



**TUGAS AKHIR - KS184822**

**KLASIFIKASI SENTIMEN ULASAN FILM  
INDONESIA MENGGUNAKAN METODE  
*CONVOLUTIONAL NEURAL NETWORK (CNN)***

**NADHIFA AYU SHAFIRRA  
NRP 062116 4000 0064**

**Dosen Pembimbing  
Irhamah, S.Si, M.Si, Ph.D**

**PROGRAM STUDI SARJANA  
DEPARTEMEN STATISTIKA  
FAKULTAS SAINS DAN ANALITIKA DATA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2020**





**TUGAS AKHIR - KS184822**

**KLASIFIKASI SENTIMEN ULASAN FILM  
INDONESIA MENGGUNAKAN METODE  
*CONVOLUTIONAL NEURAL NETWORK (CNN)***

**NADHIFA AYU SHAFIRRA  
NRP 062116 4000 0064**

**Dosen Pembimbing  
Irhamah, S.Si, M.Si, Ph.D**

**PROGRAM STUDI SARJANA  
DEPARTEMEN STATISTIKA  
FAKULTAS SAINS DAN ANALITIKA DATA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2020**





**FINAL PROJECT - KS184822**

**SENTIMENT CLASSIFICATION ON INDONESIAN  
MOVIE REVIEWS USING CONVOLUTIONAL  
NEURAL NETWORK (CNN)**

**NADHIFA AYU SHAFIRRA  
SN 062116 4000 0064**

**Supervisor  
Irhamah, S.Si, M.Si, Ph.D**

**UNDERGRADUATE PROGRAMME  
DEPARTMENT OF STATISTICS  
FACULTY OF SCIENCE AND DATA ANALYTICS  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2020**



**LEMBAR PENGESAHAN**

**KLASIFIKASI SENTIMEN ULASAN FILM INDONESIA  
MENGUNAKAN METODE *CONVOLUTIONAL NEURAL  
NETWORK (CNN)***

**TUGAS AKHIR**

Diajukan untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Statistika  
pada  
Program Studi Sarjana Departemen Statistika  
Fakultas Sains dan Analitika Data  
Institut Teknologi Sepuluh Nopember

Oleh:

**Nadhifa Ayu Shafirra**  
NRP. 062116 4000 0064

Disetujui oleh Pembimbing:  
**Irhamah, S.Si, M.Si, Ph.D**  
NIP. 19780406 200112 2 002

( *Irhamah* )

Mengetahui,  
Kepala Departemen Statistika



**Dr. Dra. Kartika Fithriasari, M.Si**  
NIP. 19691212 199303 2 002

**SURABAYA, JANUARI 2020**





# **KLASIFIKASI SENTIMEN ULASAN FILM INDONESIA MENGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)**

**Identitas Mahasiswa** : Nadhifa Ayu Shafirra  
**NRP** : 062116 4000 0064  
**Departemen** : Statistika-FSAD-ITS  
**Dosen Pembimbing** : Irhamah, S.Si, M.Si, Ph.D

## **Abstrak**

*Ulasan film adalah sebuah opini yang bersifat subjektif. Ulasan film memiliki media yang beragam, seperti tulisan, audio, dan video. Ulasan film dapat diolah dengan menggunakan klasifikasi sentimen, agar ucapan seseorang terkait film dapat ditentukan sebagai sentimen tertentu. Di masa sekarang, data memiliki berbagai bentuk, pemilihan jenis data yang lebih baik juga dapat mempengaruhi klasifikasi sentimen. Data video dapat diekstraksi menggunakan Mel-Frequency Cepstral Coefficients (MFCC) dan dikonversi menjadi data teks dengan bantuan Speech-to-Text (STT). Fitur Ekstraksi MFCC digunakan karena keunikan skala Mel memungkinkan audio yang ada dibedakan berdasarkan suaranya. Data teks digunakan karena kata atau kalimat dapat dibedakan secara negatif atau positif. Sehingga, kedua jenis data tersebut dibandingkan dan dipilih model dengan jenis data yang memiliki hasil klasifikasi terbaik. Dengan menggunakan metode Convolutional Neural Network, didapatkan bahwa data teks memiliki nilai AUC lebih baik dibandingkan data MFCC. Model terbaik yang dipilih adalah model klasifikasi sentimen dengan data teks yang dimodelkan berdasarkan aspek dari penilaian film.*

***Kata kunci: Convolutional Neural Network, MFCC, Speech-to-Text, Ulasan Film***

*(Halaman ini sengaja dikosongkan)*

# SENTIMENT CLASSIFICATION ON INDONESIAN MOVIE REVIEWS USING CONVOLUTIONAL NEURAL NETWORK (CNN)

**Name** : Nadhifa Ayu Shafirra  
**Student Number** : 062116 4000 0064  
**Department** : Statistics  
**Supervisor** : Irhamah, S.Si, M.Si, Ph.D

## **Abstract**

*Movie review is an opinion that is subjective. Movie reviews have a variety of media, such as writing, audio, and video. Movie reviews can be processed using sentiment classification, so that someone's sayings related to a particular film can be determined as certain sentiments. At present, data takes a variety of forms, and selection of better data types can also affect sentiment classification. Video data can be extracted using Mel-Frequency Cepstral Coefficients (MFCC) and converted to text data with the help of Speech-to-Text (STT). The MFCC extraction feature is used because the uniqueness of the Mel scale allows existing audio to be differentiated based on sound. Text data is used because words or sentences can be distinguished negatively or positively. Thus, the two types of data are compared and the model with the best classification results is selected. By using the Convolutional Neural Network method, it was found that the text data has better AUC scores than MFCC data. The best model chosen is the sentiment classification model with text data that are modeled based on aspects of movie evaluation.*

***Keywords: Convolutional Neural Network, MFCC, Movie Review, Speech-to-Text***

*(Halaman ini sengaja dikosongkan)*

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah melimpahkan segala rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir dengan judul **“Klasifikasi Sentimen Ulasan Film Indonesia Menggunakan Metode *Convolutional Neural Network* (CNN)”** dengan lancar.

Pembuatan Laporan Tugas Akhir ini tidak akan terselesaikan dengan baik tanpa adanya bantuan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada :

1. Ibu Irhamah, S.Si, M.Si, Ph.D selaku Dosen Wali yang telah memberikan saran, dukungan, serta arahan selama proses belajar di Departemen Statistika ITS. Serta selaku dosen pembimbing Tugas Akhir yang telah meluangkan waktu memberikan bimbingan, mendukung, serta memberi masukan selama penyusunan Tugas Akhir.
2. Ibu Dra. Wiwiek Setya Winahju, M.Si. dan Ibu Santi Puteri Rahayu, M.Si, Ph.D. selaku dosen penguji atas saran, kritik, serta evaluasi untuk Tugas Akhir ini.
3. Dr. Dra. Kartika Fithriasari, M.Si selaku Kepala Departemen Statistika FSAD ITS.
4. Ibu Santi Wulan Purnami, M.Si, Ph.D. selaku Sekretaris Departemen 1 Bidang Akademik dan Kemahasiswaan Departemen Statistika FSAD ITS.
5. Seluruh dosen dan civitas akademika Departemen Statistika ITS untuk waktu dan ilmu yang telah diberikan selama perkuliahan.
6. Teman-teman Statistika ITS angkatan 2016 yang selalu memberikan bantuan, dukungan, dan semangat selama melaksanakan tugas akhir sampai dengan saat ini.
7. Oktavia Ramadhani dan Anisa Ramadhani sebagai teman satu kos yang ikut mendukung saya dalam pengerjaan tugas akhir.
8. Teman satu dosen pembimbing, Muhammad Abid As Sarofi dan Ni Luh Putu Ika Candrawendi yang menjadi teman seperjuangan dalam proses pengerjaan tugas akhir.

9. Mas Azmi dan Mbak Ulfa yang ikut membantu dan menjawab pertanyaan yang kurang penulis pahami terkait Tugas Akhir, serta Mas Adam yang membantu memberikan ide dalam proses pembuatan Tugas Akhir.
10. Teman-teman angkatan 2016 pejuang PW 121 yang membuat penulis termotivasi untuk mengerjakan dan menyelesaikan Tugas Akhir.
11. Orang tua saya, Papa Suyud Mama Liza Deviyanti serta kedua adik laki-laki saya Narendra Dhafa Ilyaza dan Rasyid Akhtar Ilyaza yang selalu mendoakan serta memberikan kasih sayang, semangat, serta dukungan kepada penulis.
12. Semua pihak yang mendukung yang tidak dapat penulis sebutkan satu persatu dalam penyelesaian Tugas Akhir ini.

Semoga laporan yang penulis susun dapat memberikan manfaat serta dapat digunakan sebagai rujukan penelitian ke depan. Penulis memohon maaf apabila terdapat banyak kekurangan dalam laporan yang telah penulis susun. Atas perhatian dan dukungannya, penulis mengucapkan terima kasih.

Surabaya, Januari 2020

Penulis

## DAFTAR ISI

	Halaman
<b>LEMBAR PENGESAHAN</b> .....	iii
<b>ABSTRAK</b> .....	v
<b>ABSTRACT</b> .....	vii
<b>KATA PENGANTAR</b> .....	ix
<b>DAFTAR ISI</b> .....	xi
<b>DAFTAR GAMBAR</b> .....	xiii
<b>DAFTAR TABEL</b> .....	xv
<b>DAFTAR LAMPIRAN</b> .....	xvii
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Tujuan.....	5
1.4 Manfaat.....	5
1.5 Batasan Masalah.....	6
<b>BAB II TINJAUAN PUSTAKA</b> .....	7
2.1 Analisis Sentimen.....	7
2.2 <i>Speech Recognition</i> .....	7
2.3 <i>Text Mining</i> .....	12
2.4 <i>Text Preprocessing</i> .....	12
2.6 <i>Word Embedding</i> .....	13
2.7 <i>K-Fold Cross Validation</i> .....	15
2.8 <i>Synthetic Minority Oversampling Technique</i> (SMOTE).....	15
2.9 <i>Convolutional Neural Network</i> .....	17
2.10 Evaluasi Kinerja Klasifikasi.....	22
2.11 <i>Word Cloud</i> .....	23
2.12 <i>Youtube</i> .....	23
2.13 Ulasan Film.....	23
<b>BAB III METODOLOGI PENELITIAN</b> .....	25
3.1 Sumber Data.....	25
3.2 Variabel Penelitian dan Struktur Data Hasil Ekstraksi MFCC.....	27

3.3	Variabel Penelitian dan Struktur Data Teks .....	27
3.4	Langkah Analisis .....	28
<b>BAB IV ANALISIS DAN PEMBAHASAN.....</b>		<b>33</b>
4.1	Karakteristik Ulasan Film .....	33
4.1.1	<i>Data Preprocessing</i> Menggunakan MFCC .....	33
4.1.2	<i>Data Preprocessing</i> Menggunakan <i>Speech-To-Text</i> .....	37
4.1.3	Hasil Karakteristik Ulasan Film.....	41
4.2	Hasil Klasifikasi Data MFCC dengan Metode CNN ....	43
4.2.1	Hasil Klasifikasi Keseluruhan Data MFCC.....	44
4.2.2	Hasil Klasifikasi Data MFCC pada Tiap Aspek.....	45
4.3	Hasil Klasifikasi Data Teks dengan Metode CNN .....	46
4.3.1	Hasil Klasifikasi Keseluruhan Data Teks .....	48
4.3.2	Hasil Klasifikasi Data Teks pada Tiap Aspek.....	49
4.4	Perbandingan Evaluasi Kinerja Klasifikasi .....	50
4.5	Hasil Visualisasi Persentase Nilai Ulasan Film.....	51
<b>BAB V PENUTUP.....</b>		<b>53</b>
5.1	Kesimpulan .....	53
5.2	Saran .....	54
<b>DAFTAR PUSTAKA .....</b>		<b>55</b>
<b>LAMPIRAN .....</b>		<b>63</b>



## DAFTAR GAMBAR

	Halaman
<b>Gambar 2.1</b> Ilustrasi Tahap <i>Filter Banks</i> pada MFCC .....	11
<b>Gambar 2.2</b> Ilustrasi Visualisasi Nilai MFCC .....	12
<b>Gambar 2.3</b> Ilustrasi <i>K-Fold Cross Validation</i> dengan $k=5$ .....	15
<b>Gambar 2.4</b> Ilustrasi SMOTE dalam <i>K-Fold Cross Validation</i> .....	16
<b>Gambar 2.5</b> Ilustrasi Text Mining Menggunakan CNN .....	17
<b>Gambar 2.6</b> Ilustrasi Audio Classification dengan MFCC Menggunakan CNN .....	18
<b>Gambar 2.7</b> Ilustrasi <i>Convolutional Layer</i> .....	19
<b>Gambar 2.8</b> Ilustrasi <i>Pooling Layer</i> .....	20
<b>Gambar 2.9</b> Ilustrasi Word Cloud .....	23
<b>Gambar 3.1</b> Diagram Alir .....	31
<b>Gambar 4.1</b> Amplitudo Data Audio Review Film .....	33
<b>Gambar 4.2</b> Amplitudo Data Audio Setelah <i>Pre-Emphasis</i> .....	34
<b>Gambar 4.3</b> Arsitektur <i>Word2Vec</i> pada Kata ‘Raja’ dan ‘Pria’ .....	40
<b>Gambar 4.4</b> <i>Pie Chart</i> Persentase Sentimen Negatif dan Positif.....	41
<b>Gambar 4.5</b> Plot Sinyal Potongan Audio Bersentimen (a) Negatif (b) Positif.....	42
<b>Gambar 4.6</b> <i>Word Cloud</i> Ulasan Film Bersentimen (a) Negatif (b) Positif.....	42
<b>Gambar 4.7</b> Arsitektur CNN dalam Klasifikasi Data MFCC ....	43
<b>Gambar 4.8</b> Arsitektur CNN <i>Unigram</i> dalam Klasifikasi Data Teks .....	46
<b>Gambar 4.9</b> Arsitektur CNN <i>Bigram</i> dalam Klasifikasi Data Teks .....	47
<b>Gambar 4.10</b> Arsitektur CNN <i>Trigram</i> dalam Klasifikasi Data Teks .....	47
<b>Gambar 4.11</b> Grafik Persentase Sentimen Film Bumi Manusia Aspek (a) Pemeran (b) Direktur (c) Jalan Cerita (d) Adegan (e) Musik (f) Keseluruhan .....	51

*(Halaman ini sengaja dikosongkan)*

## DAFTAR TABEL

	Halaman
<b>Tabel 2.1</b> Kata Kunci dan Aspek yang Dinilai dalam Ulasan Film.....	24
<b>Tabel 3.1</b> Daftar Ulasan yang Digunakan sebagai Data .....	25
<b>Tabel 3.2</b> Jumlah Potongan Ulasan Tiap Aspek .....	26
<b>Tabel 3.3</b> Variabel Penelitian Hasil Ekstraksi MFCC.....	27
<b>Tabel 3.4</b> Struktur Data Hasil Ekstraksi MFCC.....	27
<b>Tabel 3.5</b> Variabel Penelitian Data Teks .....	28
<b>Tabel 3.6</b> Struktur Data Teks .....	28
<b>Tabel 4.1</b> Hasil Framing .....	34
<b>Tabel 4.2</b> Hasil Windowing pada Tiap Frame .....	35
<b>Tabel 4.3</b> Hasil Nilai <i>Power Spectrum</i> pada Tiap <i>Frame</i> .....	36
<b>Tabel 4.4</b> Hasil Koefisien <i>Filter Banks</i> pada Tiap <i>Frame</i> .....	36
<b>Tabel 4.5</b> Nilai MFCC pada Tiap <i>Frame</i> .....	37
<b>Tabel 4.6</b> Ilustrasi <i>Speech-to-Text</i> dan <i>Text Preprocessing</i> .....	38
<b>Tabel 4.7</b> Ilustrasi <i>Word Embedding</i> dengan <i>Word2Vec</i> .....	39
<b>Tabel 4.8</b> Kata dan <i>Neighbor</i> dengan <i>window size</i> = 2 .....	39
<b>Tabel 4.9</b> <i>Encoding</i> pada Kata dan <i>Neighbor</i> .....	40
<b>Tabel 4.10</b> Hasil <i>Embedding</i> pada Tiap Kata .....	41
<b>Tabel 4.11</b> Nilai Kinerja Klasifikasi Keseluruhan Data MFCC .....	44
<b>Tabel 4.12</b> <i>Confusion Matrix</i> Klasifikasi Keseluruhan Data MFCC .....	45
<b>Tabel 4.13</b> Nilai Kinerja Klasifikasi Data MFCC Tiap Aspek ...	45
<b>Tabel 4.14</b> Nilai AUC Klasifikasi Keseluruhan Data Teks .....	48
<b>Tabel 4.15</b> <i>Confusion Matrix</i> Klasifikasi Keseluruhan Data MFCC .....	49
<b>Tabel 4.16</b> Nilai Kinerja Klasifikasi Data Teks Tiap Aspek .....	49
<b>Tabel 4.17</b> Evaluasi Kinerja Klasifikasi Model Tiap Data .....	50
<b>Tabel 4.18</b> Evaluasi Kinerja Klasifikasi Terbaik Model Tiap Data.....	51
<b>Tabel 4.19</b> Persentase Sentimen Positif dan Negatif Tiap Film.....	52

*(Halaman ini sengaja dikosongkan)*

## DAFTAR LAMPIRAN

	Halaman
<b>Lampiran 1</b> Daftar <i>Channel</i> dan <i>Link</i> Video Ulasan.....	63
<b>Lampiran 2</b> Data Hasil Ekstraksi MFCC.....	64
<b>Lampiran 3</b> Data Teks .....	66
<b>Lampiran 4</b> Hasil Grafik Persentase Sentimen Tiap Film .....	68
<b>Lampiran 5</b> Hasil <i>Confusion Matrix</i> .....	73
<b>Lampiran 6</b> <i>Syntax Python Split on Silence</i> .....	74
<b>Lampiran 7</b> <i>Syntax Python</i> Konversi ke Teks dengan STT .....	75
<b>Lampiran 8</b> <i>Syntax Python Import</i> Data untuk <i>Filter</i> .....	75
<b>Lampiran 9</b> <i>Syntax Python Text Pre-Processing</i> .....	76
<b>Lampiran 10</b> <i>Syntax Python Filter</i> Data Berdasarkan Aspek ...	77
<b>Lampiran 11</b> <i>Syntax Python Export</i> Data Audio dan Teks .....	79
<b>Lampiran 12</b> <i>Syntax Python</i> Ekstraksi MFCC .....	80
<b>Lampiran 13</b> <i>Syntax Python</i> Model <i>Word2Vec</i> .....	82
<b>Lampiran 14</b> <i>Syntax Python</i> Model <i>CNN</i> untuk <i>MFCC</i> .....	83
<b>Lampiran 15</b> <i>Syntax Python</i> Model <i>CNN</i> untuk Data Teks .....	84
<b>Lampiran 16</b> <i>Syntax Python</i> Klasifikasi Sentimen <i>MFCC</i> .....	87
<b>Lampiran 17</b> <i>Syntax Python</i> Klasifikasi Sentimen Data Teks ..	91
<b>Lampiran 18</b> <i>Output</i> Model <i>CNN</i> Data <i>MFCC</i> .....	96
<b>Lampiran 19</b> <i>Output</i> Model <i>CNN</i> Data Teks.....	96
<b>Lampiran 20</b> Surat Pernyataan Data Sekunder .....	99

*(Halaman ini sengaja dikosongkan)*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Film menjadi salah satu alternatif hiburan bagi masyarakat. Tidak hanya di Indonesia, tetapi sebagian besar masyarakat di seluruh dunia juga menikmati film-film yang beredar. Menurut Effendi (1986), film adalah hasil budaya dan alat ekspresi kesenian. Di mana film sebagai komunikasi massa merupakan gabungan dari berbagai teknologi seperti fotografi dan rekaman suara, kesenian baik seni rupa, seni teater, sastra, arsitektur, serta seni musik. Film baru selalu saja muncul terus menerus. Berbagai macam film hadir dengan genre yang berbeda-beda. Masyarakat selalu memiliki banyak pilihan untuk film yang ingin mereka tonton. Terkadang, saking banyaknya film yang hadir di bioskop, masyarakat seringkali bingung bagaimana memilih film yang sesuai dengan mereka. Salah satu permasalahan bagi sebagian masyarakat adalah keresahan jika nantinya film yang akan ditonton tidak memenuhi ekspektasi mereka. Sehingga, banyak orang mulai mencari cara agar bisa memilih film yang dapat memuaskan keinginan mereka.

Salah satu cara yang sering dilakukan adalah membaca dan menonton ulasan film dari orang lain. Sekarang banyak orang yang menulis maupun membicarakan ulasan dari film yang telah ditonton. Ulasan film telah beredar di berbagai media sosial, misal ulasan film dalam bentuk tulisan seringkali disebar melalui *blog* pribadi serta *website* penyedia rating film. Sedangkan ulasan film dalam bentuk video juga sudah sangat banyak di Youtube, Instagram, dan lain sebagainya. Ulasan film berbentuk video yang sudah banyak beredar di internet mulai diminati oleh masyarakat. Banyak orang yang melihat video ulasan film untuk menilai film. Namun durasi video yang bervariasi dari video singkat sekitar 5 menit hingga video berdurasi 20 menit atau lebih akan memakan waktu lama dalam menontonnya. Ulasan film dalam bentuk video juga tidak menuju langsung ke intinya karena di dalam video pasti selalu ada bagian pembuka terlebih dahulu, dan terkadang hal ter-

sebut berisi obrolan yang kurang penting dan tidak berhubungan dengan film yang akan diulas.

Ulasan film adalah sebuah opini yang bersifat subjektif dan tentu saja tiap orang pasti akan berbeda-beda dalam memberikan opininya terkait film yang telah ditonton. Seringkali antar dua ulasan memiliki poin yang berbeda dalam menilai suatu film. Perbedaan persepsi antar *reviewer* dapat menghasilkan pemahaman yang berbeda tiap individu. Sehingga dibutuhkan suatu metode yang dapat merangkum opini berbagai *reviewer* dari suatu film untuk menentukan apakah film tersebut bagus atau tidak. Hal tersebut dibutuhkan untuk mengetahui bagaimana klasifikasi sentimen dari tiap *reviewer* yang berbeda terhadap film yang diulas. Alternatif untuk menyelesaikan permasalahan ini adalah analisis sentimen. Analisis sentimen atau bisa juga disebut *opinion mining* adalah suatu bidang yang mempelajari cara menganalisis opini, sentimen, evaluasi, pujian, sikap, dan emosi seseorang terhadap suatu entitas seperti produk, jasa, organisasi, individu, isu, acara, topik, dan lain sebagainya (Liu, 2012).

Data di masa sekarang tidak hanya dalam bentuk angka saja, mulai dari tulisan, audio, gambar, ataupun video sekarang dapat menjadi sumber data yang dapat diolah. Analisis sentimen juga dapat diterapkan dalam data audio. Namun, data audio tidak bisa langsung digunakan sebagai *input* untuk klasifikasi sentimen. Data audio tersebut harus diekstraksi terlebih dahulu ke bentuk yang lain. Salah satu fitur ekstraksi yang sering digunakan dalam mengekstraksi audio adalah *Mel-Frequency Cepstral Coefficients*, di mana MFCC mewakili audio dengan cara mereplikasi telinga manusia sebagai model matematika (Maghilnan & Rajesh, 2017). Konstanta MFCC menandakan energi yang berkaitan dengan Skala Mel yang unik untuk setiap pembicara. Keunikan ini memungkinkan kita untuk mengidentifikasi pembicara berdasarkan suaranya (Kinnunen & Li, 2010). MFCC dikembangkan dari *Fourier Transform* atau *Discrete Cosine Transform* (Bhattacharya, 2006). Data audio yang ada juga dapat diubah menjadi data teks yang selanjutnya digunakan untuk analisis sentimen. Proses perubahan data su-



ara menjadi data teks seringkali disebut *Speech-to-Text* (Amos, 2018). Banyak API yang menyediakan fasilitas *Speech-to-Text* secara gratis, salah satu contohnya adalah *Google Cloud Speech*.

Data yang telah diekstraksi selanjutnya dianalisis menggunakan metode klasifikasi. Karena ulasan film memiliki durasi dan panjang teks yang berbeda-beda akan kurang efektif jika melakukan klasifikasi sentimen dengan variabel dependen berupa ulasan. Sehingga untuk membuat klasifikasi sentimen lebih efektif, ulasan yang ada dapat dibagi berdasarkan aspek-aspek penilaian seorang *reviewer* dalam menilai suatu film. Aspek penilaian yang digunakan adalah film secara keseluruhan, pemeran, direktor (sutradara), jalan cerita, adegan, dan musik (Thet, Na, & Khoo, 2010). Sehingga, tiap ulasan akan memiliki enam variabel dependen berdasarkan jenis aspeknya.

Metode klasifikasi yang akan digunakan adalah *deep learning*. *Deep learning* adalah salah satu metode dalam *machine learning* yang didasarkan pada pembelajaran representasi data (Sewak, Karim, & Pujari, 2018). Menurut Cui, Lin, & Utsuro (2018), hasil analisis sentimen data *tweet* menggunakan CNN lebih baik dibandingkan metode SVM untuk keseluruhan kelas. Tetapi, pada kelas netral kedua metode tersebut masih belum cukup baik dibandingkan kelas yang lain. Berdasarkan Lim, Jang, & Taejin (2016), antara metode *convolutional neural network* (CNN), LSTM, dan *distributed CNN*, dalam melakukan klasifikasi emosi pada data hasil *speech recognition* didapat nilai akurasi paling baik adalah *distributed CNN* sebesar 86,65%, CNN sebesar 86,06%, dan terakhir LSTM sebesar 78,31%. Berdasarkan penelitian terdahulu untuk klasifikasi sentimen pada data audio oleh Bardou, Zhang, & Ahmad (2018) tentang klasifikasi suara paru-paru dari 7 kelas yang ada, nilai presisi metode CNN lebih baik di 4 kelas dibandingkan metode SVM, dengan presisi terendah sebesar 0,88 dibanding SVM yang memiliki presisi paling rendah sebesar 0,64. Berdasarkan beberapa penelitian terdahulu dapat disimpulkan bahwa metode yang akan digunakan dalam penelitian ini adalah *convolutional neural network* (CNN).

*Convolutional Neural Network* (ConvNet/CNN) adalah algoritma *deep learning* yang dapat menggunakan gambar sebagai *input*, menetapkan kepentingan (bobot dan bias yang dapat dipelajari) untuk berbagai aspek dan objek dalam gambar agar dapat membedakan satu dengan yang lain (Saha, 2018). Penelitian terdahulu terkait klasifikasi sentimen pada ulasan film diteliti oleh Wollmer, dkk. (2013) tentang analisis sentimen pada ulasan film di Youtube di mana penelitian ini membandingkan hasil klasifikasi berdasarkan ekstraksi data teks dengan data gabungan teks, video, dan audio menggunakan metode SVM dan BLSTM dengan hasil klasifikasi data teks memiliki nilai F1 tertinggi sebesar 73, data audio sebesar 64,4, serta gabungan antara data teks dan audio memiliki nilai F1 sebesar 72,4. Setelah mendapatkan hasil klasifikasi menggunakan metode CNN dari masing-masing jenis data, hasil klasifikasi tersebut dihitung menggunakan evaluasi kinerja klasifikasi untuk menentukan hasil klasifikasi terbaik antara kedua data yang telah digunakan. Berdasarkan hasil klasifikasi tersebut, model yang ada dapat digunakan untuk menentukan sentimen dari ulasan film yang ada.

