media mining* dengan mengimpor *stopwords* dan perancangan representasi matriks masukan yang pada tahapan ini merancang fungsi-fungsi implementatif ke dalam sistem analisis dengan menggunakan bahasa pemrograman *python*.

2.3 Perancangan Proses Klasifikasi dan Analisis Sentimen

Perancangan proses klasifikasi dengan merancang fungsi-fungsi dari sistem dengan *optimizer variable* dan *learning state N-Gram Multichannel CNN* yang kita definisikan bagaimana melakukan optimisasi *network loss function*. Tensorflow mempunyai beberapa *built-in optimizers* yakni dengan menggunakan *Adam optimizer* yang akan digunakan dalam bahasa pemrograman *python*, kemudian merancang algoritma analisis sentimen dengan membentuk kelas klasifikasi opini positif dan negatif.

3. Pra-proses Data

Pra-proses data merupakan proses dimana kita mempersiapkan terlebih dahulu data yang akan dipergunakan. Beberapa langkah yang digunakan oleh penulis adalah:

- *Case Folding*

Seluruh kata pada artikel opini akan diubah menjadi huruf kecil (*lowercase*). Hal ini bertujuan untuk memperkecil kemungkinan terjadinya multi data pada kata yang sama. Multi data adalah keadaan suatu kata dianggap memiliki arti yang berbeda karena terdapat perbedaan penulisan huruf kapital (contoh: Surabaya dan surabaya).

- *Tokenization*

Tokenization adalah tugas 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. Sebagai contoh, tokenisasi dari kalimat "Aku sudah belajar bab *machine learning*." menghasilkan enam token, yakni: "Aku", "sudah", "belajar", "bab", "*machine*", "*learning*".

- *Filtering*

Pada tahap ini akan difilter kata-kata penting dari opini atau bias bisa dikatakan membuang kata-kata yang tidak berpengaruh pada arti utama dari kalimat opini. Kata yang dihapus termasuk misalnya tanda hubung, url, emotikon serta *punctuation*. Penghilangan angka, kata yang dianggap tidak bertendensi yang biasa disebut *stopword*. Contoh terdapat kalimat "Aku ini sudah mempelajarinya lho! 😊" maka dapat difiltrasi sehingga menghasilkan kalimat "Aku mempelajarinya".

- *Stemming*

Proses perubahan menjadi kata dasar. Kata yang telah dilakukan *filtering* kemudian akan diubah kedalam bentuk kata dasar. Dengan kata lain pada tahap ini akan dihilangkan *prefix* dan *suffix* dari sebuah kata. Hal ini akan memudahkan proses pengenalan kata, sehingga kata yang memiliki kata dasar sama akan dianggap sama. Contoh ketika terdapat

kalimat “Aku mempelajarinya” maka akan dirubah ke kalimat “aku belajar”.

4. **Labelling**

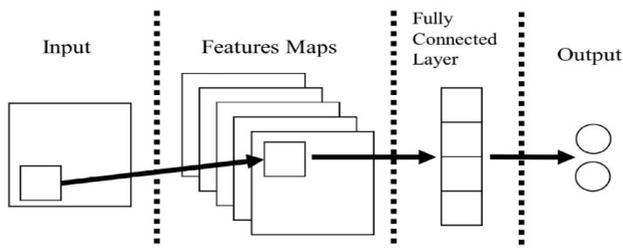
Pada tahap labeling ini akan menggunakan metode HIT (*Human Intelligence Task*) dimana proses ini memiliki tingkat persetujuan lebih besar dari 95% [24]. Setiap HIT akan dilakukan oleh 3 orang untuk setiap 10.000 kalimat opini. Masing-masing anotator akan memberikan label pada kalimat opini dengan tujuan mempertimbangkan asumsi masing-masing anotator. Untuk mendapatkan label akhir dari masing-masing kalimat opini apabila dua orang atau lebih memilih label yang sama, maka label akhirnya adalah berdasarkan pemilihan tersebut.

5. **Feature Extraction**

Setelah didapatkan kata-kata yang “bersih” maka akan diubah menjadi bentuk vektor sehingga dapat dicerna oleh komputer. Vektor kata adalah representasi kedekatan antar kata-kata. Vektor yang digunakan adalah *Word2Vec*. Untuk kata-kata yang tidak terkandung di dalamnya akan dihasilkan secara acak. Setelah tahap ini kita akan mendapatkan representasi matriks kata dari *tweet* dan postingan teks.

6. **Implementasi N-Gram-Multichannel CNN**

Pada tahap ini dilakukan implementasi sistem seperti pada ilustrasi gambar berikut:



Gambar 3.1 Implementasi Model

Perancangan pada tahap ini melanjutkan dari tahap sebelumnya dengan menggunakan data yang didapatkan dari tahap *feature extraction* dimana pada proses Implementasi Algoritma ini terdiri dari tahap uji dan latih. Tahap latih (*training*) digunakan untuk mendapatkan parameter yang sesuai dan optimal, sedangkan tahap uji (*testing*) untuk melihat keberhasilan dari parameter yang telah didapatkan setelah penelitian, beberapa proses yang dilakukan meliputi:

a. *Convolutional Layer* dengan N-Gram

Pada tahap ini Konvolusi adalah lapisan pertama dalam arsitektur jaringan N-Gram dan CNN yang mengekstraksi fitur data input menggunakan *one-dimensional CNN* dimana layer konvolusi dikembangkan dengan mengubah ukuran kernel. Ukuran kernel pada lapisan konvolusional menentukan jumlah kata pada input teks, memberikan parameter pengelompokan yang diimplementasikan berdasar resolusi yang berbeda atau berbeda n-gram (kelompok kata) nantinya akan dicari *training process* yang mengintegrasikan interpretasi sistem konvolusi dengan baik. Pada proses *convolution*, dilakukan operasi *dot product* antara input *layer* dan *kernel / filter*. Operasi tersebut akan dilakukan secara terus menerus hingga semua nilai pada dimensi vektor terkalkulasi. Selanjutnya dengan menggunakan fungsi non linear yang digunakan yakni fungsi *Rectified Linear Unit* (ReLU) maka vektor yang bernilai negatif dirubah menjadi 0.

b. *Pooling Layer*

Layer selanjutnya adalah *pooling layer*. Fungsi dari pooling layer adalah sebagai down sampling yang non-linear. Pada tahap ini dilakukan *MaxPooling1D* untuk mengkonsolidasikan output dari lapisan konvolusi. *Max Pooling*, membagi output vektor *convolutional layer* ke

beberapa *grid* vektor dan mengambil nilai maksimal dari setiap *grid*.

c. *Fully Connected Layer*

Pada tahap ini dilakukan *Flatten*, *Concatenate Flat*, dan *Dense*. *Flatten Layer* sendiri digunakan untuk mengubah *Feature map* yang dihasilkan masih berbentuk *multidimensional array*, sehingga harus melakukan “*flatten*” atau *reshape feature map* menjadi sebuah vektor agar bisa digunakan sebagai input dari *fully-connected layer*. Output *feature map* tiga dimensi menjadi dua dimensi yang nantinya akan dilakukan konkatenasi (penggabungan).

7. Implementasi Analisis Sentimen Klasifikasi Data

Layer selanjutnya setelah *max-pool* yakni dilakukan konkatenasi maka masuk ke dalam sebuah *long feature vector*, menambahkan *dropout regularization* yang prosesnya dapat mencegah terjadinya *overfitting* dan juga mempercepat proses learning, serta mengklasifikasikan hasil menggunakan *softmax layer*.

8. Evaluasi

Pada tahap ini akan dilakukan evaluasi kecocokan model sistem dengan memprediksi sentimen yang belum diketahui pada semua ulasan di data uji. Menggunakan fungsi pemusatan data yang dikembangkan di bagian sebelumnya, kita bisa *load* dan *encode* baik data uji maupun data latihan dimana nantinya menghasilkan angka akurasi dari sistem.

9. Analisa Hasil dan Pengambilan Keputusan

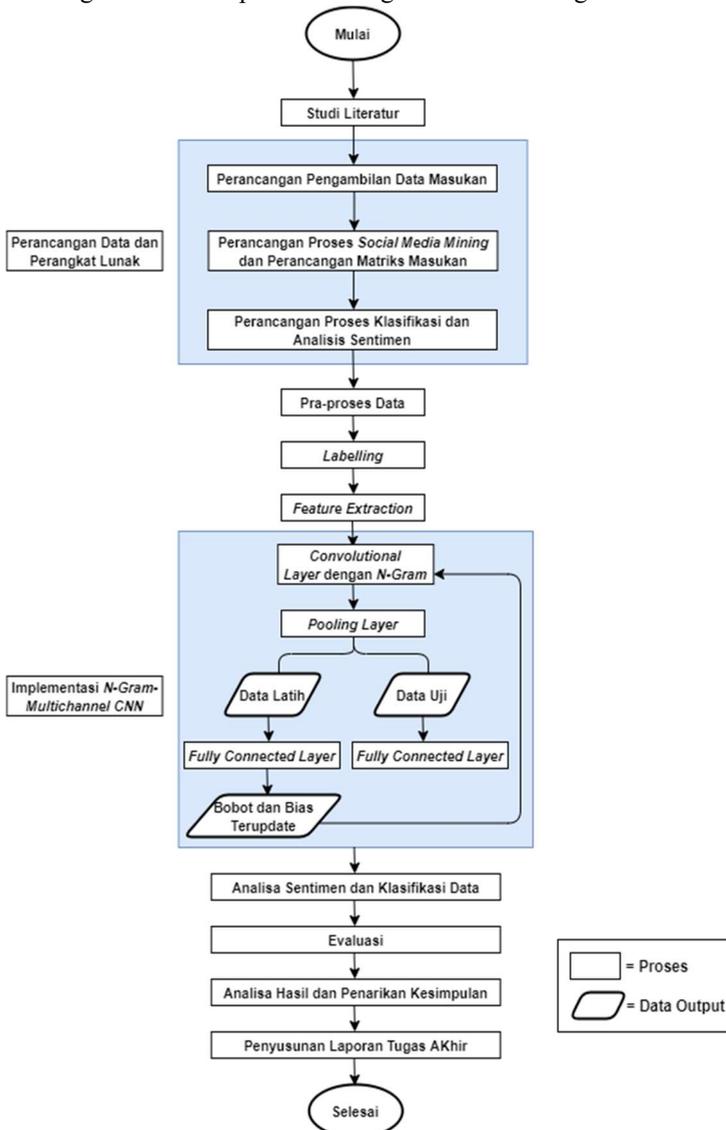
Pada tahap ini dilakukan penarikan kesimpulan dari hasil keluaran (output) klasifikasi opini-opini yang akan diklasifikasikan ke dalam kelas positif dan negatif dengan menganalisis akurasi sistem.

10. Penyusunan Laporan Tugas Akhir

Pada tahap ini semua proses serta pembahasan dirangkum dan ditulis pada buku Tugas Akhir.

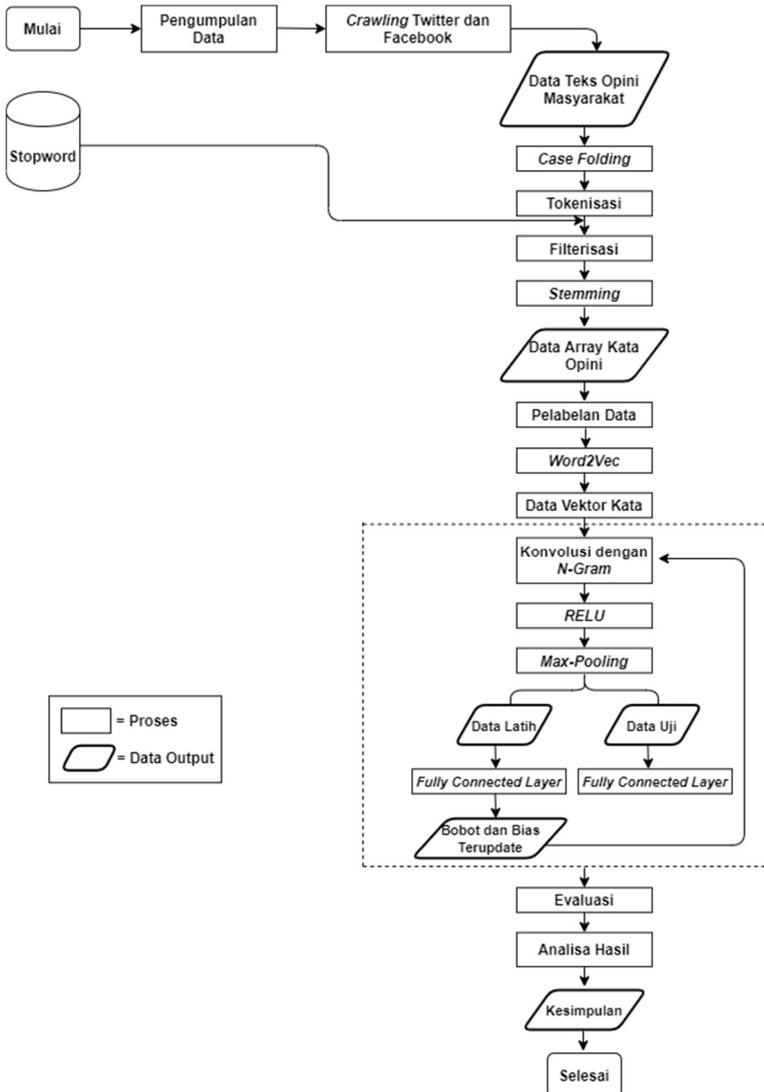
3.5 Diagram Alir Penelitian

Diagram Alir dari penelitian Tugas Akhir ini sebagai berikut:



Gambar 3.2 Diagram Alir Metodologi Penelitian

Perincian tahapan uji coba dan pembahasan dapat diperhatikan pada diagram alir pada gambar berikut ini:



Gambar 3.3 Perincian Diagram Alir Penelitian

BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM

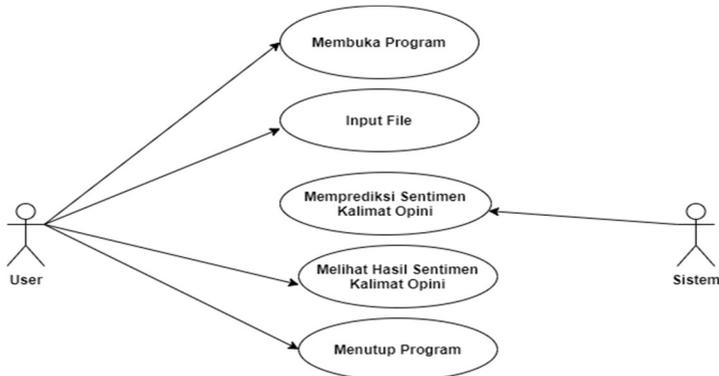
Pada bab ini akan menjelaskan mengenai rancangan berupa Analisa dari desain sistem yang digunakan sebagai acuan untuk implementasi sistem. Perancangan yang akan dibuat berupa perancangan data, pra-pemrosesan data, dan pembuatan model *N-Gram-Multichannel CNN*. Adapun ilustrasi penjelasan proses adalah dengan menggunakan data *dummy* untuk memudahkan dalam interpretasi.

4.1 Analisis 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

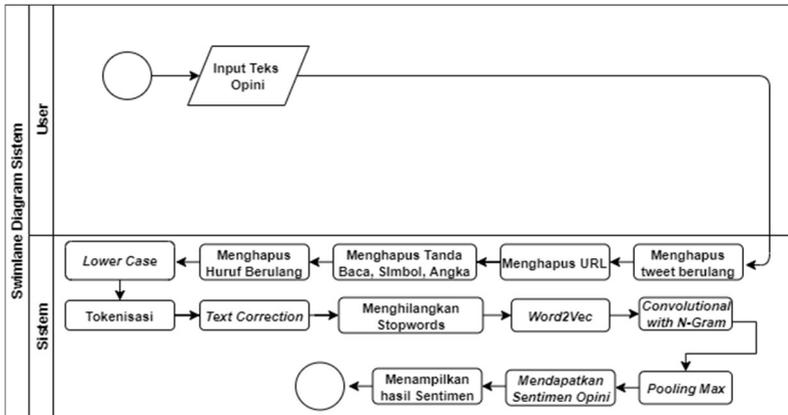
Use Case Diagram adalah gambaran analisa optimalisasi terhadap suatu sistem dalam bentuk *graphical* dari beberapa atau semua *user*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara rinci tentang penggunaan *use case*, tetapi hanya memberi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



Gambar 4.1 *Use Case Diagram* Sistem Analisa Sentimen dengan Algoritma *N-Gram-Multichannel CNN*

4.1.2 *Swimlane Diagram*

Swimlane diagram adalah sebuah diagram yang merepresentasikan alur proses yang menggambarkan interaksi dari beberapa bagian yang berbeda dan bagaimana perkembangan proses melalui beberapa fase yang berbeda. *Swimlane* membagi aktivitas dalam beberapa kelompok dimana setiap kelompok yang telah dibagi dibuat untuk dapat merepresentasikan organisasi yang bertanggung jawab untuk aktivitas tersebut. Gambar 4.2 merupakan *Swimlane diagram* dari penelitian ini.



