



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IS4853

**KLASIFIKASI SENTIMEN ASPEK LEVEL OPERATOR
TELEKOMUNIKASI PADA TWITTER MENGGUNAKAN
MULTI GRAINED ATTENTION NETWORK**

***ASPECT-LEVEL SENTIMENT CLASSIFICATION OF
TELECOMMUNICATION OPERATOR IN TWITTER
USING MULTI-GRAINED ATTENTION NETWORK***

MAGRID PRISCA JUWITA
NRP 05211540000124

Dosen Pembimbing:
Renny Pradina , S.T., M.T., SCJP

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

TUGAS AKHIR - IS4853

**KLASIFIKASI SENTIMEN ASPEK LEVEL
OPERATOR TELEKOMUNIKASI PADA TWITTER
MENGUNAKAN MULTI GRAINED ATTENTION
NETWORK**

MAGRID PRISCA JUWITA
NRP 0521154000124

Dosen Pembimbing
Renny Pradina K., ST, MT, SCJP

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

UNDERGRADUATE THESIS - KS141501

***ASPECT-LEVEL SENTIMENT CLASSIFICATION
OF TELECOMMUNICATION OPERATOR IN
TWITTER USING MULTI-GRAINED ATTENTION
NETWORK***

MAGRID PRISCA JUWITA

NRP 05211540000124

Dosen Pembimbing

Renny Pradina K., ST, MT, SCJP

DEPARTEMEN SISTEM INFORMASI

Fakultas Teknologi Informasi dan Komunikasi

Institut Teknologi Sepuluh Nopember

Surabaya 2019

LEMBAR PENGESAHAN

**KLASIFIKASI SENTIMEN ASPEK LEVEL OPERATOR
TELEKOMUNIKASI PADA TWITTER
MENGUNAKAN MULTI GRAINED ATTENTION
NETWORK**

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

MAGRID PRISCA JUWITA
NRP 05211540000124

Surabaya, 9 Juli 2019

**KEPALA
DEPARTEMEN SISTEM INFORMASI**



Mahendrawati ER, ST, M.Sc, Ph.D
197610112006042001

HALAMAN PERSETUJUAN

KLASIFIKASI SENTIMEN ASPEK LEVEL OPERATOR TELEKOMUNIKASI PADA TWITTER MENGUNAKAN MULTI GRAINED ATTENTION NETWORK

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Pada


Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:


MAGRID PRISCA JUWITA
NRP 05211540000124

Disetujui Tim Penguji : Tanggal Ujian : 9 Juni 2019
Periode Wisuda : September 2019

Renny Pradina K., ST, MT, SCJP


(Pembimbing I)

Nur Aini Rakhmawati S.Kom, M.Sc.Eng


(Penguji I)

Faisal Johan Atletiko, S.Kom, M.Kom


(Penguji II)



KLASIFIKASI SENTIMEN ASPEK LEVEL OPERATOR TELEKOMUNIKASI PADA TWITTER MENGUNAKAN MULTI GRAINED ATTENTION NETWORK

Nama Mahasiswa : Magrid Prisca Juwita
NRP : 05211540000124
Departemen : Sistem Informasi
Pembimbing I : Renny Pradina K., ST, MT, SCJP

ABSTRAK

Sosial media merupakan suatu hal yang sulit untuk terlepas dalam setiap kegiatan sehari-hari di era modern ini, salah satu sosial media tersebut adalah Twitter. Setiap orang dapat memberikan pendapat atau komentarnya secara bebas di Twitter. Pendapat atau penilaian terhadap suatu objek disebut sentimen. Dalam melakukan analisis sentimen aspek level terdapat dua pengelompokan, yaitu aspek berdasarkan kategori dan aspek berdasarkan terms. Pendapat yang disampaikan melalui sosial media dimanfaatkan oleh perusahaan untuk mengetahui pendapat masyarakat terhadap produk atau jasa miliknya. Saat ini industri telekomunikasi semakin berkembang dan sangat dibutuhkan oleh masyarakat Indonesia yang memiliki sifat konsumtif terhadap penggunaan handphone dan gadget yang membutuhkan koneksi internet. Hal itu menyebabkan banyaknya pendapat dan komentar terhadap operator telekomunikasi di sosial media Twitter.