## 1.2 Rumusan Masalah

Ulasan film bersifat subjektif dan dapat menghasilkan persepsi yang berbeda-beda antar *reviewer*. Permasalahan tersebut dapat diatasi dengan klasifikasi sentimen pada ulasan film yang ada di internet. Di masa sekarang, data tidak hanya dalam bentuk angka, namun dapat juga berbentuk tulisan, audio, video, dan gambar. Lantas, menentukan jenis data yang lebih baik digunakan dalam klasifikasi sentimen juga menjadi faktor penting dalam kebaikan model klasifikasi sentimen. Karena klasifikasi sentimen tidak dapat dihitung secara langsung dari data video tersebut, maka audio yang ada di video tersebut perlu diekstraksi terlebih dahulu agar dapat dianalisa. Ekstraksi yang digunakan adalah ekstraksi *Mel-Frequency Cepstral Coefficients* (MFCC) untuk *audio mining*. Sedangkan untuk *text mining* data audio dikonversi menjadi data teks melalui *Speech-To-Text* (STT). Metode klasifikasi yang akan digunakan adalah *Convolutional Neural Network* (CNN). Hasil

klasifikasi dari dua data dibandingkan apakah data hasil ekstraksi MFCC lebih baik daripada data teks atau sebaliknya. Selanjutnya hasil terbaik digunakan untuk mengetahui persentase sentimen dari ulasan film tersebut.

### 1.3 Tujuan

Berdasarkan rumusan masalah yang telah dijelaskan, maka penelitian ini memiliki tujuan sebagai berikut.

1. Mendapatkan hasil karakteristik data ulasan film yang telah melalui *preprocessing data* dengan fitur ekstraksi MFCC dan konversi menggunakan STT.
2. Mengklasifikasikan sentimen dari keseluruhan ulasan film dan ulasan film yang dikelompokkan berdasarkan aspeknya (keseluruhan, pemeran, direktor, jalan cerita, adegan, dan musik) melalui *audio mining* dengan fitur ekstraksi MFCC menggunakan metode *Convolutional Neural Network* (CNN).
3. Mengklasifikasikan sentimen dari keseluruhan ulasan film dan hasil yang telah dikelompokkan berdasarkan aspeknya (keseluruhan, pemeran, direktor, jalan cerita, adegan, dan musik) melalui *text mining* dengan bantuan STT menggunakan metode *Convolutional Neural Network* (CNN).
4. Membandingkan kinerja klasifikasi sentimen antara *audio mining* dan *text mining* dengan menggunakan metode *Convolutional Neural Network* (CNN).
5. Memvisualisasikan hasil pemilihan model klasifikasi terbaik untuk melihat nilai ulasan dari suatu film.

### 1.4 Manfaat

Penelitian ini diharapkan dapat menjadi referensi untuk penelitian selanjutnya dalam membandingkan klasifikasi sentimen antara data teks dan data ekstraksi audio menggunakan metode *Convolutional Neural Network* (CNN). Bagi pembaca yang ingin melakukan penelitian ini, diharapkan dapat menjadi referensi dalam penelitian bidang analisis sentimen dengan data audio yang diekstraksi menggunakan fitur ekstraksi *Mel Frequency Cepstral Coefficient* (MFCC) dan data teks yang diperoleh melalui konversi

dengan bantuan *Speech-To-Text* (STT). Bagi masyarakat diharapkan model terbaik yang terpilih dapat digunakan dalam membuat persentase nilai sentimen dari ulasan suatu film sehingga dapat dimanfaatkan dan dilihat oleh masyarakat.

### **1.5 Batasan Masalah**

Batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Data yang digunakan berupa ulasan 5 film berbahasa Indonesia dengan masing-masing filmnya terdiri dari 10 video ulasan yang ada di Youtube.
2. Data audio dipotong berdasarkan keheningan atau jeda dalam audio dengan jeda (*silence*) minimal 350 ms dan audio tersebut dikatakan hening jika amplitudo suara lebih kecil dari -25 hingga -30 dBFS.
3. *Labelling* sentimen positif dan negatif dilakukan secara manual.
4. Jumlah aspek penilaian film yang akan digunakan adalah film secara keseluruhan, pemeran, direktor (sutradara), jalan cerita, adegan, dan musik.

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini berisi pembahasan mengenai proses analisis sentimen dengan metode *Convolutional Neural Network*. Selain itu juga terdapat pembahasan mengenai *speech recognition* menggunakan fitur ekstraksi *Mel Frequency Cepstral Coefficients* dan konversi data audio ke data teks dengan *Speech-to-Text*.

#### **2.1 Analisis Sentimen**

Analisis sentimen, atau bisa juga disebut dengan *opinion mining*, adalah bidang studi yang menganalisis opini, sentimen, penilaian, sikap, dan emosi seseorang terhadap entitas dan atribut yang dinyatakan dalam teks tertulis. Entitas dapat berupa produk, layanan, organisasi, individu, peristiwa, masalah, atau topik. Banyak nama yang berhubungan di bidang ini dengan tugas yang mirip, misalnya, analisis sentimen, *opinion mining*, analisis opini, ekstraksi opini, *sentiment mining*, analisis subjektivitas, analisis pengaruh, *emoticon analysis*, dan *review mining*, di mana bidang-bidang tersebut termasuk di analisis sentimen (Liu, 2015).

Istilah analisis sentimen pertama kali muncul di Nasukawa dan Yi (2003), dan istilah *opinion mining* pertama kali muncul di Dave dkk. (2003). Namun, penelitian tentang sentimen dan opini telah dimulai sebelumnya. Bahkan pekerjaan terkait telah ada sebelumnya seperti interpretasi metafora, ekstraksi kata sifat sentimen, komputasi afektif, dan analisis subjektivitas, sudut pandang, dan pengaruh. Sejak penelitian di bidang tersebut hadir, penerapan analisis sentimen berfokus terutama pada teks tertulis, sehingga menjadi bidang penelitian aktif pada *natural language processing* (NLP). Namun, topik ini juga telah banyak dipelajari dalam *data mining*, *web mining*, dan *information retrieval* karena banyak peneliti di bidang ini yang menangani data teks (Liu, 2015).

#### **2.2 Speech Recognition**

*Speech recognition* atau pengenalan suara adalah kapabilitas suatu teknologi dalam memahami suatu ucapan (Christensson,

2014). Pengenalan suara berawal dari suatu penelitian yang dilakukan oleh *Bell Labs* pada awal tahun 1950-an. Sistem awal ini terbatas pada satu penutur dan memiliki kosakata terbatas sekitar selusin kata. Sistem pengenalan suara modern telah berkembang sejak saat itu. Sistem ini dapat mengenali pembicaraan dari banyak penutur dan memiliki banyak kosakata dalam berbagai bahasa. Komponen pertama dari pengenalan suara adalah ucapan. Suara ini harus dikonversikan dari suara fisik ke sinyal listrik dengan mikrofon, kemudian ke data digital dengan konverter. Setelah didigitalkan, beberapa model dapat digunakan untuk menyalin audio ke teks. Hal ini sering disebut juga *Speech-to-Text* (STT). Pendekatan ini bekerja berdasarkan asumsi bahwa sinyal suara, bila dilihat pada skala waktu yang cukup singkat, dapat diperkirakan secara wajar sebagai proses stasioner (Amos, 2018). Sudah banyak API (*Application Programming Interface*) yang tersedia untuk memenuhi kebutuhan *speech recognizing* dalam pemrograman.

Data audio juga dapat diekstraksi menggunakan metode lain selain menggunakan *Speech-To-Text*. Jika *speech-to-text* mengubah data audio menjadi teks, fitur ekstraksi yang biasanya digunakan untuk audio mengubah bentuk gelombang suara menjadi bentuk representasi parametrik pada dimensi data yang relatif lebih rendah sehingga dapat digunakan untuk pemrosesan dan analisis selanjutnya (Alim & Rashid, 2018).

Fitur ekstraksi yang sering digunakan salah satunya adalah *Mel Frequency Cepstral Coefficients* (MFCC). MFCC adalah fitur ekstraksi yang digunakan dalam *speech recognition*, *speaker recognition*, dan juga dalam banyak algoritma berbasis suara lainnya. MFCC mereplikasi telinga manusia sebagai model matematika (Maghilan & Rajesh, 2017). Konstanta MFCC menandakan energi yang berkaitan dengan Skala Mel yang unik untuk setiap pembicara. Keunikan ini memungkinkan kita untuk mengidentifikasi pembicara berdasarkan suaranya (Kinnunen & Li, 2010). Langkah-langkah dasar yang harus diikuti dalam menurunkan MFCC di antaranya adalah *pra-emphasis*, *framing*, *windowing*, *fourier-transform*, *power spectrum*, *filter banks*, dan peng-

hitungan koefisien MFCC menggunakan *discrete cosine transform* (DCT).

*Pra-Emphasis* digunakan untuk meningkatkan magnitudo di bagian berfrekuensi tinggi pada sinyal (Huang, dkk., 2018). Tahapan ini perlu dilakukan untuk menyeimbangkan spektrum frekuensi karena frekuensi yang tinggi biasanya memiliki magnitudo yang lebih kecil dibandingkan frekuensi yang lebih rendah (Fayek, 2016). *Pre-emphasis* dapat diterapkan pada sinyal  $x$  menggunakan persamaan berikut.

$$y(t) = x(t) - \alpha x(t - 1), \quad (2.1)$$

keterangan:

$y(t)$  = magnitudo hasil *pra-emphasis* pada sinyal ke- $t$

$x(t)$  = magnitudo awal pada sinyal ke- $t$

$\alpha$  = koefisien filter (0,97).

Nilai koefisien filter yang biasa digunakan bernilai 0,95 dan 0,97 dan yang akan digunakan pada penelitian ini adalah koefisien filter sebesar 0,97 (Fayek, 2016).

*Framing* diperlukan untuk membagi sinyal menjadi *frame* jangka pendek. Alasan di balik tahapan ini karena frekuensi di dalam sinyal berubah dari waktu ke waktu, sehingga dalam jika melakukan transformasi Fourier di seluruh sinyal kemungkinan besar akan menghilangkan kontur frekuensi sinyal. Untuk mengatasi masalah tersebut dapat diasumsikan bahwa frekuensi dalam suatu sinyal adalah diam selama periode waktu yang sangat singkat. Oleh karena itu *framing* akan menggabungkan *frame* yang berdekatan agar memperoleh perkiraan yang baik dari kontur frekuensi sinyal yang ada (Fayek, 2016).

Tahapan selanjutnya adalah menerapkan fungsi *windowing*. Fungsi yang digunakan adalah *Hamming window* untuk diterapkan pada setiap *frame* (Fayek, 2016). *Hamming window* memiliki formula sebagai berikut.

$$w[n] = 0,54 - 0,46 \cos\left(\frac{2\pi n}{N-1}\right), \quad (2.2)$$

keterangan:

$w[n]$  = amplitudo hasil *windowing* pada *window length* ke- $n$

$N$  = window length, di mana  $0 \leq n \leq N - 1$

Tahapan *fourier-transform* mengaplikasikan *N-point FFT* pada setiap *frame* untuk menghitung spektrum frekuensi atau bisa disebut sebagai *Short-Time Fourier-Transform (STFT)*. Nilai  $N$  yang digunakan dalam penelitian ini adalah 512 (Fayek, 2016). Nilai FFT untuk tiap *frame* didapatkan dengan persamaan yang ditunjukkan pada persamaan (2.3).

$$FFT(x_k) = \sum_{n=0}^{N-1} x_n e^{-\frac{iz\pi kn}{N}}, \quad (2.3)$$

keterangan:

$X_k$  = nilai FFT ke- $k$ , di mana  $0 \leq k \leq \frac{N}{2} + 1$

$x_n$  = amplitudo hasil *windowing* ke- $n$

Selanjutnya, setelah mendapatkan nilai FFT di tiap-tiap *frame* dapat dihitung *power spectrum* (peridogram) dengan persamaan berikut.

$$P = \frac{|FFT(x_i)|^2}{N}, \quad (2.4)$$

di mana  $x_i$  adalah amplitudo *windowing* ke- $i$  di *frame* tertentu.

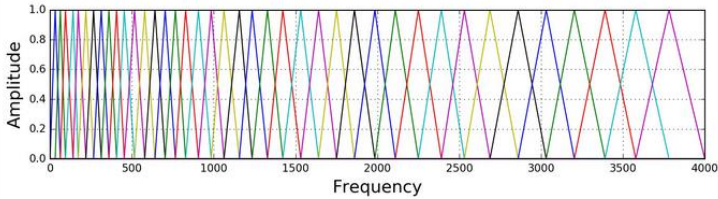
*Filter Banks* merupakan tahapan yang mengaplikasikan *triangular filters*, dengan nilai filter sebanyak 40 pada skala *mel* ke *power spektrum* dalam melakukan ekstraksi terhadap *frequency bands*. Skala *mel* ditujukan untuk meniru persepsi suara telinga manusia yang non-linear, dengan menjadi lebih diskriminatif terhadap frekuensi rendah dan kurang diskriminatif pada frekuensi yang lebih tinggi. Berikut adalah persamaan dalam mengkonversi *Hertz (f)* dan *Mel (m)*.

$$m = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad (2.5)$$

$$f = 700 \left( 10^{m/2595} - 1 \right) \quad (2.6)$$

Nilai  $f$  saat menkonversikan ke dalam satuan *Mel* adalah setengah dari jumlah *sample rate*. Persamaan akan menentukan nilai frekuensi terendah dan nilai frekuensi tertinggi sebagai batas nilai satuan *mel*. Setiap *filter* berbentuk segitiga yang memiliki respons 1 pada bagian tengah frekuensi dan menurun secara linear menuju 0. Hal ini ditunjukkan pada Gambar 2.1.





**Gambar 2.1** Ilustrasi Tahap *Filter Banks* pada MFCC

Selanjutnya tahapan *filter banks* dapat dimodelkan berdasarkan persamaan 2.7.

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & f(m-1) \leq k < f(m) \\ 1 & k = f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)} & f(m) < k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \quad (2.7)$$

Setelah mengaplikasikan *filter banks* ke dalam *power spectrum*, selanjutnya hasil *filter banks* tersebut dikonversi ke dalam satuan desibel dengan persamaan sebagai berikut.

$$fbanks(k) = 20 \log_{10} H_m(k) \quad (2.8)$$

Tahapan terakhir dalam proses fitur ekstraksi MFCCs adalah menerapkan *Discrete Cosine Transform* (DCT) untuk menghasilkan representasi dari *filter bank*. Dalam *automatic speech recognition* (ASR), nilai koefisien *cepstrum* yang digunakan berada di rentang 2 hingga 13 dan yang lain akan dibuang (Fayek, 2016). Persamaan dari DCT adalah sebagai berikut.

$$y(k) = f(k) \times 2 \sum_{n=0}^{N-1} x_n \cos\left(\pi k \left(\frac{2n+1}{2N}\right)\right), \quad 0 \leq k < N \quad (2.9)$$

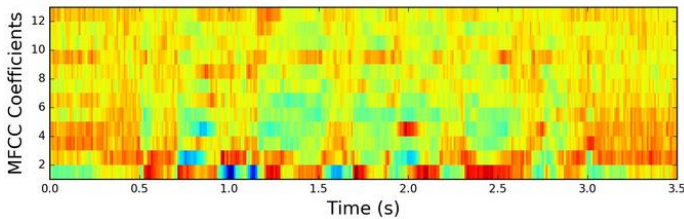
$$f(k) = \begin{cases} \sqrt{\frac{1}{4N}}, & k = 0 \\ \sqrt{\frac{1}{2N}}, & otherwise \end{cases} \quad (2.10)$$

keterangan:

$y(k)$  = nilai MFCC pada waktu ke- $k$

$f(k)$  = *scaling factor*

Nilai koefisien *cepstrum* yang akan digunakan adalah 12. Nilai ini digunakan karena telah mewakili perubahan cepat dalam koefisien *filter bank*. Selain itu, Nilai koefisien dari *filter bank* juga tidak terlalu mempengaruhi proses *Automatic Speech Recognition* (Fayek, 2016). Hasil MFCC divisualisasikan pada Gambar 2.2.



**Gambar 2.2** Ilustrasi Visualisasi Nilai MFCC

### 2.3 Text Mining

*Text mining* adalah serangkaian teknik dan metode yang digunakan untuk pemrosesan otomatis untuk data teks bahasa (*natural language text*) yang tersedia dalam jumlah yang cukup besar, dengan tujuan mengekstraksi dan menyusun konten dan tema, untuk keperluan analisis yang cepat, penemuan data yang tersembunyi, atau pengambilan keputusan secara otomatis. Ini berbeda dari *stylometry*, yang mempelajari gaya teks untuk mengidentifikasi seorang penulis atau menentukan tanggal suatu karya, tetapi memiliki banyak kesamaan dengan *lexicometry* atau statistik leksikal yang lebih dikenal sebagai ‘statistik linguistik’ atau ‘linguistik kuantitatif’ (Tuffery, 2011).

*Text mining* sebagian berasal dari tanggapan terhadap data teks berukuran besar yang dibuat dan disebarakan di masyarakat (seperti hukum, perintah, peraturan, atau kontrak), dan sebagian untuk tujuan generalisasi *input* dan penyimpanan data di sistem komputer (Tuffery, 2011).

### 2.4 Text Preprocessing

Sebagian besar metode *text preparation* dan *preprocessing* memiliki dasar dari *natural language processing* (NLP). *Text preprocessing* mengambil *input raw text* dan mengembalikan *output* berupa *token* yang telah dibersihkan. *Token* adalah kata

tunggal atau kelompok kata yang dihitung berdasarkan frekuensinya dan berfungsi sebagai variabel (*features*) dalam analisis. Tahapan *text preprocessing* terdiri dari *case folding*, *stemming*, *stopwords removal*, dan *tokenizing* (Gurusamy & Kannan, 2014).

*Case folding* adalah tahapan yang berfungsi untuk mengubah *font*, serta mengubah semua huruf menjadi huruf *lowercase*. Langkah ini merupakan langkah dasar yang paling banyak digunakan dalam *natural language processing* (Lutfi, Permanasari, & Fauziati, 2018). *Stopwords removal* adalah tahapan *text preprocessing* yang akan menghilangkan *stopwords* dalam suatu teks. *Stopwords* adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna (Ledeneva, 2008). Contoh *stopword* dalam bahasa Indonesia adalah “yang”, “dan”, “di”, dan lain sebagainya. *Stemming* adalah proses untuk mengganti kata-kata menjadi kata dasarnya (Ledeneva, 2008). Proses *stemming* dalam bahasa Indonesia menghilangkan semua kata imbuhan baik itu sufiks dan prefiks (Nugroho, 2018). *Tokenizing* adalah proses untuk memisahkan kata per kata dalam suatu teks (Liu, 2012)

## 2.5 *N-gram*

Teknik berbasis *n-gram* adalah metode yang dominan digunakan dalam *natural language processing* dan aplikasinya. *N-gram* merupakan urutan elemen yang muncul dalam teks. “N” yang dimaksud dalam *N-gram* adalah jumlah elemen yang diambil dari teks tersebut (Sidorov, dkk., 2012). *N-gram* didefinisikan sebagai fonem yang berturut-turut dalam *window* N. Nilai N biasanya berkisar dari 1 sampai 5 (Mikros & Macutek, 2015). Contoh dari *N-gram* untuk *unigram* (1-gram) adalah “Film”, “Film Indonesia” untuk *bigram* (2-gram), dan “Film Indonesia bagus” untuk *trigram* (3-gram).

## 2.6 *Word Embedding*

*Word Embedding* adalah pendekatan untuk vektor dari representasi kata-kata berdasarkan makna kata tersebut. *Word*

*embedding* merupakan peningkatan dari skema model *bag-of-word* yang lebih sederhana seperti jumlah dan frekuensi kata sehingga menghasilkan vektor besar yang sebagian besar berisi nilai 0 (Brownlee, 2019). Salah satu metode *word embedding* yang bias digunakan adalah *Word2Vec*.

*Word2Vec* adalah *word embedding* di mana kata-kata dipetakan dalam ruang vektor tetap, sehingga kemiripan *cosinus* antara dua kata adalah estimator yang baik dari kesamaan *semantic*. Manfaat utama *Word2Vec* adalah bahwa metode *training* tidak perlu membangun dan memvalidasi model yang kompleks secara manual, tetapi model *Word2Vec* belajar dengan mengekstraksi makna kata dengan mempertimbangkan kata-kata yang berdekatan dalam satu set kalimat (Charles, dkk., 2017). Persamaan yang digunakan dalam *Word2Vec* adalah *cosine similarity* yang ditunjukkan pada Persamaan 2.11.

$$S(\mathbf{w}_1, \mathbf{w}_2) = \cos(\mathbf{w}_1, \mathbf{w}_2) = \frac{\mathbf{w}_1 \times \mathbf{w}_2}{\|\mathbf{w}_1\| \times \|\mathbf{w}_2\|} \quad (2.11)$$

keterangan:

$\mathbf{w}_1$  = vektor *word embedding* dari kata pertama,

$\mathbf{w}_2$  = vektor *word embedding* dari kata kedua.

*Word2Vec* menghitung kesamaan *semantic* melalui kemunculan bersama. Terkadang, kata-kata yang didapat bukan sinonim melainkan kata-kata yang berkaitan. Dalam hal ini, relevansi kata di setiap domain dengan fitur umum dapat dihitung, dan vektor kata dari domain yang berbeda akan diperoleh (Zhou, dkk., 2019). Persamaan untuk menghitung *similarity* antara *feature* umum dan kata yang lain adalah sebagai berikut.

$$\text{sim}_{g_i}(w) = \begin{cases} \cos(w_{g_i}, w_w) & \text{jika } \cos(w_{g_i}, w_w) > 0, \\ 0 & \text{lainnya,} \end{cases} \quad (2.12)$$

keterangan:

$g$  = *generic feature* pada  $G$ ,

$w$  = kata dari salah satu domain  $a$  atau  $b$ .

Setelah menghitung *similarity* antara semua kata dan *feature* umum, matriks kata *fuzzy*  $H$  diperoleh berdasarkan persamaan 2.13.

$$H = \begin{bmatrix} \text{sim}_{g_1}(w_1) & \text{sim}_{g_1}(w_2) & \text{sim}_{g_1}(w_3) & \dots & \text{sim}_{g_1}(w_v) \\ \text{sim}_{g_2}(w_1) & \text{sim}_{g_2}(w_2) & \text{sim}_{g_2}(w_3) & \dots & \text{sim}_{g_2}(w_v) \\ \dots & \dots & \dots & \dots & \dots \\ \text{sim}_{g_m}(w_1) & \text{sim}_{g_m}(w_2) & \text{sim}_{g_m}(w_3) & \dots & \text{sim}_{g_m}(w_v) \end{bmatrix} \quad (2.13)$$

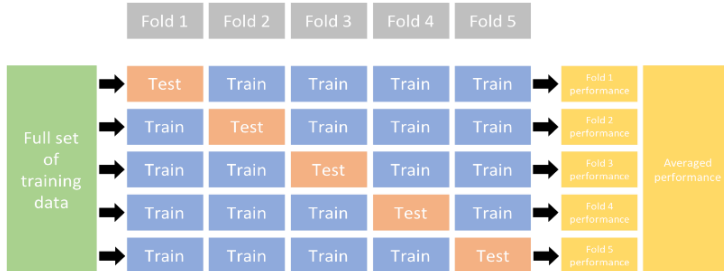
keterangan:

$m$  = ukuran dari  $G$ ,

$v$  = ukuran dari  $W$ .

## 2.7 K-Fold Cross Validation

*K-Fold Cross Validation* berfungsi untuk meminimalkan bias yang terkait dengan pengambilan sampel acak dari data *training* dan data sampel dalam membandingkan akurasi prediksi dua metode atau lebih. *K-Fold CV* juga sering disebut estimasi rotasi, di mana kumpulan data lengkap dibagi secara acak menjadi  $k$  *subset* dengan ukuran yang sama. Model klasifikasi yang ada diuji sebanyak  $k$  kali. Setiap *fold* di-*training* kecuali satu *fold* dan diuji untuk satu *fold* yang tersisa (Olson & Delen, 2008).



Gambar 2.3 Ilustrasi *K-Fold Cross Validation* dengan  $k=5$

## 2.8 Synthetic Minority Oversampling Technique (SMOTE)

*Resampling* adalah salah satu teknik *preprocessing* yang menyesuaikan distribusi dari suatu *dataset* yang tidak seimbang (*imbalanced*) hingga hampir seimbang, sebelum memasukkannya dalam *classifier* (Bunhumpornpat, Sinapiromsaran, & Lursinsap, 2009). Salah satu teknik *resampling* adalah *Synthetic Minority Oversampling Technique* (SMOTE) yang dirancang oleh Chawla, N., dkk., W. (2002). Teknik ini mengambil sampel berlebihan di kelas

minoritas dengan memilih tiap contoh positif dan menghasilkan data *synthetic* sepanjang segmen garis dengan menggabungkan tetangga terdekat berdasarkan jarak *euclidean* antara dua data (Baesens, Vlasselaer, & Verbeke, 2015). Cara menentukan data *synthetic* dirumuskan dalam persamaan berikut.

$$x_{syn} = x_i + \delta(x_{knn} - x_i), \tag{2.14}$$

keterangan:

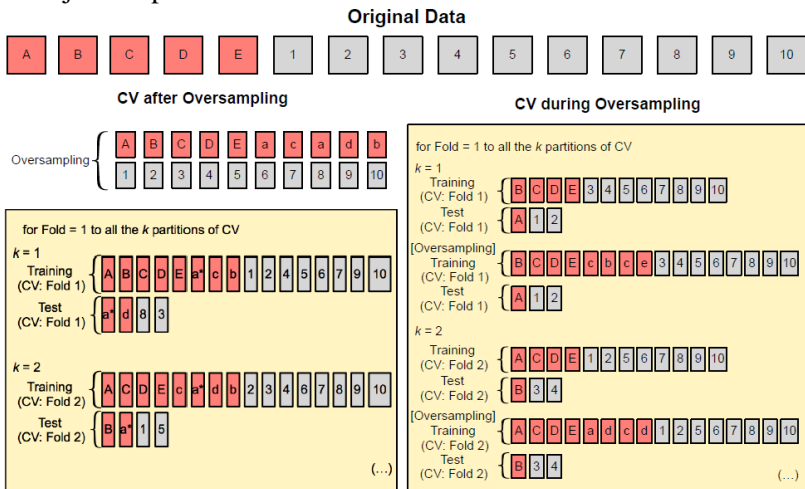
$x_{syn}$  = nilai bobot *synthetic*

$x_i$  = nilai bobot data ke- $i$  di kelas minoritas

$x_{knn}$  = nilai bobot dari data di kelas minoritas yang memiliki jarak terdekat dengan  $x_i$

$\delta$  = bilangan acak antara 0 dan 1

SMOTE akan diterapkan di prosedur *K-Fold Cross Validation* pada masing-masing data *training* di setiap *fold*. Jika *K-Fold CV* dilakukan bersamaan dengan SMOTE, hanya pola data *training* yang dipertimbangkan untuk menghasilkan pola baru dan model dari data *training*, sehingga dapat menghindari *overoptimistic* (Santos, dkk., 2018). Ilustrasi penerapan SMOTE pada *K-Fold CV* ditunjukkan pada Gambar 2.4.