Gambar 4.2 Use Case Diagram Sistem Analisa Sentimen dengan Algoritma N-Gram-Multichannel CNN

4.2 Analisis Kebutuhan Fungsional Sistem

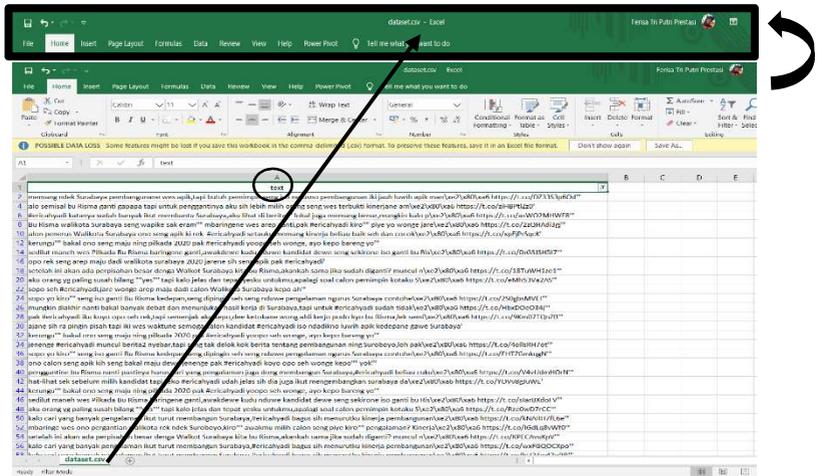
Berdasarkan pada analisis sistem menggunakan *use case diagram* dan *swimlane diagram* berikut maka output sistem yang akan dibangun diharapkan mampu mengembangkan kompetensi sebagai berikut:

1. Sistem mampu melakukan pembelajaran dengan menggunakan algoritma *N-Gram-Multichannel CNN* pada data latih.
2. Sistem mampu menerapkan proses pembelajaran dengan menggunakan algoritma *N-Gram Multichannel CNN* pada data uji dengan melakukan klasifikasi pada algoritma yang akan dibangun melalui sistem.
3. Sistem mampu mengklasifikasikan input data opini dalam bentuk teks yang sudah di *pre-processing* untuk mendapatkan output kelas biner yaitu kelas opini positif ataupun kelas opini negatif.
4. Sistem mampu menyimpan data hasil klasifikasi opini yang telah diproses.

4.3 Perancangan Data

Data pada sistem dapat dibedakan menjadi tiga bagian, yakni data masukan sistem, data proses, serta data hasil pengolahan.

1. Data masukan sistem merupakan data berupa komentar facebook dan *tweet* bahasa Indonesia yang sudah disimpkan dalam format .csv dengan *sheet* dan kolom tunggal dengan *header* berjudul “text”. Bila data masukan sudah sesuai, maka program akan berjalan sesuai dengan prosedur. Contoh data masukan yang sudah disimpan ditunjukkan pada Gambar 4.3.



Gambar 4.3 Contoh Data Input berupa Kalimat Opini

2. Data proses merupakan data yang digunakan dalam proses pengolahan data masukan hingga hasil dari proses tersebut. Data proses sistem ini terbagi menjadi beberapa tahapan dapat ditunjukkan pada Tabel 4.1 berikut:

Tabel 4.1 Tabel Data Proses

No	Proses	Masukan	Luaran
1	Persiapan Data Masukan	Data mentah kumpulan opini berbahasa Indonesia dari facebook dan twitter berformat .csv	Data awal yaitu data berupa opini dan label
2	Menghapus duplikasi data	Data opini dan label yang kembar	Data opini dan label yang tunggal dan berbeda satu sama lain
3	Menghapus URL	Data opini tunggal dengan bercampur URL di dalam kalimat	Data opini tunggal dan bersih dari URL
4	<i>Case Folding</i>	List kalimat opini dari data awal	List kalimat opini dengan format <i>lowercase</i>
5	<i>Tokenisasi</i>	List kalimat opini dengan format <i>lowercase</i>	List kalimat opini yang telah dipisah per-kata
6	<i>Filtering</i>	List kalimat opini yang telah dipisah per-kata	List kalimat opini tanpa ada angka dan tanda baca

Lanjutan Tabel 4.1

7	<i>Stemming</i>	List kalimat opini tanpa ada angka dan tanda baca	list kalimat opini dengan hanya kata dasar
8	Ekstraksi Fitur	list kalimat opini dengan hanya kata dasar	array vektor representasi kata dari kalimat opini
9	<i>N-Gram-Multichannel CNN</i>	array vektor representasi kata dari kalimat opini	probabilitas prediksi kelas dari kalimat opini

3. Data hasil pengolahan pada sistem ini yaitu akurasi pembelajaran menggunakan algoritma *N-Gram-Multichannel CNN* dan berupa kumpulan artikel opini yang telah dilakukan *pre-processing* dan telah dilakukan klasifikasi kelas pada tiap opini, sehingga muncul kelas opini positif dan kelas opini negatif.

Tahap awal pada perancangan dimulai dengan mengumpulkan seluruh data berupa kalimat opini mengenai bakal calon Walikota Surabaya 2020 yang dibutuhkan dari media sosial facebook dan 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. Terdapat masing-masing regulasi dalam pengambilan data dari media sosial baik dari Developer Twitter dan Developer Facebook.

Pada Developer Twitter sendiri, 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 ketersediaan. Sedangkan pada Developer Facebook, durasi token API yang disediakan oleh Graph API facebook membatasi setiap *generate* token hanya berlaku selama waktu 1 jam.

Akses untuk mengambil data dari twitter memerlukan 4 kode rahasia yaitu : *customer key*, *customer secret*, *token access*, dan *token access secret* yang didapatkan dari situs dev.twitter.com dengan mendaftar sebagai developer, sedangkan pada facebook memerlukan 3 kode yakni *App_id*, *App Secret*, dan *token access*. Berikut merupakan kode API Twitter yang didapatkan dari dev.twitter.com dan Kode Token Graph API Facebook <https://developers.facebook.com/apps> yang dapat dilihat pada Tabel 4.2 dan Tabel 4.3 pada halaman selanjutnya.

Tabel 4.2 Kode API Twitter

<i>customer key</i>	e5E1w5mimsnmqf4gRM2RJMHWYZ
<i>customer secret</i>	tQYWigKTckslEBjg0txbvuyYzvXC6e9Tqc3KQaats0GBdTk8na
<i>token access</i>	251485653- hWoRe8xmHq6mvsLBZIJv5gXEUx6idi3UDoNMdwJms
<i>token access secret</i>	TulGSPMMfewNj1dbKPdbUY5XIJQnLsuwb4YodGeixSRDfd

Tabel 4.3 Kode Token Graph API Facebook

<i>App_id</i>	28301420970778042
<i>App Secret</i>	a65ce47d53c837a73404c8d70592cbe02
<i>token access</i>	EAAPPnbFkvPMBALxuququdeMy4bo MNwkB8UurrQxP4ugFEK1SffaVuMXqKPZAAtMT2 59SnhSqY8sb1ZCZB88CoQto2 C5Ipv0kLtgptl5fr8LElGrgQrwEZByxjEZBhk8ni9ST52 NoUGxXIbcupVxCQL8DZA2 TGVs7h2en4abys0IMW4yQg9XcICybCNV8RRmry8Mp0x3rKU2jh2 2HZBf0YLEChGg9WiiapsK8eAG9wWr8YTwjIIAZDZD2

Pada proses *crawling* dari Developer facebook dibuat menggunakan *library request* untuk mengirim *request* pada sistem Facebook dan *library facebook* untuk membaca *token access* yang telah diperoleh dari facebook developer. Dalam menampilkan data, facebook menggunakan API yang mengembalikan data ulasan berupa data .json dari setiap data yang akan diakses, sehingga dalam proses *crawling* akan dilakukan *request* dari url API ulasan menggunakan *token access*. Hasil *request* akan dikembalikan berupa data ulasan .json yang berisi sesuai dengan izin yang diminta kepada pihak Developer facebook yakni *user_status*, *user_posts*, dan *user_public_profile*. Data .json yang diperoleh selanjutnya diolah agar dapat diubah dalam bentuk .csv untuk mempermudah penyimpanan dan penggunaan.

Berikut *Syntax* yang digunakan dalam proses *crawling* Twitter dan Facebook:

```

twitter_key = 'e5E1w5mimsnmqf4gRM2RJMHwY'
twitter_secret =
'tQYWigKTcks1EBjg0txbvuysYzvXC6e9Tqc3KQaats0GBdTk8n
'
access_token = '251485653-
hWoRe8xmHq6mvsLBZlJv5gXEUx6idi3UDoNmDwJm'
access_secret =
'TulGSPMMfewNj1dbKPdbUY5XlJQnLsuwb4YodGeixSRdf'
auth = tweepy.OAuthHandler(twitter_key,
twitter_secret)
auth.set_access_token(access_token, access_secret)
api = tweepy.API(auth,wait_on_rate_limit=True)
csvFile = open('crawl9april.csv', 'a')
csvWriter = csv.writer(csvFile)
for tweet in
tweepy.Cursor(api.search,q="ericahyadi",
,lang="id").items():
    print (tweet.created_at, tweet.text)
    csvWriter.writerow([tweet.created_at,
tweet.text.encode('utf-8')])

```

Gambar 4.4 *Syntax Crawling Data Twitter*

Gambar 4.4 memberikan *code* perintah yang digunakan untuk melakukan pengambilan data twitter sesuai dengan *keyword* pencarian yang di inisiasi. Kemudian pada proses ini juga menghasilkan file CSV. Proses ini akan berulang ke dalam masing-masing keyword nama bakal calon Walikota Surabaya 2020. Proses ini menggunakan search API twitter dimana menjelaskan proses kode sebuah *instance* yang menghubungkan client menggunakan *API_Key* dan *API_Secret* yang telah didapatkan dari website Twitter API. Selanjutnya berikut kode *crawling* facebook:

```

token =
"EAAPPnbFkvPMBANyuAi5dvOxEEn4JXPP7g7RFuYVbZAz19Phji
dpLqA5mVsDe2EsAErc4YZBC50J3gzoCcssPGMxEn8UfEd4tN7ZCZ
AjNnoEr3CMvdlOOFuOZBskDsovknQtPZADYhr6pBozFWBf5zwaM
UmoVZAUneJyFzAmFFnpuiVZAx5lWFBBrMVCLZBvK9MDV60zjFU5a
0fy38OKZAJp4XW7Ohz4dGhWZBMaz9SlcXx3p2GdwZDZD"
user_id='1072701123116275'
page_id='2168667396585351'
keyword='ericahyadi'
#load access token
graph = fb.GraphAPI(token)
fb_data = graph.get_object(user_id)
fb_data.keys()
post=fb_data['posts']['data']
post_data = pd.DataFrame(post)

```

Gambar 4.5 *Syntax Crawling Data Facebook*

Hampir sama dengan prinsip *crawling* pada twitter, pada facebook dapat diperhatikan pada Gambar 4.5 berisi *code* perintah yang digunakan untuk melakukan pengambilan data facebook juga sesuai dengan *keyword* pencarian yang di inisiasi dengan menggunakan input token yang sudah diperoleh dari developer.facebook.com.

Contoh kutipan kalimat opini yang diposting masyarakat terkait dengan bakal calon Walikota Surabaya 2020 ada sebagai berikut:

setelah ini akan ada perpisahan besar dengan Walikota Surabaya kita bu Risma, apakah sama jika sudah diganti? muncul nama #ericahyadi, diriku jadi yakin pasti akan sama dan bahkan akal menjadi lebih baik lagi, kenapa? karena kinerja dan pengalamannya, orang ahli kerja sopo yo kiro" seng iso ganti Bu Risma kedepan, seng dipingin seh seng nduwe pengalaman ngurus Surabaya contohe #ericahyadi beliau yo tau apik mbangun Surabaya, lek wes pengalaman lebih meyakinkan mbarangene tahun ngarep wes ono pilkada maneh rek opo kalian wes nemu kandidat gawe awakmu dewe, #ericahyadi jare wonge apik, lan jelas kerjone yoopo, penasaran kambek wonge? podo..

kalo semisal bu Risma ganti gapapa tapi untuk penggantian aku sih lebih milih orang seng wes terbukti kinerjane ambek wes terbukti pengalaman buat ngurus Surabaya #ericahyadi

Adapun berikut disajikan data *dummy* yang berjumlah empat tanggapan dari masyarakat dari beberapa bakal calon Walikota Surabaya 2020 untuk mempermudah dalam interpretasi. Tabel 4.4 merupakan contoh data *dummy*:

Tabel 4.4 Data *Dummy*

jaman sekarang jangan asal memilih ya, cari yg pengalaman dan sudah banyak kinerja yang bagus untuk bisa diyakini bisa maju,#ericahyadi menurutku juga sudah jelas dan bakal ga salah memilihnya 😊 https://t.co/DZ33S3p6Od
machfud arifin kiranya memang bapak ini mempunyai pengalaman yang sangat banyak sekaliiiiii https://t.co/asWO2MHWF
Lahhh program sembako murah spt ini 'maaf' model kampanye basi dok gamal albinsaid. Utk seorg inovator yg millennial spt dokter hrsnya model kampanye kyk gini g masuk dlm list. Kita mau program kerja yg realistis. Misal klo para ibu mjd lbh berdaya, shg sembako mahl pun ttp kebeli. https://t.co/eMh53Va2A
Auto milih Pak Whisnu di Pilwali Surabaya nantiiii!!! https://t.co/2S0gbsMVEI Auto milih Pak Whisnu di Pilwali Surabaya nantiiii!!! https://t.co/2S0gbsMVEI

4.3 Perancangan Pra-pemrosesan Data

Adapun perancangan pra-proses data akan dilakukan proses persiapan awal guna menyiapkan data agar dapat menjadi masukan pada proses algoritma *N-Gram-Multichannel CNN*. Proses ini sangat penting karena akan mempengaruhi performansi dari sistem yang akan digunakan dalam klasifikasi kalimat opini.

Setelah mendapatkan data opini yang masih mentah yakni masih tidak terstruktur dan mengandung banyak *noise*, maka harus diproses terlebih dahulu agar lebih mudah untuk melakukan klasifikasi sentimennya. Data kalimat opini masih belum baku, memuat angka, tanda baca, kata-kata yang berbahasa jawa, masih terdapat kata dengan penulisan yang salah maupun yang kurang bermakna untuk dijadikan fitur. Beberapa proses awal sebelum mengklasifikasikan sentimen atau yang biasa di sebut dengan *Normalization Text*. Tahapan tersebut adalah sebagai berikut.

4.3.1 *Load Data Opini dalam Python*

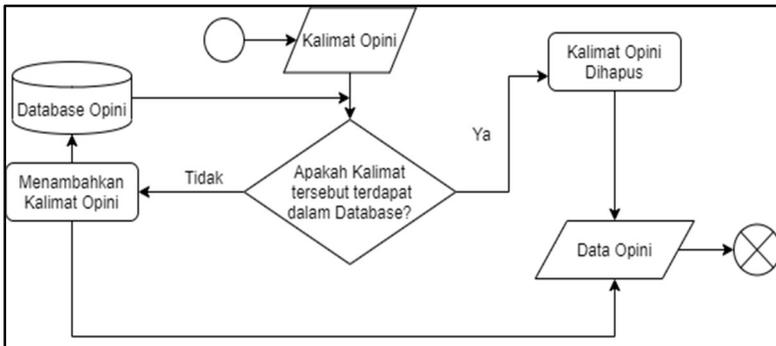
Data opini yang sebelumnya sudah disimpan dalam format *.csv* di-*load* dalam python dan ditampilkan dalam bentuk *list*. Proses ini bertujuan untuk membuka file yang disimpan dalam *.csv* ke python. Hal itu dilakukan karena dalam pengolahan kalimat opini tersebut menggunakan Bahasa pemrograman python. Berikut merupakan *Syntax* yang digunakan untuk *load data*:

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

Gambar 4.6 *Code Load Data Opini*

4.3.2 *Menghapus Opini yang Berulang*

Pada tahapan ini dilakukan penghapusan duplikasi data karena data yang didapatkan dari hasil *crawling* twitter dan *crawling* facebook masih ditemukan kalimat opini yang sama, oleh karena itu harus dihapus salah satunya hingga setiap data yang ada merupakan data yang unik. Proses ini dapat ditunjukkan pada Gambar 4.7.



Gambar 4.7 Flowchart Penghapusan Duplikasi Data Opini

Berikut merupakan *Syntax* yang digunakan untuk menghapus data yang kembar atau duplikat dengan menghapus salah satu menggunakan fungsi `set()` dalam python:

```
data_awal= set(dataset)
print(data_awal)
```

Gambar 4.8 *Syntax* Penghapusan Duplikasi Data Opini

4.3.3 Menghapus URL

Data opini 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 kalimat opini yang telah di miliki. Berikut *Syntax* menghapus URL:

```

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()

```

Gambar 4.9 *Syntax* Penghapusan URL

Data *dummy* yang telah dilakukan proses penghapusan duplikasi dan penghapusan URL dapat ditunjukkan pada tabel 4.5 berikut:

Tabel 4.5 Hasil Penghapusan URL Data *Dummy*

jaman sekarang jangan asal memilih ya, cari yg pengalaman dan sudah banyak kinerja yang bagus untuk bisa diyakini bisa maju,#ericahyadi menurutku juga sudah jelas dan bakal ga salah memilihnya 😊

machfud arifin kiranya memang bapak ini mempunyai pengalaman yang sangat banyak sekaliiiiii

Lahhh program sembako murah spt ini 'maaf' model kampanye basi dok gamal albinsaid. Utk seorg inovator yg millennial spt dokter hrsnya model kampanye kyk gini g masuk dlm list. Kita mau program kerja yg realistis. Misal

klo para ibu mjd lbh berdaya, shg sembako mahl pun ttp kebeli
Auto milih Pak Whisnu di Pilwali Surabaya nantiiii!!!