Dalam menerapkan analisis sentimen berdasarkan aspek level dibutuhkan algoritma untuk dapat melakukan klasifikasi teks. Pada penelitian ini akan digunakan metode Attention Network untuk melakukan klasifikasi sentimen aspek level berbahasa Indonesia menggunakan data dari Twitter. Mekanisme attention dalam Neural Network berfungsi untuk mengarahkan persepsi serta mengakses memori. Attention akan melihat dari memori yang sebelumnya telah disimpan maupun yang baru saja didapatkan. Attention yang akan digunakan memiliki dua tahap yaitu coarse-grained attention dan fine-

grained attention, kedua tahap tersebut digabungkan sehingga menjadi Multi Grained Attention Network (MGAN). MGAN membutuhkan input yang berupa vektor untuk dapat melakukan training data yang menghasilkan sebuah model. Sehingga untuk mendapatkan input berupa vektor maka akan dibutuhkan library word2vec.

Selain menggunakan MGAN, pada penelitian ini juga menggunakan algoritma Long Short Term Memory (LSTM). LSTM merupakan sebuah algoritma yang akan melakukan klasifikasi dengan menggunakan aspek-aspek yang telah disimpan dan akan membuang aspek yang tidak digunakan. Algoritma ini digunakan untuk membandingkan hasil model yang didapatkan MGAN dengan model LSTM.

Data yang digunakan berjumlah 2000 tweet, dengan pembagian 20% data test dan 80% data train. Masing masing tweet dibeikan label dengan 4 jenis pembagian label sentimen yaitu negatif, netral, positif dan konflik. Akurasi test terbaik menggunakan model MGAN adalah 78%. Sedangkan akurasi test terbaik menggunakan model LSTM yaitu 76%.

Kata kunci: Analisis sentimen, aspek level, Twitter, telekomunikasi, Attention Network, Coarse-grained, Fine-grained, word2vec

**ASPECT-LEVEL SENTIMENT CLASSIFICATION OF
TELECOMUNICATION OPERATOR IN TWITTER
USING MULTI-GRAINED ATTENTION NETWORK**

Name : Magrid Prisca Juwita
NRP : 0521154000124
Department : Information System FTIK-ITS
Supervisor : Renny Pradina K., ST, MT, SCJP

ABSTRACT

Social media is a difficult thing to get rid of in every day's activities in this modern era, one of these social media is Twitter. Everyone can give their opinions or comments freely on Twitter. Opinion or judgment of an object is called sentiment. In conducting aspect level sentiment analysis there are two groupings, namely aspects based on categories and aspects based on terms. Opinions conveyed through social media are used by companies to find out people's opinions on their products or services. Currently the telecommunications industry is growing and is needed by Indonesian people who have consumptive nature towards the use of mobile phones and gadgets that require internet connections. It caused many opinions and comments on telecommunications operators on social media Twitter.

In applying sentiment analysis based on the aspect level, an algorithm is needed to be able to classify text. In this study, the Attention Network method will be used to classify aspects of Indonesian language sentiment using data from Twitter. The mechanism of attention in the Neural Network serves to direct perception and access memory. Attention will see from memory that has been previously saved as well as the one just obtained. Attention that will be used has two stages, namely coarse-grained attention and fine-grained attention, these two stages are combined to become a Multi Grained Attention Network (MGAN). MGAN requires vector input for training to do data

training that produces a model. So to get a vector input, the word2vec library will be needed.

In addition to using MGAN, this study also uses the Long Short Term Memory (LSTM) algorithm. LSTM is an algorithm that will classify using saved aspects and forget the unused aspects. This algorithm is used to compare the results of the model obtained by MGAN with the LSTM model.

The data used amounted to 2000 tweets, with a share of 20% test data and 80% data train. Each tweet is labeled with 4 types of sentiment labeling namely negative, neutral, positive and conflict. The best test accuracy using the MGAN model is 78%. While the best test accuracy using the LSTM model is 76%.

Keywords: Sentiment analysis, level aspects, Twitter, telecommunications, Attention Network, Coarse-grained, Fine-grained, word2vec

KATA PENGANTAR

Dengan mengucapkan rasa syukur kepada Allah SWT atas izin-Nya penulis dapat menyelesaikan buku Tugas Akhir dengan judul Klasifikasi Sentimen Aspek Level Operator Telekomunikasi pada Twitter Menggunakan Multi Grained Attention Network ini dengan lancar dan tepat waktu. Dalam penyusunan pengerjaan Tugas Akhir ini tidak terlepas dari bantuan berbagai pihak kepada penulis. Maka pada kesempatan ini penulis mengucapkan terimakasih kepada pihak-pihak yang telah membantu penulis hingga terselesaikannya penyusunan buku Tugas Akhir ini, yang antara lain kepada:

1. Allah SWT yang telah melimpahkan rahmat serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir.
2. Bapak Jumadi dan Ibu Wartini selaku kedua orang tua Magrid Prisca Juwita yang tiada henti memberikan doa, bimbingan, dan semangat dalam pelaksanaan Tugas Akhir
3. Mayka Maharani dan Andri Widodo selaku kakak kanding dan kakak ipar yang selalu mendukung, memberi arahan dan semangat dalam pengerjaan Tugas Akhir.
4. Ibu Renny Pradina K., S.T., M.T., SCJP, selaku dosen pembimbing yang senantiasa meluangkan waktu, membagikan ilmu dan arahan, serta memotivasi untuk kelancaran tugas akhir.
5. Ibu Mahendrawathi ER. S.T., M.Sc., Ph.D. selaku Kepala Departemen Sistem Informasi ITS Surabaya
6. Ibu Nur Aini Rakhmawati S.Kom, M.Sc.Eng dan Bapak Radityo Prasetyanto W., S.Kom, M.Kom selaku dosen penguji yang telah memberikan kritik dan saran yang membuat kualitas penelitian ini lebih baik lagi.
7. Segenap dosen dan karyawan Departemen Sistem Informasi yang telah membantu kelancaran tugas akhir.

8. Anindya Dwi dan Fia Afina selaku sahabat penulis yang selalu mendukung dan menyemangati penulis dikala lelah mengerjakan tugas akhir
9. Weny, Evia, Andira, Supri, Yoga, Salim dan Azzam yang telah berbagi ilmu tentang materi pada topik tugas akhir penulis
10. Mbak Santi dan Mbak Nailis selaku teman kos yang *mensupport* penulis mengerjakan tugas akhir dalam bentuk makanan, dukungan dan doa
11. Teman teman di KSR PMI ITS dan Lannister yang telah memberikan banyak kenangan manis dan pahit semasa kuliah
12. Pihak lainnya yang tidak bisa disebutkan satu persatu yang telah berkontribusi dalam menyelesaikan tugas akhir

Penyusunan tugas akhir ini masih jauh dari kata sempurna, untuk itu penulis menerima segala kritik dan saran yang membangun sebagai upaya menjadi lebih baik lagi ke depannya. Semoga buku tugas akhir ini dapat memberikan manfaat untuk pembaca.

Surabaya, 9 Juli 2019

Penulis,

Magrid Prisca Juwita

DAFTAR ISI

ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
DAFTAR KODE	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Relevansi	4
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Sebelumnya	5
2.2 Dasar Teori	7
2.2.1 Natural Language Processing	7
2.2.2 Sentimen Analisis	8
2.2.3 Word Embeddings	9
2.2.4 <i>Word2vec</i>	9
2.2.5 <i>Long-Short Term Memory</i>	10
2.2.6 <i>Attention Network</i>	11
2.2.7 Operator Telekomunikasi	13
2.2.8 <i>Twitter</i>	13
BAB III METODOLOGI	15
3.1 Diagram Metodologi	15
3.2 Arsitektur Penelitian	16
3.3 Uraian Metodologi	16
3.2.1. Studi Literatur.....	17
3.2.2. Pengumpulan Data	17
3.2.3. Pre-processing Data	17
3.2.3.1 Data Tagging	17
3.2.3.2 Pembersihan Data	18

3.2.3.3	<i>Case folding</i> dan Tokenisasi	19
3.2.3.4	Word Embedding menggunakan Word2vec	19
3.2.4.	Training dan Testing Data.....	20
3.2.5.	Analisis Model	20
3.2.6.	Dokumentasi	21
BAB IV	PERANCANGAN	23
4.1	Perancangan Dataset.....	23
4.2	Perancangan <i>Preprocessing</i> Data	23
4.2.1	Pemberian Label Data.....	23
4.2.2	Pembersihan Dataset.....	26
4.2.3	<i>Case folding</i> dan Tokenisasi	26
4.2.4	<i>Word Embedding</i>	26
4.3	Perancangan Model MGAN.....	27
4.3.1	Perancangan Training Model MGAN.....	27
4.3.2	Perancangan Testing Model MGAN	29
4.4	Perancangan Demo Prediksi.....	29
BAB V	IMPLEMENTASI.....	31
5.1	Lingkungan Implementasi.....	31
5.2	<i>Preprocessing</i> Data.....	32
5.2.1	Pemberian Label Data.....	32
5.2.2	Pembersihan Data.....	32
5.2.3	<i>Case Folding</i> dan Tokenisasi	33
5.2.4	Word Embedding.....	35
5.3	Pembuatan Model MGAN	36
5.3.1	Inisiasi Argumen	37
5.3.2	Pembuatan <i>Class</i> MGAN.....	39
5.3.3	Pembuatan <i>Training</i> model MGAN	40
5.3.4	Pembuatan <i>Testing</i> model MGAN	42
5.4	Demo Prediksi	45
BAB VI	HASIL DAN PEMBAHASAN	49
6.1	<i>Preprocessing</i>	49
6.1.1	Hasil Pemberian Label Dataset	49
6.1.2	Pembersihan Data.....	49
6.1.3	<i>Case Folding</i> dan Tokenisasi	50
6.1.4	Word Embedding.....	52
6.2	Hasil Pembuatan Model MGAN	53