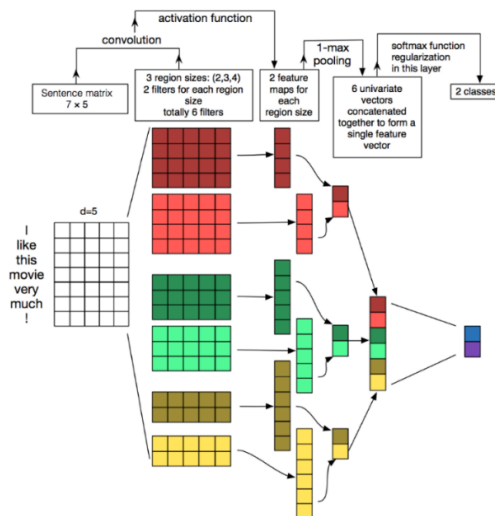


**Gambar 2.4** Ilustrasi SMOTE dalam *K-Fold Cross Validation*

## 2.9 Convolutional Neural Network

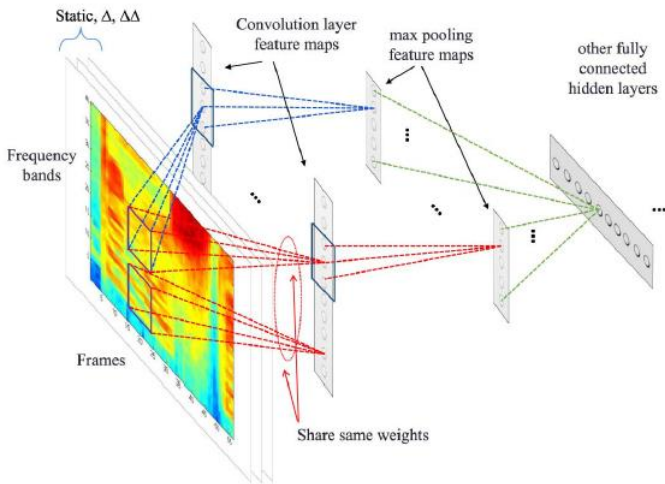
*Convolutional Neural Network* (ConvNet/CNN) adalah algoritma *deep learning* yang dapat menggunakan gambar sebagai *input*, menetapkan kepentingan (bobot dan bias yang dapat dipelajari) untuk berbagai aspek dan objek dalam gambar agar dapat membedakan satu dengan yang lain. Arsitektur dari CNN analog dengan pola konektivitas neuron di otak manusia dan organisasi *visual cortex*. Neuron individu merespons rangsangan hanya di daerah terbatas bidang visual atau yang dikenal sebagai bidang reseptif (Saha, 2018).

CNN termasuk dalam jaringan saraf tiruan *deep feed-forward* (di mana koneksi antar *node* tidak membentuk siklus) dan menggunakan variasi dari *multilayer perceptron* yang dirancang untuk memerlukan *preprocessing* yang tidak banyak. CNN umumnya digunakan dalam *computer vision*, namun CNN juga mulai banyak diterapkan pada berbagai tugas *Natural Language Processing* (Collobert & Weston, 2008). Penggunaan CNN pada data teks ditunjukkan pada diagram berikut.



**Gambar 2.5** Ilustrasi *Text Mining* Menggunakan CNN

Hasil setiap belokan (*convolution*) akan menyala ketika pola khusus terdeteksi. Ukuran kernel dapat divariasikan lalu disatukan hasil *output*-nya, sehingga nantinya dapat mendeteksi pola-pola dengan ukuran yang berkelipatan (2, 3, atau 5 kata yang berdekatan). Pola bisa berupa ekspresi dan karena itu CNN dapat mengidentifikasinya dalam kalimat tertentu terlepas posisi kata-kata tersebut (Maheshwari, 2018). Selain itu CNN juga dapat diterapkan pada hasil ekstraksi menggunakan MFCC. Ilustrasi CNN pada hasil MFCC ditunjukkan pada Gambar 2.6.



**Gambar 2.6** Ilustrasi *Audio Classification* dengan MFCC Menggunakan CNN

Berdasarkan kedua ilustrasi tersebut *layer* yang digunakan dalam CNN terdiri dari 3 jenis yaitu *convolutional layer*, *pooling layer*, dan *fully-connected layer*. *Layer* pertama adalah *convolutional layer* berisi serangkaian *filter* dengan ukuran yang tetap untuk digunakan dalam melakukan konvolusi pada data. *Output* dari *layer* ini adalah matriks baru yang disebut *feature maps*. *Filter* ini berfungsi untuk melihat beberapa pola yang dapat berguna di *layer* selanjutnya (Ferreira & Giraldi, 2017). Persamaan operasi *convolutional* ditunjukkan pada Persamaan 2.15.

$$FM_{a,b} = bias + \sum_c^C \sum_d^D Z_{c,d} \times X_{a+c-1,b+d-1}, \quad (2.15)$$



keterangan:

$FM_{a,b}$  = *feature map* ke- $a,b$ ,

*bias* = bias pada *feature map*,

$Z_{c,d}$  = bobot pada *convolution kernel* ke- $c,d$ ,

$X$  = *input*,

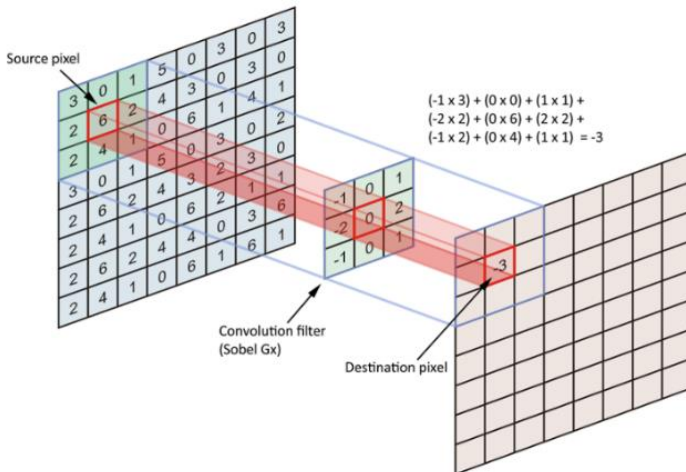
$a$  = 1,2, ...,  $A$ . Di mana  $A$  adalah panjang *pixel feature map*,

$b$  = 1,2, ...,  $B$ . Di mana  $B$  adalah lebar *pixel feature map*,

$c$  = 1,2, ...,  $C$ .  $C$  adalah ukuran panjang *convolution kernel*,

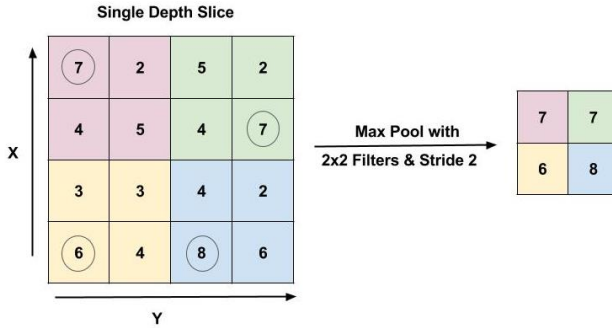
$d$  = 1,2, ...,  $D$ .  $D$  adalah ukuran lebar *convolution kernel*.

Berdasarkan perhitungan tersebut berikut ditunjukkan ilustrasi proses pada *convolutional layer* pada Gambar 2.7.



**Gambar 2.7** Ilustrasi *Convolutional Layer*

*Pooling layer* ini memastikan bahwa jaringan hanya berfokus pada pola yang paling penting. Pada *pooling layer*, data dirangkum dengan menggesar jendela melintasi *feature maps*, lalu menerapkan beberapa operasi linear atau non-linear pada data yang ada di dalam jendela. *Pooling layer* berfungsi untuk mengurangi dimensi dari *feature maps* yang akan digunakan di *layer* selanjutnya (Ferreira & Giraldo, 2017). Ilustrasi *pooling layer* ditunjukkan pada Gambar 2.8.



**Gambar 2.8** Ilustrasi *Pooling Layer*

*Layer* terakhir yang digunakan adalah *fully-connected layer* untuk memahami pola yang dihasilkan oleh *layer* sebelumnya. Neuron di *layer* ini memiliki koneksi penuh ke semua aktivasi di *layer* sebelumnya. *Fully-connected layers* juga sering disebut *inner product layers* (Ferreira & Giraldi, 2017). Metode CNN juga menggunakan fungsi aktivasi yang dilakukan ketika berada di antara *convolutional layer* dan *pooling layer*. Aktivasi di antara kedua *layer* tersebut menggunakan fungsi aktivasi ReLU. Sedangkan untuk fungsi aktivasi *output* menggunakan *sigmoid*. ReLU (*Rectified Linear Unit*) dapat digunakan untuk konvergensi yang lebih cepat pada *training* metode CNN karena fungsi ini mengatasi permasalahan *gradient* yang menghilang dengan menjaga *gradient* lebih atau kurang konstan di semua *layer* jaringan (Ferreira & Giraldi, 2017). Persamaan fungsi aktivasi ReLU adalah sebagai berikut.

$$f_h(0, FM_{a,b}) = \max(0, FM_{a,b}) = \begin{cases} FM_{a,b}, & \text{jika } FM_{a,b} \geq 0, \\ 0, & \text{jika } FM_{a,b} < 0, \end{cases} \quad (2.16)$$

keterangan:

$FM_{a,b}$  = *feature map* ke- $a, b$ ,

$a$  = 1, 2, ...,  $A$ .  $A$  merupakan panjang *pixel* pada *feature map*,

$b$  = 1, 2, ...,  $B$ .  $B$  merupakan panjang *pixel* pada *feature map*,

Fungsi aktivasi pada *output* termasuk menggunakan fungsi *sigmoid* yang dikhususkan pada klasifikasi dengan dua kelas

karena memiliki *range* nilai 0 hingga 1 (Gautam & Ravi, 2017). Persamaan fungsi aktivasi *sigmoid* adalah sebagai berikut.

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.17)$$

Fungsi terakhir adalah *loss function* untuk menghitung *loss* (nilai *error*) dengan menggunakan *binary cross-entropy*. Persamaan dari *loss function* adalah sebagai berikut.

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \times \log(p(y_i)) + (1 - y_i) \times \log(1 - p(y_i)) \quad (2.18)$$

keterangan:

$N$  = jumlah banyak data,

$y_i$  = kelas pada klasifikasi (0 atau 1),

$p(y_i)$  = nilai peluang  $y_i$ .

Setelah mencapai *layer* terakhir dan menghitung fungsi aktivasi *output*, tahapan selanjutnya adalah melakukan optimasi parameter untuk meminimumkan nilai *loss* menggunakan *Adam optimizer*. *Adam optimizer* adalah metode optimisasi stokastik yang diterapkan pada model CNN untuk memperbarui parameter pada jaringan sehingga fungsi tujuan dapat dioptimalkan (Li, dkk., 2016). *Adam* menggunakan *gradient*, estimasi momen pertama dan kedua, serta melakukan koreksi melalui *bias correction*. Tahap pertama adalah menghitung *gradient* dari *loss function* terhadap parameter bias dan pembobot (Kingma & Ba, 2015).

Langkah-langkah algoritma dari optimasi Adam adalah sebagai berikut.

1. Menentukan nilai  $\beta_1$  dan  $\beta_2$  secara *default* berturut-turut sebesar 0,9 dan 0,999.
2. Menentukan fungsi  $f(x)$  yang merupakan *stochastic objective function*.
3. Menginisiasi parameter *moment* pertama dan kedua ( $m_0, v_0$ ).
4. Menghitung gradien dari fungsi (2.18).
5. Melakukan *update* pada *moment* pertama dan kedua dengan formula berikut.

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1)g_t \quad (2.19)$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2)g_t \odot g_t \quad (2.20)$$

6. Menghitung *bias correction* dengan formula (2.21) dan (2.22).

$$\hat{m}_t = \frac{m_t}{(1-\beta_1^t)}, \quad (2.21)$$

$$\hat{v}_t = \frac{v_t}{(1-\beta_2^t)}. \quad (2.22)$$

7. Melakukan *update parameter* jika konvergen maka iterasi berhenti, jika tidak maka mengulang langkah ke-4. Persamaan untuk menghitung tahap terakhir ini ditunjukkan pada persamaan 2.23.

$$\theta_t = \theta_{t-1} - \frac{\eta \cdot \hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}, \quad (2.23)$$

$\theta$  merupakan parameter yang terdiri dari *bias pada input layer* ( $a_j$ ), bobot antara *input layer* dan *hidden layer* ( $w_{i,j}$ ), bias pada *hidden layer* ( $b_k$ ), dan bobot antara *input layer* dan *output layer* ( $v_{j,k}$ ).

## 2.10 Evaluasi Kinerja Klasifikasi

Evaluasi kinerja klasifikasi dibuat sesuai dengan rumus berikut (Al Amrani, Lazaar, & El Kadiri, 2018).

Nilai *True Positive*:

$$TP\ Rate = \frac{TP}{TP+FN} \quad (2.24)$$

*TP Rate* menghasilkan nilai perbandingan antara jumlah kejadian positif yang diklasifikasikan dengan benar dar jumlah elemen keseluruhan yang diklasifikasikan dengan benar.

Nilai *False Positive*.

$$FP\ Rate = \frac{FP}{FP+TN} \quad (2.25)$$

*FP Rate* berhubungan secara simetris dengan nilai *True Positive*. Nilai dari *TP Rate* dan *FP Rate* dibentuk menjadi suatu matriks untuk kelas tertentu yang disebut dengan *confusion matrix*.

Akurasi adalah ukuran umum untuk kinerja klasifikasi dan menghasilkan angka yang proporsional berdasarkan hasil klasifikasi yang benar dari jumlah keseluruhan kejadian. Di mana tingkat kesalahan menggunakan klasifikasi yang salah daripada yang benar.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.26)$$

Nilai AUC digunakan sebagai ukuran untuk klasifikasi biner. Metrik ROC-AUC menunjukkan area di bawah kurva yang diperoleh kurva ROC. AUC menghindari subjektivitas dalam proses pemilihan *threshold* (Lobo, Jiménez-Valverde, & Real, 2008).

$$AUC = \frac{TP\ Rate + (1 - FP\ Rate)}{2} \quad (2.27)$$

### 2.11 Word Cloud

*Word cloud* adalah teknik yang memvisualisasikan kata kunci dalam teks untuk memahami makna atau konsepnya. Sebagai contoh, ukuran kata yang sering disebutkan akan ditingkatkan tergantung pada frekuensi kata tersebut digunakan. Metode ini digunakan untuk menyimpulkan karakteristik data ketika menganalisis *big data* (Park, dkk., 2018). Beberapa contoh dari *word cloud* ditunjukkan pada Gambar 2.9.



Gambar 2.9 Ilustrasi *Word Cloud*

### 2.12 Youtube

YouTube adalah layanan berbagi video di mana pengguna dapat menonton, menyukai, berbagi, berkomentar, dan mengunggah video mereka sendiri. Layanan video dapat diakses di PC, laptop, tablet, dan melalui ponsel (Webwise, 2019).

### 2.13 Ulasan Film

Film terdiri dari gambar bergerak yang telah direkam sehingga dapat ditampilkan di bioskop atau di televisi. Sebuah film menceritakan sebuah kisah atau menunjukkan situasi nyata (Collins, 2019). Ulasan film adalah analisis dari suatu film yang dibuat oleh satu orang atau secara berkelompok lalu mengungkapkan pendapatnya tentang film tersebut (Best Essays, 2019). Media

dari sebuah ulasan beragam, mulai dari tulisan, suara, ataupun melalui video. Ulasan ditujukan untuk menunjukkan pandangan atau penilaian terhadap suatu film serta membantu para pembaca dan penonton untuk mengetahui isi dari suatu film (Syahid, 2019). Para *reviewer* membuat ulasan berdasarkan aspek-aspek tertentu, sehingga klasifikasi sentimen nantinya tidak hanya untuk keseluruhan ulasan tapi berdasarkan aspek yang dinilai dalam ulasan tersebut. Aspek yang digunakan berdasarkan (Thet, Na, & Khoo, 2010) dan (Anand & Naorem, 2016) adalah sebagai berikut.

**Tabel 2.1** Kata Kunci dan Aspek yang Dinilai dalam Ulasan Film

<b>Aspek</b>	<b>Kata Kunci</b>
Pemeran	akting, aktris, aktor, peran, karakter, penjahat, kinerja, performansi, dimainkan, pemeran, pemain, <i>chemistry</i> , pesona
Direktor	arah, arahan, penyutradaraan, sutradara, difilmkan, pembuatan film, pembuat film, sinematik, edisi, sinematografi, direktor
Jalan Cerita	jalan cerita, cerita, kisah, roman, dialog, skrip, bercerita, berakhir, <i>ending</i> , menceritakan, balas dendam, pengkhianatan, plot, penulisan, <i>twist</i> , drama, misteri, <i>thriller</i> , horor
Adegan	pemandangan, animasi, kekerasan, skenario, aksi, efek khusus, <i>special effect</i> , <i>stunt</i> , visual, alat peraga, properti, kamera, grafik, efek, kamera
Musik	lirik, suara, musik, audio, <i>musical</i> , trek, efek suara, <i>sound effect</i> , <i>soundtrack</i>
Keseluruhan	film, <i>remake</i> , adaptasi, <i>sequel</i> , hiburan

## BAB III METODOLOGI PENELITIAN

### 3.1 Sumber Data

Data merupakan data sekunder yang diambil dari Youtube. Data audio berupa suara orang yang mengulas film di-*download* dan dikonversi menjadi data teks serta diekstraksi menggunakan MFCC. Data audio yang digunakan terdiri dari 10 video *review* tiap film. Film yang dipilih adalah 5 film Indonesia yang tayang di tahun 2019. Film dan nama *channel* yang mengulas film tersebut adalah sebagai berikut.

**Tabel 3.1** Daftar Ulasan yang Digunakan sebagai Data

Judul Film	Nama Channel	Judul Film	Nama Channel
Bumi Manusia	#sumatranbigfoot	Dilan 1991	#sumatranbigfoot
	abibayu		Cine Crib
	Cine Crib		Ibnuism - Daily Students
	Kawan Review		Kepin Helmy
	Kepin Helmy		KINCIR - Cinema Club
	KINCIR - Cinema Club		ML Studios
	Red n Blue Cafe		Narasi Entertainment
	The Lazy Cinema		Pure Pixels
	vloggerpedia		Putu Bahagiana
Wandy Oentu	vloggerpedia		
Dua Garis Biru	#sumatranbigfoot	Gundala	#sumatranbigfoot
	Cine Crib		AIIZWEL
	Grid ID		Bang Ray
	Kawan Review		Cine Crib
	Kepin Helmy		Cleansound Studio
	KINCIR - Cinema Club		Kepin Helmy
	LIANT		Pure Pixels

**Tabel 3.1** Daftar Ulasan yang Digunakan sebagai Data (Lanjutan)

Judul Film	Nama Channel	Judul Film	Nama Channel
Dua Garis Biru	Narasi Entertainment	Gundala	RH Entertainment
	Red n Blue Cafe		Wandy Oentu
	Wandy Oentu		Yofie Setiawan
Makmum	#sumatranbigfoot	Makmum	Kawan Review
	Adhe Tora TV		Kepin Helmy
	Cine Crib		Pure Pixels
	Demem Nonton		Red n Blue Cafe
	Erick Antonieta		The Lazy Cinema

Data yang ada diolah melalui *preprocessing* dan selanjutnya akan didapatkan 242 potongan ulasan. Potongan ulasan tersebut telah dikelompokkan berdasarkan aspeknya. Dari 10 ulasan di tiap film, terdapat aspek yang tidak ada di beberapa ulasan sehingga jumlah potongan ulasan bisa berjumlah kurang dari 10.

**Tabel 3.2** Jumlah Potongan Ulasan Tiap Aspek

Aspek	Film	Jumlah Potongan Ulasan
Aspek 1 (Pemeran)	Bumi Manusia	9
	Dilan 1991	7
	Dua Garis Biru	10
	Gundala	9
	Makmum	9
Aspek 2 (Direktor)	Bumi Manusia	9
	Dilan 1991	6
	Dua Garis Biru	9
	Gundala	7
	Makmum	9
...	...	...
Aspek 6 (Keseluruhan)	Bumi Manusia	10
	Dilan 1991	10
	Dua Garis Biru	10
	Gundala	10
	Makmum	10



### 3.2 Variabel Penelitian dan Struktur Data Hasil Ekstraksi MFCC

Variabel penelitian yang akan digunakan terdiri dari variabel prediktor dan variabel respon. Variabel respon adalah sentimen dari aspek tiap ulasan yang ditentukan melalui proses *labelling*.

**Tabel 3.3** Variabel Penelitian Hasil Ekstraksi MFCC

Variabel	Keterangan	Skala Data
Y	Kelas sentimen 0 = Positif 1 = Negatif	Nominal
X	Koefisien MFCC	Interval

Struktur data setelah dilakukan *preprocessing* menggunakan ekstraksi MFCC ditunjukkan pada Tabel 3.4. Ukuran *frame* maksimum sebesar 55.472 *frame* sehingga data dengan ukuran *frame* kurang dari nilai tersebut akan ditambah kolomnya dengan vektor nol (*padding*).

**Tabel 3.4** Struktur Data Hasil Ekstraksi MFCC

Potongan Ulasan Ke-	Kelas Sentimen (y)	Koefisien MFCC		
1	$y_1$	$x_{1,1,1}$	...	$x_{1,1,55472}$
		$x_{1,2,1}$	...	$x_{1,2,55472}$
		:	:	:
		$x_{1,12,1}$	...	$x_{1,12,55472}$
2	$y_2$	$x_{2,1,1}$	...	$x_{2,1,55472}$
		$x_{2,2,1}$	...	$x_{2,2,55472}$
		:	:	:
		$x_{2,12,1}$	...	$x_{2,12,55472}$
...	...	...	...	...
242	$y_{242}$	$x_{242,1,1}$	...	$x_{242,1,55472}$
		$x_{242,2,1}$	...	$x_{242,2,55472}$
		:	:	:
		$x_{242,12,1}$	...	$x_{242,12,55472}$

### 3.3 Variabel Penelitian dan Struktur Data Teks

Variabel penelitian yang akan digunakan untuk klasifikasi sentimen menggunakan data teks ditunjukkan pada Tabel 3.5.

**Tabel 3.5** Variabel Penelitian Data Teks

Variabel	Keterangan	Skala Data
Y	Kelas sentimen 0 = Positif 1 = Negatif	Nominal
X	<i>Word Embedding</i>	Rasio

Struktur data setelah dilakukan *text preprocessing* dan *word embedding* ditunjukkan pada Tabel 3.6. Jumlah kata terbanyak dalam satu ulasan sebanyak 848 kata, sehingga ulasan dengan kata kurang dari itu akan melalui proses *padding* yaitu menambah vektor nol sehingga ukuran matriks akan sama untuk tiap ulasan.

**Tabel 3.6** Struktur Data Teks

Potongan Ulasan Ke-	Kelas Sentimen (y)	<i>Word Embedding</i>		
1	$y_1$	$x_{1,1,1}$	...	$x_{1,1,200}$
		$x_{1,2,1}$	...	$x_{1,2,200}$
		:	:	:
		$x_{1,848,1}$	...	$x_{1,848,200}$
2	$y_2$	$x_{2,1,1}$	...	$x_{2,1,200}$
		$x_{2,2,1}$	...	$x_{2,2,200}$
		:	:	:
		$x_{2,848,1}$	...	$x_{2,848,200}$
...	...	...	...	...
242	$y_{242}$	$x_{242,1,1}$	...	$x_{242,1,200}$
		$x_{242,2,1}$	...	$x_{242,2,200}$
		:	:	:
		$x_{242,848,1}$	...	$x_{242,848,200}$

### 3.4 Langkah Analisis

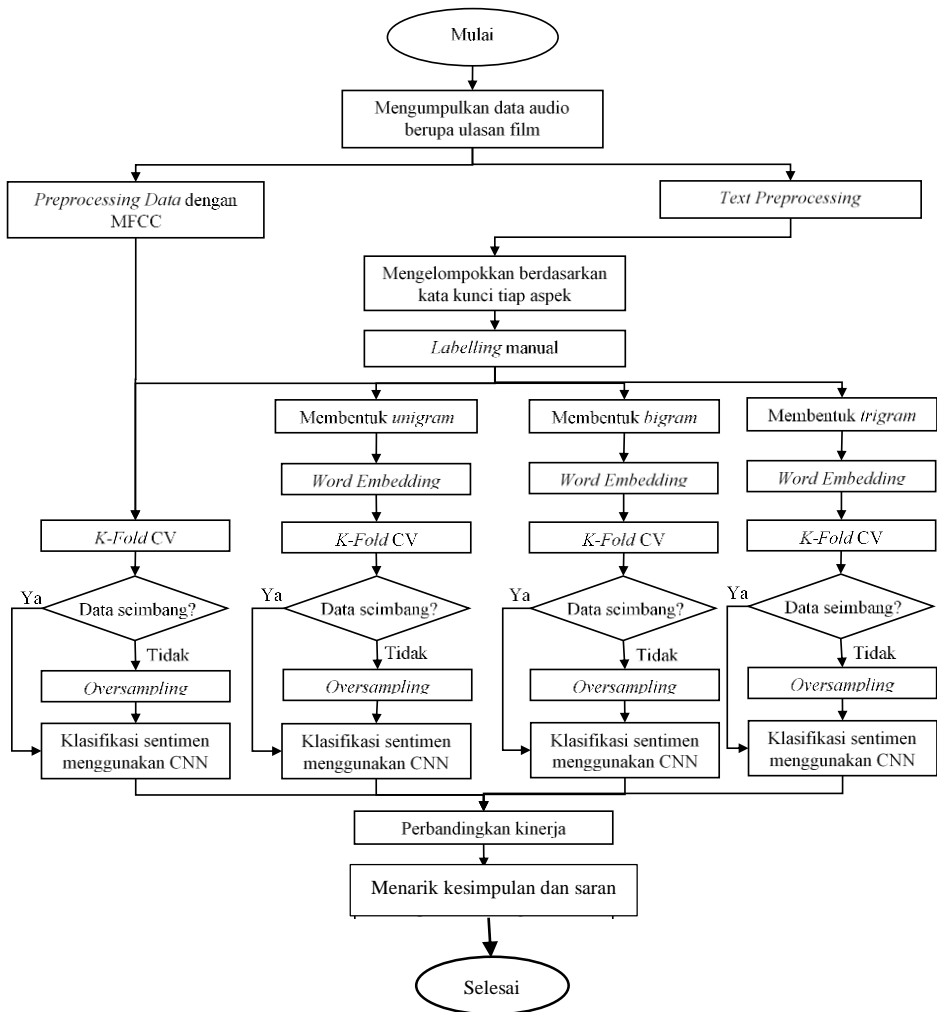
Langkah analisis yang akan dilakukan pada penelitian ini adalah sebagai berikut.