4.3.4 Menghapus Tanda Baca, Simbol, dan Angka

Menghilangkan karakter yang tidak perlu dan karakter khusus/spesial hal penting pada Normalisasi Teks. Alasan utama karena sering tanda baca atau karakter khusus tidak memiliki arti ketika kita menganalisis teks dan menggunakannya untuk mengekstraksi fitur atau informasi. Berikut *Syntax* menghapus Tanda Baca, Simbol, dan Angka:

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

Gambar 4.10 *Syntax* Penghapusan Tanda Baca, Simbol, dan Angka

Data *dummy* yang telah dilakukan proses penghapusan duplikasi dan penghapusan URL dapat ditunjukkan pada tabel 4.6 berikut:

Tabel 4.6 Hasil Penghapusan Tanda Baca, Simbol, dan Angka

jaman sekarang jangan asal memilih ya cari yg pengalaman dan sudah banyak kinerja yang bagus untuk bisa diyakini bisa maju ericahyadi menurutku juga sudah jelas dan bakal ga salah memilihnya
machfud arifin kiranya memang bapak ini mempunyai pengalaman yang sangat banyak sekaliiiii
Lahhh program sembako murah spt ini maaf model kampanye basi dok gamal albinsaid Utk seorg inovator yg millennial spt dokter hrsnya model kampanye kyk gini g masuk dlm list Kita mau program kerja yg realistis Misal klo para ibu mjd lbh berdaya shg sembako mahl pun ttp kebeli
Auto milih Pak Whisnu di Pilwali Surabaya nantiiii

4.3.5 Menghapus Huruf yang Berulang

Menghilangkan huruf yang berulang dilakukan pengulangan huruf ini tidak memiliki arti ketika kita menganalisis teks dan menggunakannya untuk mengekstraksi fitur atau informasi. Pada proses ini dibatasi untuk setiap huruf hanya boleh berulang dua kali. Berikut *Syntax* menghapus huruf yang berulang:

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

Gambar 4.11 *Syntax* Penghapusan Huruf Berulang

Data *dummy* yang telah dilakukan proses penghapusan duplikasi huruf dapat ditunjukkan pada tabel 4.7 berikut:

Tabel 4.7 Hasil Penghapusan Huruf Berulang

jaman sekarang jangan asal memilih ya cari yg pengalaman dan sudah banyak kinerja yang bagus untuk bisa diyakini bisa maju ericahyadi menurutku juga sudah jelas dan bakal ga salah memilihnya
machfud arifin kiranya memang bapak ini mempunyai pengalaman yang sangat banyak sekali
Lah program sembako murah spt ini maaf model kampanye basi dok gamal albinsaid Utk seorg inovator yg millennial spt dokter hrsnya model kampanye kyk gini g masuk dlm list Kita mau program kerja yg realistis Misal klo para ibu mjd lbh berdaya shg sembako mahl pun ttp kebeli
Auto milih Pak Whisnu di Pilwali Surabaya nanti

4.3.6 Case Folding

Dalam pra-pemrosesan data yakni tahap *case folding* ini akan dilakukan penyamaan bentuk penulisan. Dalam dataset kalimat opini ini, seluruh kata akan diubah menjadi bentuk penulisan *lowercase* atau dirubah ke huruf kecil semua. Hal ini

bertujuan agar tidak terjadi data ganda apabila ada penulisan dengan huruf kapital ataupun tidak. Berikut *Syntax case folding*:

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

Gambar 4.12 *Syntax Case Folding*

Data *dummy* yang telah dilakukan proses *case folding* dapat ditunjukkan pada tabel 4.8 berikut:

Tabel 4.8 Hasil *Case Folding*

jaman sekarang jangan asal memilih ya cari yg pengalaman dan sudah banyak kinerja yang bagus untuk bisa diyakini bisa maju ericahyadi menurutku juga sudah jelas dan bakal ga salah memilihnya
machfud arifin kiranya memang bapak ini mempunyai pengalaman yang sangat banyak sekali
lah program sembako murah spt ini maaf model kampanye basi dok gamal albinsaid utk seorg inovator yg millennial spt dokter hrsnya model kampanye kyk gini g masuk dlm list kita mau program kerja yg realistis misal klo para ibu mjd lbh berdaya shg sembako mahl pun ttp kebeli
auto milih pak whisnu di pilwali surabaya nanti

4.3.7 Tokenisasi

Pada tahap tokenisasi dilakukan suatu proses pemecahan tiap kalimat opini 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. Berikut *Syntax* tokenisasi:

```

from nltk.tokenize import word_tokenize
def token(datasetresult):
    take_token = [word_tokenize(i) for i in
datasetresult]
    for i in take_token:
        print (i)
    return i

```

Gambar 4.13 *Syntax* Tokenisasi

Data *dummy* yang telah dilakukan proses penghapusan duplikasi dan penghapusan URL dapat ditunjukkan pada tabel 4.9 berikut:

Tabel 4.9 Hasil Tokenisasi

['jaman', 'sekarang', 'jangan', 'asal', 'memilih', , 'ya', 'cari', 'yg', 'pengalaman', 'dan', 'sudah', , 'banyak', 'kinerja', 'yang', 'bagus', 'untuk', 'bi sa', 'diyakini', 'bisa', 'maju', 'ericahyadi', 'me nurutku', 'juga', 'sudah', 'jelas', 'dan', 'bakal', , 'ga', 'salah', 'memilihnya']
['machfud', 'arifin', 'kiranya', 'memang', 'bapak', , 'ini', 'mempunyai', 'pengalaman', 'yang', 'sang at', 'banyak', 'sekali']
['lah', 'program', 'sembako', 'murah', 'spt', 'ini', , 'maaf', 'model', 'kampanye', 'basi', 'dok', 'gam al', 'albinsaid', 'utk', 'seorg', 'inovator', 'yg', , 'millennial', 'spt', 'dokter', 'hrsnya', 'model', , 'kampanye', 'kyk', 'gini', 'g', 'masuk', 'dlm', 'li st', 'kita', 'mau', 'program', 'kerja', 'yg', 'real istis', 'misal', 'klo', 'para', 'ibu', 'mjd', 'lbh', , 'berdaya', 'shg', 'sembako', 'mahl', 'pun', 'ttp', , 'kebeli']
['auto', 'milih', 'pak', 'whisnu', 'di', 'pilwali', , 'surabaya', 'nanti']

4.3.8 *Spelling Normalization*

Data opini masyarakat yang diambil dari data di sosial media tentu tidak memperhatikan kaidah Bahasa yang 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* dalam python. Tujuan utama dari proses *Spelling Normalization* 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. Berikut *Syntax spelling normalization*:

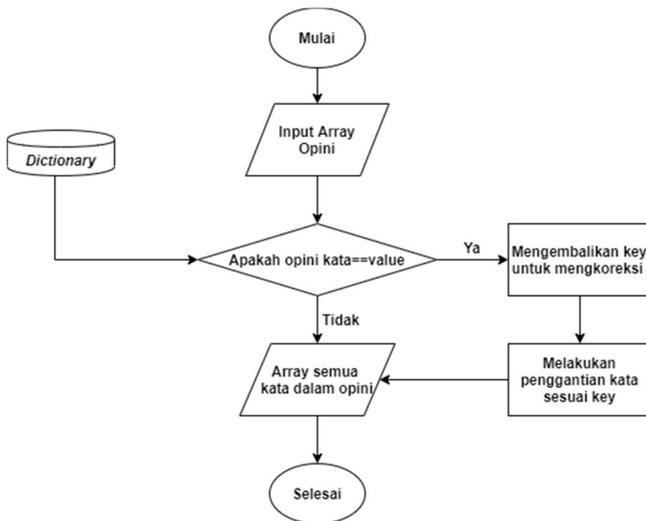
```
d = {}
with open("dictionary.txt") as text:
    for line in text:
        if line.strip():
            key, val = line.split(None, 1)
            d[key]=val.split()
def mencaritypo(kata):
    for key in d:
        list1=d.get(key)
        if kata in list1:
            return key
    return kata
def replacetypo(tupel):
    temp_
    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
```

Gambar 4.14 *Syntax Spelling Normalization*

Korpus dalam tahap ini dapat dilihat dalam Lampiran B, dimana berikut potongan contoh *corpus* yang digunakan sebagai *dictionary* dari beberapa kata yang digunakan dalam proses ini. *Dictionary* adalah struktur data yang bentuknya seperti kamus. Ada kata *key* dan ada *value*. Kata *key* menunjukkan kata yang benar, sedangkan *value list* kata-kata yang kurang tepat dalam menuliskan kata pada *key*.

```
{'sekarang': ['skr', 'skarang', 'skrng',
'skrg'], 'sebelum': ['sblm', 'sbelum',
'sblum', 'seblum', 'sebelm'], 'tidak':
['gak', 'gk', 'tdk', 'gx', 'ga', 'nggak',
'enggak', 'g', 'engga', 'ngga', 'tyda',
'tydac', 'tydak', 'ngak', 'ngk', 'kagak',
'nggk'], 'lebih': ['lbh', 'lrbih',
'lbih'],'selamat': ['slmt', 'slamat',
'met', 'slamet', 'selamet'],'yang':
['yg']}
```

Gambar 4.15 Potongan *corpus*



Gambar 4.16 Flowchart *Spelling Normalization*

Data *dummy* yang telah dilakukan proses *Spelling Normalization* dapat ditunjukkan pada tabel 4.10 berikut:

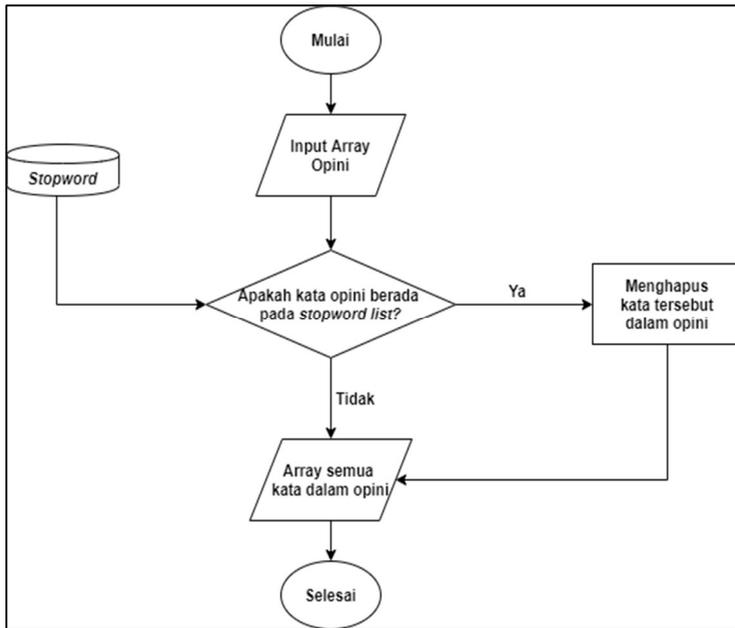
Tabel 4.10 Hasil *Spelling Normalization*

['jaman', 'sekarang', 'jangan', 'asal', 'memilih', 'ya', 'cari', 'yang', 'pengalaman', 'dan', 'sudah', 'banyak', 'kinerja', 'yang', 'bagus', 'untuk', 'bisa', 'diyakini', 'bisa', 'maju', 'ericahyadi', 'menurutku', 'juga', 'sudah', 'jelas', 'dan', 'bakal', 'nggak', 'salah', 'memilihnya']
['machfud', 'arifin', 'sekiranya', 'memang', 'bapak', 'ini', 'mempunyai', 'pengalaman', 'yang', 'sangat', 'banyak', 'sekali']
['lho', 'program', 'sembako', 'murah', 'seperti', 'ini', 'maaf', 'model', 'kampanye', 'basi', 'dok', 'gamal', 'albinsaid', 'untuk', 'seorang', 'inovator', 'yang', 'millenial', 'seperti', 'dokter', 'seharusnya', 'model', 'kampanye', 'kayak', 'gini', 'nggak', 'masuk', 'dalam', 'list', 'kita', 'mau', 'program', 'kerja', 'yang', 'realistis', 'misal', 'kalau', 'para', 'ibu', 'menjadi', 'lebih', 'berdaya', 'sehingga', 'sembako', 'mahal', 'pun', 'tetap', 'kebeli']
['auto', 'milih', 'pak', 'whisnu', 'di', 'pilwali', 'surabaya', 'nanti']

4.3.9 Filterisasi

Pada tahap filterisasi akan dilakukan penghilangan kata atau entitas yang dirasa tidak penting dalam penentuan sentimen kalimat opini. Dalam Tugas Akhir ini penulis menggunakan library *stopwords* milik *nlTK* untuk melakukan filterisasi pada artikel yang telah berbentuk token. Library yang digunakan dalam proses filterisasi ini sudah tersedia untuk Bahasa Indonesia.

Proses ini dapat ditunjukkan oleh diagram alir pada Gambar 4.17. Untuk daftar *stopwords* dapat dilihat pada Lampiran C.



Gambar 4.17 Flowchart Filterisasi

Berikut *Syntax* Filterisasi dimana prosesnya menghilangkan *stopwords* atau mengoreksi kata:

```

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
  
```

Gambar 4.18 Syntax Filterisasi

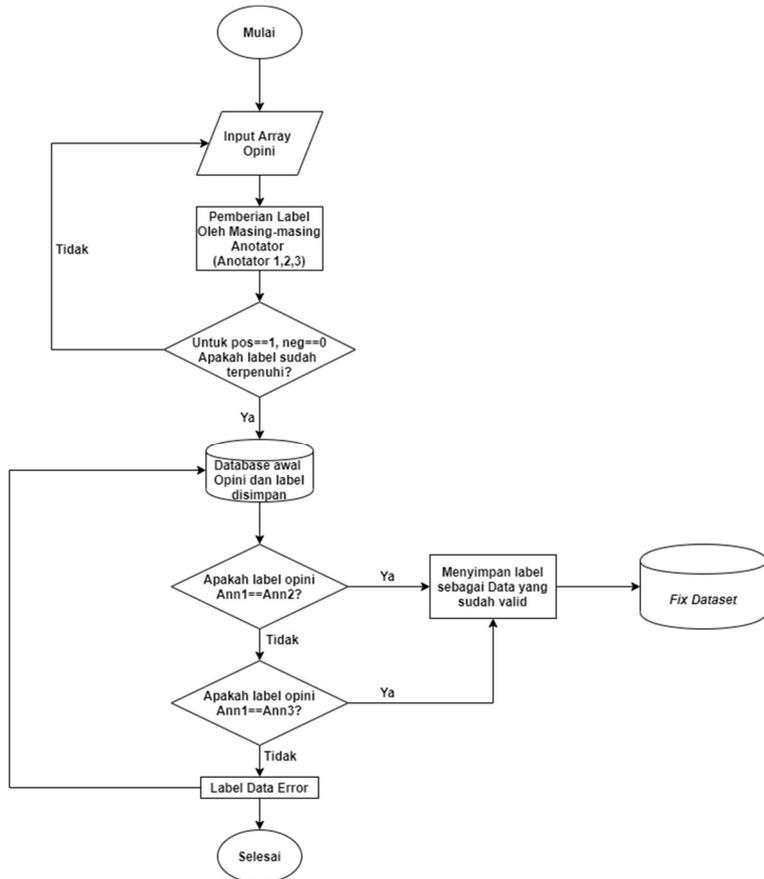
Data *dummy* yang telah dilakukan proses Filterisasi dapat ditunjukkan pada tabel 4.11 berikut:

Tabel 4.11 Hasil Filterisasi Menghilangkan *Stopwords*

['jaman','sekarang','asal','memilih','car i','pengalaman','sudah','banyak','kinerja ,','bagus','bisa','diyakini','maju','erica hyadi','menurutku','bakal','nggak','salah ,','memilihnya']
['machfud','arifin','bapak','mempunyai',' pengalaman','sangat','banyak','sekali']
['program','sembako','murah','seperti','m odel','kampanye','basi','dok','gamal','al binsaid','inovator','yang','millennial','s eperti','dokter','model','kampanye','ngga k','masuk','list','mau','program','kerja ,','realistis','misal','para','ibu','menjad i','berdaya','sehingga','sembako','mahal ,','tetap','kebeli']
['auto','milih','pak','whisnu','pilwali', 'surabaya','nanti']

4.4 Pelabelan Data

Dataset yang berasal dari facebook dan twitter dan sudah melalui proses pembersihan pada tahap pra-pemrosesan data akan dilakukan proses pelabelan. Jumlah anotator yang akan memberikan label tweet berjumlah tiga orang untuk setiap 10.000 *instances* dataset. 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 bertanda 0 dan label positif bertanda 1. Alur untuk menentukan label akhir dari sebuah tweet dapat ditunjukkan pada Gambar 4.19 berikut:



Gambar 4.19 Flowchart Anotasi Label Opini

Dibawah ini Tabel 4.12 contoh hasil pelabelan dataset *dummy* oleh Anotator yang nantinya masing-masing setiap 10.000 kalimat opini akan dianotasi oleh tiga orang:

Tabel 4.12 Hasil Pelabelan Data *Dummy* oleh Anotator

Opini	Label
jaman sekarang jangan asal memilih ya cari yg pengalaman dan sudah banyak kinerja yang bagus untuk bisa diyakini bisa maju ericahyadi menurutku juga sudah jelas dan bakal ga salah memilihnya	Positif
machfud arifin kiranya memang bapak ini mempunyai pengalaman yang sangat banyak sekaliiii	Positif
Lahhh program sembako murah spt ini maaf model kampanye basi dok gamal albinsaid Utk seorg inovator yg millennial spt dokter hrsnya model kampanye kyk gini g masuk dlm list Kita mau program kerja yg realistis Misal klo para ibu mjd lbh berdaya shg sembako mahl pun ttp kebeli	Negatif
Auto milih Pak Whisnu di Pilwali Surabaya nantiiii	Positif

4.5 Ekstraksi Fitur

Proses ekstraksi fitur yaitu pengubahan kata menjadi bentuk vektor. Proses ekstraksi fitur dalam penelitian ini menggunakan metode *Word2Vec*. Pertama akan dibuat algoritma yang berfungsi merubah kata menjadi vektor-vektor kata berdasarkan hubungan keterkaitan antar kata. Sampel aplikasi *Word2Vec* misalkan terdapat kalimat “seorang pegawai sedang ajak cawali”, “seorang pegawai mencalonkan seorang cawali”, “seorang cawali mengajukan ajak warga”. Kalimat tersebut memiliki *corpus vocabulary* sebanyak 8 kata yaitu “seorang”, “pegawai”, “sedang”, “ajak”, “cawali”, “mencalonkan”, “mengajukan”, “warga”. *Corpus* adalah himpunan dari kalimat yang berisikan *vocab/suku* kata. Kemudian *corpus* tersebut diurutkan berdasarkan alfabet maka menjadi “ajak”, “cawali”,

“mencalonkan”, “mengajukan”, “pegawai”, “sedang”, “seorang”, “warga”. Maka berdasarkan *corpus* ini diinisialisasi *neural network* memiliki 8 input neuron dan 8 output neuron, dengan inialisasi menggunakan 3 neuron pada *hidden layer*. Didapatkan matriks WI (koneksi dari layer input ke *hidden layer*) dan WO (koneksi dari *hidden layer* ke layer output) yang merupakan representasi dari V kata dan N dimensi kata menjadi ukuran dari matriks WI dan WO yaitu $V \times N$ dan $N \times V$ untuk *corpus* ini:

$$WI = 8 \times 3 \quad (1)$$

$$WO = 3 \times 8 \quad (2)$$

Sebelum *training* dijalankan kedua matriks ini diinisialisasi ke bobot yang *random* dalam pelatihan *neural network*. Contoh:

$$WI = \begin{bmatrix} -0.944491 & -0.443977 & 0.313917 \\ -0.490796 & -0.229903 & 0.065460 \\ 0.072921 & 0.172246 & -0.357751 \\ 0.104514 & -0.463000 & 0.079367 \\ -0.226080 & -0.154659 & -0.038422 \\ 0.406115 & -0.192794 & -0.441992 \\ 0.181755 & 0.088268 & 0.277574 \\ -0.055334 & 0.491792 & 0.263102 \end{bmatrix}$$

$$WO = \begin{bmatrix} 0.023074 & 0.479901 & 0.432148 & 0.375480 & -0.364732 & -0.119840 & 0.266070 & -0.351000 \\ -0.368008 & 0.424778 & -0.257104 & -0.148817 & 0.033922 & 0.353874 & -0.144942 & 0.130904 \\ 0.422434 & 0.364503 & 0.467865 & -0.020302 & -0.423890 & -0.438777 & 0.268529 & -0.446787 \end{bmatrix}$$

Dari matriks WI dan WO diatas misalkan akan dicari hubungan antara kata “cawali” dan “mengajukan” artinya, *network* harus menunjukkan probabilitas yang tinggi untuk “mengajukan” ketika “cawali” diinputkan ke *network*. Dalam istilah *word embedding*, kata “cawali” disebut sebagai kata konteks dan kata “mengajukan ” disebut sebagai kata target. Maka input vektor X menjadi $[0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]t$. Dalam matriks input tersebut hanya komponen kedua dari vektor yang bernilai 1. Ini karena kata input “cawali” yang memegang posisi nomor dua dalam daftar *corpus* yang sudah diurutkan.