1. Mengumpulkan audio dari video ulasan film Indonesia yang ada di Youtube.
2. Memotong audio menjadi potongan audio berdasarkan keheningan audio dengan ketentuan durasi keheningan (*silence*) mi-

- nimal 350 ms dan audio tersebut dikatakan hening jika amplitudo suara lebih kecil dari -25 hingga -30 dBFS.
3. Melakukan proses *Speech-to-Text* pada data audio dengan bantuan API *Google Speech Cloud* untuk mengambil data teks.
  4. Melakukan *text preprocessing* dengan tahap-tahap sebagai berikut.
    - a. *Case folding*,
    - b. *Stemming*,
    - c. *Stopword Removal*,
    - d. *Tokenizing*.
  5. Mengelompokkan potongan teks tersebut berdasarkan aspek ulasan yang disebutkan pada Tabel 2.1.
  6. Pelabelan sentimen data ulasan film secara manual.
  7. Melakukan pembobotan menggunakan *word embedding* dengan algoritma *Word2Vec*.
  8. Melakukan ekstraksi audio menggunakan MFCC dengan tahapan sebagai berikut.
    - a. Melakukan *import data* audio dan mendapatkan nilai amplitudo.
    - b. Melakukan tahapan *pre-emphasis* pada masing-masing *file audio*.
    - c. Membagi sinyal ke dalam *frame-frame* yang lebih kecil dengan ukuran *frame* 25 ms dan 10 ms *stride* (15 ms *overlap*).
    - d. Menerapkan *window* pada setiap *frame* sesuai dengan persamaan (2.2).
    - e. Mengaplikasikan *FFT* pada setiap *frame* untuk menghitung spektrum frekuensi dengan  $N = 512$  berdasarkan persamaan (2.3) dan (2.4).
    - f. Melakukan *filter banks* sebanyak 40 pada skala *mel* menjadi *power spectrum* untuk mendapatkan koefisien *filter banks* dengan persamaan (2.5), (2.6), (2.7), dan (2.8).
    - g. Menghitung nilai MFCC menggunakan *Discrete Cosine Transform* (DCT) dengan nilai koefisien *cepstrum* adalah 12 berdasarkan persamaan (2.9) dan (2.10).

9. Membagi data menjadi data *training* dan data *testing* menggunakan *K-Fold Cross Validation*.
10. Melakukan teknik SMOTE jika pembagian data *training* dan data *testing* tidak seimbang (*imbalanced*).
11. Klasifikasi sentimen ulasan film Indonesia di Youtube menggunakan metode CNN untuk vektor kata dan nilai MFCC dengan langkah-langkah sebagai berikut.
  - a. Menentukan ukuran *convolution kernel* dan *pooling*.
  - b. Melakukan perhitungan operasi konvolusi berdasarkan persamaan (2.15).
  - c. Menghitung fungsi aktivasi ReLU sesuai persamaan (2.16).
  - d. Melakukan operasi *max-pooling*.
  - e. Melakukan perhitungan berdasarkan persamaan (2.16) dan (2.17) di mana input yang digunakan adalah vektor dari langkah sebelumnya dengan jumlah *convolutional layer* sebanyak 6 dengan 1 *pooling layer* di antara *layer* kedua dan ketiga.
  - f. Jika nilai *loss* kurang dari nilai minimum yaitu 0,05 maka lanjut ke langkah h. Jika tidak, maka lanjut ke langkah i.
  - g. Jika iterasi telah mencapai batas maksimum yaitu 100 maka lanjut ke langkah h. Jika tidak maka lanjut ke langkah i.
  - h. Melakukan *update* parameter (bias dan bobot) menggunakan parameter optimasi *Adam*.
  - i. Parameter baru digunakan kembali untuk iterasi berikutnya dari langkah c hingga langkah g.
  - j. Iterasi *training* selesai. Model, bias, bobot, serta nilai *loss* yang telah didapat diuji pada data *testing*.
12. Memilih model hasil *text mining* dan hasil *audio mining* dengan metode CNN berdasarkan nilai AUC dan akurasi.
13. Interpretasi, mengambil kesimpulan dan saran.

Langkah-langkah penelitian tersebut digambarkan dalam bentuk diagram yang diilustrasikan pada Gambar 3.1.



**Gambar 3.1** Diagram Alir

*(Halaman ini sengaja dikosongkan)*

## BAB IV ANALISIS DAN PEMBAHASAN

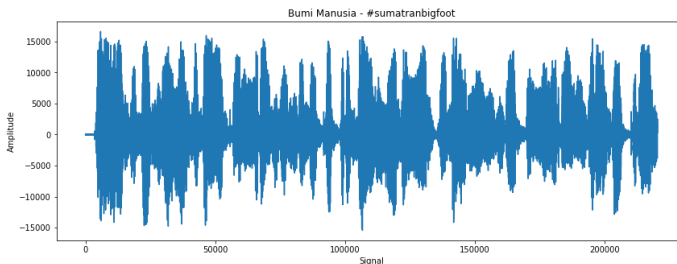
Bab ini membahas tentang hasil analisis dari klasifikasi sentimen menggunakan metode *Convolutional Neural Network*. Bab ini menjelaskan proses pengolahan data audio hingga menjadi data hasil ekstraksi *Mel Frequency Cepstral Coefficient* dan data teks yang akan digunakan sebagai data untuk klasifikasi sentimen.

### 4.1 Karakteristik Ulasan Film

Karakteristik data diperoleh setelah melalui preprocessing. Data yang digunakan dalam klasifikasi sentimen adalah data audio, sehingga dibutuhkan *preprocessing* karena data audio tidak bisa langsung diolah menggunakan metode klasifikasi. Data audio akan diekstraksi terlebih dahulu menggunakan MFCC dan dikonversi menjadi data teks dengan bantuan *speech-to-text*.

#### 4.1.1 Data Preprocessing Menggunakan MFCC

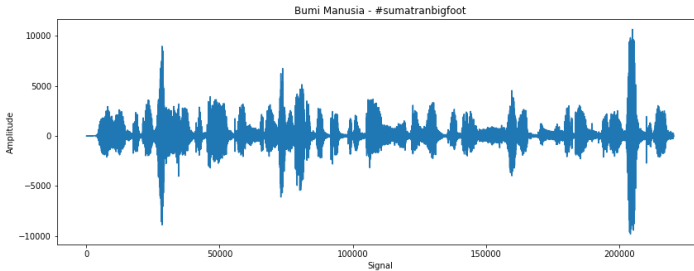
Ilustrasi ini menampilkan proses tahapan fitur ekstraksi MFCC dengan salah satu potongan audio *review* film Bumi Manusia oleh #sumatranbigfoot. Audio yang diambil berdurasi 5 detik. Sinyal yang dihasilkan sejumlah 220.500 dengan grafik amplitudo sebagai berikut.



**Gambar 4.1** Amplitudo Data Audio *Review* Film

#### 1. *Pre-Emphasis*

Pada tahapan pertama, formula yang digunakan adalah berdasarkan Persamaan 2.1. Setelah melakukan proses *pre-emphasis* didapatkan amplitudo baru dengan grafik yang ditunjukkan pada Gambar 4.2.



**Gambar 4.2** Amplitudo Data Audio Setelah *Pre-Emphasis*

## 2. Framing

Tahapan *framing* membagi sinyal ke dalam *frame* yang lebih kecil. *Frame size* yang akan digunakan sebesar 25 ms dan *frame stride* yang digunakan adalah 10 ms. Jumlah *frame* yang akan digunakan dihitung menggunakan formula berikut.

$$\text{number of frames} = \frac{|\text{signal length} - \text{frame length}|}{\text{frame step}}$$

keterangan:

*signal length* : jumlah sinyal hasil *pre-emphasis*

*frame length* : *frame size* x *sample rate*

*frame step* : *frame stride* x *sample rate*

Berdasarkan formula tersebut didapatkan jumlah *frame* yang digunakan pada perhitungan berikut.

$$\text{number of frames} = \frac{|220500 - 1102|}{441} = 498$$

Jumlah *frame* sebesar 498 sehingga data magnitudo yang ada dibagi dengan  $n$  sebesar 1102 untuk jumlah sinyal tiap *frame*.

**Tabel 4.1** Hasil *Framing*

$n$	<i>Frame</i> 1	<i>Frame</i> 2	<i>Frame</i> 3	...	<i>Frame</i> 496	<i>Frame</i> 497	<i>Frame</i> 498
0	-23	4,66	-1,09	...	-56,98	307,51	-7,62
1	-1,69	4,78	-0,12	...	-79,89	294,08	-29,43
2	-0,72	0,9	-1,12	...	-101,4	265,96	-35,87
...	...	...	...	...	...	...	...
1100	-0,11	1,21	-1,05	...	-157,09	-220,54	-23,2
1101	1,92	-0,76	1,95	...	-176,21	-202,77	21,95



### 3. Windowing

Tahap ketiga melakukan *Hamming Windowing* menggunakan Persamaan 2.2. Contoh hasil perhitungan *windowing* pada data pertama dan data kedua di *frame* 1 adalah sebagai berikut.

$$w[0] = (-23)0,54 - 0,46 \cos\left(\frac{2\pi(0)}{1102 - 1}\right) = -1,840$$

$$w[1] = (-1,69)0,54 - 0,46 \cos\left(\frac{2\pi(1)}{1102 - 1}\right) = -0,135$$

$N$  adalah panjang *frame* dengan nilai sebesar 1102 dan data pertama dimulai dari  $n = 0$ . Hasil *windowing* selengkapnya ditampilkan pada Tabel 4.2 berikut.

**Tabel 4.2** Hasil *Windowing* pada Tiap *Frame*

$n$	<i>Frame</i> 1	<i>Frame</i> 2	<i>Frame</i> 3	...	<i>Frame</i> 496	<i>Frame</i> 497	<i>Frame</i> 498
<b>0</b>	-1,840	0,373	-0,087	...	-4,558	24,601	-0,610
<b>1</b>	-0,135	0,382	-0,010	...	-6,392	23,529	-2,355
<b>2</b>	-0,058	0,072	-0,090	...	-8,115	21,285	-2,871
<b>3</b>	0,28	-1,1	-1,15	...	-111,45	235,73	-25,46
...	...	...	...	...	...	...	...
<b>1100</b>	-0,009	0,097	-0,084	...	-12,568	-17,645	-1,856
<b>1101</b>	0,154	-0,061	0,156	...	-14,097	-16,222	1,756

### 4. Fourier-Transform dan Power Spectrum

Contoh perhitungan *fourier transform* dan *power spectrum* untuk data pertama di *frame* pertama adalah sebagai berikut.

$$|FFT(x_0)| = \left| \sum_{n=0}^{511} x_n e^{-\frac{j2\pi(0)n}{512}} \right| = 43,109$$

Setelah mendapatkan nilai FFT dilanjutkan dengan mengitung *power spectrum* seperti pada persamaan berikut.

$$P = \frac{|FFT(x_0)|^2}{N} = \frac{(43,109)^2}{512} = 3,630$$

Perhitungan *power spectrum* dilakukan pada masing-masing *frames*. Nilai *power spectrum* secara keseluruhan ditampilkan pada Tabel 4.3.

**Tabel 4.3** Hasil Nilai *Power Spectrum* pada Tiap *Frame*

<i>n</i>	<i>Frame</i> <b>1</b>	<i>Frame</i> <b>2</b>	<i>Frame</i> <b>3</b>	...	<i>Frame</i> <b>496</b>	<i>Frame</i> <b>497</b>	<i>Frame</i> <b>498</b>
<b>0</b>	3,630	4,630	0,121	...	9411,280	5638,026	18595,539
<b>1</b>	0,209	4,259	7,584	...	54130,104	12969,192	12492,473
<b>2</b>	6,092	1,220	8,692	...	83160,078	161118,042	92009,659
...	...	...	...	...	...	...	...
<b>255</b>	0,455	0,176	0,090	...	3,355	0,834	7,471
<b>256</b>	0,004	0,160	0,061	...	5,879	0,791	10,146

### 5. *Filter Banks*

Tahapan ini diawali dengan mengkonversi nilai frekuensi tertinggi menjadi satuan *mel* dengan persamaan sebagai berikut.

$$m = 2595 \log_{10} \left( 1 + \frac{\left( \frac{44100}{2} \right)}{700} \right) = 3923,34$$

Selanjutnya satuan *mel* terendah yaitu 0 dan satuan *mel* tertinggi senilai 3923,34 menjadi batas bawah dan batas atas untuk *filter*. Karena jumlah *filter* yang digunakan 40 maka dalam rentang itu dibagi menjadi 40 nilai sama besar (selain kedua nilai tersebut) dan selanjutnya dikonversikan menjadi satuan Hz. Setelah mendapatkan rentang satuan Hz dilanjutkan menghitung nilai pada  $f(m)$  yang berupa fungsi berdasarkan persamaan berikut.

$$f(m) = \frac{(512 + 1) \times f}{44100}$$

Hasil persamaan berikut akan menghasilkan suatu *array* di mana  $f(m)$  adalah nilai dari fungsi tersebut pada *array* ke- $m$ . Selanjutnya koefisien *filter banks* yang dihasilkan ditampilkan pada Tabel 4.4.

**Tabel 4.4** Hasil Koefisien *Filter Banks* pada Tiap *Frame*

<i>n</i>	<i>Frame</i> <b>1</b>	<i>Frame</i> <b>2</b>	<i>Frame</i> <b>3</b>	...	<i>Frame</i> <b>496</b>	<i>Frame</i> <b>497</b>	<i>Frame</i> <b>498</b>
<b>0</b>	11,197	13,312	-18,321	...	79,473	75,023	85,388
<b>1</b>	-13,581	12,586	17,598	...	94,669	82,258	81,933
<b>2</b>	15,695	1,730	18,783	...	98,398	104,143	99,277

**Tabel 4.4** Hasil Koefisien *Filter Banks* pada Tiap *Frame* (Lanjutan)

<i>n</i>	<i>Frame</i> <b>1</b>	<i>Frame</i> <b>2</b>	<i>Frame</i> <b>3</b>	...	<i>Frame</i> <b>496</b>	<i>Frame</i> <b>497</b>	<i>Frame</i> <b>498</b>
...	...	...	...	...	...	...	...
<b>38</b>	8,822	8,047	8,486	...	39,639	23,954	41,423
<b>39</b>	14,104	9,731	10,888	...	40,174	22,281	41,771

#### 6. *Discrete Cosine Transform*

Tahap terakhir dalam fitur ekstraksi ini adalah menerapkan *Discrete Cosine Transform*. Perhitungan menggunakan DCT pada data kedua ( $k=1$ ) di *frame* pertama dituliskan pada persamaan di halaman selanjutnya.

$$y(1) = f(k) \times 2 \sum_{n=0}^{11} x_n \cos\left(\pi k \left(\frac{2n+1}{2(12)}\right)\right) = 24,844$$

Hasil perhitungan nilai MFCC selengkapnya ditunjukkan pada Tabel 4.5.  $k$  menunjukkan nilai koefisien ke- $k$ .

**Tabel 4.5** Nilai MFCC pada Tiap *Frame*

<i>k</i>	<i>Frame</i> <b>1</b>	<i>Frame</i> <b>2</b>	<i>Frame</i> <b>3</b>	...	<i>Frame</i> <b>496</b>	<i>Frame</i> <b>497</b>	<i>Frame</i> <b>498</b>
<b>0</b>	24,844	38,396	21,197	...	120,435	152,063	119,787
<b>1</b>	-50,444	-33,299	-22,901	...	-19,544	-31,809	-22,462
<b>2</b>	16,594	-17,793	-27,740	...	22,526	14,142	-9,685
<b>3</b>	-9,453	17,381	34,881	...	42,357	38,487	28,684
<b>4</b>	-36,541	-15,647	3,534	...	4,001	8,023	24,749
...	...	...	...	...	...	...	...
<b>10</b>	-5,999	3,915	-3,417	...	-14,041	-20,501	-10,898
<b>11</b>	-8,951	-8,031	-27,553	...	-8,856	-0,585	-9,583

Hasil Nilai MFCC yang telah melalui beberapa tahapan yang telah diilustrasikan digunakan sebagai data untuk proses klasifikasi sentimen ulasan video dengan *Convolutional Neural Network*.

#### 4.1.2 *Data Preprocessing Menggunakan Speech-To-Text*

Ilustrasi ini menampilkan tahapan *text pre-processing* dan *word embedding* dengan salah satu potongan audio ulasan film

Bumi Manusia oleh #sumatranbigfoot. Audio tersebut dikonversi menjadi data teks menggunakan *Speech-to-Text* dan melalui tahap *pre-processing* dengan tahapan pada Tabel 4.6.

**Tabel 4.6** Ilustrasi *Speech-to-Text* dan *Text Preprocessing*

<b>Tahap</b>	<b>Output</b>
<i>Speech-to-Text</i>	kita akan merasa sedikit terganggu dengan tidak mulus siap menggunakan green screen dan render anti patah-patah yang pada akhirnya membuat film ini terlihat sedikit murah
<i>Case Folding</i>	kita akan merasa sedikit terganggu dengan tidak mulus siap menggunakan green screen dan render anti patah-patah yang pada akhirnya membuat film ini terlihat sedikit murah
<i>Stopwords Remover</i>	akan merasa sedikit terganggu tidak mulus siap menggunakan green screen render anti patah-patah pada akhirnya membuat film terlihat sedikit murah
<i>Stemming</i>	akan rasa sedikit ganggu tidak mulus siap guna green screen render anti patah pada akhir buat film lihat sedikit murah
<i>Tokenizing</i>	['akan', 'rasa', 'sedikit', 'ganggu', 'tidak', 'mulus', 'siap', 'guna', 'green', 'screen', 'render', 'anti', 'patah', 'pada', 'akhir', 'buat', 'film', 'lihat', 'sedikit', 'murah']

Tahapan selanjutnya adalah *word embedding* dengan algoritma *word2vec* di mana tiap kata yang muncul akan diubah menjadi angka berbentuk vektor. Sebelumnya *word2vec* perlu dimodelkan terlebih dahulu untuk mendapatkan nilai vektor pada masing-masing kata. Model didapatkan setelah melakukan *training Word2Vec* menggunakan *corpus Wikipedia*. Setelah model didapat, hasil *text preprocessing* diubah melalui *word embedding* dengan menggunakan algoritma *Word2Vec*.

Masing-masing kata memiliki nilai vektor berukuran 200. Ukuran sebesar 200 ditentukan karena jumlah kata yang ada tidak sedikit sehingga semakin besar ukuran vektor maka akan mengurangi kemungkinan nilai vektor yang sama antar kata. Hasil dari *word embedding* ini akan digunakan sebagai data untuk klasifikasi sentimen menggunakan *Convolutional Neural Network*.

**Tabel 4.7** Ilustrasi *Word Embedding* dengan *Word2Vec*

akan	rasa	sedikit	ganggu	...	sedikit	murah
1,431	0,540	-0,298	-0,432	...	-0,298	1,607
1,504	1,500	1,739	0,073	...	1,739	2,546
...	...	...	...	...	...	...
-0,622	-1,480	-0,185	-0,001	...	-0,185	0,246
0,887	2,153	0,770	0,706	...	0,770	-0,060

*Word2Vec* adalah *word embedding* dengan *similarity* yang dihitung dari kata-kata yang berdekatan. Ilustrasi sederhana jika terdapat dua kalimat “raja pria berani” dan “ratu perempuan cantik”. Pengelompokkan kata dari kedua kalimat tersebut menggunakan *window size* sebesar 2 maka didapatkan tabel kata dan *neighbor* sebagai berikut.

**Tabel 4.8** Kata dan *Neighbor* dengan *window size* = 2

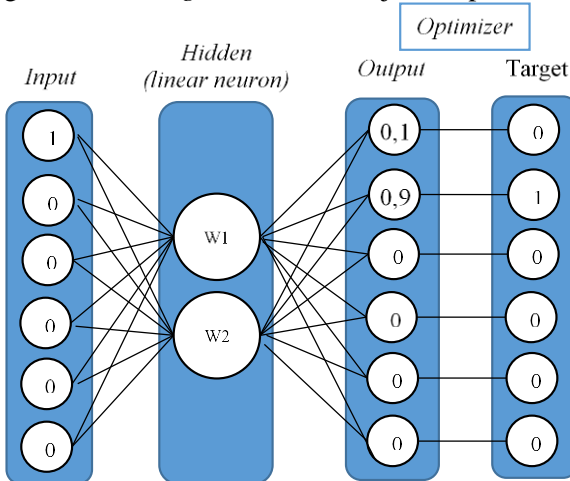
Kata	Neighbor
Raja	Berani
Raja	Pria
Berani	Raja
Berani	Pria
Pria	Raja
Pria	Berani
Ratu	Cantik
Ratu	Perempuan
Cantik	Ratu
Cantik	Perempuan
Perempuan	Ratu
Perempuan	Cantik

Selanjutnya, *encoding* dilakukan pada setiap kata yang ada baik pada tiap kata dan *neighbor* dengan menandai tiap kata dengan vektor yang terdiri dari angka satu dan nol seperti yang ditunjukkan pada Tabel 4.9.

**Tabel 4.9** *Encoding pada Kata dan Neighbor*

Kata	One Hot Encoding	Neighbor	One Hot Encoding
Raja	[1,0,0,0,0,0]	Berani	[0,0,1,0,0,0]
Raja	[1,0,0,0,0,0]	Pria	[0,1,0,0,0,0]
Berani	[0,0,1,0,0,0]	Raja	[1,0,0,0,0,0]
Berani	[0,0,1,0,0,0]	Pria	[0,1,0,0,0,0]
Pria	[0,1,0,0,0,0]	Raja	[1,0,0,0,0,0]
Pria	[0,1,0,0,0,0]	Berani	[0,0,1,0,0,0]
Ratu	[0,0,0,1,0,0]	Cantik	[0,0,0,0,0,1]
Ratu	[0,0,0,1,0,0]	Perempuan	[0,0,0,0,1,0]
Cantik	[0,0,0,0,0,1]	Ratu	[0,0,0,1,0,0]
Cantik	[0,0,0,0,0,1]	Perempuan	[0,0,0,0,1,0]
Perempuan	[0,0,0,0,1,0]	Ratu	[0,0,0,1,0,0]
Perempuan	[0,0,0,0,1,0]	Cantik	[0,0,0,0,0,1]

*One hot encoding* yang telah dibuat digunakan sebagai *input layer* dan *output layer* dalam *word2vec*. Misalkan ‘raja’ sebagai *input layer* maka terdapat 2 target yang mungkin yaitu ‘berani’ dan ‘pria’. Algoritma *training word2vec* ditunjukkan pada Gambar 4.3.

**Gambar 4.3** Arsitektur *Word2Vec* pada Kata ‘Raja’ dan ‘Pria’

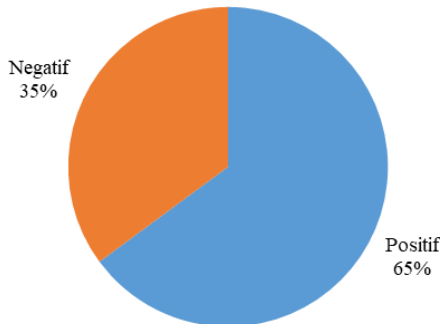
Karena jumlah *linear neuron* sebanyak dua, maka vektor tiap kata nanti akan memiliki dua nilai. Hasil vektor kata inilah yang nantinya digunakan sebagai data dalam proses klasifikasi sentimen. Setelah proses *Word2Vec* dilakukan maka tiap kata akan memiliki vektor *embedding* dengan ilustrasi sebagai berikut.

**Tabel 4.10** Hasil *Embedding* pada Tiap Kata

<b>Kata</b>	<b><i>Embedding</i></b>
Raja	[1,1]
Pria	[1,2]
Berani	[1,3]
Ratu	[5,5]
Perempuan	[5,6]
Cantik	[5,7]

#### 4.1.3 Hasil Karakteristik Ulasan Film

Data yang telah melalui *preprocessing* digunakan untuk melihat karakteristik dari ulasan film. Penentuan *label* sentimen positif dan negatif dilakukan secara *manual* dengan membaca hasil ulasan yang telah dipraproses dan dikelompokkan. Ulasan tersebut akan ditentukan positif atau negatifnya berdasarkan kecenderungan tulisan tersebut. Jumlah data yang akan digunakan sebanyak 242 sentimen ulasan film dengan persentase antara sentimen positif dan negatif ditunjukkan pada grafik berikut.

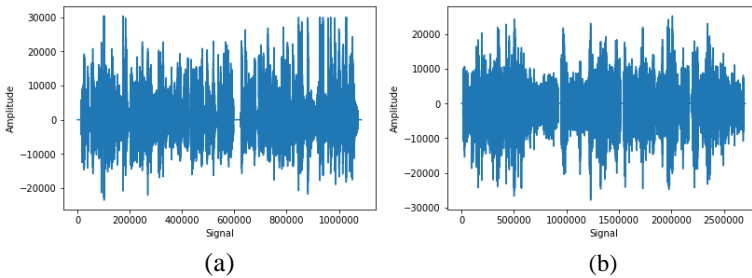


**Gambar 4.4** *Pie Chart* Persentase Sentimen Negatif dan Positif

Berdasarkan Gambar 4.4, sentimen positif yang digunakan memiliki jumlah yang lebih besar dibandingkan sentimen negatif. Hal ini menyebabkan data tidak seimbang sehingga nanti ketika

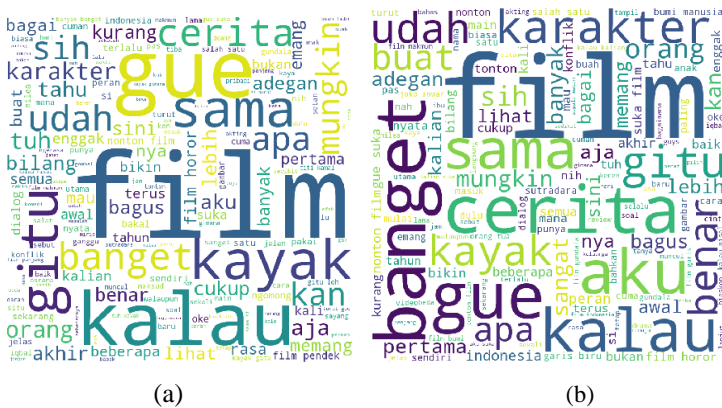
dilakukan pemodelan klasifikasi sentimen, metode SMOTE akan digunakan dalam pembagian data *training* dan *testing* untuk menyeimbangkan data *imbalance*.

Ilustrasi sinyal dari potongan data audio yang digunakan ditunjukkan pada Gambar 4.5. Potongan audio tersebut jika dilihat secara visual, perbedaan antara audio dengan sentimen positif dan sentimen negatif tidak terlalu nampak. Sehingga akan sulit membedakan audio tersebut positif atau negatif berdasarkan plot sinyal.



**Gambar 4.5** Plot Sinyal Potongan Audio Bersentimen (a) Negatif (b) Positif

Berdasarkan Gambar 4.6, teks ulasan film yang memiliki sentimen negatif dan positif tidak terlalu memiliki perbedaan. Sehingga, metode klasifikasi dapat membantu menentukan sentimen dari ulasan film yang ada.

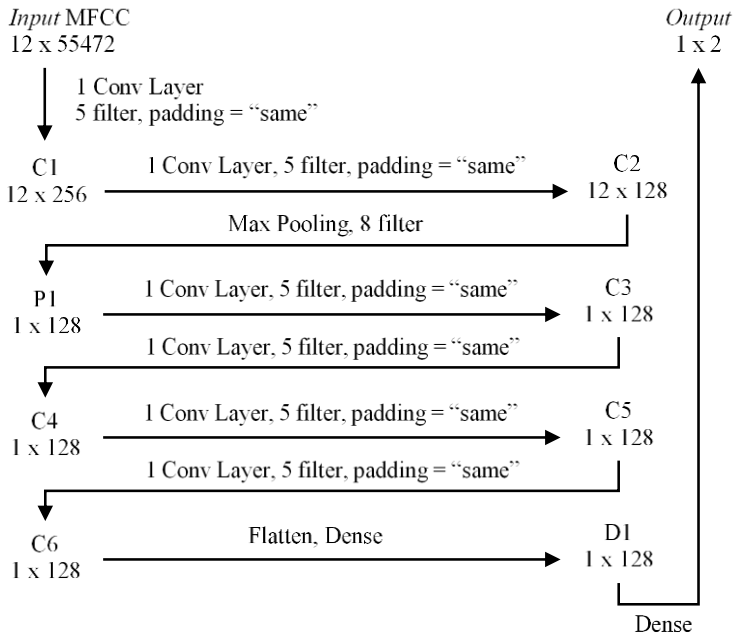


**Gambar 4.6** Word Cloud Ulasan Film Bersentimen (a) Negatif (b) Positif



## 4.2 Hasil Klasifikasi Data MFCC dengan Metode CNN

Klasifikasi sentimen dapat menggunakan data hasil ekstraksi MFCC dengan menggunakan metode CNN. Pada penelitian kali ini, klasifikasi sentimen yang dilakukan menggunakan keseluruhan data ulasan dan mengklasifikasikan sentimen berdasarkan tiap-tiap aspek penilaian film. Arsitektur CNN yang digunakan diilustrasikan pada Gambar 4.7.



**Gambar 4.7** Arsitektur CNN dalam Klasifikasi Data MFCC

*Convolutional layer* yang digunakan sebanyak 6 layer dengan *max pooling* berada di antara layer kedua dan ketiga. Karena CNN yang digunakan adalah CNN 1D (1 Dimensi), maka *convolution window* hanya bergerak 1 arah saja. Dengan ukuran *filter* 5 maka hasil *convolution layer* pertama seharusnya sebanyak 8 layer namun dilakukan proses *padding* yang menambah kolom nol sebanyak jumlah baris di *input* sebelumnya. *Output max pooling* hanya berukuran  $1 \times 128$  karena jumlah baris sebelumnya hanya sebe-

sar 12 sehingga baris terambil dalam proses *max pooling* hanya satu baris saja.

Berikut adalah salah satu persamaan dari *convolution layer* pertama untuk model menggunakan data MFCC.

$$Y_{1,1} = -0,0034 + (-0,0144)X_{1,1} + (-0,0106)X_{2,1} + (-0,0017)X_{3,1} \\ + (-0,0103)X_{4,1} + (-0,0112)X_{5,1}$$

$$Y_{1,2} = -0,0019 + (-0,0031)X_{1,2} + 0,0038X_{2,2} + 0,0141X_{3,2} \\ + (-0,0021)X_{4,2} + (-0,0150)X_{5,2}$$

$$\dots \\ Y_{1,256} = -0,0018 + 0,0085X_{1,256} + (-0,0392)X_{2,256} \\ + (-0,0133)X_{3,256} + (-0,0133)X_{4,256} \\ + (-0,0098)X_{5,256}$$

Persamaan dari *convolution layer* akan berjalan hingga  $Y_{8,256}$  karena ukuran *output* dari *layer* pertama menjadi (8,256) pada tahapan selanjutnya akan dilakukan *padding* dengan menambahkan vektor nol agar *output* menjadi ukuran (12,256).

#### 4.2.1 Hasil Klasifikasi Keseluruhan Data MFCC

Data hasil ekstraksi MFCC digunakan untuk klasifikasi sentimen menggunakan metode CNN. Didapatkan nilai evaluasi kerja klasifikasi tiap *fold* yang ditunjukkan pada Tabel 4.11.

**Tabel 4.11** Nilai Kinerja Klasifikasi Keseluruhan Data MFCC

K	AUC		Accuracy	
	Training	Testing	Training	Testing
2	1,000	0,533	1,000	0,624
3	0,997	0,563	0,998	0,612
4	1,000	0,570	1,000	0,591
5	1,000	0,548	1,000	0,619
6	0,998	0,530	0,998	0,615
7	0,999	0,538	0,999	0,554
<b>8</b>	<b>0,999</b>	<b>0,585</b>	<b>0,999</b>	<b>0,652</b>
9	0,997	0,569	0,998	0,628
10	0,999	0,565	1,000	0,627

Berdasarkan hasil kinerja evaluasi, model terbaik dengan nilai AUC data *testing* tertinggi sebesar 0,585 pada *fold* 8. Akurasi dari model tersebut sebesar 0,652. Sehingga model yang akan digunakan untuk dibandingkan dengan model lain adalah model CNN pada *fold* 8. *Confusion matrix* untuk hasil klasifikasi sentimen pada keseluruhan data hasil ekstraksi MFCC ditunjukkan pada Tabel 4.12.

**Tabel 4.12** *Confusion Matrix* Klasifikasi Keseluruhan Data MFCC

<b>Kelas Aktual</b>	<b>Kelas Prediksi</b>		<b>Total</b>
	Positif	Negatif	
Positif	82	3	<b>85</b>
Negatif	7	150	<b>157</b>
<b>Total</b>	<b>89</b>	<b>153</b>	<b>242</b>

Model klasifikasi sentimen untuk seluruh data ekstraksi MFCC dapat memprediksi dengan tepat 232 sentimen positif maupun negatif. Hanya 10 data yang belum bisa diprediksi secara tepat.

#### 4.2.2 Hasil Klasifikasi Data MFCC pada Tiap Aspek

Klasifikasi selanjutnya adalah klasifikasi pada data hasil ekstraksi MFCC yang dikelompokkan berdasarkan aspek penilaiannya. Aspek yang digunakan adalah aspek pemeran, direktor, jalan cerita, adegan, musik, dan keseluruhan. Tiap model dipilih berdasarkan nilai AUC tertinggi pada masing-masing aspek dan dihasilkan rata-rata nilai AUC dan akurasi yang baru. Hasil *fold* terbaik untuk masing-masing aspek ditunjukkan pada Tabel 4.13.

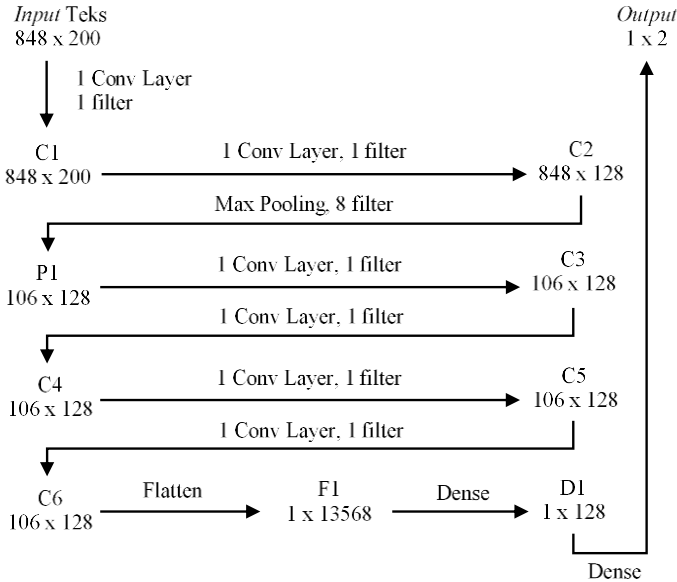
**Tabel 4.13** Nilai Kinerja Klasifikasi Data MFCC Tiap Aspek

<b>Aspek</b>	<b>K</b>	<b>Fold</b>	<b>AUC</b>		<b>Accuracy</b>	
			<b>Training</b>	<b>Testing</b>	<b>Training</b>	<b>Testing</b>
Pemeran	7	1	1,000	0,690	1,000	0,650
Direktor	10	1	0,997	0,521	0,997	0,475
Jalan Cerita	3	3	1,000	0,696	1,000	0,792
Adegan	9	6	1,000	0,617	1,000	0,522
Musik	2	2	0,917	0,500	0,950	0,805
Keseluruhan	8	2	0,555	0,550	0,700	0,661
<b>Rata-Rata</b>			0,912	0,596	0,941	0,651

Model terbaik untuk keenam aspek menurut Tabel 4.12 memiliki rata-rata AUC sebesar 0,596 untuk data *testing*, dengan rata-rata akurasi klasifikasi data *testing* sebesar 0,651. Hasil *confusion matrix* dari klasifikasi sentimen untuk data MFCC di tiap aspek dilampirkan pada Lampiran 5.

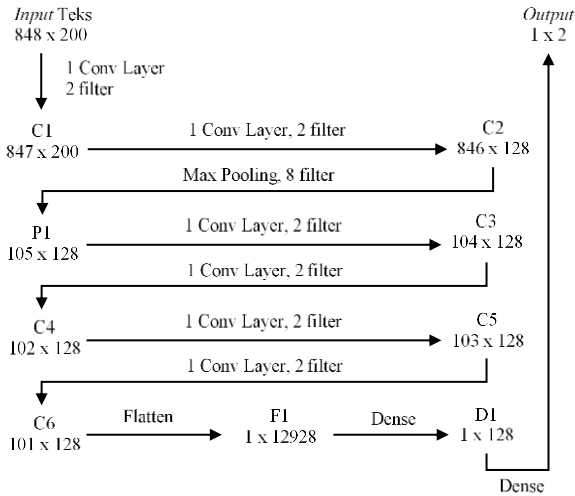
### 4.3 Hasil Klasifikasi Data Teks dengan Metode CNN

Klasifikasi sentimen pada data teks dibagi menjadi dua yaitu dengan menggunakan semua ulasan yang ada dan pemodelan klasifikasi menggunakan data pada tiap-tiap aspek yang ada. Arsitektur CNN untuk masing-masing N-gram berbeda-beda.



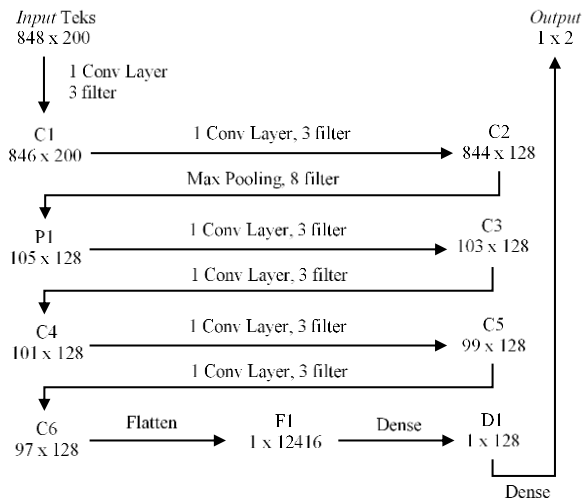
**Gambar 4.8** Arsitektur CNN *Unigram* dalam Klasifikasi Data Teks

Arsitektur CNN unigram yang ditunjukkan pada Gambar 4.8 terdiri dari 6 *convolution layer* dengan ukuran *filter* sebesar 1 sehingga *convolution window* bergeser tiap 1 kata dalam teks ulasan tersebut. Sehingga ukuran *output* dari *convolution layer* tidak berubah dari sebelumnya.



**Gambar 4.9** Arsitektur CNN *Bigram* dalam Klasifikasi Data Teks

Berdasarkan Gambar 4.9, arsitektur CNN *bigram* memiliki 6 *convolution layer* dengan ukuran *filter* 2, di mana *convolution window* bergeser tiap dua kata.



**Gambar 4.10** Arsitektur CNN *Trigram* dalam Klasifikasi Data Teks

Arsitektur CNN untuk *trigram* ditunjukkan pada Gambar 4.10 di mana tiap *convolution layer* memiliki ukuran *filter* sebesar 3. *Convolution window* akan bergeser setiap tiga kata dan ukuran *output* dari tiap *convolution layer* akan memiliki jumlah *input* dikurangi dengan dua.

Contoh perhitungan dari *convolution layer* pertama dalam model *trigram* adalah sebagai berikut.

$$\begin{aligned}
 Y_{1,1} &= -0,0018 + 0,0314X_{1,1} + (-0,0227)X_{2,1} + 0,0203X_{3,1} \\
 Y_{1,2} &= -0,0050 + (-0,0279)X_{1,2} + 0,0567X_{2,2} + (-0,0520)X_{3,2} \\
 &\quad \dots \\
 Y_{1,200} &= -0,0079 + 0,0085X_{1,200} + (-0,0392)X_{2,200} \\
 &\quad \quad \quad + (-0,0133)X_{3,200} \\
 &\quad \quad \quad \dots \\
 Y_{846,200} &= -0,0079 + 0,0145X_{846,200} + (-0,0036)X_{847,200} \\
 &\quad \quad \quad + (-0,0780)X_{848,200}
 \end{aligned}$$

#### 4.3.1 Hasil Klasifikasi Keseluruhan Data Teks

Klasifikasi sentimen menggunakan data teks dibagi menjadi tiga bagian di mana kalimat ulasan dipotong menjadi *unigram* (1 kata), *bigram* (2 kata), dan *trigram* (3 kata). Berikut adalah hasil nilai akurasi pada model unigram, bigram, dan trigram dengan *k-fold* 2 hingga 10.

**Tabel 4.14** Nilai AUC Klasifikasi Keseluruhan Data Teks

K	Unigram		Bigram		Trigram	
	Training	Testing	Training	Testing	Training	Testing
2	0,941	0,545	0,936	0,518	0,863	0,559
3	0,952	0,569	<b>0,917</b>	<b>0,602</b>	<b>0,947</b>	<b>0,628</b>
4	0,944	0,588	0,941	0,593	0,947	0,570
5	0,944	0,569	0,949	0,557	0,957	0,563
6	<b>0,944</b>	<b>0,624</b>	0,939	0,559	0,948	0,603
7	0,952	0,572	0,953	0,528	0,957	0,606
8	0,949	0,573	0,945	0,586	0,957	0,583
9	0,952	0,525	0,947	0,535	0,932	0,577
10	0,949	0,513	0,951	0,583	0,957	0,587

Berdasarkan Tabel 4.14, model terbaik dengan nilai AUC tertinggi adalah model *trigram* dengan *fold* sebesar 3. Akurasi dari

data *training* model ini sebesar 1 dan untuk akurasi data *testing* memiliki nilai sebesar 0,579. Model *trigram* dengan *fold* 3 ini akan dibandingkan dengan hasil klasifikasi MFCC. *Confusion matrix* untuk hasil klasifikasi keseluruhan data teks ditampilkan pada Tabel 4.15.

**Tabel 4.15** *Confusion Matrix* Klasifikasi Keseluruhan Data Teks

Kelas Aktual	Kelas Prediksi		Total
	Positif	Negatif	
Positif	77	8	85
Negatif	14	143	157
<b>Total</b>	<b>91</b>	<b>151</b>	<b>242</b>

Hasil klasifikasi sentimen dengan keseluruhan data teks yang ditunjukkan pada Tabel 4.15 dapat memprediksi 220 sentimen dari 242 sentimen dengan tepat. Sedangkan masih terdapat 22 potongan ulasan yang belum bisa diprediksi dengan benar.

### 4.3.2 Hasil Klasifikasi Data Teks pada Tiap Aspek

Klasifikasi selanjutnya menggunakan data teks yang dikelompokkan pada tiap aspeknya. Aspek penilaian yang digunakan sebanyak 6 aspek yaitu pemeran, direktor, jalan cerita, adegan, musik, dan secara keseluruhan. Berdasarkan AUC dari data *testing* tertinggi, model terbaik dari masing-masing aspek dipilih dan model terpilih di tiap aspek dengan nilai AUC dan akurasinya ditunjukkan pada Tabel 4.16.

**Tabel 4.16** Nilai Kinerja Klasifikasi Data Teks Tiap Aspek

Aspek	K	Fold	Model	AUC		Akurasi	
				<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>
Pemeran	8	2	Unigram	0,996	0,720	0,997	0,721
Direktor	9	1	Trigram	0,997	0,599	0,997	0,483
Jalan Cerita	2	2	Bigram	1,000	0,590	1,000	0,646
Adegan	9	6	Unigram	1,000	0,608	1,000	0,550
Musik	3	1	Trigram	1,000	0,717	1,000	0,619
Keseluruhan	3	1	Trigram	1,000	0,655	1,000	0,680
<b>Rata-Rata</b>				0,999	0,648	0,999	0,617

Rata-rata AUC data *testing* dari masing-masing model tiap aspek bernilai sebesar 0,648. Sedangkan untuk rata-rata akurasi da-

ta *testing* berdasarkan Tabrl 4.14 sebesar 0,617. Kumpulan model klasifikasi dari tiap aspek ini akan dibandingkan dengan model yang lain. Hasil klasifikasi sentimen dalam bentuk *confusion matrix* untuk tiap aspek dengan menggunakan data teks dilampirkan pada Lampiran 5.

#### 4.4 Perbandingan Evaluasi Kinerja Klasifikasi

Model-model yang terpilih dari masing-masing jenis data dibandingkan untuk menentukan model terbaik berdasarkan nilai AUC dan akurasi dari data *testing*.

**Tabel 4.17** Evaluasi Kinerja Klasifikasi Model Tiap Data

Jenis Data		AUC		Akurasi	
		<i>Training</i>	<i>Testing</i>	<i>Training</i>	<i>Testing</i>
MFCC	Semua	0,999	0,585	0,999	0,652
	Aspek	0,912	0,596	0,941	0,651
Teks	Semua	0,947	0,628	1,000	0,579
	Aspek	0,999	0,648	0,999	0,617

Model dengan nilai AUC dari data *testing* tertinggi menurut Tabel 4.17 adalah model tiap aspek dengan menggunakan data teks. Meskipun nilai akurasinya tidak tertinggi, namun nilai tersebut berada di atas 0,6. Klasifikasi menggunakan data MFCC memiliki akurasi yang lebih tinggi namun tidak dengan nilai AUC-nya, sehingga data teks lebih baik digunakan ketika data sentimen yang ada bersifat *imbalanced*.

Selanjutnya, model tersebut dijalankan kembali berdasarkan nilai K dan hasil pada *fold* model terbaik dari sebelumnya digunakan untuk model baru. Hasil pemilihan *fold* dengan nilai AUC tertinggi ditunjukkan pada Tabel 4.18.

Rata-rata akurasi dengan model data teks di tiap aspek yang ditunjukkan pada Tabel 4.18 memiliki nilai sebesar 83,7%. Dengan rata-rata nilai AUC data *testing* yang diperoleh sebesar 0,845. Aspek dengan nilai AUC dan akurasi tertinggi adalah aspek pemeran dan adegan. Sedangkan untuk aspek dengan nilai AUC dan akurasi terendah adalah aspek keseluruhan.

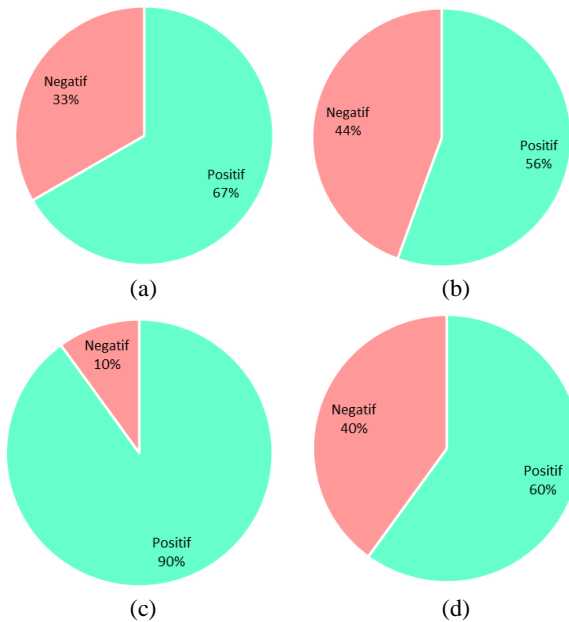


**Tabel 4.18** Evaluasi Kinerja Klasifikasi Terbaik Model Tiap Data

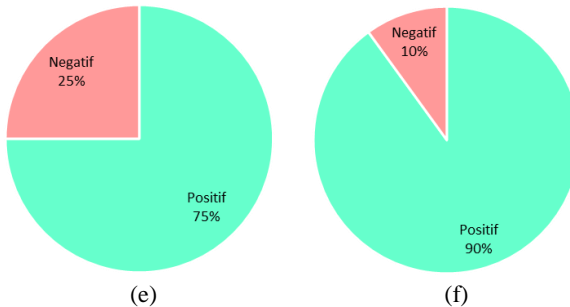
Aspek	K	Model	AUC		Akurasi	
			Training	Testing	Training	Testing
Pemeran	8	Unigram	1,000	1,000	1,000	1,000
Direktor	9	Trigram	1,000	0,750	1,000	0,750
Jalan Cerita	2	Bigram	1,000	0,668	1,000	0,708
Adegan	9	Unigram	1,000	1,000	1,000	1,000
Musik	3	Trigram	1,000	0,916	1,000	0,857
Keseluruhan	3	Trigram	1,000	0,733	1,000	0,706
<b>Rata-Rata</b>			1,000	0,845	1,000	0,837

#### 4.5 Hasil Visualisasi Persentase Nilai Ulasan Film

Model yang telah terpilih adalah klasifikasi sentimen dari data teks pada tiap aspek dengan menggunakan metode CNN. Prosentase sentimen positif dan negatif dari tiap divisualisasikan untuk melihat nilai ulasan dari setiap film.



**Gambar 4.11** Grafik Persentase Sentimen Film Bumi Manusia Aspek (a) Pemeran (b) Direktor (c) Jalan Cerita (d) Adegan (e) Musik (f) Keseluruhan



**Gambar 4.11** Grafik Persentase Sentimen Film Bumi Manusia Aspek (a) Pemeran (b) Direktur (c) Jalan Cerita (d) Adegan (e) Musik (f) Keseluruhan (Lanjutan)

Persentase sentimen positif dan negatif ditampilkan untuk menunjukkan nilai tiap aspek di suatu film. Pada Gambar 4.11, film Bumi Manusia memiliki persentase sentimen positif yang lebih besar di semua aspek. Sehingga, hasil visualisasi tersebut dapat digunakan sebagai rujukan masyarakat dalam mengetahui nilai dari film yang ada. Hasil grafik persentase untuk keempat film yang lain selengkapnya dapat dilihat di Lampiran 4. Tabel 4.19 merupakan hasil persentase sentimen positif dan negatif dari keseluruhan ulasan untuk tiap film.

**Tabel 4.19** Persentase Sentimen Positif dan Negatif Tiap Film

Nama Film	Persentase Sentimen Positif	Persentase Sentimen Negatif	Jumlah Sentimen
Bumi Manusia	74%	26%	47
Dilan 1991	50%	50%	44
Dua Garis Biru	71%	29%	52
Gundala	67%	33%	48
Makmum	55%	45%	51

Hasil persentase sentimen pada Tabel 4.19 menunjukkan dari kelima film, film Bumi Manusia memiliki persentase sentimen positif tertinggi. Sedangkan untuk film dengan persentase sentimen positif terendah adalah film Dilan 1991.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan penelitian yang telah dikerjakan didapatkan kesimpulan sebagai berikut.

1. Data audio yang telah dikelompokkan menghasilkan 242 potongan audio dengan jumlah sentimen positif sebesar 65% dan sentimen negatif sebesar 35%.
2. Pada klasifikasi sentimen menggunakan metode CNN dengan data MFCC, model terbaik adalah model klasifikasi sentimen berdasarkan aspeknya dengan AUC sebesar 0,596 dan dapat memprediksi dengan tepat sebanyak 224 potongan ulasan dengan akurasi sebesar 92,6%.
3. Arsitektur CNN untuk data teks dibagi menjadi tiga yaitu *uni-gram*, *bigram*, dan *trigram*. Model terbaik untuk klasifikasi sentimen dengan data teks hasil konversi *Speech-to-Text* adalah model berdasarkan aspek dari penilaian film dengan AUC sebesar 0,648. Model klasifikasi per aspek tersebut dapat memprediksi dengan benar sebanyak 219 potongan ulasan dengan akurasi sebesar 90,5%.
4. Dari keempat model klasifikasi sentimen di tiap data yang ada dibandingkan dan terpilih model dari data teks di tiap aspek. Setelah memilih model terbaik di tiap *fold* didapatkan rata-rata nilai AUC dan akurasi dari data *testing* berturut-turut sebesar 0,845 dan 0,837. Akurasi model awal yang terpilih memiliki akurasi dari hasil klasifikasi data *testing* di tiap aspek pemeran, direktor, jalan cerita, adegan, musik, dan keseluruhan berturut-turut sebesar 100%, 75%, 70,8%, 100%, 85,7%, dan 70,6%.
5. Grafik persentase dibuat berdasarkan hasil klasifikasi sentimen model terpilih dan diperoleh persentase sentimen positif terbesar adalah film Bumi Manusia dengan nilai 74%. Sedangkan persentase sentimen positif terendah adalah film Dilan 1991 sebesar 50%.

## 5.2 Saran

Saran yang dapat dilakukan di penelitian selanjutnya adalah diharapkan dapat memilih model arsitektur CNN yang dapat mengklasifikasikan sentimen dengan lebih baik karena jenis dan jumlah *layer* yang digunakan dapat mempengaruhi kebaikan model. Selain itu, jumlah data dapat ditambah agar bisa menangani *imbalance* serta agar data tidak *overfitting*. Metode untuk klasifikasi sentimen dapat ditambahkan sehingga tidak hanya perbandingan jenis data saja, namun juga terdapat perbandingan metode untuk menentukan metode klasifikasi manakah yang lebih baik.

## DAFTAR PUSTAKA

- Abburi, H. (2017). *Audio and Text based Multimodal Sentiment Analysis using Features Extracted from Selective Regions and Deep Neural Networks*. Hyderabad.
- Al Amrani, Y., Lazaar, M., & El Kadiri, K. E. (2018). Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis. *Procedia Computer Science*, 511-520.
- Alessa, A., & Faezipour, M. (2018). Flu Tweet Classification Using Sentiment Features and TF-IDF Weighting. *Machine Learning and Data Mining in Pattern Recognition*. New York: Springer.
- Alim, S. A., & Rashid, N. K. (2018). Some Commonly Used Speech Feature Extraction Algorithms. *From Natural to Artificial Intelligence*.
- Amos, D. (2018). *The Ultimate Guide To Speech Recognition With Python*. Diambil kembali dari Real Python: <https://realpython.com/python-speech-recognition/#appendix-recognizing-speech-in-languages-other-than-english>
- Anand, D., & Naorem, D. (2016). Semi-supervised Aspect Based Sentiment Analysis for Movies using Review Filtering. *Procedia Computer Science*, 86-93.
- Anandarajan, M., Hill, C., & Nolan, T. (2019). *Practical Text Analytics: Maximizing the Value of Text Data*. Cham: Springer.
- Baesens, B., Vlasselaer, V. V., & Verbeke, W. (2015). *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques*. Hoboken: John Wiley & Sons, Inc.
- Bardou, D., Zhang, K., & Ahmad, S. M. (2018). Lung Sounds Classification using Convolutional Neural Networks`. *Artificial Intelligence in Medicine*.