Selanjutnya kata “mengajukan” disebut sebagai kata target, maka vektor targetnya Y menjadi $[0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]t$. Berdasarkan vektor input yang mewakili “cawali”, output pada neuron *hidden layer* dapat dihitung sebagai:

$$H_t = W I X_t$$

$$= [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \begin{bmatrix} -0.944491 & -0.443977 & 0.313917 \\ -0.490796 & -0.229903 & 0.065460 \\ 0.072921 & 0.172246 & -0.357751 \\ 0.104514 & -0.463000 & 0.079367 \\ -0.226080 & -0.154659 & -0.038422 \\ 0.406115 & -0.192794 & -0.441992 \\ 0.181755 & 0.088268 & 0.277574 \\ -0.055334 & 0.491792 & 0.263102 \end{bmatrix}$$

$$H_t = [-0.490796 \quad -0.229903 \quad 0.065460]$$

Dapat kita lihat pada vektor H pada neuron *hidden layer* menyerupai bobot baris kedua karena memang dasarnya *flow* inputan ke *hidden layer* hanya dengan menyalin vektor dari input kata pada layer input. Dengan melakukan hal yang sama maka dapat pula diperoleh vektor aktivasi untuk layer output dari *hidden layer* sebagai berikut:

$$= H_t W O$$

$$\begin{bmatrix} 0.023074 & 0.479901 & 0.432148 & 0.375480 & -0.364732 & -0.119840 & 0.266070 & -0.351000 \\ -0.368008 & 0.424778 & -0.257104 & -0.148817 & 0.033922 & 0.353874 & -0.144942 & 0.130904 \\ 0.422434 & 0.364503 & 0.467865 & -0.020302 & -0.423890 & -0.438777 & 0.268529 & -0.446787 \end{bmatrix} \begin{bmatrix} -0.490796 \\ -0.229903 \\ 0.065460 \end{bmatrix} x$$

$$= [0.100934 \quad -0.309331 \quad -0.122361 \quad -0.151399 \quad 0.143463 \quad -0.051262 \quad -0.079686 \quad 0.112928]$$

Karena, tujuannya adalah menghasilkan probabilitas untuk kata-kata di lapisan output, $Pr(\text{kata ke-}k \mid \text{konteks kata})$ untuk $k = 1, V$, untuk mencerminkan hubungan kata berikutnya dengan kata konteks pada input, kita memerlukan jumlah output neuron dalam lapisan output untuk ditambahkan menjadi satu. *Word2Vec* mencapai hasil ini dengan mengonversi nilai aktivasi neuron

lapisan keluaran ke probabilitas menggunakan fungsi *softmax*. Dengan demikian, output dari neuron k-th dihitung dengan ekspresi berikut di mana aktivasi (n) mewakili nilai aktivasi neuron lapisan keluaran ke-n:

$$y_k = P_r(\text{word}_k) | \text{word}_{\text{context}} = \frac{\exp(\text{activation}(k))}{\sum_{n=1}^V \exp(\text{activation}(n))}$$

Maka diperoleh probabilitas dari 8 kata pada *corpus* berikut:
[0.143073 0.094925 0.114441 0.111166 0.149289 0.122874 0.119431 0.144800]

Probabilitas yang dicetak tebal berikut merupakan kata target yang dipilih diawal inisialisasi yaitu “mengajukan”. Mengingat vektor target $[0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]t$ vektor maka *error* untuk lapisan keluaran mudah dihitung dengan mengurangi vektor probabilitas dari vektor target. Setelah *error* diketahui, bobot dalam matriks *WO* dan *WI* dapat diperbarui menggunakan *backpropagation*. Dengan demikian, *training* dapat dilanjutkan dengan menghadirkan pasangan kata target konteks yang berbeda dari *corpus*. Penjelasan contoh diatas merupakan *flow* bagaimana *Word2Vec* mempelajari hubungan antara kata-kata dan dalam proses mengembangkan representasi vektor untuk kata-kata dalam *corpus*. Sumber basis data vektor kata yang digunakan merupakan hasil dari proses pembuatan model *Word2Vec* ini menggunakan library *Gensim* dengan ukuran dimensi untuk tiap kata adalah 50. Proses ini menggunakan algoritma Skip-gram.

```
from gensim.models.Word2Vec import Word2Vec
feature_num = 100
sizewindow = 7
count_min = 2
workers_num = 2
subsampling = 1e-3
```

Gambar 4.20 Inisialisasi Parameter Ekstraksi Fitur *Word2Vec*

Berikut merupakan penjelasan dari parameter yang digunakan untuk pembelajaran model *Word2Vec* :

1. `feature_num` : Dimensi dari vektor kata.
2. `sizewindow` : Jarak maksimum antara kata saat ini dengan prediksi dalam sebuah kalimat.
3. `count_min`: Parameter yang menentukan bahwa sebuah kata akan dilakukan pembelajaran apabila kata tersebut muncul minimal dalam jumlah yang ditentukan.
4. `workers_num` : banyak utilitas ini untuk *training* model.
5. `subsampling`: Parameter ini merupakan salah satu parameter penting karena akan menentukan kandidat-kandidat yang dihasilkan dari proses prediksi menggunakan model yang telah terbentuk.

```
#Membaca Dataset CSV
with open('dataset.csv','r') as data:
    data_fix = data.readlines()
#Memanggil fungsi Token
tokenized_sents = token(data_fix)
model = Word2Vec(tokenized_sent, workers,
size, min_count, window, sample = subsampling)
vocab_w2v = 'Word2Vec_model'
#menyimpan model
model.save(vocab_w2v)
```

Gambar 4.21 *Syntax* Model *training* Ekstraksi Fitur *Word2Vec*
Code pada gambar di atas bertujuan untuk membentuk *library* vektor kata yang kemudian akan disimpan ke dalam file bernama `vocab_w2v` dengan keterangan sebagai berikut:

```
Word2Vec(vocab=2300, size=50, alpha=0.025)
```

Proses model *training* diatas 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 didapatkan

dari data latih model. Kemudian setiap kata akan diberikan nilai vektor dengan besar dimensi yang telah ditentukan. Kata yang terdapat dalam library *Word2Vec* akan langsung diubah ke dalam bentuk vektor sesuai dengan nilai yang tertera pada library. Namun, untuk kata yang tidak terdapat dalam library akan tetap diubah ke dalam bentuk vektor nol berukuran 1×50

Proses di atas juga akan melakukan pengecekan, apakah panjang array artikel tersebut sudah sesuai dengan panjang array artikel maksimal atau belum. Jika belum maka code di atas akan melakukan penambahan array nol berukuran 1×50 sebanyak kekurangannya. Berikut hasil dari ekstraksi fitur dengan data *dummy*:

```
[ -1.12407207e-02, -6.30088570e-03, -7.27833249e-03, ...
  1.24360221e-02, -4.49447567e-03,  8.02137703e-03],
[  8.13056249e-03,  2.46114377e-03, -8.43414757e-03, ...
  8.06766190e-03, -3.49627179e-03, -7.40810018e-03],
[  4.02027136e-03, -5.62059507e-03, -1.02112386e-02, ...
  1.60642981e-03, -1.92127621e-03,  7.69580901e-03],
...,
[  0.00000000e+00,  0.00000000e+00,  0.00000000e+00, ...
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00],
[  0.00000000e+00,  0.00000000e+00,  0.00000000e+00, ...
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00],
[  0.00000000e+00,  0.00000000e+00,  0.00000000e+00, ...
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00]],
```

Gambar 4.22 Array Vektor dari data *dummy* kalimat opini

4.6 Desain Model *N-Gram-Multichannel CNN*

Model untuk klasifikasi opini disini menggunakan layer *Embedding* sebagai input, diikuti oleh jaringan saraf *convolutional* satu dimensi, pooling layer, dan kemudian layer output prediksi. Ukuran kernel di lapisan konvolusi menentukan jumlah kata yang perlu dipertimbangkan ketika konvolusi dilewatkan di dokumen teks input, serta memberikan parameter pengelompokan. Jaringan saraf *convolutional multi-channel* untuk klasifikasi opini melibatkan penggunaan beberapa versi

dari model standar dengan ukuran kernel yang berbeda. Ini memungkinkan dokumen untuk diproses pada resolusi yang berbeda atau n-gram (kelompok kata) yang berbeda pada satu waktu, sementara model belajar bagaimana cara terbaik mengintegrasikan interpretasi ini.

Algoritma ini menentukan model dengan tiga saluran input untuk memproses 4-gram, 6-gram, dan 8-gram teks opini. Setiap saluran terdiri dari elemen-elemen berikut:

- *Input layer* yang mendefinisikan panjang urutan input.
- Lapisan *embedding* diatur ke ukuran kosa kata dan representasi menggunakan *Word2Vec*.
- Lapisan konvolusi satu dimensi dengan 32 filter dan ukuran kernel diatur ke jumlah kata untuk dibaca sekaligus.
- *Max Pooling Layer* untuk mengkonsolidasikan output dari lapisan *convolutional*.
- *Flatten Layer* untuk mengurangi output tiga dimensi menjadi dua dimensi untuk penggabungan.

Keluaran dari ketiga *layer* tersebut disatukan menjadi satu vektor dan diproses oleh *Dense Layer* dan sebuah *output layer*. Fungsi di bawah ini mendefinisikan dan mengembalikan model. Sebagai bagian dari pendefinisian model, ringkasan dari model yang ditentukan dicetak dan sebidang grafik model dibuat dan disimpan ke file. Arsitektur yang dibangun menggunakan model *N-Gram Multichannel CNN* dapat ditunjukkan oleh Kode 4.23 berikut:

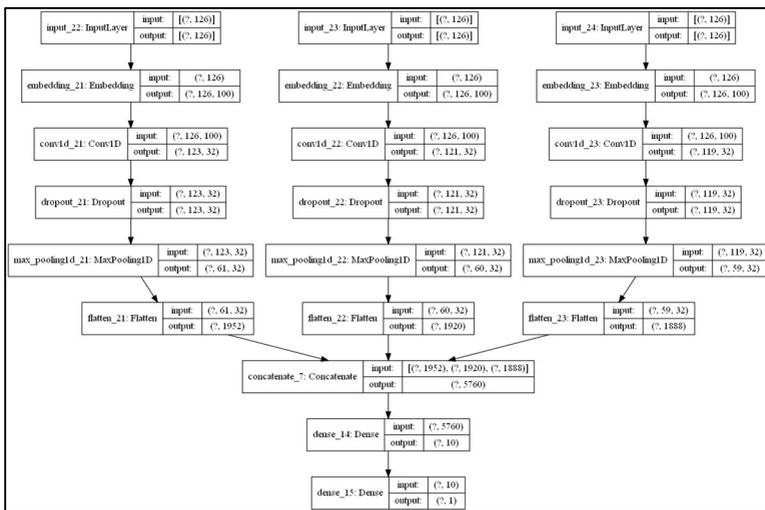
```

def define_model(length, vocab_size):
    # channel 1
    inputs1 = Input(shape=(length,))
    embedding1 = Embedding(vocab_size, 100)(inputs1)
    conv1 = Conv1D(filters=32, kernel_size=4,
activation='relu')(embedding1)
    drop1 = Dropout(0.2)(conv1)
    pool1 = MaxPooling1D(pool_size=2)(drop1)
    flat1 = Flatten()(pool1)
    # channel 2
    inputs2 = Input(shape=(length,))
    embedding2 = Embedding(vocab_size, 100)(inputs2)
    conv2 = Conv1D(filters=32, kernel_size=6,
activation='relu')(embedding2)
    drop2 = Dropout(0.2)(conv2)
    pool2 = MaxPooling1D(pool_size=2)(drop2)
    flat2 = Flatten()(pool2)
    # channel 3
    inputs3 = Input(shape=(length,))
    embedding3 = Embedding(vocab_size, 100)(inputs3)
    conv3 = Conv1D(filters=32, kernel_size=8,
activation='relu')(embedding3)
    drop3 = Dropout(0.2)(conv3)
    pool3 = MaxPooling1D(pool_size=2)(drop3)
    flat3 = Flatten()(pool3)
    # merge
    merged = concatenate([flat1, flat2, flat3])
    # interpretation
    dense1 = Dense(10, activation='relu')(merged)
    outputs = Dense(1, activation='sigmoid')(dense1)
    model = Model(inputs=[inputs1, inputs2, inputs3],
outputs=outputs)
    # compile
    model.compile(loss='binary_crossentropy',
optimizer='adam', metrics=['accuracy'])
    # summarize
    model.summary()
    plot_model(model, show_shapes=True,
to_file='model.png')
    return model
    # fit model
    model.fit([trainX, trainX, trainX],
array(trainLabels), epochs=7, batch_size=16)
    model.save('model.h5')

```

Gambar 4.23 *Syntax Model N-Gram-Multichannel CNN*

Proses diatas merupakan inisialisasi parameter untuk model CNN dengan konvolusi menggunakan N-Gram melalui beberapa *channel*. Plot dari model yang diinisialisasi disimpan ke file, dengan jelas menunjukkan tiga saluran input untuk model. Dengan perancangan model tersebut plot model dapat dilihat pada gambar berikut:



Gambar 4.24 Desain Model *N-Gram-Multichannel CNN*

Berikut merupakan penjelasan dari parameter yang digunakan untuk pembelajaran model *N-Gram-Multichannel CNN* :

1. `length`: Panjang dari kalimat dimana kita akan mengeset semua kalimat mempunyai panjang yang sama.
2. `vocab_size`: Ukuran dari kosakata. Ini diperlukan untuk mendefinisikan ukuran dari embedding layer, yang akan memiliki bentuk `[vocabulary_size, embedding_size]`.
3. `Embedding`: Ukuran dimensi vektor kata dari *word embedding*
4. `filters`: Jumlah kata yang kita inginkan untuk convolutional filters. Kita akan memiliki `num_filters` untuk setiap ukuran

spesifik. Sebagai contoh, [3, 4, 5] yang artinya kita akan mempunyai filter slide 3, 4 dan 5 untuk masing-masing kata, sebagai total dari $3 * \text{num_filters}$ filter.

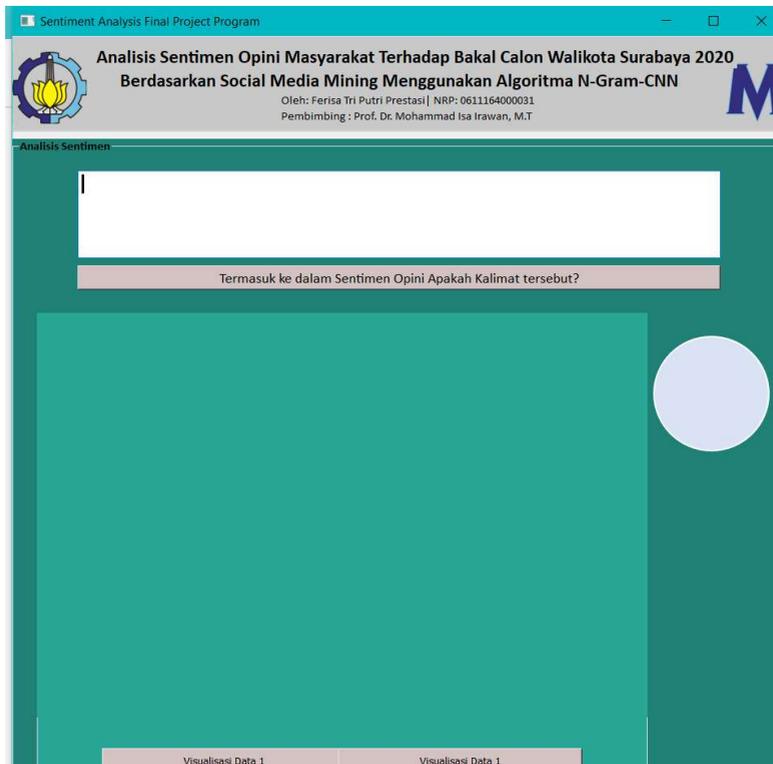
5. `kernel_size`: Jumlah dan ukuran dari *filter* kata yang digunakan.