- Best Essays. (2019). *Glossary - Movie Review*. Diambil kembali dari bestessays: [https://www.bestessays.com/glossary/movie\\_review.php](https://www.bestessays.com/glossary/movie_review.php)
- Bhattacharya, J. (2006). *Technology in Government*. New Delhi: GIFT Publishing.
- Brownlee, J. (2019). *Deep Learning fo Natural Language Processing: Develop Deep Learning Models for Natural Language in Python*. Jason Brownlee.
- Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling Technique. *Advances in Knowledge Discovery and Data Mining* (hal. 475-482). Bangkok: Springer.
- Charles, D. S., Carmona, J., Mulero, V. M., & Sole, M. (2017). Reducing Event Variability in Logs by Clustering of Word Embeddings. *Business Process Management Workshops* (hal. 191-203). Barcelona: Springer.
- Christensson, P. (2014). *Speech Recognition Definition*. Diambil kembali dari Technical Terms: <https://techterms.com>
- Cichosz, P. (2015). *Data Mining Algorithms: Explained Using R*. Chichester: John Wiley & Sons, Ltd.
- Collins. (2019). Diambil kembali dari Collins Dictionary: <https://www.collinsdictionary.com/dictionary/english/film>
- Collobert, R., & Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. *Proceedings of the 25th International Conference on Machine Learning* (hal. 160–167). New York: ACM.
- Cui, H., Lin, Y., & Utsuro, T. (2018). Sentiment Analysis of Tweets by CNN utilizing Tweets with Emoji as Training Data.
- Datta, A. K., Solanki, S. S., Sengupta, R., Chakraborty, S., Mahto, K., & Patranabis, A. (2017). *Signal Analysis of Hindustani Classical Music*. Singapore: Springer.

- Fayek, H. (2016, April 21). *Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between*. Diambil kembali dari <https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html>
- Ferreira, A., & Giraldo, G. (2017). Convolutional Neural Network Approaches to Granite Tiles Classification. *Expert Systems With Applications*, 1-11.
- Gautam, C., & Ravi, V. (2017). Auto Associative Extreme Learning Machine Based Hybrids for Data Imputation. Dalam A. Kumar, M. K. Dash, S. K. Trivedi, & T. K. Panda, *Handbook of Research on Intelligent Techniques and Modeling Applications in Marketing Analytics*. Hershey: IGI Global.
- Gurusamy, V., & Kannan, S. (2014). Preprocessing Techniques for Text Mining.
- Huang, Y., Tan, S., Li, B., & Huang, J. (2018). VPCID - A VoIP Phone Call Identification Database. *Digital Forensics and Watermarking* (hal. 307-321). Jeju Island: Springer.
- JJ. (2016, March 23). *MAE and RMSE — Which Metric is Better?* Diambil kembali dari Medium: <https://medium.com/human-in-a-machine-world/>
- Kingma, D. P., & Ba, J. L. (2015). Adam: A Method for Stochastic Optimization. *ICLR 2015*.
- Kinnunen, T., & Li, H. (2010). An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, (hal. 12-40).
- Kuznetsov, I. (2019, May 9). *Metrics for Imbalanced Classification*. Diambil kembali dari Towards Data Science: <https://towardsdatascience.com/metrics-for-imbalanced-classification-41c71549bbb5>
- Ledeneva, Y. (2008). Effect of Preprocessing on Extractive Summarization with Maximal Frequent Sequences. *MICAI 2008: Advances in Artificial Intelligence* (hal. 123-132). Heidelberg: Springer.

- Li, H., Xu, B., Wang, N., & Liu, J. (2016). Deep Convolutional Neural Networks for Classifying Body Constitution. *Artificial Neural Network and Machine Learning* (hal. 128-135). Barcelona: Springer.
- Lim, W., Jang, D., & Taejin, L. (2016). Speech Emotion Recognition using Convolutional and Recurrent Neural Networks. *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. Jeju: IEEE.
- Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*.
- Liu, B. (2015). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. New York: Cambridge University Press.
- Lobo, J. M., Jiménez-Valverde, A., & Real, R. (2008). AUC: a misleading measure of the performance of predictive distribution models.
- Lutfi, A. A., Permanasari, A. E., & Fauziati, S. (2018). Sentiment Analysis in the Sales Review of Indonesian Marketplace by Utilizing Support Vector Machine. *Journal of Information Systems Engineering and Business Intelligence*, 57-64.
- Maghilnan, S., & Rajesh, M. (2017). Sentiment Analysis on Speaker Specific Speech Data. *International Conference on Intelligent Computing and Control*.
- Maheshwari, A. (2018, July 17). *Report on Text Classification using CNN, RNN & HAN*. Diambil kembali dari Medium: <https://medium.com/jatana/>
- Masada, T., Fukagawa, D., Takasu, A., Shibata, Y., & Oguri, K. (2010). Modeling Topical Trends over Continuous Time with Priors. *Advances in Neural Network - ISNN 2010* (hal. 302-311). Shanghai: Springer.
- Mikros, G. K., & Macutek, J. (2015). *Sequences in Language and Text*. Berlin: De Gruyter Mouton.
- Muhury, S., Neogi, G., Debnath, P., & Dastidar, J. G. (2016). Design of a Voice-Based System by Recognizing Speech



- using MFCC. *Computational Science and Engineering* (hal. 77-80). Kolkata: CRC Press.
- Mushtaq, R. (2011, August 17). *Augmented Dickey Fuller Test*. Diambil kembali dari SSRN: <https://ssrn.com/abstract=1911068>
- Nugroho, K. S. (2018, June). *Dasar Text Preprocessing dengan Python*. Diambil kembali dari Medium: <https://medium.com/@ksnugroho/dasar-text-preprocessing-dengan-python-a4fa52608ffe>
- Olson, D. L., & Delen, D. (2008). *Advanced Data Mining Techniques*. Heidelberg: Springer.
- Ostertagová, E., & Ostertag, O. (2014). THE SIMPLE EXPONENTIAL SMOOTHING MODEL. *Modelling of Mechanical and Mechatronic Systems 2011*. Košice: Faculty of Mechanical Engineering, Technical University of Košice.
- Otoritas Jasa Keuangan. (2015). *Buku Saku Otoritas Jasa Keuangan* (2nd ed.). Jakarta.
- Park, J. J., Loia, V., Yi, G., & Sung, Y. (2018). *Advances in Computer Science and Ubiquitous Computing*. Singapore: Springer.
- PT Telkom Indonesia (Persero) Tbk. (2017). *Telkom*. Diambil kembali dari TENTANG TELKOMGROUP: [https://www.telkom.co.id/servlet/tk/about/id\\_ID/stocklanning/profil-dan-riwayat-singkat.html](https://www.telkom.co.id/servlet/tk/about/id_ID/stocklanning/profil-dan-riwayat-singkat.html)
- Saha, S. (2018, December 16). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Diambil kembali dari Towards Data Science: <https://towardsdatascience.com/>
- Saminan. (1995). *Kemampuan Memahami Grafik dalam Fisika*. Bandung: tidak diterbitkan.
- Sandy, K. f. (2017). *Sindonews.com*. Dipetik November 24, 2018, dari <https://ekbis.sindonews.com/read/1188315/178/ojk-terus-tingkatkan-peran-industri-keuangan-non-bank-1489491425>

- Santos, M. S., Soares, J. P., Abreu, P. H., Araujo, H., & Santos, J. (2018). Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches. *IEEE Computational Intelligence Magazine*.
- Sewak, M., Karim, R., & Pujari, P. (2018). *Practical Convolutional Neural Network*. Birmingham: Packt Publishing.
- Shumway, R., & Stoffer, D. (2006). *Time Series Analysis and Its Application with R Examples*. Springer.
- Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., & Hernandez, L. C. (2012). Syntactic Dependency-based N-grams as Classification Features.
- Syahid, B. (2019, July 05). *Pengertian Teks Ulasan, Contoh, Ciri, Tujuan, Struktur, dan Kaidahnya*. Diambil kembali dari Guru Pendidikan: <https://www.gurupendidikan.co.id/teks-ulasan/>
- Thet, T. T., Na, J.-C., & Khoo, C. S. (2010). Aspect-Based Sentiment Analysis of Movie Reviews on Discussion Boards. *Journal of Information Science*, 1-26.
- Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of Sentiment Reviews using N-gram Machine Learning Approach. *Expert Systems With Applications*, 57, 117-126.
- Tuffery, S. (2011). *Data Mining and Statistics for Decision Making*. Chichester: John Wiley & Sons.
- Unuth, N. (2019, April 23). *What Is Speech Recognition?* Diambil kembali dari Lifewire: <https://www.lifewire.com/>
- Vandeput, N. (2019, January 25). *Croston Forecast Model for Intermittent Demand*. Diambil kembali dari Analytics Vidhya: <https://medium.com/analytics-vidhya/>
- Walpole, R. E. (2011). *Probability and Statistics for Engineering and Scientist*. Boston: Prentice Hall, Pearson.
- Webwise. (2019). *Explained: What is Youtube?* Diambil kembali dari Webwise: <https://www.webwise.ie/parents/what-is-youtube/>
- Wei, W. W. (2006). *Time Series Analysis: Univariate and Multivariate Methods*. Pearson Education, Inc.

- Wikipedia. (t.thn.). *Telkom Indonesia - Wikipedia Bahasa Indonesia, Ensiklopedia Bebas*. Dipetik 17, 2019, dari Wikipedia:  
[https://id.wikipedia.org/wiki/Telkom\\_Indonesia](https://id.wikipedia.org/wiki/Telkom_Indonesia)
- Wollmer, M., Weninger, F., Knaup, T., Schuller, B., Sun, C., Sagae, K., & Morency, L.-P. (2013). Youtube Movie Reviews: Sentiment Analysis in an Audio-Visual Context. *IEEE Intelligent Systems*, 46-53.
- Yayasan Pendidikan Telkom. (2017, July 7). *Kemeriahan HUT Telkom ke-52*. Diambil kembali dari Yayasan Pendidikan Telkom: <http://ypt.or.id/kemeriahan-hut-telkom-ke-52/>
- Yin, Y., Zhang, Y., Liu, X., Zhang, Y., Xing, C., & Chen, H. (2014). HealthQA: A Chinese QA Summary System for Smart Health. *Smart Health* (hal. 51-62). Beijing: Springer.
- YouTube. (2019). Diambil kembali dari YouTube: <https://www.youtube.com/about/>
- Zhou, W., Wang, H., Sun, H., & Sun, T. (2019). A Domain-Adapting Word Representation Method for Word Clustering. *Recent Trends in Intelligent Computing, Communication and Devices* (hal. 141-148). Singapore: Springer.

*(Halaman ini sengaja dikosongkan)*

## LAMPIRAN

### Lampiran 1. Daftar *Channel* dan *Link* Video Ulasan

<b>Judul Film</b>	<b>Nama Channel</b>	<b>Link Video</b>
Bumi Manusia	#sumatranbigfoot	<a href="https://www.youtube.com/watch?v=lymbzI_YSHU">https://www.youtube.com/watch?v=lymbzI_YSHU</a>
	abibayu	<a href="https://www.youtube.com/watch?v=4or-Y-KC6_I">https://www.youtube.com/watch?v=4or-Y-KC6_I</a>
	Cine Crib	<a href="https://www.youtube.com/watch?v=G6kRRFyERvM">https://www.youtube.com/watch?v=G6kRRFyERvM</a>
	Kawan Review	<a href="https://www.youtube.com/watch?v=u6a0MQUIP3c">https://www.youtube.com/watch?v=u6a0MQUIP3c</a>
	Kepin Helmy	<a href="https://www.youtube.com/watch?v=2Br6kWHNtSA">https://www.youtube.com/watch?v=2Br6kWHNtSA</a>
	KINCIR - Cinema Club	<a href="https://www.youtube.com/watch?v=dBo44mDSmRo">https://www.youtube.com/watch?v=dBo44mDSmRo</a>
	Red n Blue Cafe	<a href="https://www.youtube.com/watch?v=vBHju25HJo8">https://www.youtube.com/watch?v=vBHju25HJo8</a>
	The Lazy Cinema	<a href="https://www.youtube.com/watch?v=Fn7GjtogJbk">https://www.youtube.com/watch?v=Fn7GjtogJbk</a>
	vloggerpedia	<a href="https://www.youtube.com/watch?v=vvtQRIH2dC4">https://www.youtube.com/watch?v=vvtQRIH2dC4</a>
Wandy Oentu	<a href="https://www.youtube.com/watch?v=EgasDCGStko">https://www.youtube.com/watch?v=EgasDCGStko</a>	
Dilan 1991	#sumatranbigfoot	<a href="https://www.youtube.com/watch?v=590Oz1GB9NU">https://www.youtube.com/watch?v=590Oz1GB9NU</a>
	Cine Crib	<a href="https://www.youtube.com/watch?v=jwPmGeBse2Y">https://www.youtube.com/watch?v=jwPmGeBse2Y</a>
	Ibnuism - Daily Students	<a href="https://www.youtube.com/watch?v=p0I7IL-IFRg">https://www.youtube.com/watch?v=p0I7IL-IFRg</a>
	Kepin Helmy	<a href="https://www.youtube.com/watch?v=qZYNzaOodr0">https://www.youtube.com/watch?v=qZYNzaOodr0</a>
	KINCIR - Cinema Club	<a href="https://www.youtube.com/watch?v=shbEO7i7AHs">https://www.youtube.com/watch?v=shbEO7i7AHs</a>
	ML Studios	<a href="https://www.youtube.com/watch?v=GShb6dBjer4">https://www.youtube.com/watch?v=GShb6dBjer4</a>
	Narasi Entertainment	<a href="https://www.youtube.com/watch?v=ANnNVZ1x4B4">https://www.youtube.com/watch?v=ANnNVZ1x4B4</a>
	Pure Pixels	<a href="https://www.youtube.com/watch?v=jpDjZ4IZAgY">https://www.youtube.com/watch?v=jpDjZ4IZAgY</a>
	Putu Bahagiana	<a href="https://www.youtube.com/watch?v=YySIgQ15egc">https://www.youtube.com/watch?v=YySIgQ15egc</a>
vloggerpedia	<a href="https://www.youtube.com/watch?v=D-TrWUwzLSA">https://www.youtube.com/watch?v=D-TrWUwzLSA</a>	
Dua Garis Biru	#sumatranbigfoot	<a href="https://www.youtube.com/watch?v=T2sQzHFKICg">https://www.youtube.com/watch?v=T2sQzHFKICg</a>
	Cine Crib	<a href="https://www.youtube.com/watch?v=IQ9Gxwov288">https://www.youtube.com/watch?v=IQ9Gxwov288</a>
	Grid ID	<a href="https://www.youtube.com/watch?v=SXhi3yDcCOU">https://www.youtube.com/watch?v=SXhi3yDcCOU</a>
	Kawan Review	<a href="https://www.youtube.com/watch?v=wHjiU76VT0A">https://www.youtube.com/watch?v=wHjiU76VT0A</a>
	Kepin Helmy	<a href="https://www.youtube.com/watch?v=AZo6QnDt2fw">https://www.youtube.com/watch?v=AZo6QnDt2fw</a>
	KINCIR - Cinema Club	<a href="https://www.youtube.com/watch?v=ONd3-FC8Lbg">https://www.youtube.com/watch?v=ONd3-FC8Lbg</a>
	LIANT	<a href="https://www.youtube.com/watch?v=w_1kcm0ebus">https://www.youtube.com/watch?v=w_1kcm0ebus</a>

**Lampiran 1. Daftar Channel dan Link Video Ulasan (Lanjutan)**

<b>Judul Film</b>	<b>Nama Channel</b>	<b>Link Video</b>
Dua Garis Biru	Narasi Entertainment	<a href="https://www.youtube.com/watch?v=QRUM1ge_9y0">https://www.youtube.com/watch?v=QRUM1ge_9y0</a>
	Red n Blue Cafe	<a href="https://www.youtube.com/watch?v=Bes8Zv8Y7pY">https://www.youtube.com/watch?v=Bes8Zv8Y7pY</a>
	Wandy Oentu	<a href="https://www.youtube.com/watch?v=SuRbJvDMRiw">https://www.youtube.com/watch?v=SuRbJvDMRiw</a>
Gundala	#sumatranbigfoot	<a href="https://www.youtube.com/watch?v=qbb5M8ba058">https://www.youtube.com/watch?v=qbb5M8ba058</a>
	AIIZWEL	<a href="https://www.youtube.com/watch?v=Mnqj7TUeD8I">https://www.youtube.com/watch?v=Mnqj7TUeD8I</a>
	Bang Ray	<a href="https://www.youtube.com/watch?v=_xDY72Uni4w">https://www.youtube.com/watch?v=_xDY72Uni4w</a>
	Cine Crib	<a href="https://www.youtube.com/watch?v=U2SNFpj5BAo">https://www.youtube.com/watch?v=U2SNFpj5BAo</a>
	Cleansound Studio	<a href="https://www.youtube.com/watch?v=bAO9YJ_qzLU">https://www.youtube.com/watch?v=bAO9YJ_qzLU</a>
	Kepin Helmy	<a href="https://www.youtube.com/watch?v=R2R28fCNGfQ">https://www.youtube.com/watch?v=R2R28fCNGfQ</a>
	Pure Pixels	<a href="https://www.youtube.com/watch?v=HnIz9dL-2fA">https://www.youtube.com/watch?v=HnIz9dL-2fA</a>
	RH Entertainment	<a href="https://www.youtube.com/watch?v=Lu23eV-Xtn8">https://www.youtube.com/watch?v=Lu23eV-Xtn8</a>
	Wandy Oentu	<a href="https://www.youtube.com/watch?v=L-dZfwNM-G4">https://www.youtube.com/watch?v=L-dZfwNM-G4</a>
	Yofie Setiawan	<a href="https://www.youtube.com/watch?v=fsyq0Fs6aiI">https://www.youtube.com/watch?v=fsyq0Fs6aiI</a>
	Makmum	#sumatranbigfoot
Adhe Tora TV		<a href="https://www.youtube.com/watch?v=Ha9uL9ouS-4">https://www.youtube.com/watch?v=Ha9uL9ouS-4</a>
Cine Crib		<a href="https://www.youtube.com/watch?v=cppx8TgCycE">https://www.youtube.com/watch?v=cppx8TgCycE</a>
Demem Nonton		<a href="https://www.youtube.com/watch?v=W_AT_Bp8t1Y">https://www.youtube.com/watch?v=W_AT_Bp8t1Y</a>
Erick Antonieta		<a href="https://www.youtube.com/watch?v=tHfd5JimF7Y">https://www.youtube.com/watch?v=tHfd5JimF7Y</a>
Kawan Review		<a href="https://www.youtube.com/watch?v=eMnpR61DX6c">https://www.youtube.com/watch?v=eMnpR61DX6c</a>
Kepin Helmy		<a href="https://www.youtube.com/watch?v=epmzNK-MIwI">https://www.youtube.com/watch?v=epmzNK-MIwI</a>
Pure Pixels		<a href="https://www.youtube.com/watch?v=nbetzBav_10">https://www.youtube.com/watch?v=nbetzBav_10</a>
Red n Blue Cafe		<a href="https://www.youtube.com/watch?v=zUY8QweV42Y">https://www.youtube.com/watch?v=zUY8QweV42Y</a>
The Lazy Cinema	<a href="https://www.youtube.com/watch?v=am3IEFpVG8Y">https://www.youtube.com/watch?v=am3IEFpVG8Y</a>	

**Lampiran 2. Data Hasil Ekstraksi MFCC**

<b>MFCC</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>...</b>	<b>9</b>	<b>10</b>	<b>11</b>
1	0.000	0.000	0.000	...	0.000	0.000	0.000
	0.000	0.000	0.000	...	0.000	0.000	0.000
	0.000	0.000	0.000	...	0.000	0.000	0.000
	...	...	...	...	...	...	...
	5.340	13.946	-13.281	...	7.412	11.821	5.333

**Lampiran 2. Data Hasil Ekstraksi MFCC (Lanjutan)**

MFCC	0	1	2	...	9	10	11
	54.826	-47.946	-17.143	...	-18.318	-10.207	0.829
	59.348	-37.185	-4.196	...	0.913	0.313	-0.994
	...	...	...	...	...	...	...
	0.000	0.000	0.000	...	0.000	0.000	0.000
	0.000	0.000	0.000	...	0.000	0.000	0.000
	0.000	0.000	0.000	...	0.000	0.000	0.000
2	0.000	0.000	0.000	...	0.000	0.000	0.000
	0.000	0.000	0.000	...	0.000	0.000	0.000
	0.000	0.000	0.000	...	0.000	0.000	0.000
	...	...	...	...	...	...	...
	2.717	-10.679	-26.277	...	-16.133	-9.589	4.404
	74.476	-33.993	29.378	...	-26.655	-0.668	1.763
	77.981	-59.631	37.996	...	-45.810	-11.252	-14.369
	...	...	...	...	...	...	...
	0.000	0.000	0.000	...	0.000	0.000	0.000
	0.000	0.000	0.000	...	0.000	0.000	0.000
0.000	0.000	0.000	...	0.000	0.000	0.000	
...	...						
242	0.000	0.000	0.000	...	0.000	0.000	0.000
	0.000	0.000	0.000	...	0.000	0.000	0.000
	0.000	0.000	0.000	...	0.000	0.000	0.000
	...	...	...	...	...	...	...
	-75.163	-15.513	-27.360	...	11.348	5.359	1.200
	-82.510	-27.189	14.139	...	4.547	11.288	2.422
	-56.288	-15.157	17.150	...	3.753	9.535	2.665
	...	...	...	...	...	...	...
	0.000	0.000	0.000	...	0.000	0.000	0.000
	0.000	0.000	0.000	...	0.000	0.000	0.000
0.000	0.000	0.000	...	0.000	0.000	0.000	

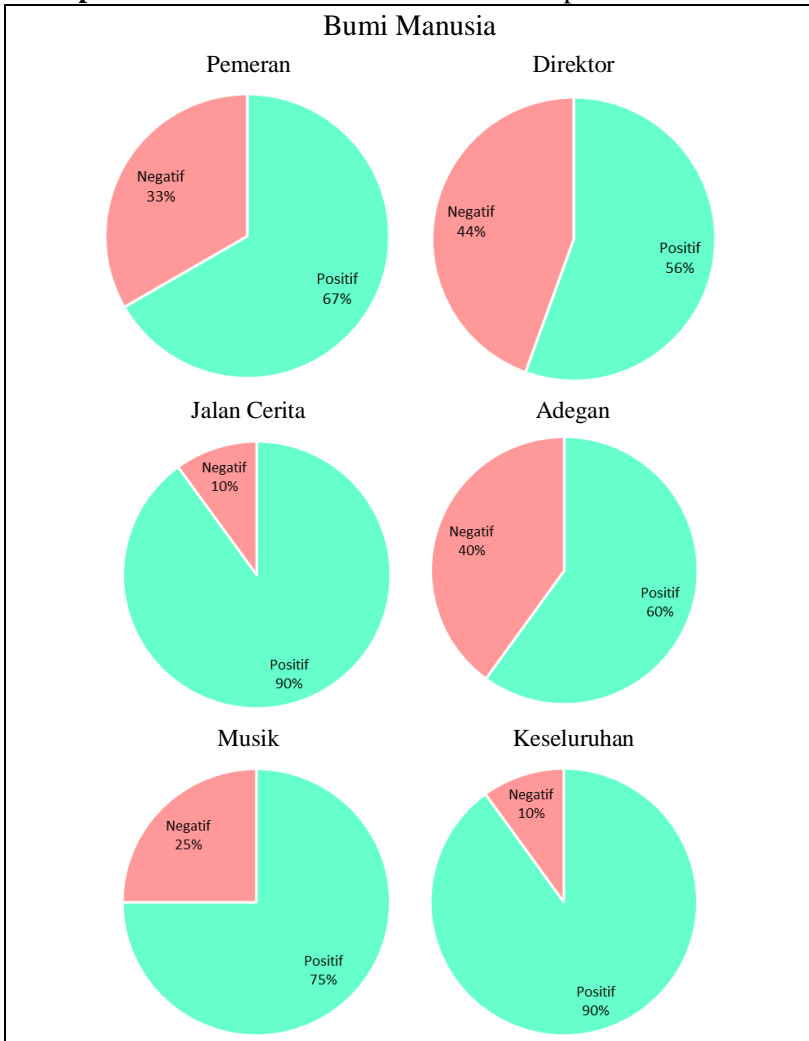
### Lampiran 3. Data Teks

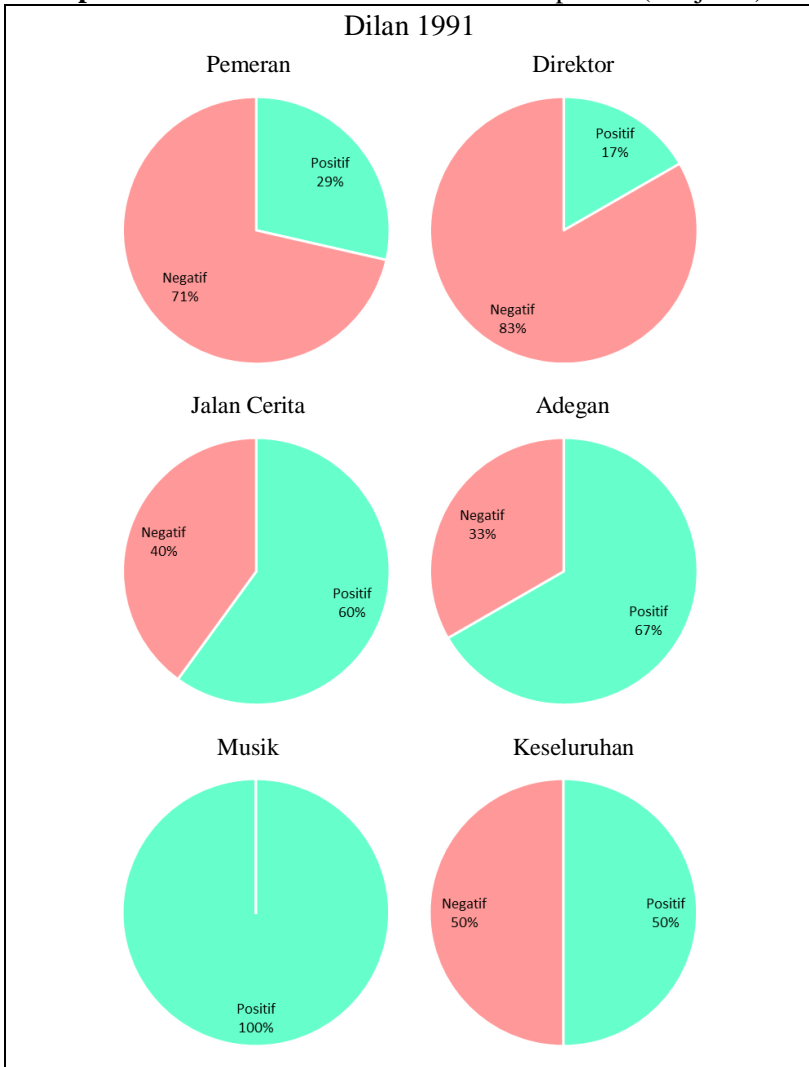
No	Ulasan	Sentimen
1	<p>gubah musik andhika triyadi sering kali tambal beberapa poin minor cukup elegan seringkali buat departemen akting seringkali tanya banyak orang sejak tunjuk tahun lalu mampu iqbaal dhiafakhri ramadan sukses besar peran minke meski nilai sangat relatif pribadi berangkat tonton film novel iqbal jelas beri performa cukup kecewa level cukup vokal lugas tampil akting luap emosi lebih dialog kasual daya kurang jiwa gerak adegan main mimik lebih laku inner dialog sayang seringkali buat sedikit puruk jomplang sensen adu akting aktor utama mawar eva sains febrianti ayu laksmi bahan jerome kurnia panjang menang semua orang iqbal ramadan mulai pikir sekolah akting tuju terima kasih teman subscribe nyala tombol notifikasi</p>	0
2	<p>ngeliat iqbal benar apa sih gue kayak ngelihat aktor gitu gitu kaya cuman satu nyai ontosoroh kompleks banget gimana sih hidup karakter perempuan gambar bagai asal</p>	0
3	<p>karakter karakter baik ngobrol adegan tudepoin datang hadap ngobrol kan adegan selesai efisien benar bagus apa buat adegan hindu justru letak apa benar cerita we flashback imut banget ngobrol buka pintu nyata cerita film luar biasa dinamis karakter sangat struktural naik turun tunjuk buah sabar buah dominan sesuai zaman film sesuai bagaimana karakter milik trayek apa nih bahasa belanda kapan bahasa jawa gue ngomong sama orang keluarin apa karakter dimensi ad review bumi manusia share jam kalian tonton peran rasa jangan lupa share video temen lu emang suka nonton film hanung bramantyo suka ngikutin jalan karir akting iqbal Ramadhan</p>	1
4	<p>manusia cerita peran iqbaal ramadhan analisis peran mawar eva hadap angkuh hukum kolonial coba hancur hubung latar belakang nike analisis beda minke pemuda pribumi jawa toto sedang ali orang gadis blaster indo belanda anak nyai ontosoroh peran saina febrianti engkau juang mati lawan ketidakadilan terus banyak tolak bumi manusia akhir pilih iqbaal ramadhan peran bagai nikah per utama film benar aku pribadi salah pilih iqbaal ramadhan peran karakter gin nih kelas baik sebentar pakai emosi memang biasa tampil iqbal bagai anggota boyband cilik cjr tiba percaya peran bagai fakta iqbal hasil bawa karakter ikonik novel best seller pidi baiq sebut alter ego orang iqbal sekarang posisi sama sekian kali ragam respon gak sedikit skeptis iqbal peran bagai bingkai bumi manusia tuju iqbal ramadhan hasil bikin aku kan visualisasi siswa hbs cerdas wibawa gambar jelas sosok iqbal ramadhan hasil toreh nama baris bakal calon aktor indonesia patut hitung potensi dunia film tanah air mudah bagus bikin aku kagum panjang film sosok nyai ontosoroh gila sisain febrina lihat sepenuh hati taruh asa sama karakter duit aja soal punya pesan sendiri nominasi bagai aktris baik tahun jangan tinggal peran alis peran mawar eva polos natural tampil bagai analis patut hitung meski kam karakter lebih lambat aktor udah aku sebutin barusan</p>	1

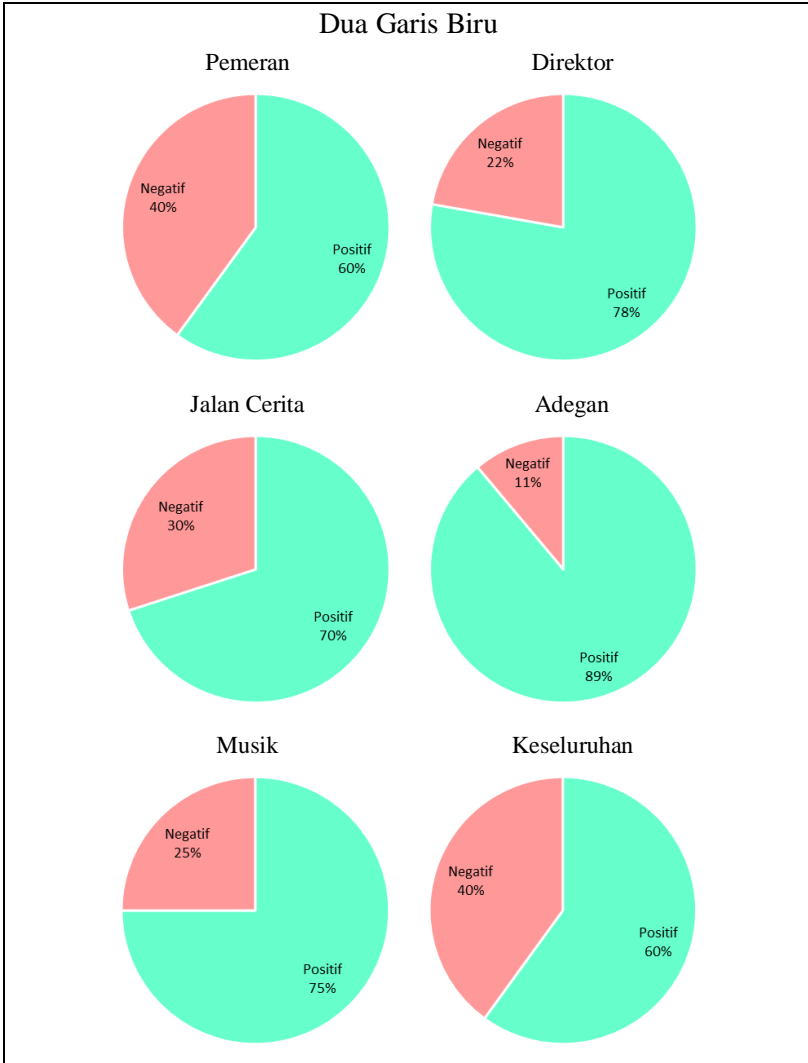


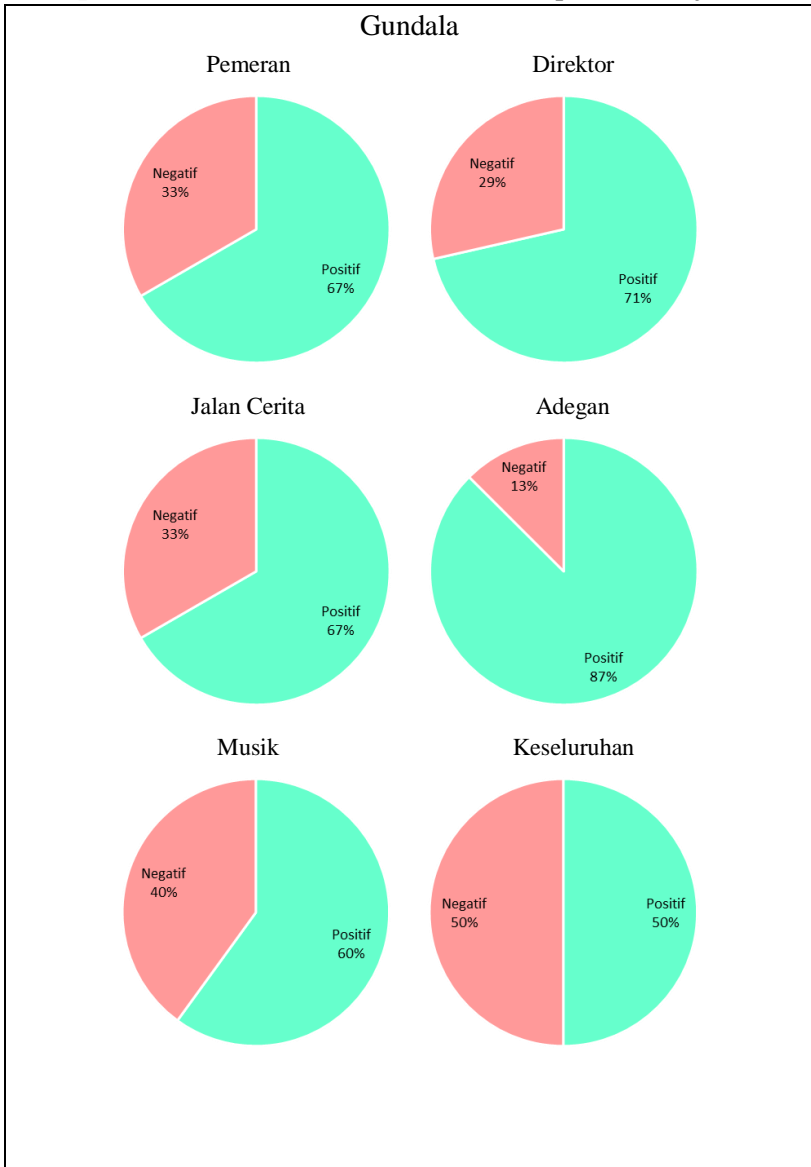
**Lampiran 3. Data Teks (Lanjutan)**

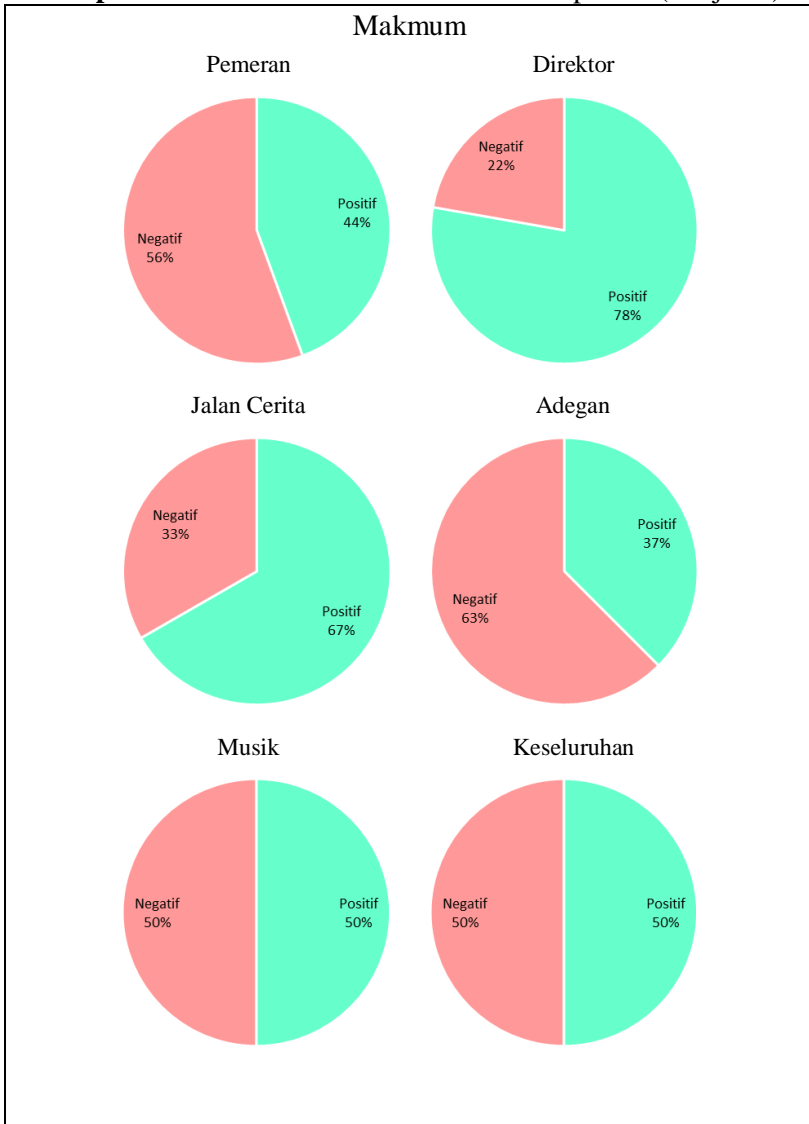
No	Ulasan	Sentimen
5	berani banget punya cerita bagus sih sebenarnya gue tahu film peranin sama iqbal kira film bakal lulus sensor rating nyata enggak dong nyata film walaupun cuma adegan singkat detik tetep aja pakai baju sesuai karakter kayak makan orang eropa pakai baju orang eropa misal pesta makan siang gitu arti pakai baju eropa pas neo mewah gitu lima gue suka film soal akting semua sini mungkin awal gak bakal lebay segala macam nyata enggak gue bagus udah usaha maksimal mungkin sih gue tahu banget emang kalau misal kalian bukan orang suka nonton drama romance tiba harap film peperangan wah gitu kalian salah emang bis film drama romance wajar aja ketik	1
...	...	
242	film horor indo buah segmen cukup cari film hari mau mereview film makmum buah film cukup parah banget kalo gue lihat cerita kalau kalau lihat film sebenarnya sih banding film horor sama kreator kalau tonton tuh lu kayak horor zaman dulu warna mirip sama kalau gue bilang sih sebenarnya kalo lu punya punya ekspektasi ken nonton film horor indonesia banget bukan film joko anwar film joko anwar memang kalau tuh film horor kalau lu bilang film memang indonesia gin lah film horor digangguin pesan lu bayangin nih film pendek menit tarik satu tengah jam gitu persis kayak gitu kalau lihat tuh bener bukan ngomongin kulit film indonesia dulu gimana kalau sekarang kan lu prestasi kalau udah punya tuju solid lama cuman ngelihat baru bagai orang bakau satu asrama digangguin ganggu tuh kayak film spongebob gitu percaya masa aneh sih cuman untung aktor bagus emang kamu tahu kalau coba diri gitu inti film cukup buat nyalain tv takut kalau nonton cukup akhir aku lihat film indo cukup film horor rem ngagetin doang cukup tebak murah tuju ga kalau film memang kalau segi film kalau horor indonesia kali banding malam oke lah mungkin md film bagus jangan dedek paten lah dedek seksi kelapa gading kelapa gading tuh kalau pilih salah satu potensial nonton hati harta natrium gitu kan tanda film enak titi kamal film	0

**Lampiran 4. Grafik Persentase Sentimen Tiap Film**

**Lampiran 4. Grafik Persentase Sentimen Tiap Film (Lanjutan)**

**Lampiran 4. Grafik Persentase Sentimen Tiap Film (Lanjutan)**

**Lampiran 4. Grafik Persentase Sentimen Tiap Film (Lanjutan)**

**Lampiran 4. Grafik Persentase Sentimen Tiap Film (Lanjutan)**

**Lampiran 5.** Hasil *Confusion Matrix***CONFUSION MATRIKS** KLASIFIKASI SENTIMEN TIAP ASPEK  
Aspek 1 (Pemeran)

<b>Kelas Aktual</b>	Data MFCC		<b>Kelas Aktual</b>	Data Teks		<b>Total</b>
	<b>Kelas Prediksi</b> Positif	Negatif		<b>Kelas Prediksi</b> Positif	Negatif	
Positif	17	1	Positif	18	0	<b>18</b>
Negatif	0	26	Negatif	2	24	<b>26</b>
<b>Total</b>	<b>17</b>	<b>27</b>	<b>Total</b>	<b>20</b>	<b>24</b>	<b>44</b>

## Aspek 2 (Direktor)

<b>Kelas Aktual</b>	Data MFCC		<b>Kelas Aktual</b>	Data Teks		<b>Total</b>
	<b>Kelas Prediksi</b> Positif	Negatif		<b>Kelas Prediksi</b> Positif	Negatif	
Positif	17	0	Positif	14	3	<b>17</b>
Negatif	1	22	Negatif	1	22	<b>23</b>
<b>Total</b>	<b>18</b>	<b>22</b>	<b>Total</b>	<b>15</b>	<b>25</b>	<b>40</b>

## Aspek 3 (Jalan Cerita)

<b>Kelas Aktual</b>	Data MFCC		<b>Kelas Aktual</b>	Data Teks		<b>Total</b>
	<b>Kelas Prediksi</b> Positif	Negatif		<b>Kelas Prediksi</b> Positif	Negatif	
Positif	14	2	Positif	12	4	<b>16</b>
Negatif	4	28	Negatif	2	30	<b>32</b>
<b>Total</b>	<b>18</b>	<b>30</b>	<b>Total</b>	<b>14</b>	<b>34</b>	<b>48</b>

## Aspek 4 (Adegan)

<b>Kelas Aktual</b>	Data MFCC		<b>Kelas Aktual</b>	Data Teks		<b>Total</b>
	<b>Kelas Prediksi</b> Positif	Negatif		<b>Kelas Prediksi</b> Positif	Negatif	
Positif	13	0	Positif	12	1	<b>13</b>
Negatif	1	25	Negatif	0	26	<b>26</b>
<b>Total</b>	<b>14</b>	<b>25</b>	<b>Total</b>	<b>12</b>	<b>27</b>	<b>39</b>

## Aspek 5 (Musik)

<b>Kelas Aktual</b>	Data MFCC		<b>Kelas Aktual</b>	Data Teks		<b>Total</b>
	<b>Kelas Prediksi</b> Positif	Negatif		<b>Kelas Prediksi</b> Positif	Negatif	
Positif	1	3	Positif	4	0	<b>4</b>
Negatif	0	17	Negatif	3	14	<b>17</b>
<b>Total</b>	<b>1</b>	<b>20</b>	<b>Total</b>	<b>7</b>	<b>14</b>	<b>21</b>

**Lampiran 5.** Hasil *Confusion Matrix* (Lanjutan)

Aspek 6 (Keseluruhan)

Data MFCC			Data Teks			Total
Kelas Aktual	Kelas Prediksi		Kelas Aktual	Kelas Prediksi		
	Positif	Negatif		Positif	Negatif	
Positif	12	5	Positif	15	2	<b>17</b>
Negatif	1	32	Negatif	5	28	<b>33</b>
<b>Total</b>	<b>13</b>	<b>37</b>	<b>Total</b>	<b>20</b>	<b>30</b>	<b>50</b>

**Lampiran 6.** *Syntax Python Split on Silence*

```

from pydub import AudioSegment
from pydub.silence import split_on_silence
def match_target_amplitude(aChunk, target_dBFS):
    change_in_dBFS = target_dBFS - aChunk.dBFS
    return aChunk.apply_gain(change_in_dBFS)
song = AudioSegment.from_wav("nama audio.wav")
song = song.set_channels(1)
chunks = split_on_silence (
    song,
    min_silence_len = 350,
    silence_thresh = -29,
    keep_silence=300
)
target_length = 10 * 1000
output_chunks = [chunks[0]]
for chunk in chunks[1:]:
    if len(output_chunks[-1]) < target_length:
        output_chunks[-1] += chunk
    else:
        output_chunks.append(chunk)
for i, chunk in enumerate(output_chunks):
    silence_chunk = AudioSegment.silent(duration=300)
    audio_chunk = silence_chunk + chunk + silence_chunk
    normalized_chunk = match_target_amplitude(audio_chunk,
-20.0)
    print("Exporting chunk{0}.wav.".format(i))
    normalized_chunk.export(
        "chunk{0}.wav".format(i),
        bitrate = "192k",
        format = "wav")

```



### Lampiran 7. *Syntax Python* Konversi ke Teks dengan STT

```

import speech_recognition as sr
r = sr.Recognizer()
import glob #untuk mengambil data musik
import numpy as np
import pandas
import matplotlib.pyplot as plt
import os
sound_files=sorted(glob.glob('nama file/*.wav'),
key=os.path.getmtime)
all_features=[]
for f in sound_files:
    features=sr.AudioFile(f)
    all_features.append(features)
all_audio=[]
for features in all_features:
    with features as source:
        audio = r.record(source)
        all_audio.append(audio)
all_input_str = []
for audio in all_audio:
    try:
        input_str = r.recognize_google(audio,language='id')
        all_input_str.append(input_str)
    except:
        pass
with open('nama file.txt', 'w') as f:
    for hasil in all_input_str:
        f.write("%s\n" % hasil)

```

### Lampiran 8. *Syntax Python* Import Data untuk Filter

```

import glob
import numpy as np
import pandas
import os
ulasan=sorted(glob.glob('STT/*.txt'))
all_ulasan=[]
for f in ulasan:
    features = open(f, 'r')
    content = features.read().splitlines()
    all_ulasan.append(content)
all_sound=[]

```

**Lampiran 8.** *Syntax Python Import Data untuk Filter (Lanjutan)*

```

folder_list = ['folder 1', 'folder 2',..., 'folder 50']
import os
for a in folder_list:
    path = 'trim/'+a
    sound_files=sorted(glob.glob(path+"/*.*wav"),
key=os.path.getmtime)
    all_sound.append(sound_files)

```

**Lampiran 9.** *Syntax Python Text Pre-Processing*

```

import nltk
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.tokenize import word_tokenize
from nltk.tokenize.treebank import TreebankWordDetokenizer

factory = StemmerFactory()
stemmer = factory.create_stemmer()
factory = StopWordRemoverFactory()
more_stopword = ['jadi', 'dil', 'bareng', 'gua', 'lagi']
data = factory.get_stop_words()+more_stopword
words = set(nltk.corpus.words.words())

all_ulasan_clean=[]
for ulasan in all_ulasan:
    ulasan_indo=[]
    for teks in ulasan:
        w = teks.lower()
        hasil = w.strip()
        stemm = stemmer.stem(hasil)
        word_tokens = word_tokenize(stemm)
        filtered_sentence = []
        for w in word_tokens:
            if w not in data:
                filtered_sentence.append(w)
        new =
TreebankWordDetokenizer().detokenize(filtered_sentence)
        ulasan_indo.append(new)
    all_ulasan_clean.append(ulasan_indo)
import re
all_ulasan_new=[]

```

### Lampiran 9. *Syntax Python Text Pre-Processing (Lanjutan)*

```
for ulasan in all_ulasan_clean:
    ulasan_indo=[]
    for teks in ulasan:
        rem_num = re.sub('[0-9]+', '', teks)
        ulasan_indo.append(rem_num)
    all_ulasan_new.append(ulasan_indo)
```

### Lampiran 10. *Syntax Python Filter Data Per Aspek*

```
# fungsi untuk memfilter potongan audio ulasan berdasarkan
kata kunci
def filterulasan(all_ulasan,all_audio):
    all_aspek1=[]
    all_aspek2=[]
    all_aspek3=[]
    all_aspek4=[]
    all_aspek5=[]
    all_aspek6=[]
    all_song1=[]
    all_song2=[]
    all_song3=[]
    all_song4=[]
    all_song5=[]
    all_song6=[]
    #mendefinisikan kata kunci masing-masing aspek
    pemeran = ['akting', 'aktris', 'aktor', 'peran',
'memerankan', 'karakter', 'penjahat', 'kinerja',
'performansi', 'dimainkan', 'pemeran', 'pemain',
'chemistry', 'pesona']
    direktor = ['arah', 'arahan', 'penyutradaraan',
'sutradara', 'difilmkan', 'pembuatan film', 'pembuat film',
'sinematik', 'edisi', 'sinematografi', 'direktor']
    jlncerita = ['jalan cerita', 'cerita', 'kisah',
'roman', 'dialog', 'skrip', 'bercerita', 'berakhir',
'ending', 'menceritakan', 'balas dendam', 'pengkhianatan',
'plot', 'penulisan', 'twist', 'drama', 'misteri',
'thriller', 'horor']
    adegan = ['pemandangan', 'animasi', 'screen',
'kekerasan', 'skenario', 'aksi', 'efek khusus', 'special
effect', 'stunt', 'visual', 'alat peraga', 'properti',
'kamera', 'grafik', 'efek', 'kamera']
```

**Lampiran 10. Syntax Python Filter Data Per Aspek (Lanjutan)**

```

musik = ['lirik', 'suara', 'musik', 'audio',
'musical', 'trek', 'efek suara', 'sound effect',
'soundtrack']
lainnya = ['film', 'remake', 'adaptasi', 'sekuel',
'prekuel', 'hiburan']
for ulasan, audio in zip(all_ulasan, all_audio):
    aspek1=[]
    aspek2=[]
    aspek3=[]
    aspek4=[]
    aspek5=[]
    aspek6=[]
    song_aspek1=[]
    song_aspek2=[]
    song_aspek3=[]
    song_aspek4=[]
    song_aspek5=[]
    song_aspek6=[]
    for sentence, chunks in zip(ulasan, audio):
        if any(word in sentence for word in pemeran):
            a = sentence
            aspek1.append(a)
            song = chunks
            song_aspek1.append(song)
        if any(word in sentence for word in direktor):
            a = sentence
            aspek2.append(a)
            song = chunks
            song_aspek2.append(song)
        if any(word in sentence for word in jlncerita):
            a = sentence
            aspek3.append(a)
            song = chunks
            song_aspek3.append(song)
        if any(word in sentence for word in adegan):
            a = sentence
            aspek4.append(a)
            song = chunks
            song_aspek4.append(song)
        if any(word in sentence for word in musik):
            a = sentence

```

### Lampiran 10. *Syntax Python Filter Data Per Aspek (Lanjutan)*

```

        aspek5.append(a)
        song = chunks
        song_aspek5.append(song)
    if any(word in sentence for word in lainnya):
        a = sentence
        aspek6.append(a)
        song = chunks
        song_aspek6.append(song)
all_aspek1.append(aspek1)
all_aspek2.append(aspek2)
all_aspek3.append(aspek3)
all_aspek4.append(aspek4)
all_aspek5.append(aspek5)
all_aspek6.append(aspek6)
all_song1.append(song_aspek1)
all_song2.append(song_aspek2)
all_song3.append(song_aspek3)
all_song4.append(song_aspek4)
all_song5.append(song_aspek5)
all_song6.append(song_aspek6)
return
all_aspek1,all_aspek2,all_aspek3,all_aspek4,all_aspek5,all_
aspek6,all_song1,all_song2,all_song3,all_song4,all_song5,al
l_song6
all_aspek1,all_aspek2,all_aspek3,all_aspek4,all_aspek5,all_
aspek6,all_song1,all_song2,all_song3,all_song4,all_song5,al
l_song6 = filterulasan(all_ulasan_new,all_sound)
all_aspek =
all_aspek1+all_aspek2+all_aspek3+all_aspek4+all_aspek5+all_
aspek6
all_song =
all_song1+all_song2+all_song3+all_song4+all_song5+all_song6

```

### Lampiran 11. *Syntax Python Export Data Audio dan Teks*

```

from pydub import AudioSegment
all_review=[]
for song in all_song:
    ulasan_audio=[]
    for chunks in song:
        song = AudioSegment.from_wav(chunks)
        song = song.set_channels(1)

```

### Lampiran 11. *Syntax Python Export Data Audio dan Teks* (Lanjutan)

```

        ulasan_audio.append(song)
    all_review.append(ulasan_audio)
for i, audio in enumerate(all_review):
    review=AudioSegment.empty()
    for chunk in audio:
        review += chunk
    print("Exporting review{0}.wav.".format(i))
    review.export(
        "trim/review{0}.wav".format(i),
        bitrate = "192k",
        format = "wav"
    )
for i, ulasan in enumerate(all_aspek):
    print("Exporting review{0}.txt.".format(i))
    with open('trim/review{0}.txt'.format(i), 'w') as f:
        for hasil in ulasan:
            f.write("%s\n" % hasil)

```

### Lampiran 12. *Syntax Python Ekstraksi MFCC*

```

import numpy
import scipy.io.wavfile
from scipy.fftpack import dct
import pandas as pd
import glob
import os
sound_files=sorted(glob.glob('trim/*.wav'),
key=os.path.getmtime)
for i, audio in enumerate(sound_files):
    try:
        sample_rate, signal = scipy.io.wavfile.read(audio)
        #pre-emphasis
        pre_emphasis = 0.97
        emphasized_signal = numpy.append(signal[0],
signal[1:] - pre_emphasis * signal[:-1])
        #framing
        frame_size = 0.025
        frame_stride = 0.01
        frame_length, frame_step = frame_size * sample_rate,
frame_stride * sample_rate # Convert from seconds to samples
        signal_length = len(emphasized_signal)

```

**Lampiran 12. Syntax Python Ekstraksi MFCC (Lanjutan)**

```

        frame_length = int(round(frame_length))
        frame_step = int(round(frame_step))
        num_frames
        =
int(numpy.ceil(float(numpy.abs(signal_length
-
frame_length)) / frame_step)) # Make sure that we have at
least 1 frame
        pad_signal_length = num_frames * frame_step +
frame_length
        z = numpy.zeros((pad_signal_length - signal_length))
        pad_signal = numpy.append(emphasized_signal, z) #
Pad Signal to make sure that all frames have equal number of
samples without truncating any samples from the original
signal
        indices = numpy.tile(numpy.arange(0, frame_length),
(num_frames, 1)) + numpy.tile(numpy.arange(0, num_frames *
frame_step, frame_step), (frame_length, 1)).T
        frames = pad_signal[indices.astype(numpy.int32,
copy=False)]
        #Hamming
        frames *= numpy.hamming(frame_length)
        #FFT and Power Spectrum
        NFFT = 512
        mag_frames = numpy.absolute(numpy.fft.rfft(frames,
NFFT)) # Magnitude of the FFT
        pow_frames = ((1.0 / NFFT) * ((mag_frames) ** 2)) #
Power Spectrum
        #Filter Banks
        nfilt = 40
        low_freq_mel = 0
        high_freq_mel = (2595 * numpy.log10(1 + (sample_rate
/ 2) / 700)) # Convert Hz to Mel
        mel_points = numpy.linspace(low_freq_mel,
high_freq_mel, nfilt + 2) # Equally spaced in Mel scale
        hz_points = (700 * (10**(mel_points / 2595) - 1)) #
Convert Mel to Hz
        bin = numpy.floor((NFFT + 1) * hz_points /
sample_rate)

        fbank = numpy.zeros((nfilt, int(numpy.floor(NFFT /
2 + 1))))
        for m in range(1, nfilt + 1):

```

**Lampiran 12. Syntax Python Ekstraksi MFCC (Lanjutan)**

```

        f_m_minus = int(bin[m - 1]) # left
        f_m = int(bin[m]) # center
        f_m_plus = int(bin[m + 1]) # right

        for k in range(f_m_minus, f_m):
            fbank[m - 1, k] = (k - bin[m - 1]) / (bin[m]
- bin[m - 1])
        for k in range(f_m, f_m_plus):
            fbank[m - 1, k] = (bin[m + 1] - k) / (bin[m]
+ 1] - bin[m])
        filter_banks = numpy.dot(pow_frames, fbank.T)
        filter_banks = numpy.where(filter_banks == 0,
numpy.finfo(float).eps, filter_banks) # Numerical Stability
        filter_banks = 20 * numpy.log10(filter_banks) # dB
        #DCT untuk Koefisien MFCC
        num_ceps = 12
        mfcc = dct(filter_banks, type=2, axis=1,
norm='ortho')[:, 1 : (num_ceps + 1)] # Keep 2-13
        print("Exporting review{0}.csv.".format(i))
        import pandas as pd

pd.DataFrame(mfcc).to_csv("trim/review{0}.csv".format(i))
    except:
        pass

```

**Lampiran 13. Syntax Python Model Word2Vec**

```

import gensim
import numpy as np
import os
import matplotlib.pyplot as plt

from sklearn.decomposition import PCA, KernelPCA
import io
import time
from datetime import timedelta

import gensim

start_time = time.time()
print('Streaming wiki...')