6. `epochs`: Jumlah iterasi yang dilakukan selama pembelajaran model.

7. `num_classes` : Jumlah dari kelas di output layer, 2 dalam kasus kita (positif dan negatif).

4.7 Implementasi Desain Antarmuka atau *Graphic User Interface (GUI)*

Pada proses berjalannya sistem, untuk memudahkan pengguna dalam mengoperasikan sistem maka dirancang desain GUI (*Graphic User Interface*) atau desain antarmuka agar kemudahan dalam implementasi sistem. Dengan implementasi pengguna dapat melakukan prediksi nilai ulasan yang diberikan serta mencoba sistem verifikasi ulasan yang diterapkan. Pembuatan GUI menggunakan library *PyQt5-tools*. Tampilan desain antarmuka atau GUI dari sistem yang akan dibangun dimana sistem akan dapat diakses oleh *user* untuk mengisi kalimat opini pada kotak yang telah disediakan setelah itu menekan tombol Proses untuk mengetahui sentimen dari kalimat opini tersebut. Untuk lebih jelasnya desain dasar dapat dilihat pada Gambar 4.25 yang merupakan tampilan awal program.



Gambar 4.25 *Interface Design* Analisa Sistem

BAB V

UJI COBA DAN ANALISA PEMBAHASAN

Pada bab ini dijelaskan proses pengujian program pada data, dan pengujian hasil, beserta pembahasannya.

5.1 Deskripsi Data Uji Coba

Sebagaimana disampaikan pada bab sebelumnya, bahwa data yang digunakan untuk uji coba adalah data hasil *crawling* dari dua media sosial yaitu Facebook dan Twitter dengan menggunakan *keyword* sesuai yang ditunjukkan pada bab 1. Proses *crawling* ini dilakukan terhitung selama 77 hari atau 11 minggu sejak tanggal 1 Februari 2020 sampai dengan 17 April 2020. Proses *crawling* ini mendapatkan kalimat opini sebanyak 45.725 namun setelah di lakukan pra-proses data menjadi 42.891 kalimat. Tabel 5.1 menunjukkan rincian jumlah kalimat opini tiap *keyword* yakni nama masing-masing bakal calon Walikota Surabaya 2020.

Tabel 5.1 Rincian Jumlah Opini Masing-masing *keyword*

No	Nama Bakal Calon Walikota Surabaya 2020	Jumlah
1	Whisnu Sakti Buana	9.875
2	Dyah Katarina	1.002
3	Gamal Albinsaid	5.400
4	Machfud Arifin	8.762
5	Zahrul Azhar Asad	532
6	Hariyanto	32
7	Achmad Zakaria	254
8	Reni Astuti	121
9	Sigit Sosiantomo	103
10	Hanif Dhakiri	198
11	M. Sholeh	4.521

Tabel lanjutan dari Tabel 5.1

12	Eri Cahyadi	11.554
13	Fandi Eko Utomo	84
14	Untung Suropati	197
15	Arif Afandi	101
16	Anwar Sadad	98
17	Toni Tamatompol	57

Berdasarkan Tabel 5.1 dapat ditunjukkan bahwa nama bakal calon Walikota Surabaya 2020 yang mendapatkan urutan tiga teratas jumlah opini terbanyak dari hasil *crawling* media sosial Facebook dan Twitter adalah Eri Cahyadi dengan jumlah kalimat opini sebanyak 11.554 menempati posisi pertama, kedua Whisnu Sakti Buana dengan jumlah kalimat opini sebanyak 9.875. Kemudian disusul Machfud Arifin dengan jumlah kalimat opini sebanyak 8.762. Visualisasi dari data pada tabel 5.1 dapat dijelaskan pada gambar berikut:

```
In [40]: import pandas as pd
df = pd.read_csv('opini eri cahyadi.csv', sep=',', header=0)
df.index = df.index + 1
df

Out[40]:
```

	text	label
1	setelah ini akan ada perpisahan besar dengan W...	0
2	sopo yo kiro" seng iso ganti Bu Risma kedepan ...	0
3	mbaringene tahun ngarep wes ono pilkada maneh ...	0
4	eri cahyadi melu ngewangi pembangunan GBT gaw...	0
5	kalo semisal bu Risma ganti gapapa tapi untuk ...	0
...
11550	yang mampu untuk menunjukan hasil kerja di kot...	1
11551	beberapa masyarakat akan mendukung Cak kepal...	1
11552	pilihlah pilih pilih calon kandidat kepala bap...	1
11553	calon penerus walikota rakyat sby harus bisa k...	1
11554	bagaimana pendapat kalian rek kota sby tercin...	1

```
11554 rows × 2 columns
```

Gambar 5.1 *Dataframe* Opini Nama Bakal Calon Walikota Surabaya 2020

Dengan menggunakan *Dataframe* pandas kita juga bisa mengetahui seberapa banyak data opini masing-masing bagian positif dan negatif. Untuk lebih jelasnya dapat dilihat pada gambar 5.2 dan 5.3 berikut:

```
In [41]: df.loc[df['label'] == 0]
```

Out[41]:

	text	label
1	setelah ini akan ada perpisahan besar dengan W...	0
2	sopo yo kiro" seng iso ganti Bu Risma kedepan ...	0
3	mbaringene tahun ngarep wes ono pilkada maneh ...	0
4	eri cahyadi melu ngewangi pembangunan GBT gaw...	0
5	kalo semisal bu Risma ganti gapapa tapi untuk ...	0
...
1998	surabaya sedang dilanda kebingungan dimana aku...	0
1999	eri cahyadi bukan bekerja untuk kepentingan ra...	0
2000	kepentingan rakyat diatas segalanya diatas kep...	0
2001	pak eri cahyadi tidak pantas mlanjutkan estafe...	0
2002	pak eri cahyadi mundur saja tidak perlu mencal...	0

2002 rows × 2 columns

Gambar 5.2 *Dataframe* Opini Negatif Nama Bakal Calon Walikota Surabaya 2020

Berdasarkan output dari *Dataframe* berikut dapat dilihat bahwa opini dengan berlabel 0 (nol) atau yang termasuk ke dalam Opini Negatif sebanyak 2.002 baris. Sedangkan banyak opini positif bakal calon sebagai berikut:

```
In [42]: df.loc[df['label'] == 1]
```

Out[42]:

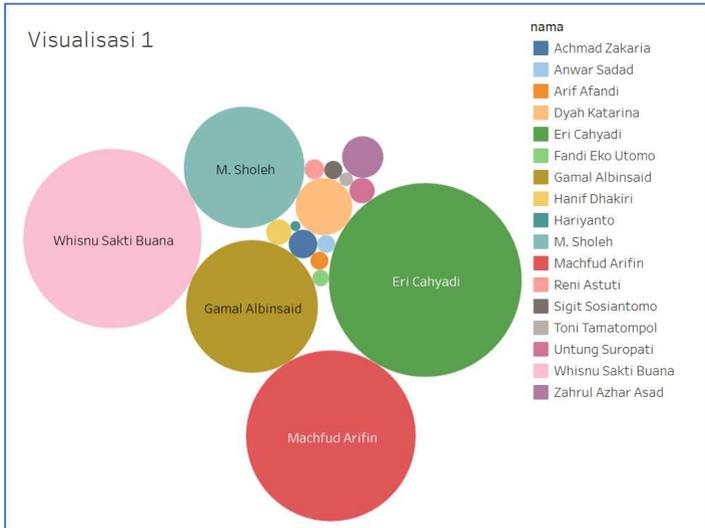
	text	label
2003	harus ada yang bisa melanjutkan Bu Risma bagus...	1
2004	Bu Risma walikota Surabaya mau berakhir masa k...	1
2005	ono calon seng bagus kih seng bakal maju dewe...	1
2006	kita sudah sama-sama merasakan peran Bu Risma ...	1
2007	ono calon seng bagus kih seng bakal maju de...	1
...
11550	yang mampu untuk menunjukan hasil kerja di kot...	1
11551	beberapa masyarakat akan mendukung Cak kapal...	1
11552	pilihlah pilih pilih calon kandidat kepala bap...	1
11553	calon penerus walikota rakyat sby harus bisa k...	1
11554	bagaimana pendapat kalian rek kota sby tercint...	1

9552 rows × 2 columns

Gambar 5.3 *Dataframe* Opini Positif Nama Bakal Calon Walikota Surabaya 2020

Berdasarkan output dari *Dataframe* berikut dapat dilihat bahwa opini dengan berlabel 1 (satu) atau yang termasuk ke

dalam Opini Positif sebanyak 9.552 baris. Selain untuk mengetahui berapa banyak masing-masing opini positif dan negatif bakal calon, maka untuk menentukan siapa bakal calon yang ramai diperbincangkan di Media sosial dalam kurun waktu 1 Februari 2020 sampai dengan 17 April 2020 dapat dilihat jelas intensitasnya pada gambar visualisasi berikut:



Gambar 5.4 Visualisasi Data Jumlah Opini Masing-masing Nama Bakal Calon Walikota Surabaya 2020

Visualisasi di atas merupakan jenis *circle bar* dimana menginterpretasikan lingkaran terbesar yang terbentuk dengan nama bakal calon di tengah lingkaran menjelaskan bahwa Bakal Calon tersebut yang ramai diperbincangkan. Selain dengan menggunakan metode visualisasi berikut, kepadatan kata yang ramai diutarakan masyarakat terkait topik Pilwali Surabaya ini dapat pula dilihat menggunakan Visualisasi *WordCloud* seperti pada gambar 5.5 berikut:

Tabel 5.2 Rincian Jumlah Opini Positif dan Negatif

No	Nama Bakal Calon Walikota Surabaya 2020	Jumlah	Opini Positif	Opini Negatif
1	Whisnu Sakti Buana	9.875	8.803	1.072
2	Dyah Katarina	1.002	679	323
3	Gamal Albinsaid	5.400	4.147	1.253
4	Machfud Arifin	8.762	7.134	1.628
5	Zahrul Azhar Asad	532	318	214
6	Hariyanto	32	17	15
7	Achmad Zakaria	254	189	65
8	Reni Astuti	121	87	34
9	Sigit Sosiantomo	103	73	30
10	Hanif Dhakiri	198	103	95
11	M. Sholeh	4.521	3.136	1.385
12	Eri Cahyadi	11.554	9.552	2.002
13	Fandi Eko Utomo	84	67	17
14	Untung Suropati	197	101	96
15	Arif Afandi	101	65	36
16	Anwar Sadad	98	77	21
17	Toni Tamatompol	57	31	26

Selanjutnya untuk potongan hasil *crawling* data oleh salah satu *keyword* nama bakal calon Walikota Surabaya 2020 yang diambil dari media sosial Facebook dan Twitter dan disimpan dalam format .CSV ditunjukkan oleh Gambar 5.6 berikut.

1	text
2	setelah ini akan ada perpindahan besar dengan Walikota Surabaya kita bu Risma akankah sama jika sudah diganti? muncul nama Rerichyadi diriku jadi yakin pasti akan sama dan bahkan akal menjadi lebih baik lagi kenapa? bu
3	sopo yo kiro? seng iso ganti Bu Risma kedapan seng dipingih seh seng nduwe pengalaman ngurus Surabaya contoh Rerichyadi bellau yo tau apik mbongan Surabaya lek wes pengalaman lebih meyakinkan
4	mbaringene tahan ngarep wes ono pilkada maneh rek opo kalian wes nemu kandidat gawe awakmu deve Rerichyadi jare wong apik lan jelas kerjone wong penasaran kambek wong? zodo...
5	Rerichyadi melu ngevangi pembangunan GBT gawe pidun wah mantap sih itu. Jarene juga mau maju nyalon Walikota Surabaya wadhah ga kebayang bakal tambah apik Surabaya wong Kinerjane dadi pejabat Surabaya yo
6	kalo semisal bu Risma gati gapapa tapi untuk pengantiannya aku sih lebih milih orang seng wes terbukti kinerjane amek wes terbukti pengalaman buat ngurus Surabaya Rerichyadi
7	lek emange wes wayah yo gapapa seng penting pengantiannya aku sih lebih milih orang seng wes terbukti kinerjane amek wes terbukti pengalaman buat ngurus Surabaya Rerichyadi
8	Bu Risma walikota Surabaya seng wajihe sak eram" mbaringene wes arep ganti pak Rerichyadi kiro" piye yo wong jarene sih apik semoga temen apik yo rek
9	ono calon seng apik kih seng bakal maju deve jenenge pak Rerichyadi koyo opo seh wong kepo" yok
10	jenenge Rerichyadi muncul berita nyabar tapi seng tak delok kok berita tentang pembangunan ning Surabaya loh pak iki sopo gatau muncul tapi berita e wes apik" mantap lah pak iki lek semisal maju nyalon wes cocok r
11	Rerichyadi kepala bapekko surabaya seng milenial lan nduwe pengalaman ngurus Surabaya dngkno ae wong melu dadi cavalkotta Surabaya pasti bakal dadi penerus bu Risma ki bu Risma kan yo wong bapekko mbiy
12	calon penerus Walikota Surabaya ono seng apik ki rek Rerichyadi setaku memang kinerja befaul baik seh dan cocok karena pengalaman pisan ketekke joss
13	Rerichyadi katanya sudah banyak rek membantu Surabaya aku lihat di berita" lokal juga memang benar mungkin kalo pak ini ikut maju nyalon jadi walikota Surabaya sudah pas nih
14	surabaya habis ini ada Pilkada pasti ada nih kandidat" yang pasti sudah kelihatan baik dan jelas kinerjanya aku aku pernah denger pak Rerichyadi soal dia ikut ngebangin pembangunan Surabaya joss nih pak Rerichyadi
15	sopo rek seng arep maju dadi walikota surabaya 2020 jarene sih seng apik pak Rerichyadi"
16	kalo dilhat" bu Risma iku walikota paling muantep tapi dibalik itu juga ada jajaran seng ikut mengembangkan Surabaya contoh pak Rerichyadi bellau juga ikut dalam perkembangan mantap sih pak Rerichyadi kalo ada yo
17	sedulit maneh wes Pilkada Bu Risma baringene ganti awakdeke kudu nduwe kandidat deve seng sekirone iso ganti bu Risma lek aku Rerichyadi iso soale wong yo melu mbongan surabaya dadi luwih apik dadi karang o
18	mbaringene wes ono pergantian walikota rek ndek Surabaya kiro" awakmu milih calon seng piye kiro" pengalaman? Kinerja wes terbukti bagus? Rerichyadi wes nduwe kabeh menurutu mantap wes pilkaha engko
19	lek semisal ono seng meyakinkan gawe ganti bu Risma yo ora popo Rerichyadi wong pengalaman dan ikut turut serta ning pembangunan iki iso didadlno contoh kinerjane apik iso dicek deve lek kepo
20	surabaya habis ini ada Pilkada pasti ada nih kandidat" yang pasti sudah kelihatan baik dan jelas kinerjanya aku aku pernah denger pak Rerichyadi soal dia ikut ngebangin pembangunan Surabaya joss nih pak Rerichyadi
21	lihat-lihat sek sebelum milih kandidat tapi teko Rerichyadi udah jelas sih dia juga ikut mengembangkan surabaya dan perkembangannya juga cukup blin Surabaya makin maju joss deh kalo Rerichyadi jadi kandidat
22	Rerichyadi arep maju nyalon dadi walikota surabaya hemm menurutu sih cocok ae soale pasti ora asal"an
23	rek lek kalian ga tega ndelok perjuangane bu Risma mbongan Surabaya sampe apik ngegne mene wayah pemilihan walikota ojo salah pilih yo rek pilih seng pengalaman dan pasti kerjone deloken pak Rerichyadi mungkin
24	kalo cari yang banyak pengalaman ikut turut membangun Surabaya Rerichyadi bagus sih menurut kinerja pembangunan di Surabaya juga wes banyak bisa tuh jadi kandidat untuk calon Walikota surabaya
25	jaman sekarang jangan asal memilih ya cari yg pengalaman dan sudah banyak kinerja yang bagus untuk bisa dyakini bisa maju Rerichyadi menurutu juga sudah jelas dan bakal ga salah memilihnya
26	sopo yo kiro? seng iso ganti Bu Risma kedapan seng dipingih seh seng nduwe pengalaman ngurus Surabaya contoh Rerichyadi bellau yo tau apik mbongan Surabaya lek wes pengalaman lebih meyakinkan
27	calon penerus Walikota Surabaya ono seng apik ki rek Rerichyadi setaku memang kinerja befaul baik seh dan cocok karena pengalaman pisan ketekke joss
28	seng digoloki pemimpin ih wong seng ga banyak bicara tapi langsung kerja dan kerjanya nyata menurutu sopo rek seng cocok dadi penganti bu Risma? seng tak sebutno mau Rerichyadi beritane model ngono penae
29	aku orang yg paling susah bilang "yes" tapi kalo jelas dan tepat yesku untukmu apalagi soal calon pemimpin kotaku Surabaya semenjeng denger Rerichyadi mau maju hemm katta #yes ku tiba" muncul itu menandakan kalo

Gambar 5.6 Potongan data hasil *crawling* format CSV

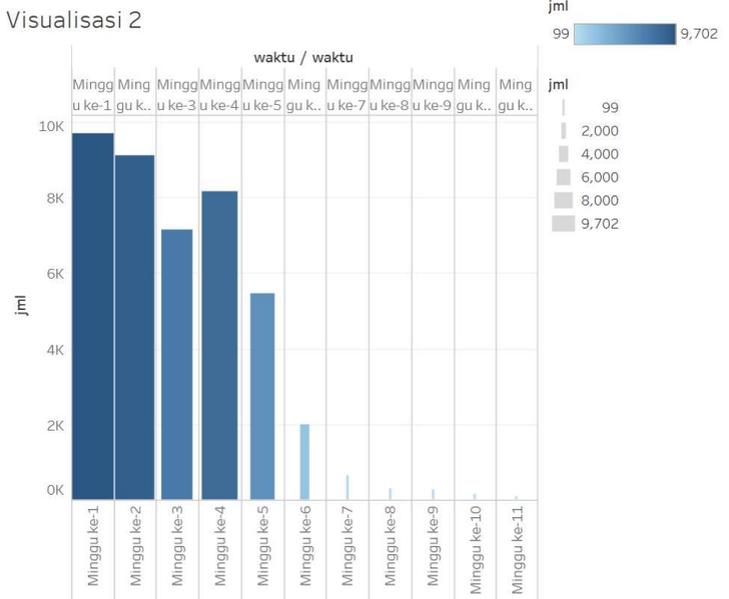
Detail perolehan jumlah data kalimat opini dari postingan dan komentar facebook serta *tweet* dari Twitter berdasarkan urutan waktu adalah ditunjukkan pada Tabel 5.2 berikut:

Tabel 5.3 Jumlah Data *Tweet* per-Minggu

No	Minggu ke-	Jumlah
1	Minggu ke-1	9702
2	Minggu ke-2	9111
3	Minggu ke-3	7154
4	Minggu ke-4	8176
5	Minggu ke-5	5460
6	Minggu ke-6	1987
7	Minggu ke-7	654
8	Minggu ke-8	311
9	Minggu ke-9	276
10	Minggu ke-10	141
11	Minggu ke-11	99

Berdasarkan Tabel 5.3 dapat ditunjukkan Visualisasi *record* data tersebut dapat dilihat pada gambar berikut:

3



Gambar 5.7 Record Data setiap Minggu

Berdasarkan statistik data berikut diketahui bahwa perolehan jumlah data kalimat opini dari postingan dan komentar facebook serta *tweet* dari Twitter dari *keyword* nama bakal calon Walikota Surabaya 2020 fluktuatif namun memang cenderung menurun drastis pada minggu ke-7 sampai dengan minggu ke-11 dikarenakan berdasarkan data research statistik Twitter dari Drone Emprit *Tren Mention topic* semua mengarah ke topik *Corona Virus* atau Covid-19. Pada penelitian ini digunakan data opini sebagai uji dengan contoh seperti pada tabel 5.4 dibawah ini:

Tabel 5.4 Data Kalimat Opini

kalo cari yang banyak pengalaman ikut turut membangun Surabaya, #ericahyadi bagus sih menurutku kinerja pembangunan
'Loh inisiatif warga ya, sudah mulai banyak yg dukung #ericahyadi #meneruskankebaikan #walikotasurabaya
'Whisnu Sakti Buana pun bertekad untuk melanjutkan keberhasilan dan berbagai capaian membanggakan
Indonesia butuh Pemimpin muda yg cerdas & penuh prestasi positif & Pemimpin Muda itu akan di awali oleh Anak Muda yg Mendunia dr. Gamal Albinsaid yg in shaa allah diberikan Amanah sbg Walikota Surabaya 2020-2025 Bismillah Aamiin Allahumaamiin

5.2 Proses Uji Coba

Untuk menampilkan detail dari berjalannya sistem yang sudah diimplementasikan pada data uji coba, maka ditampilkan hasil uji coba pada masing-masing proses.

5.2.1 Hasil Uji Coba Impor Data

Uji coba impor data kumpulan kalimat opini yang diperoleh dari media sosial Facebook dan Twitter dengan menggunakan *library* pandas dalam bentuk *Dataframe* adalah sebagai berikut:

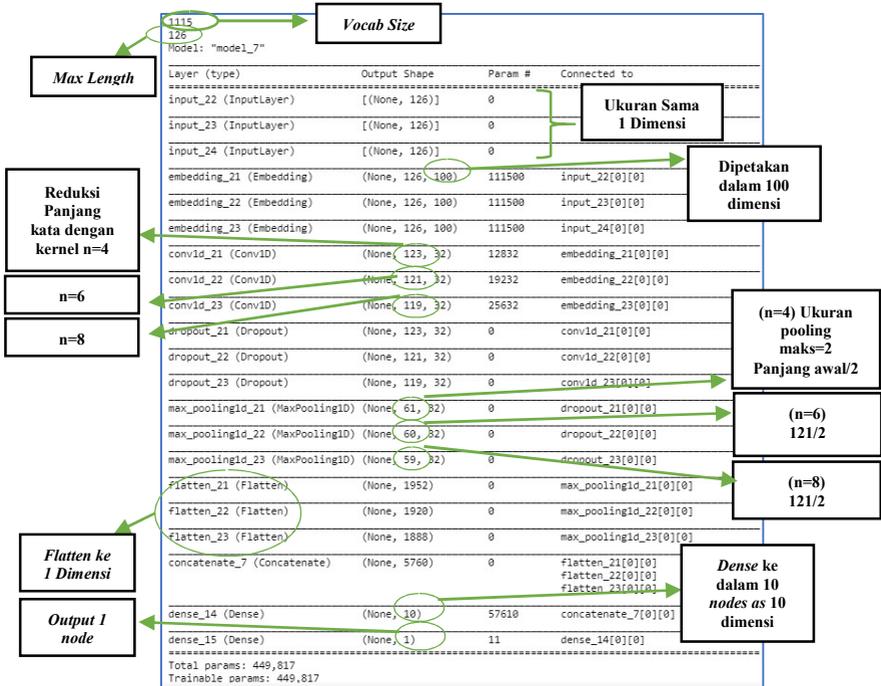
Hasil uji coba praproses data sebagaimana proses berjalannya dijelaskan pada bab sebelumnya dan disimpan dalam *list* baru adalah sebagai berikut.

```
[ 'seng digoleki pemimpin iku seng ga bicara kerjanya menurutmu sopo rek seng cocok dadi penggantine bu
u risma seng sebutno beritane model ngono ', 'org yg diidolakan yg pengalaman gercep memimpin surabay
a ga diragukan bu risma ', 'penggantine bu risma pastinya cari pengalaman membangun surabaya beliau p
engalaman berita baca positif pembangunan surabaya ', 'sahabat santri eri cahyadi ramadhan mulia penu
h keberkahan dimana amal kebaikan manusia dilipatgandakan allah swt berbagi tengah pandemi ', 'suraba
ya wes apik ngene yo gara bu risma dibalik iku ono seng melu ngewangi salah satune wonge nduwe pengal
aman akeh gawe mbangun suroboyo rek lek maju pilkada wonge wes cocok iki ', 'rek lek ga tega ndelok p
erjuangane bu risma mbangun suroboyo sampe apik ngene mene wayahe pemilihan walikota ojo salah pilih
yo rek pilih seng pengalaman kerjone deloken iso gawe pilihanmu ', 'dukung ya wuhh dukung sampeyan jg
', 'surabaya habis pilkada kandidat kinerjanya denger ngembangin pembangunan surabaya joss ', 'cak op
o isihndigoreng lek warga wes seneng yo ga salah soale cak iki wes nduwe pengalaman gawe noto suroboy
o terbukti ga ngomong tok iso bu risma ', 'yg susah bilang yes kalo yesku untukmu calon pemimpin kota
ku surabaya semenjak denger maju ', 'sedilut maneh wes pilkada bu risma baringene ganti awakdewe kudu
nduwe kandidat dewe seng sekirone iso ganti bu risma lek iso soale wonge yo melu mbangun suroboyo dad
i luwih apik dadi opo ', 'selamat pendidikan nasional ning ngarso sung tuladha teladan ning madya man
gun karsa tengah membangun semangat ntut wuri handayani dorongan ki hajar dewantara ', 'dirumah aja m
engurangi penyebaran virus covid rungokno bapake rek xc xa xe xa ', 'jenenge muncul berita nyebar sen
g delok berita pembangunan ning suroboyo iki sopo gatau muncul berita wes apik mantap iki ', 'kelihat
anya surabaya menemukan calon risma surabaya wes rame gini kepala bapekko ya semoga yg diharapkan war
ga terwujud ', 'membantu surabaya berita lokal kalo maju nyalon walikota surabaya pas ', 'ndek suraba
ya pembangunan wes apik butuh pemimpin seng iso neruso pembangunan iki luwih apik wes melu ngewangi
```

Gambar 5.10 *List* Hasil Praproses Data

5.2.3 Hasil Uji Coba Algoritma

Setelah dilakukan proses pembelajaran maka didapatkan model yang telah menyesuaikan data yang dimasukkan. Model *N-Gram-Multichannel CNN* tersebut perlu diuji akurasiya sehingga dapat disimpulkan bahwa model yang dihasilkan merupakan model yang sesuai atau tidak. Berikut ringkasan (*summary*) model *N-Gram-Multichannel CNN* :



Gambar 5.11 Hasil Ringkasan Model

Untuk penjelasan lebih lanjut mengenai sampel perhitungan model dapat dilihat pada Lampiran D. Untuk training awal didefinisikan terdapat 2 komponen penting yakni teks yang berisikan *alphabet* dan *Max length* atau Panjang maksimum dokumen untuk membuat semua dokumen teks mempunyai ukuran yang sama yang nantinya akan ditaruh pada *batch*.

Pengujian/validasi dilakukan dengan menggunakan data uji yang telah dikelompokkan sebelumnya. Data uji dan data latih merupakan data yang saling independen sehingga dapat dilihat kemampuan model untuk menangani data yang baru. Dengan menggunakan Algoritma *N-Gram-Multichannel CNN* umumnya beberapa algoritma klasifikasi memerlukan beberapa parameter

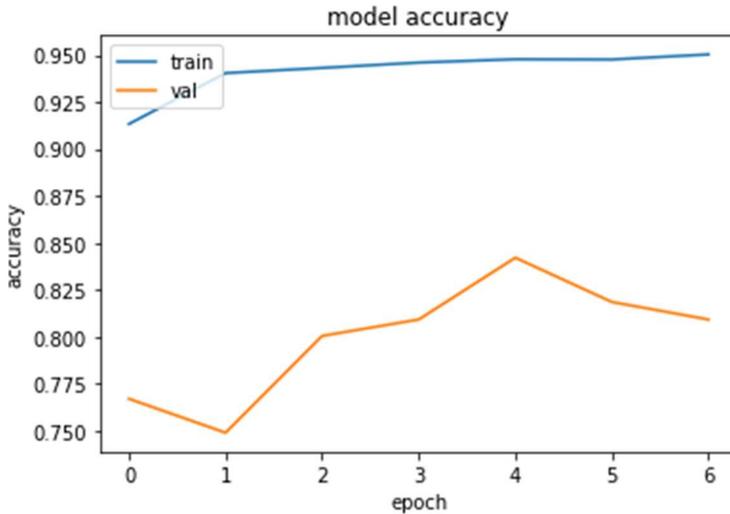
yakni jumlah *hidden layer* dan *learning rate* pada *neural network*. Pada *training* model ini data latih dilakukan *splitting* untuk data validasi yang digunakan untuk mencari parameter yang paling baik untuk sebuah algoritma klasifikasi sebesar 10% dari data latih dengan *Syntax validation_split=0.1*

Berdasarkan hasil implementasi yang dilakukan didapatkan keakurasian tiap iterasi untuk tiap jenis data. Dalam Tugas Akhir dengan beberapa percobaan. Pada tabel 5.18 diperoleh akurasi dan *loss* berdasarkan *running* model pertama menggunakan 10 *epoch*/10 iterasi menunjukkan hasil akurasi model terhadap data latih dan data uji dataset yang dilakukan pada percobaan pertama.

Tabel 5.5 Hasil akurasi model terhadap data latih dan data uji dataset percobaan pertama

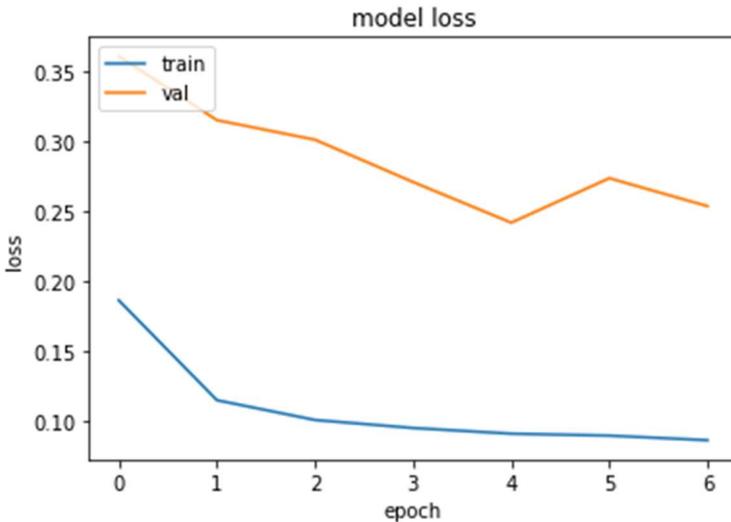
<i>Epoch</i>	<i>Train accuracy</i>	<i>Train loss</i>	<i>Validation accuracy</i>	<i>Validation Loss</i>
1	0.9133	0.1862	0.7669	0.3606
2	0.9404	0.1148	0.7488	0.3149
3	0.9432	0.1007	0.8004	0.3010
4	0.9460	0.0950	0.8091	0.2707
5	0.9478	0.0909	0.8421	0.2417
6	0.9477	0.0895	0.8185	0.2735
7	0.9504	0.0863	0.8091	0.2535

Untuk gambaran Kurva Pembejaran (*Learning Curve*) yang menggambarkan akurasi pada model yang dibuat pada percobaan *running* model pertama dapat dilihat pada gambar berikut:



Gambar 5.12 Grafik akurasi model terhadap data latih dan data uji dataset percobaan pertama

Dengan melihat pergerakan akurasi model pada percobaan pertama berikut, selanjutnya dengan menggunakan kurva pembelajaran yang melihat *loss* yang merupakan suatu ukuran dari sebuah error yang dibuat oleh network, dan tujuannya adalah untuk meminimalisirnya. *Loss* dihitung pada metrik dimana parameter model dioptimalkan.



Gambar 5.13 Grafik pergerakan *loss* pada percobaan pertama

Akurasi model yang diperoleh paling besar dihasilkan sebesar 93.93% untuk data latih dan 96.47% untuk data uji. Hasil tersebut diperoleh pada iterasi ke 6. Dapat dilihat pada gambar berikut :

```
# evaluate model on training dataset
_, acc = model.evaluate([Xtrain,Xtrain,Xtrain], ytrain, verbose=0)
print(' Train Accuracy: %f ' % (acc*100))

Train Accuracy: 93.929237

# evaluate model on test dataset
_, acc = model.evaluate([Xtest,Xtest,Xtest], ytest, verbose=0)
print(' Test Accuracy: %f ' % (acc*100))

Test Accuracy: 96.468121
```

Gambar 5.14 Akurasi pada data latih dan data validasi ke-1

Berdasarkan hasil diatas dapat disimpulkan bahwa model awal *overfit* yakni dapat diidentifikasi dari *learning curve* dari plot *training loss* terus berkurang dengan epoch sampai dengan 4 kemudian naik lagi namun belum menunjukkan stabilitas. *Overfitting* mengacu pada model yang telah mempelajari dataset

training terlalu baik, termasuk noise statistik atau fluktuasi acak dalam dataset *training*. Masalah dengan *overfitting*, adalah bahwa semakin terspesialisasi model pada training data, semakin kurang baik untuk bisa digeneralisasikan ke data baru, menghasilkan peningkatan error generalisasi. Peningkatan error generalisasi ini dapat diukur dengan kinerja model pada data validasi.

Selanjutnya, pada percobaan kedua training model ini data latih dilakukan *dropout regularization*. *Dropout* adalah teknik regularisasi jaringan syaraf dimana beberapa neuron akan dipilih secara acak dan tidak dipakai selama pelatihan. Neuron-neuron ini dapat dibuang secara acak. Hal ini berarti bahwa kontribusi neuron yang dibuang akan diberhentikan sementara jaringan dan bobot baru juga tidak diterapkan pada neuron pada saat melakukan backpropagation. *Dropout* merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Dropout* mengacu kepada menghilangkan neuron yang berupa *hidden* maupun *layer* yang *visible* di dalam jaringan.

Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara dari jaringan yang ada. Neuron yang akan dihilangkan akan dipilih secara acak. Setiap neuron akan diberikan probabilitas yang bernilai antara 0 dan 1. Selain itu juga dengan dirubah berupa penambahan 10% *splitting* untuk data validasi yang digunakan untuk mencari proporsi representasi yang paling baik untuk sebuah algoritma klasifikasi menjadi sebesar 20% dari 80% total data yang digunakan untuk data latih agar nantinya dapat merepresentasikan data validasi dari dengan `Syntax validation_split=0,2`. Berdasarkan hasil implementasi yang dilakukan didapatkan keakurasian tiap iterasi untuk tiap jenis data. Dalam Tugas Akhir dengan beberapa percobaan. Pada tabel 5.6 diperoleh akurasi dan *loss* berdasarkan *running* model kedua menggunakan 17 *epoch*/17 iterasi dengan

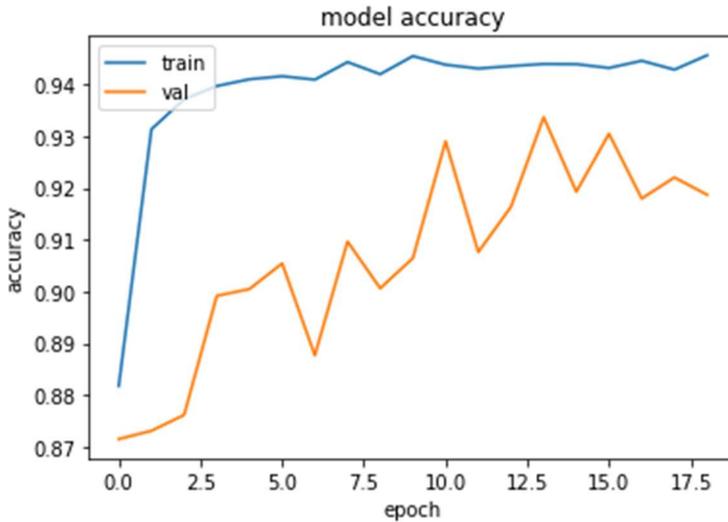
fungsi *callback early stopping Keras Tensorflow* menunjukkan akurasi model selama proses pembelajaran yang dilakukan pada percobaan kedua untuk lampiran visualisasi running model dapat dilihat pada Lampiran E, tabel penjabarannya sebagai berikut:

Tabel 5.6 Hasil akurasi model terhadap data latih dan data uji dataset percobaan kedua

<i>Epoch</i>	<i>Train accuracy</i>	<i>Train loss</i>	<i>Validation accuracy</i>	<i>Validation Loss</i>
1	0.8817	0.2723	0.8715	0.2282
2	0.9314	0.1400	0.8731	0.2232
3	0.9372	0.1182	0.8761	0.1815
4	0.9397	0.1077	0.8992	0.1517
5	0.9410	0.1032	0.9005	0.1425
6	0.9416	0.0990	0.9054	0.1273
7	0.9409	0.0968	0.8877	0.1521
8	0.9443	0.0943	0.9097	0.1121
9	0.9420	0.0927	0.9006	0.1159
10	0.9455	0.0906	0.9065	0.1138
11	0.9438	0.0899	0.9290	0.0932
12	0.9431	0.0893	0.9076	0.1111
13	0.9436	0.0890	0.9164	0.1022
14	0.9440	0.0876	0.9337	0.0921
15	0.9439	0.0860	0.9193	0.0999
16	0.9432	0.0860	0.9305	0.0892
17	0.9446	0.0848	0.9180	0.0984
18	0.9429	0.0851	0.9220	0.0973
19	0.9456	0.0831	0.9187	0.1135

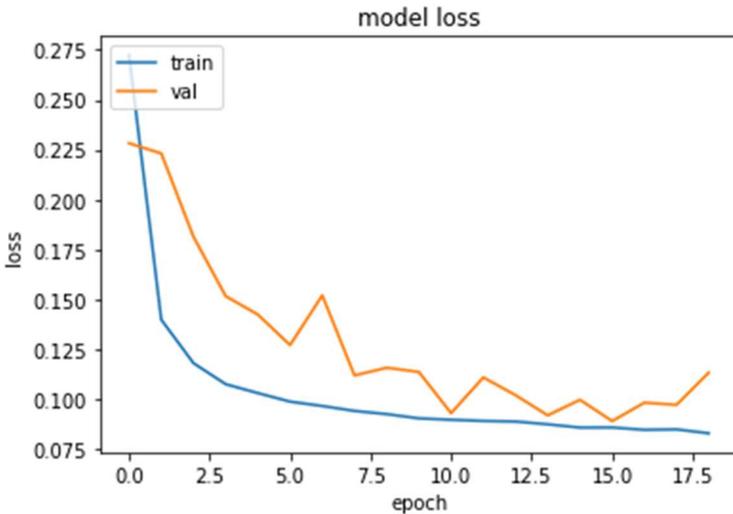
Untuk Kurva Pembelajaran (*Learning Curve*) yang menggambarkan akurasi pada model yang dibuat pada percobaan

running model kedua dapat dilihat pada gambar berikut:



Gambar 5.153 Grafik akurasi model terhadap data latih dan data uji dataset percobaan kedua

Dengan melihat pergerakan akurasi model pada percobaan kedua berikut, selanjutnya dengan menggunakan kurva pembelajaran yang melihat *loss* diperoleh gambar berikut:



Gambar 5.16 Grafik pergerakan *loss* pada percobaan kedua

Akurasi model yang diperoleh paling besar dihasilkan sebesar 96.12% untuk data latih dan 94.38% untuk data uji. Hasil tersebut diperoleh pada iterasi ke 10. Dapat dilihat pada gambar berikut :

```

In [90]: # evaluate model on training dataset
loss, acc = model.evaluate([Xtrain,Xtrain,Xtrain], ytrain, verbose=0)
print(' Train Accuracy: %f' % (acc*100))

Train Accuracy: 96.118426

In [91]: # evaluate model on test dataset
loss, acc = model.evaluate([Xtest,Xtest,Xtest], ytest, verbose=0)
print(' Test Accuracy: %f' % (acc*100))

Test Accuracy: 94.375145

```

Gambar 5.17 Akurasi pada data latih dan data validasi ke-2

Berdasarkan hasil diatas dapat disimpulkan bahwa model kedua *goodfit* yakni dapat diidentifikasi dari *learning curve* plot *train loss* menurun ke titik stabilitas. Plot *val loss* berkurang ke titik stabilitas dan antara plot *train loss* dan *val loss*

memiliki sedikit *gap* (celah). *Good fit* sendiri merupakan suatu keadaan yang diidentifikasi oleh *train loss* dan *val loss* yang menurun ke titik stabilitas dengan jarak minimum antara dua nilai *loss* akhir.

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini diuraikan kesimpulan dari serangkaian penelitian Tugas Akhir yang dilakukan. Kesimpulan dari hasil pengujian serta analisis dirangkum dan diwujudkan dalam bab ini. Selain itu saran dan pengembangan untuk penelitian di masa mendatang akan dituliskan untuk mempermudah peneliti selanjutnya dalam mengisi celah-celah penelitian Tugas Akhir ini.

6.1 Kesimpulan

Setelah melalui serangkaian tahap, dari perancangan, pengujian, hingga analisis didapatkan kesimpulan sebagai berikut.

1. Cara menerapkan analisis sentimen opini masyarakat terhadap bakal calon walikota Surabaya 2020 yaitu dengan melakukan *crawling* data, pra-pemrosesan data yang meliputi penghapusan url, penghapusan tanda, penghapusan huruf yang berulang, *case folding*, tokenisasi, *spelling normalization*, dan filterisasi, selanjutnya pelabelan data, ekstraksi fitur, dan yang terakhir pembentukan model klasifikasi menggunakan algoritma *N-Gram-Multichannel CNN* meliputi konfigurasi *Hyperparameter* pada layer konvolusi dan *embedding*, *Max Pooling*, *Flatten*, dan *Dense* dengan mengatur fungsi aktivasi. Tahapan analisis sentimen ini dapat diterapkan dengan baik yang menghasilkan data pelabelan didapatkan 8.312 dataset negatif dan 34.579 dataset positif. Kata yang sering muncul pada dataset Negatif adalah “salah” sebanyak 18.620 kata, “pencitraan” sebanyak 6.023 kata, “jelek” sebanyak 5.879 kata. Pada dataset Positif terdapat kata “bagus” sebanyak 31.418 kata,

“cocok” sebanyak 20.800 kata, dan “terbukti” sebanyak 9.679 kata.

2. Hasil akurasi model terhadap data latih dan data uji dataset opini masyarakat berdasarkan *Social Media Mining* terhadap bakal calon Walikota Surabaya 2020 dalam menganalisis sentimen menggunakan Algoritma *N-Gram-Multichannel CNN* diperoleh akurasi model terhadap data latih sebesar 94.38% untuk data latih dan sebesar 96.12% untuk data uji dengan *Learning Curve* yang menunjukkan *Good fit* setelah dilakukan *Tuning Parameter*.

6.2 Saran

Adapun saran yang dapat diberikan untuk penelitian *social media mining* dikemudian hari, dirangkum dalam pernyataan di bawah ini.

1. Hendaknya menggunakan metode yang lebih mempertimbangkan kalimat bergaya bahasa satire dalam perhitungan polaritas opini.
2. Mengembangkan (*deploy*) model dengan mengubah ukuran kernel (jumlah n-gram) yang digunakan oleh *channels* dalam model untuk melihat bagaimana hal itu berdampak pada *skill* model dan menganalisa Tuning Hyperparameter yang lain apabila di *epoch* selanjutnya terjadi *overfit*.
3. Mengembangkan analisa dengan menggunakan lebih banyak atau lebih sedikit *channel* juga dengan mengembangkan kedalaman yang dirancang dalam model guna melihat seberapa baik pengaruh terhadap model.
4. Analisis pengambilan keputusan dengan menambahkan usulan strategi terbaik dengan berpatokan pada *opportunity mining* perlu dipertimbangkan untuk melengkapi analisis yang sudah dipaparkan.

DAFTAR PUSTAKA

- [1] S. W. A. D. F. N. Sihwi, “Analisa Sentimen Masyarakat terhadap Calon Presiden Indonesia 2014 berdasarkan Opini dari Twitter Menggunakan Metode Naïve Bayes Classifier,” *Research Gate Publication*, 2016.
- [2] R. Zafarani, M. A. Abbasi dan H. Liu, *Social Media Mining : An Introduction*, London: Cambridge University Press, 2014.
- [3] M. I. Irawan, R. Wijayanto, M.L.Shahab, N.Hidayat, A.M.Rukmi, “Implementation of social media mining for decision making in product,” *Journal of Physics: Conference Series*, p. Conf. Ser. 1490 012068, 2020.
- [4] B. Y. R. F. R. J. D. T. R. M. M. Y. F. Haryanto, “Facebook Analysis of Community Sentiment on 2019 Indonesian Presidential Candidates from Facebook Opinion Data,” *ScienceDirect Procedia Computer Science The Fifth Information System International Conference 2019.*, 2019.
- [5] F. H. Listanto, “Peran Televisi Dalam Masyarakat Citraan Dewasa ini, Sejarah. 6 Perkembangan dan Pengaruhnya,” *Jurnal Desain Komunikasi Visual Nirmana, Vol.1, No.2, (2016)* , 2016.
- [6] G. A. Buntoro, “Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter,” *Researchgate, INTEGER: Journal of Information Technology 2.1*, 2019.
- [7] G. Asrofi, “Analisis Sentimen Calon Gubernur Jawa Timur 2018 dengan Metode Naïve Bayes Classifier,” *Journal of Informatics Pelita Nusantara*, 2019.
- [8] R. Ardiansyah, “Analisis Sentimen Calon Presiden dan Wakil Presiden Periode 2019-2024 Pasca Debat Pilpres di Twitter,” *ScientiCO : Computer Science and Informatics Journal*, vol. 2, no. 1, pp. E-ISSN: 2620-4118., 2019.
- [9] T. D. S. S. Y. S. Lukmana, “Analisis Sentimen Terhadap Calon Presiden 2019 dengan Support Vector Machine di Twitter,”

Seminar Nasional Penelitian Pendidikan Matematika (SNP2M) 2019 UMT, 2019.

- [10] Z. H. K. Drus, "Sentiment Analysis in Social Media and Its Application Systematic Literature Review," *ScienceDirect Procedia Computer Science The Fifth Information System International Conference 2019*, 2019.
- [11] B. I. K. J. W. K. Jang, "Word2vec Convolutional Neural Network for Classification of News Article and Tweets.," *PLoS ONE 14(8): e0220976.*, 2019.
- [12] Y. Kim, "Convolutional Neural Network for Sentence Classification," *arXiv:1408.5882*, 2014.
- [13] M. M. K. M. Rozi, "Opinion mining on book review using CNN-L2-SVM algorithm.," *Journal of Physics: Conference Series. Pg. 012004.*, vol. 974, p. 012004, 2018.
- [14] I. E. Firdausi, "Analisis Sentimen Tanggapan Pelanggan Operator Telekomunikasi di Twitter dengan Algoritma DCNN-SVM.," 2019.
- [15] I. Z. A. A. R. F. K. M. A. D. Mukhlash, "Opinion Mining on Book Review using CNN-LSTM," *International Journal of Machine Learning and Computing Pg. 437-441*, vol. 8, pp. 437-441, 2018.
- [16] W. M. M. Budiharto, "Prediction and Analysis of Indonesia Presidential Election from Twitter using Sentiment Analysis.," *SpringerLink Journal of Big Data.*, 2018.
- [17] A. D. K. Lestari, "Summarizing Nitizens' Sentiment Towards the 1st Indonesian Presidential Debate using Lexicon Sentiment Analysis.," *IOP Conference Series: Materials Science and Engineering 546 052041.*
- [18] A. M. K. a. M. Haenlein, "Users of the world, unite! the challenges and opportunities of social media," *Business horizons*, vol. 1, pp. 59-68, 2010.
- [19] D. Sarkar, "Text Analytics with Python.," *New York: Springer Science+Business Media Inc.*, 2016.

- [20] M. Bonzanini., “Mastering Social Media Mining with Python.,” *Birmingham: Packt.*, 2016.
- [21] M. S. d. D. S. R. A. C. Pandey, “Twitter sentiment analysis using hybrid cuckoo search method,” *elsevier, Vol. 53, pp. 764-779.*, vol. 53, pp. 764-779, 2017.
- [22] A. Ni'matul, “Analisis Sentimen Pada Teks Ulasan Pelanggan E-Commerce Berdasarkan Rating Menggunakan N-Gram dan Neuro-Fuzzy,” 2019.
- [23] J. W. Patihullah, “Hate Speech Detection for Indonesia Tweets Using Word,” *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, Vol. %1 dari %2Vol.13, No.1, January 2019, pp. 43~52, no. ISSN (print): 1978-1520, ISSN (online): 2460-7258, 2019.
- [24] A. R. S. R. V. S. d. F. S. P. Nakov, “SemEval-2016 Task 4 : Sentiment Analysis in Twitter,” pp. 1-18, 2016.

LAMPIRAN A. Tabel Perbandingan Penelitian Terdahulu

NO	JUDUL PENELITIAN	PENULIS	TAHUN	Akurasi
1.	<i>Analisis Sentimen Tanggapan Pelanggan Operator Telekomunikasi di Twitter dengan Algoritma DCNN-SVM</i>	Inayah Eka Firdausi	2019	63%
2.	<i>Opinion Mining on Book Review using CNN-LSTM</i>	Imam Mukhlash, dkk	2018	99.55%
3.	Analisis Sentimen Calon Gubernur DKI Jakarta 2017 di Twitter	Ghulam Asrofi Buntoro	2019	95%

4.	<i>Analisis Sentimen Calon Gubernur Jawa Timur 2018 dengan Metode Naïve Bayes Classifier</i>	Ghulam Asrofi Buntoro	2019	77%
5.	<i>Facebook Analysis of Community Sentiment on 2019 Indonesian Presidential Candidates from Facebook Opinion Data</i>	Haryanto, Budi, Yova Ruldeviyani, Fathur Rohman, dkk	2019	45%

LAMPIRAN B. *Corpus* Pengkoreksian Kata

- 'tidak': ['gak', 'gk', 'tdk', 'gx', 'ga', 'nggak', 'enggak', 'g', 'engga', 'ngga', 'tyda', 'tydac', 'tydak', 'ngak', 'ngk', 'kagak', 'nggk']
- 'lihat': ['lihat', 'liht', 'lhat', 'lht'], 'sedikit': ['dikit', 'sdkt', 'sdikit', 'sokit']
- 'sebelum': ['sblm', 'sbelum', 'sblum', 'seblum', 'sebelm']
- 'dalam': ['dlm', 'dlam', 'dalm']
- 'pencitraan': ['pncitraan', 'pnctraan', 'pnctrn']
- 'cocok': ['ccok', 'cck', 'chocok']
- 'habis': ['hbis', 'hbs', 'abis']
- 'masa': ['msa']
- 'emosi': ['esmosi']
- 'hampir': ['hmpr', 'hmpir']
- 'tidur': ['tdr', 'tdur']
- 'tahun': ['thun', 'thn', 'th']
- 'dong': ['donk']
- 'ganti': ['gnti', 'gnt', 'gonta']
- 'daripada': ['drpd', 'dripada', 'drpda', 'drpada']
- 'jabatan': ['jbtan', 'jbatan', 'jabat']
- 'kangen': ['kgn']
- 'hilang': ['ilang', 'hlang', 'hlng', 'hlg']
- 'percaya': ['prcaya', 'prcy']
- 'bong': ['boong', 'bhng', 'bhong']
- 'sekarang': ['skr', 'skarang', 'skrng', 'skrg']
- 'tanggal': ['tgl', 'tggl', 'tnggal', 'tanggl']
- 'hey': ['hai', 'hy', 'hey', 'hayy', 'haii', 'haii', 'hi']
- 'pembaruan': ['penbaruan', 'pambaruan']
- 'ohh': ['owh', 'oh', 'och', 'ouch', 'ooh', 'oohh']
- 'kita': ['qt', 'qta']
- 'pernah': ['prnah', 'prnh']
- 'favorite': ['fav', 'favorit']
- 'nonton': ['nnton']
- 'kepemimpinan': ['kpmmpinan', 'kpemimpinan', 'kpmimpnan']
- 'padahal': ['pdhl', 'pdhal', 'pdahal']
- 'pada': ['pda']

- 'telah': ['tlah', 'tlh']
- 'penuh': ['full', 'pnuh']
- 'makin': ['mkin']
- 'punya': ['pnya', 'pny']
- 'bawah': ['bwh', 'bwah']
- 'asli': ['aseli']
- 'warga': ['wrg', 'warga', 'wargo']
- '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': ['tahn', 'taah', 'taahn']
- 'begini': ['bgini', 'gini', 'ginii', 'giinii', 'giini']
- 'cacat': ['cacad']
- 'gaes': ['gaess', 'gaeess', 'gaees', 'gais', 'guys', 'guy', 'ges', 'gaiiss', 'gaiis', 'gaiiss']
- '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']
- 'kembali': ['kembali', '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']
- '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', 'kakang', 'om', 'kaka', 'ms']
- 'jadi': ['jd', 'jdi']
- 'begitu': ['gitu', 'gtu', 'gt']
- 'semua': ['smua']
- 'lama': ['lm', 'lma', 'luamaa', 'lamaa']
- 'sama': ['sm', 'ama', 'sma']
- 'agak': ['rada', 'agk']
- 'teman': ['temen', 'tmn', 'tmen']
- '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']
- 'calon': ['clon', 'cln']
- 'komplain': ['complain']
- 'bakal': ['bkl', 'bkal']

- 'beberapa': ['bbrp', 'bbrapa']
- 'berapa': ['brp', 'brapa']
- 'dan': ['n', 'dn']
- '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', 'cepat']
- 'respon': ['respond', 'rspon', 'respn', 'response', 'response']
- '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']
- 'antusias': ['antsias']
- 'jangan': ['jgn', 'jngn', 'jngan']
- 'kenapa': ['knp', 'knpa', 'knapa', 'napa', 'nape']
- 'jika': ['kalau', 'klo', 'kalo', 'kl', 'klau']
- 'deh': ['dech'], 'disini': ['dsni', 'dsini']

- 'pantas': ['pntes', 'pnts', 'pntas']
- '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', 'nyampe', 'sampe']
- 'iya': ['ya', 'iy', 'y', 'iye', 'iyak', 'iyyak', 'iyyakk', 'iyo', 'yo', 'ya',
- 'yaa', 'yyaa', 'yah', 'yahh', 'yaahh', 'yaah']
- 'pakai': ['pkai', 'pake', 'pke', 'pakek']
- 'maju': ['mju', 'majuk']
- 'pencalonan': ['pnclonan', 'pnclonan']
- 'jelek': ['jelel', 'jele', 'jelk', 'jlek']

LAMPIRAN C. *Corpus Stopword*

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, bagi, bagaikan, bagaimana, bagaimanakah, bagaimanapun, bagaimanakah, 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, berakhirilah, 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, didapatkan, 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, 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, keterlaluhan, 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, menandakan, 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, sekedar, sekadarnya, sekali, sekali-kali, sekalian, sekaligus, sekalipun, sekarang, sekaranglah, sekecil, seketika, sekiranya, sekitar, sekitarnya, sekurang-kurangnya, sekurangnyanya, 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, seperti halnya, 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, tambahannya, 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, iftt, 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, yauh, 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, hmmm.