```



**Lampiran 13.** *Syntax Python Model Word2Vec (Lanjutan)*

```

id_wiki = gensim.corpora.WikiCorpus('idwiki-latest-pages-
articles.xml.bz2', lemmatize=False, dictionary={})
article_count = 0
with io.open('idwiki.txt', 'w',encoding="utf-8") as
wiki_txt:
    for text in id_wiki.get_texts():
        wiki_txt.write(" ".join(map(str,text)) + '\n')
        article_count += 1

        if article_count % 10000 == 0:
            print('{} articles
processed'.format(article_count))

    print('total: {} articles'.format(article_count))

finish_time = time.time()
print('Elapsed time:
{}'.format(timedelta(seconds=finish_time-start_time)))
import time
import multiprocessing
from datetime import timedelta
from gensim.test.utils import common_texts, get_tmpfile
from gensim.models import word2vec

start_time = time.time()
path = get_tmpfile("idwiki_word2vec_200.model")
print('Training Word2Vec Model...')
sentences = word2vec.LineSentence('idwiki.txt')
id_w2v = word2vec.Word2Vec(sentences, size=200,
workers=multiprocessing.cpu_count()-1)
id_w2v.save("idwiki_word2vec_200.model")
finish_time = time.time()
print('Finished. Elapsed time:
{}'.format(timedelta(seconds=finish_time-start_time)))

```

**Lampiran 14.** *Syntax Python Model CNN untuk MFCC*

```

def ConvNet():
    model = Sequential()
    model.add(Conv1D(256, 5,padding='same',activation='relu
', input_shape=(12,55472))) #1

```

### Lampiran 14. *Syntax Python* Model CNN untuk MFCC (Lanjutan)

```

    model.add(Conv1D(128, 5, padding='same', activation='relu
')) #2
    model.add(MaxPooling1D(pool_size=8))
    model.add(Conv1D(128, 5, padding='same', activation='relu
')) #3
    model.add(Conv1D(128, 5, padding='same', activation='relu
')) #4
    model.add(Conv1D(128, 5, padding='same', activation='relu
')) #5
    model.add(Conv1D(128, 5, padding='same', activation='relu
')) #6
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dense(num_classes, activation='sigmoid'))
    model.compile(loss='binary_crossentropy',
                  optimizer='adam',
                  metrics=['acc'])
    model.summary()
    return model

```

### Lampiran 15. *Syntax Python* Model CNN untuk Data Teks

```

def ConvNet1(embeddings, max_sequence_length, num_words,
embedding_dim, labels_index):

    embedding_layer = Embedding(num_words,
                               embedding_dim,
                               weights=[embeddings],

input_length=max_sequence_length,
                               trainable=False)

    sequence_input = Input(shape=(max_sequence_length,),
dtype='int32')
    embedded_sequences = embedding_layer(sequence_input)

    l_conv = Conv1D(filters=200, kernel_size=1,
activation='relu')(embedded_sequences)
    l_conv = Conv1D(filters=128, kernel_size=1,
activation='relu')(l_conv)
    l_conv = MaxPooling1D(pool_size=8)(l_conv)

```

### Lampiran 15. *Syntax Python Model CNN untuk Data Teks* (Lanjutan)

```

l_conv = Conv1D(filters=128, kernel_size=1,
activation='relu')(l_conv)
l_conv = Conv1D(filters=128, kernel_size=1,
activation='relu')(l_conv)
l_conv = Conv1D(filters=128, kernel_size=1,
activation='relu')(l_conv)
l_conv = Conv1D(filters=128, kernel_size=1,
activation='relu')(l_conv)
l_conv = Flatten()(l_conv)
l_conv = Dense(128, activation='relu')(l_conv)
preds = Dense(labels_index,
activation='sigmoid')(l_conv)

model = Model(sequence_input, preds)
model.compile(loss='binary_crossentropy',
optimizer='adam',
metrics=['acc'])
model.summary()
return model
def ConvNet2(embeddings, max_sequence_length, num_words,
embedding_dim, labels_index):

    embedding_layer = Embedding(num_words,
                                embedding_dim,
                                weights=[embeddings],

input_length=max_sequence_length,
                                trainable=False)

    sequence_input = Input(shape=(max_sequence_length,),
dtype='int32')
    embedded_sequences = embedding_layer(sequence_input)

    l_conv = Conv1D(filters=200, kernel_size=2,
activation='relu')(embedded_sequences)
    l_conv = Conv1D(filters=128, kernel_size=2,
activation='relu')(l_conv)
    l_conv = MaxPooling1D(pool_size=8)(l_conv)
    l_conv = Conv1D(filters=128, kernel_size=2,
activation='relu')(l_conv)

```

### Lampiran 15. *Syntax Python Model CNN untuk Data Teks* (Lanjutan)

```

    l_conv = Conv1D(filters=128, kernel_size=2,
activation='relu')(l_conv)
    l_conv = Conv1D(filters=128, kernel_size=2,
activation='relu')(l_conv)
    l_conv = Conv1D(filters=128, kernel_size=2,
activation='relu')(l_conv)
    l_conv = Flatten()(l_conv)
    l_conv = Dense(128, activation='relu')(l_conv)
    preds = Dense(labels_index,
activation='sigmoid')(l_conv)

    model = Model(sequence_input, preds)
    model.compile(loss='binary_crossentropy',
                  optimizer='adam',
                  metrics=['acc'])
    model.summary()
    return model
def ConvNet3(embeddings, max_sequence_length, num_words,
embedding_dim, labels_index):

    embedding_layer = Embedding(num_words,
                               embedding_dim,
                               weights=[embeddings],

input_length=max_sequence_length,
                               trainable=False)

    sequence_input = Input(shape=(max_sequence_length,),
dtype='int32')
    embedded_sequences = embedding_layer(sequence_input)

    l_conv = Conv1D(filters=200, kernel_size=3,
activation='relu')(embedded_sequences)
    l_conv = Conv1D(filters=128, kernel_size=3,
activation='relu')(l_conv)
    l_conv = MaxPooling1D(pool_size=8)(l_conv)
    l_conv = Conv1D(filters=128, kernel_size=3,
activation='relu')(l_conv)
    l_conv = Conv1D(filters=128, kernel_size=3,
activation='relu')(l_conv)

```

### Lampiran 15. *Syntax Python Model CNN untuk Data Teks (Lanjutan)*

```

l_conv = Conv1D(filters=128, kernel_size=3,
activation='relu')(l_conv)
l_conv = Conv1D(filters=128, kernel_size=3,
activation='relu')(l_conv)
l_conv = Flatten()(l_conv)
l_conv = Dense(128, activation='relu')(l_conv)
preds = Dense(labels_index,
activation='sigmoid')(l_conv)

model = Model(sequence_input, preds)
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['acc'])
model.summary()
return model

```

### Lampiran 16. *Syntax Python Klasifikasi Sentimen MFCC*

```

from __future__ import division, print_function
from gensim import models
from keras.callbacks import ModelCheckpoint
from keras.layers import Dense, Reshape, Flatten,
concatenate, Input, Conv1D, MaxPooling1D, Embedding
from keras.layers.recurrent import LSTM
from keras.models import Sequential
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Model
from sklearn.model_selection import train_test_split
import os
import collections
import re
import string
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
ulasan=sorted(glob.glob('/content/drive/My Drive/Colab
Notebooks/MFCC New/*.csv'))
max_len=55472
all_mfcc=[]

```

## Lampiran 16. *Syntax Python* Klasifikasi Sentimen MFCC (Lanjutan)

```

for ulasan in ulasan:
    mfcc = np.loadtxt(open(ulas, "rb"),
delimitter=",", usecols=(1,2,3,4,5,6,7,8,9,10,11,12), skiprows
=1)
    mfcc = np.transpose(mfcc)
    all_arr=[]
    for arr in mfcc:
        arr = np.trim_zeros(arr)
        all_arr.append(arr)
    all_arr=np.array(all_arr)
    if (max_len > all_arr.shape[1]):
        pad_width = max_len - all_arr.shape[1]
        mfcc = np.pad(all_arr, pad_width=((0, 0), (0,
pad_width)), mode='constant')
    else:
        mfcc = all_arr[:, :]
    all_mfcc.append(mfcc)
all_mfcc = np.array(all_mfcc)
sentimen = pd.read_csv('/content/drive/My Drive/Colab
Notebooks/Label Sentimen New.csv')
sentiment = pd.concat([sentimen], ignore_index=False, axis=1)
sentiment.columns = ['label']
pos = []
neg = []
for l in sentiment.label:
    if l == 0:
        pos.append(0)
        neg.append(1)
    elif l == 1:
        pos.append(1)
        neg.append(0)
sentiment['Pos']= pos
sentiment['Neg']= neg

sentiment = sentiment[['label', 'Pos', 'Neg']]
sentiment.head()
from sklearn.model_selection import KFold
from imblearn.over_sampling import SMOTE
hasil_test = []
hasil_train = []

```

### Lampiran 16. *Syntax Python* Klasifikasi Sentimen MFCC (Lanjutan)

```

kf = KFold(n_splits=(2-10), random_state=0, shuffle=True)
kf.get_n_splits(sentiment)
print(kf)
for train_index, test_index in kf.split(sentiment):
    new = pd.DataFrame()
    sentiment_train, sentiment_test =
sentiment.loc[train_index], sentiment.loc[test_index]
    X_train = all_mfcc[train_index]
    X_train2d = np.reshape(X_train,(len(train_index),-1))
    y_train = sentiment_train['label'].values
    sm = SMOTE()
    X_train2d, y_train = sm.fit_sample(X_train2d, y_train)
    label_names = ['Pos', 'Neg']
    pos = []
    neg = []
    for l in y_train:
        if l == 0:
            pos.append(0)
            neg.append(1)
        elif l == 1:
            pos.append(1)
            neg.append(0)
    new['Pos']= pos
    new['Neg']= neg
    y_train = new[label_names].values
    y_test = sentiment_test[label_names].values
    X_train =
np.reshape(X_train2d,(len(X_train2d),12,55472))
    X_test = all_mfcc[test_index]
    print(X_train.shape)
    feature_dim_2 =55472
    feature_dim_1 = 12
    channel = 1
    epochs = 100
    verbose = 1
    num_classes = 2

    model = ConvNet()

```

## Lampiran 16. *Syntax Python* Klasifikasi Sentimen MFCC (Lanjutan)

```

hist = model.fit(X_train, y_train, epochs=epochs,
validation_split=0.1, shuffle=True, verbose=verbose,
batch_size=len(sentiment_train))
conv1 = np.array(model.layers[0].get_weights())
conv6 = np.array(model.layers[6].get_weights())
dense = np.array(model.layers[9].get_weights())
print(conv1)
print(conv6)
print(dense)

#TEST CNN
predictions = model.predict(X_test,
batch_size=len(sentiment_test), verbose=1)
labels = [1, 0]
prediction_labels=[]
for p in predictions:
    prediction_labels.append(labels[np.argmax(p)])
hasil_test.append(prediction_labels)

sum(sentiment_test.label==prediction_labels)/len(prediction_labels)
from sklearn.metrics import confusion_matrix
print(confusion_matrix(sentiment_test.label,
prediction_labels))
from sklearn.metrics import classification_report
print(classification_report(sentiment_test.label,
prediction_labels))
#TRAIN CNN
X_train = all_mfcc[train_index]
predictions = model.predict(X_train,
batch_size=len(sentiment_train), verbose=1)
labels = [1, 0]
prediction_labels=[]
for p in predictions:
    prediction_labels.append(labels[np.argmax(p)])
hasil_train.append(prediction_labels)

sum(sentiment_train.label==prediction_labels)/len(prediction_labels)
from sklearn.metrics import confusion_matrix

```



### Lampiran 16. *Syntax Python* Klasifikasi Sentimen MFCC (Lanjutan)

```

print(confusion_matrix(sentiment_train.label,
prediction_labels))
from sklearn.metrics import classification_report
print(classification_report(sentiment_train.label,
prediction_labels))

df_train = pd.DataFrame(hasil_train)
df_test = pd.DataFrame(hasil_test)
df_train.to_csv('nama file.csv')
df_test.to_csv('nama file.csv')

```

### Lampiran 17. *Syntax Python* Klasifikasi Sentimen Data Teks

```

ulasan=sorted(glob.glob('/content/drive/My Drive/Colab
Notebooks/Clean Text New/*.txt'))
all_ulasan=[]
for f in ulasan:
    content=' '.join(open(f,'r').read().splitlines())
    if content != '':
        all_ulasan.append(content)
all_ulasan = pd.DataFrame(all_ulasan)
all_ulasan.head()
sentimen = pd.read_csv('/content/drive/My Drive/Colab
Notebooks/Label Sentimen New.csv')
data =
pd.concat([all_ulasan,sentimen],ignore_index=False,axis=1)
data.columns = ['ulasan','label']
import re
import string
def remove_punct(text):
    text_nopunct = ''
    text_nopunct = re.sub('['+string.punctuation+']', '',
text)
    return text_nopunct
data['ulasan_c'] = data['ulasan'].apply(lambda x:
remove_punct(x))
data['ulasan_c'] = data['ulasan_c'].str.replace('\d+', '')
import nltk
nltk.download('punkt')
from nltk import word_tokenize
tokens = [word_tokenize(sen) for sen in data.ulasan_c]

```

## Lampiran 17. *Syntax Python* Klasifikasi Sentimen Data Teks (Lanjutan)

```

data['tokens']=tokens
data.head()
pos = []
neg = []
for l in data.label:
    if l == 0:
        pos.append(0)
        neg.append(1)
    elif l == 1:
        pos.append(1)
        neg.append(0)
data['Pos']= pos
data['Neg']= neg

data = data[['ulasan_c', 'tokens', 'label', 'Pos', 'Neg']]
data.head()
from gensim.test.utils import common_texts, get_tmpfile
from gensim.models import Word2Vec
model = Word2Vec.load("/content/drive/My Drive/Colab
Notebooks/idwiki_word2vec_200.model")
MAX_SEQUENCE_LENGTH = 848
EMBEDDING_DIM = 200
from sklearn.model_selection import KFold
from imblearn.over_sampling import SMOTE
hasil_test = []
hasil_train = []
teks_train = pd.DataFrame()
teks_test = pd.DataFrame()
kf = KFold(n_splits=(2-10), random_state=0, shuffle=True)
kf.get_n_splits(data)
print(kf)
for train_index, test_index in kf.split(data):
    print("TRAIN:", train_index, "TEST:", test_index)
    data_train, data_test = data.loc[train_index],
data.loc[test_index]
    all_training_words = [word for tokens in
data_train["tokens"] for word in tokens]
    training_sentence_lengths = [len(tokens) for tokens in
data_train["tokens"]]
    TRAINING_VOCAB = sorted(list(set(all_training_words)))

```

### Lampiran 17. *Syntax Python* Klasifikasi Sentimen Data Teks (Lanjutan)

```

print("%s words total, with a vocabulary size of %s" %
(len(all_training_words), len(TRAINING_VOCAB)))
print("Max sentence length is %s" %
max(training_sentence_lengths))
all_test_words = [word for tokens in data_test["tokens"]
for word in tokens]
test_sentence_lengths = [len(tokens) for tokens in
data_test["tokens"]]
TEST_VOCAB = sorted(list(set(all_test_words)))
print("%s words total, with a vocabulary size of %s" %
(len(all_test_words), len(TEST_VOCAB)))
print("Max sentence length is %s" %
max(test_sentence_lengths))
#WORD2VEC
tokenizer = Tokenizer(num_words=len(TRAINING_VOCAB),
lower=True, char_level=False)
tokenizer.fit_on_texts(data_train["ulasan_c"].tolist())
training_sequences =
tokenizer.texts_to_sequences(data_train["ulasan_c"].tolist(
))

train_word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(train_word_index))

train_cnn_data = pad_sequences(training_sequences,
maxlen=MAX_SEQUENCE_LENGTH)
train_embedding_weights =
np.zeros((len(train_word_index)+1,
EMBEDDING_DIM))

for word,index in train_word_index.items():
train_embedding_weights[index,:] = model[word] if word
in model else np.random.rand(EMBEDDING_DIM)

print(train_embedding_weights.shape)
test_sequences =
tokenizer.texts_to_sequences(data_test["ulasan_c"].tolist(
))

```

## Lampiran 17. *Syntax Python* Klasifikasi Sentimen Data Teks (Lanjutan)

```

test_cnn_data = pad_sequences(test_sequences,
maxlen=MAX_SEQUENCE_LENGTH)
#CNN
X_train = train_cnn_data
y_train = data_train['label'].values
#smote
sm = SMOTE()
x_train, y_train = sm.fit_sample(X_train, y_train)
pos = []
neg = []
for l in y_train:
    if l == 0:
        pos.append(0)
        neg.append(1)
    elif l == 1:
        pos.append(1)
        neg.append(0)
datatrainnew =
pd.concat([pd.Series(pos),pd.Series(neg)],ignore_index=False,
axis=1)
datatrainnew.columns = ['Pos', 'Neg']
label_names = ['Pos', 'Neg']
y_tr = datatrainnew[label_names].values
model_cnn = ConvNet1(train_embedding_weights,
MAX_SEQUENCE_LENGTH, len(train_word_index)+1,
EMBEDDING_DIM,
len(list(label_names)))

#TRAIN CNN
num_epochs = 100
batch_size = len(data_train)
hist = model_cnn.fit(x_train, y_tr, epochs=num_epochs,
validation_split=0.1, shuffle=True, batch_size=batch_size)
#CONFUSION MATRIX
#TEST CNN
predictions = model_cnn.predict(test_cnn_data,
batch_size=len(data_test), verbose=1)
labels = [1, 0]
prediction_labels=[]

```

## Lampiran 17. *Syntax Python* Klasifikasi Sentimen Data Teks (Lanjutan)

```

for p in predictions:
    prediction_labels.append(labels[np.argmax(p)])
hasil_test.append(prediction_labels)
sum(data_test.label==prediction_labels)/len(prediction_labels)
from sklearn.metrics import confusion_matrix
print(confusion_matrix(data_test.label,
prediction_labels))
from sklearn.metrics import classification_report
print(classification_report(data_test.label,
prediction_labels))
#TRAIN CNN
predictions = model_cnn.predict(train_cnn_data,
batch_size=len(data_train), verbose=1)
labels = [1, 0]
prediction_labels=[]
for p in predictions:
    prediction_labels.append(labels[np.argmax(p)])
hasil_train.append(prediction_labels)

sum(data_train.label==prediction_labels)/len(prediction_labels)
from sklearn.metrics import confusion_matrix
print(confusion_matrix(data_train.label,
prediction_labels))
from sklearn.metrics import classification_report
print(classification_report(data_train.label,
prediction_labels))

teks_train = pd.concat([teks_train,
data_train["ulasan_c"]], axis=1, ignore_index=True)
teks_test = pd.concat([teks_test, data_test["ulasan_c"]],
axis=1, ignore_index=True)

df_train = pd.DataFrame(hasil_train)
df_test = pd.DataFrame(hasil_test)
df_train.to_csv('nama file.csv')
df_test.to_csv('nama file.csv')
teks_train.to_csv('nama file.csv')
teks_test.to_csv('nama file.csv')

```

**Lampiran 18. Output Model CNN Data MFCC**

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 12, 256)	71004416
conv1d_2 (Conv1D)	(None, 12, 128)	163968
max_pooling1d_1 (MaxPooling1D)	(None, 1, 128)	0
conv1d_3 (Conv1D)	(None, 1, 128)	82048
conv1d_4 (Conv1D)	(None, 1, 128)	82048
conv1d_5 (Conv1D)	(None, 1, 128)	82048
conv1d_6 (Conv1D)	(None, 1, 128)	82048
flatten_1 (Flatten)	(None, 128)	0
dense_1 (Dense)	(None, 128)	16512
dense_2 (Dense)	(None, 2)	258

=====  
 Total params: 71,513,346  
 Trainable params: 71,513,346  
 Non-trainable params: 0

**Lampiran 19. Output Model CNN Data Teks**

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 848)	0
embedding_1 (Embedding)	(None, 848, 200)	150400
conv1d_1 (Conv1D)	(None, 848, 200)	40200
conv1d_2 (Conv1D)	(None, 848, 128)	25728
max_pooling1d_1 (MaxPooling1D)	(None, 106, 128)	0
conv1d_3 (Conv1D)	(None, 106, 128)	16512
conv1d_4 (Conv1D)	(None, 106, 128)	16512
conv1d_5 (Conv1D)	(None, 106, 128)	16512
conv1d_6 (Conv1D)	(None, 106, 128)	16512
flatten_1 (Flatten)	(None, 13568)	0
dense_1 (Dense)	(None, 128)	1736832
dense_2 (Dense)	(None, 2)	258

=====

## Lampiran 19. Output Model CNN Data Teks (Lanjutan)

Total params: 2,019,466

Trainable params: 1,869,066

Non-trainable params: 150,400

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	(None, 848)	0
embedding_3 (Embedding)	(None, 848, 200)	150400
conv1d_13 (Conv1D)	(None, 847, 200)	80200
conv1d_14 (Conv1D)	(None, 846, 128)	51328
max_pooling1d_3 (MaxPooling1D)	(None, 105, 128)	0
conv1d_15 (Conv1D)	(None, 104, 128)	32896
conv1d_16 (Conv1D)	(None, 103, 128)	32896
conv1d_17 (Conv1D)	(None, 102, 128)	32896
conv1d_18 (Conv1D)	(None, 101, 128)	32896
flatten_3 (Flatten)	(None, 12928)	0
dense_5 (Dense)	(None, 128)	1654912
dense_6 (Dense)	(None, 2)	258

Total params: 2,068,682

Trainable params: 1,918,282

Non-trainable params: 150,400

Layer (type)	Output Shape	Param #
input_5 (InputLayer)	(None, 848)	0
embedding_5 (Embedding)	(None, 848, 200)	150400
conv1d_25 (Conv1D)	(None, 846, 200)	120200
conv1d_26 (Conv1D)	(None, 844, 128)	76928
max_pooling1d_5 (MaxPooling1D)	(None, 105, 128)	0
conv1d_27 (Conv1D)	(None, 103, 128)	49280
conv1d_28 (Conv1D)	(None, 101, 128)	49280
conv1d_29 (Conv1D)	(None, 99, 128)	49280
conv1d_30 (Conv1D)	(None, 97, 128)	49280
flatten_5 (Flatten)	(None, 12416)	0

**Lampiran 19. Output Model CNN Data Teks (Lanjutan)**

dense_9 (Dense)	(None, 128)	1589376
dense_10 (Dense)	(None, 2)	258

=====  
Total params: 2,134,282  
Trainable params: 1,983,882  
Non-trainable params: 150,400



**Lampiran 20.** Surat Pernyataan Data Sekunder

*(Halaman ini sengaja dikosongkan)*

## BIODATA PENULIS



Nadhifa Ayu Shafirra, lahir di Kota Purwokerto pada tanggal 11 Juni 1999 merupakan anak pertama dari tiga bersaudara. Penulis adalah putri dari Ayah Suyud dan Ibu Liza Deviyanti. Penulis telah menempuh pendidikan formal di SD Muhammadiyah 1 Sidoarjo, SMP Negeri 1 Sidoarjo, dan SMA Muhammadiyah 2 Sidoarjo. Hingga selanjutnya diterima sebagai Mahasiswa Departemen Statistika FSAD ITS melalui jalur SBMPTN di tahun 2016. Di masa perkuliahan, penulis memiliki beberapa pengalaman sebagai asisten dosen di bidang statistika komputasi. Serta memiliki pengalaman di berbagai kepanitiaan, seperti Ketua *Statistic Competition* 2018, *Technical Officer* Indonesia *Scholarship Festival* (ISF) 2019, anggota Divisi Desain Dokumentasi di Olimpiade Sekolah Rakyat 2019, serta *Field Volunteer* dari acara *English in The Park* 2019. Selain itu, penulis juga aktif di organisasi dan komunitas seperti HIMASTA-ITS, BEM FMKSD ITS, serta menjabat sebagai wakil ketua Divisi *Event* di Sahabat Beasiswa Surabaya 2019. Apabila pembaca ingin memberi kritik dan saran atau diskusi lebih lanjut terkait Tugas Akhir ini, pembaca dapat menghubungi penulis melalui *e-mail*. [nadhifashafirra@gmail.com](mailto:nadhifashafirra@gmail.com).