LAMPIRAN D. Proses Model *N-Gram-Multichannel CNN*

Dalam teknik ini, dipetakan setiap kata dari 2 contoh kalimat ke vektor *embedding*. Dalam *bag of word* panjang vektor ditentukan oleh jumlah kata unik dalam korpus yang terdapat pada Lampiran C. *Word2Vec embeddings* adalah *pre-trained embedding* yang telah ditentukan dengan cara yang tidak diawasi (*unsupervised*). Tahapan ini nantinya berada pada layer konvolusi dengan adanya operasi join operator \oplus dengan persamaan sebagai berikut :

$$x_{i:n} = x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n$$

Vektor-vektor ini memiliki karakteristik yang sangat bagus, kata-kata konteks yang mirip cenderung memiliki vektor-vektor yang linear dan mengarahkan ke arah yang kira-kira sama. Penjelasan beserta ilustrasi dapat dilihat sebagai berikut:

	x_1	x_2	x_3	
“whisnu”	0.490796	-0.229903	0.065460	} <i>Word2Vec</i>
“mengajukan”	0.104514	-0.463000	0.079367	
“warga”	-0.055334	0.491792	0.263102	

Dengan menggabungkan 3 kata berikut menjadi sebuah kalimat “cawali mengajukan warga” menghasilkan vektor:

$$[0.539976 \quad -0.201111 \quad 0.407929]$$

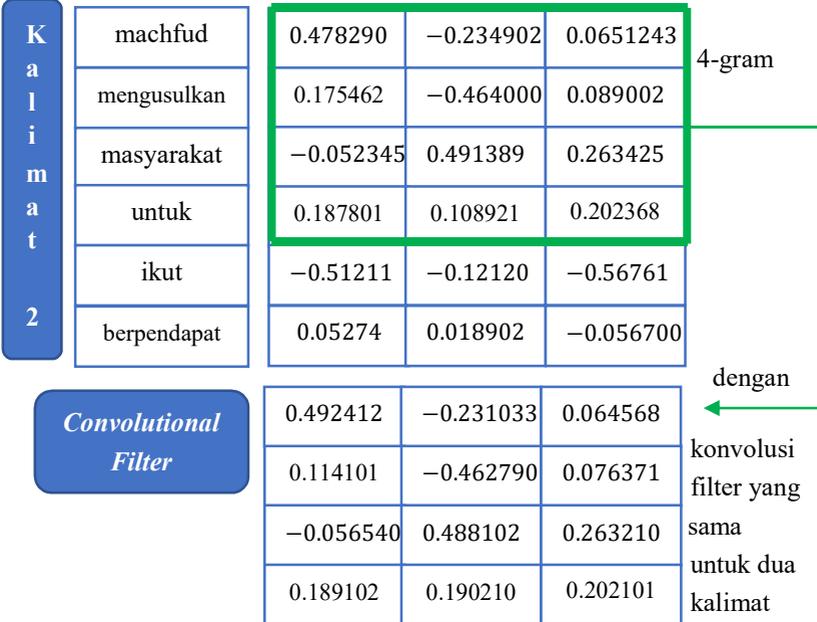
Selanjutnya masuk ke 1D *convolution*, dengan perhitungan sebagai berikut:

		<i>Words embedding</i>			
K a l i m a t	whisnu	0.490796	-0.229903	0.065460	representasi 4-gram
	mengajukan	0.104514	-0.463000	0.079367	
	warga	-0.055334	0.491792	0.263102	
	untuk	0.189031	0.103121	0.202514	dilakukan proses konvolusi
	ikut	-0.510031	-0.120345	-0.560021	
	berpendapat	0.052134	0.019002	-0.059090	

Convolutional Filter

0.492412	-0.231033	0.064568
0.114101	-0.462790	0.076371
-0.056540	0.488102	0.263210
0.189102	0.190210	0.202101

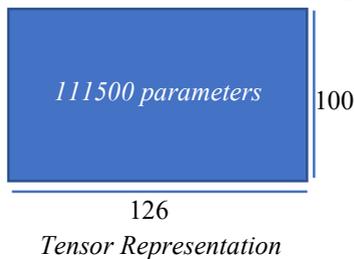
Filter konvolusional ini memiliki bobot yang akan dioptimalkan oleh model selama fase pelatihan (*training*). Dengan tidak hanya menggunakan 1 filter saja, maka dilakukan *tracking* pada *channel* yang lainnya dengan mengubah ukuran kernel contoh mengubah menjadi 6-gram, 8 gram, dan seterusnya yang akan melakukan proses pembelajaran pada masing-masing *features*. Garis penanda pada rangkaian barisan *words embedding* ini disebut sebagai filter konvolusi 1D *convolutions* karena kita hanya melakukan *sliding window* hanya 1 arah berbeda pada CNN yang diterapkan pada gambar.



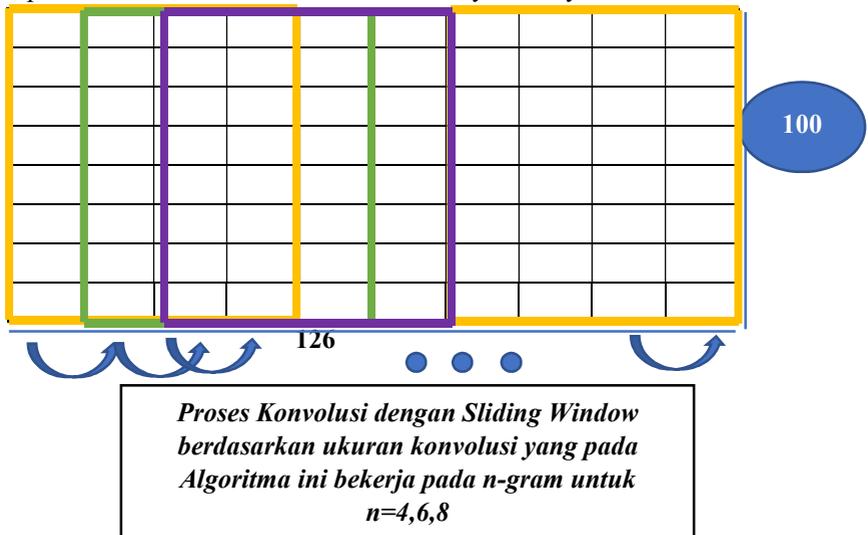
0.490796	-0.229903	0.065460
0.10451	-0.463000	0.079367
-0.055334	0.491792	0.263102
0.18903	0.10312	0.20251
-0.510031	-0.120345	-0.560021
0.05213	0.01900	-0.059090
0.490796	-0.229903	0.065460
0.10451	-0.463000	0.079367
-0.055334	0.491792	0.263102
0.18903	0.10312	0.20251
-0.510031	-0.120345	-0.560021
0.05213	0.01900	-0.059090
0.490796	-0.229903	0.065460
0.10451	-0.463000	0.079367
-0.055334	0.491792	0.263102
0.18903	0.10312	0.20251
-0.510031	-0.120345	-0.560021
0.05213	0.01900	-0.059090

Proses *sliding window* ini melibatkan berjalan pula pada kalimat kedua dengan filter konvolusi yang sama dan berulang dijalankan untuk filter konvolusi kedua dapat dimisalkan n-gram tertentu dengan misal $n=4,6,8$.

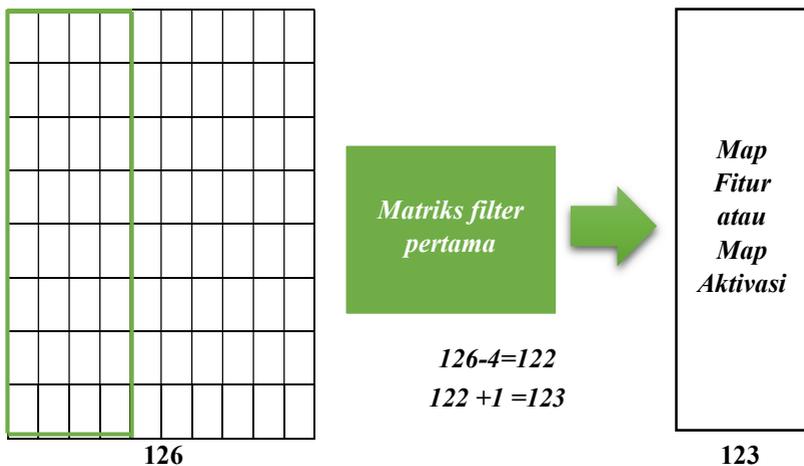
Untuk N-Gram-Multichannel-CNN dari data diatas diperoleh *embed size*= 100 dan *vocab_size*= 126, dengan *max_length*=1115 maka parameter total= $1115 \times 100 = 111500$, ilustrasi sebagai berikut:



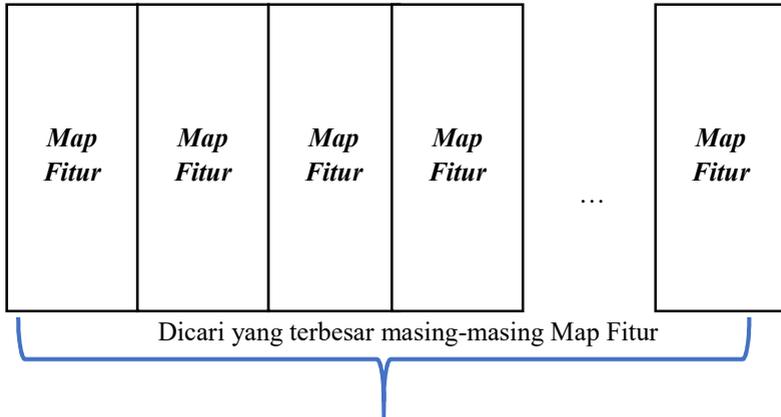
Kita misalkan baris dan kolom pada tabel dibawah adalah representasi input dari 1D-Convolutions, maka ukurannya sama yaitu 100x120



Hasil dari konvolusi merupakan Map Fitur atau Map aktivasi dengan ukuran 123 dengan kernel sebanyak 32 filter. Angka 123 didapatkan dari konvolusi pertama dengan n-gram=4 dimana berlaku:



Setelah didapatkan peta fitur maka dilakukan proses *Max-Pooling* dimana pada tahap ini dicari nilai output konvolusi berupa peta fitur yang paling besar, secara otomatis yang sebelumnya terdapat berukuran 123, maka akan dikonsolidasi diambil masing-masing paling besar maka akan ada 1 yang diambil dari 32 filter. Maka ilustrasi outputnya sebagai berikut:



Maka diperoleh Map Fitur ukuran 61 sebanyak 32 kernel hasil dari mencari nilai dari peta fitur yang paling besar setiap peta fitur pada kernelnya, dapat dilihat pada ilustrasi berikut:



Selanjutnya pada tahap *Flatten* dilakukan konkatensi dengan perhitungan:

$$\begin{aligned}
 &= \text{ukuran peta fitur} \times \text{kernel} \\
 &= 61 \times 32 \\
 &= 1952
 \end{aligned}$$

LAMPIRAN E. Visualisasi *Running Model*

```
Epoch 1/150
429/429 [=====] - 31s 73ms/step - loss: 0.2723 - accuracy: 0.8817 - val_loss: 0.2282 - val_accuracy:
0.8715
Epoch 2/150
429/429 [=====] - 32s 74ms/step - loss: 0.1480 - accuracy: 0.9314 - val_loss: 0.2232 - val_accuracy:
0.8731
Epoch 3/150
429/429 [=====] - 32s 74ms/step - loss: 0.1182 - accuracy: 0.9372 - val_loss: 0.1815 - val_accuracy:
0.8761
Epoch 4/150
429/429 [=====] - 32s 74ms/step - loss: 0.1077 - accuracy: 0.9397 - val_loss: 0.1517 - val_accuracy:
0.8992
Epoch 5/150
429/429 [=====] - 32s 75ms/step - loss: 0.1032 - accuracy: 0.9410 - val_loss: 0.1425 - val_accuracy:
0.9005
Epoch 6/150
429/429 [=====] - 32s 74ms/step - loss: 0.0990 - accuracy: 0.9416 - val_loss: 0.1273 - val_accuracy:
0.9054
Epoch 7/150
429/429 [=====] - 32s 74ms/step - loss: 0.0968 - accuracy: 0.9409 - val_loss: 0.1521 - val_accuracy:
0.8877
Epoch 8/150
429/429 [=====] - 31s 72ms/step - loss: 0.0943 - accuracy: 0.9443 - val_loss: 0.1121 - val_accuracy:
0.9097
```

```
Epoch 9/150
429/429 [=====] - 31s 73ms/step - loss: 0.0927 - accuracy: 0.9420 - val_loss: 0.1159 - val_accuracy:
0.9006
Epoch 10/150
429/429 [=====] - 31s 72ms/step - loss: 0.0906 - accuracy: 0.9455 - val_loss: 0.1138 - val_accuracy:
0.9065
Epoch 11/150
429/429 [=====] - 31s 73ms/step - loss: 0.0899 - accuracy: 0.9438 - val_loss: 0.0932 - val_accuracy:
0.9290
Epoch 12/150
429/429 [=====] - 31s 73ms/step - loss: 0.0893 - accuracy: 0.9431 - val_loss: 0.1111 - val_accuracy:
0.9076
Epoch 13/150
429/429 [=====] - 31s 73ms/step - loss: 0.0890 - accuracy: 0.9436 - val_loss: 0.1022 - val_accuracy:
0.9164
Epoch 14/150
429/429 [=====] - 31s 72ms/step - loss: 0.0876 - accuracy: 0.9440 - val_loss: 0.0921 - val_accuracy:
0.9337
Epoch 15/150
429/429 [=====] - 31s 73ms/step - loss: 0.0860 - accuracy: 0.9439 - val_loss: 0.0999 - val_accuracy:
0.9193
Epoch 16/150
429/429 [=====] - 31s 72ms/step - loss: 0.0860 - accuracy: 0.9432 - val_loss: 0.0892 - val_accuracy:
0.9305
Epoch 17/150
429/429 [=====] - 31s 72ms/step - loss: 0.0848 - accuracy: 0.9446 - val_loss: 0.0984 - val_accuracy:
0.9180
Epoch 18/150
429/429 [=====] - 31s 73ms/step - loss: 0.0851 - accuracy: 0.9429 - val_loss: 0.0973 - val_accuracy:
0.9220
Epoch 19/150
429/429 [=====] - 31s 73ms/step - loss: 0.0831 - accuracy: 0.9456 - val_loss: 0.1135 - val_accuracy:
0.9187
```

BIODATA PENULIS



Nama lengkap penulis Ferisa Tri Putri Prestasi. Penulis lahir pada tanggal 4 Nopember 1997 bertempat di Kecamatan Pare, Kabupaten Kediri, Jawa Timur. Orang tua penulis bernama Santoso Jiwo Leksono dan Gusti Ratu Rizkiah. Penulis merupakan

anak kedua dari 4 bersaudara. Riwayat Pendidikan yang telah ditempuh oleh penulis yakni di TK Kemala Bhayangkari (2003-2004), SDN 2 Pare (2004-2010), SMPN 2 Pare (2010-2013), dan SMAN 2 Pare (2013-2016). Semasa perkuliahan di Departemen Matematika ini, penulis mengambil rumpun mata kuliah Ilmu Komputer untuk mengembangkan logika dan penerapan matematika pada perkembangan teknologi. Penulis memiliki hobi olahraga seperti badminton, voli, tenis meja, selain itu berenang, bernyanyi serta membaca buku juga merupakan hobi penulis. Pada tahun kedua di perkuliahan, penulis aktif dalam beberapa kegiatan organisasi, yakni sebagai staff *Student Resources and Development* HIMATIKA ITS pada Divisi Kaderisasi (2017-2018), staff Hubungan Luar ITS Badminton Community (2017-2018), juga aktif pada UKM PSM ITS. Di tahun berikutnya penulis juga aktif sebagai Pemandu HIMATIKA ITS atau Pemandu Matriks. Penulis juga aktif dalam beberapa kegiatan kemahasiswaan, seperti Panitia Sie Acara ITS Open 2017, Penanggung Jawab Regional Surabaya OMITS 2018, Panitia Sie Acara OMITS 2018, Fasilitator GERIGI ITS 2017, Mentor GERIGI ITS 2018, Koordinator Sie Acara POMITS IBC 2017, dan lainnya. Beberapa proyek dan kegiatan komunitas yang pernah penulis ikuti dan terlibat langsung adalah Satu Data Pemerintah Kota Mojokerto, survei minat mahasiswa, aktif di komunitas Data Science Indonesia region Jawa Timur sebagai

Head of Research Development and Knowledge Management,
dan lainnya. Adapun informasi lebih lanjut terkait laporan Tugas Akhir ini, apabila berkenan memberikan saran, kritik, dan diskusi mengenai penelitian tugas akhir ini, dapat disampaikan melalui e-mail penulis di ferisatri04@gmail.com.