6.2.1	Konfigurasi Parameter Awal	53
6.2.2	Hasil Training dan Tes	54
6.3	Perbandingan Model lain.....	57
6.4	Pengukuran Performa per Sentimen Model MGAN	58
6.5	Pengukuran Performa per Sentimen Model LSTM.....	60
6.6	Hasil Demo Prediksi	62
BAB VII KESIMPULAN DAN SARAN		65
Kesimpulan.....		65
Saran.....		65
DAFTAR PUSTAKA		67
LAMPIRAN A. DATA BERLABEL		69
LAMPIRAN B. Jumlah Token Aspek		83
1.	Aspek pada Data Train	83
2.	Aspek pada Data Tes.....	84
LAMPIRAN C. Hasil Akurasi MGAN.....		85
1.	Perulangan ke-0	85
2.	Perulangan ke-1	86
3.	Perulangan ke-2	88
4.	Perulangan ke-3	89
5.	Perulangan ke-4	91
BIODATA PENULIS		93

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1 Diagram blok LSTM oleh Benny Prijono [9].....	10
Gambar 2.2 Keterangan diagram blok LSTM oleh Benny Prijono [9]	11
Gambar 2.3 Cara kerja attention oleh Skimind [12]	11
Gambar 2.4 Arsitektur attention oleh Synced [13]	12
Gambar 3.1 Metodologi	15
Gambar 3.2 Arsitektur Penelitian	16
Gambar 3.3 Flowchart pelabelan data.....	18
Gambar 3.4 Flowchart word embedding.....	19
Gambar 6.1 Hasil word embedding	52
Gambar 6.2 Percobaan Perulangan ke-0	54
Gambar 6.3 Percobaan Perulangan ke-1	55
Gambar 6.4 Percobaan Perulangan ke-2	56
Gambar 6.5 Percobaan Perulangan ke-3	56
Gambar 6.6 Percobaan Perulangan ke-4	57
Gambar 6.7 Perbandingan akurasi MGAN dan LSTM.....	57
Gambar 6.8 Confussion Matrix Model MGAN	59
Gambar 6.9 ROC AUC Model MGAN.....	60
Gambar 6.10 Confussion Matrix Model LSTM	61
Gambar 6.11 ROC AUC Model LSTM	62
Gambar 6.12 Hasil prediksi sentimen	63
Gambar 6.13 Hasil prediksi sentimen 2	63

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Literatur 1	5
Tabel 2.2 Literatur 2	6
Tabel 2.3 Literatur 3	6
Tabel 4.1 Label Topik.....	24
Tabel 4.2 Label Aspek	24
Tabel 4.3 Label Sentimen.....	25
Tabel 4.4 Parameter Model MGAN.....	27
Tabel 4.5 Parameter Training Model	28
Tabel 5.1 Spesifikasi perangkat keras	31
Tabel 5.2 Spesifikasi perangkat lunak	31
Tabel 6.1 Sampel data hasil pembersihan	49
Tabel 6.2 Sampel hasil case folding dan tokenisasi.....	50
Tabel 6.3 Jumlah label	51
Tabel 6.4 Konfigurasi awal model.....	53
Tabel 6.5 Konfigurasi awal training	54
Tabel 6.6 Perbandingan Model MGAN dan LSTM.....	58
Tabel 6.7 Performa per sentimen MGAN	59
Tabel 6.8 Performa per sentimen LSTM.....	61

Halaman ini sengaja dikosongkan

DAFTAR KODE

Kode 5.1 Pembersihan data	33
Kode 5.2 <i>Case folding</i> dan tokenisasi	34
Kode 5.3 Menyimpan hasil tokenisasi	34
Kode 5.4 <i>Loading file word2vec embbeding</i>	35
Kode 5.5 <i>Embedding</i> token	36
Kode 5.6 Persiapan data	37
Kode 5.7 Inisiasi model.....	37
Kode 5.8 Inisiasi argumen.....	38
Kode 5.9 Inisialisasi model MGAN.....	39
Kode 5.10 <i>Foward</i> MGAN	40
Kode 5.11 Inisiasi CrossEntropyLoss	40
Kode 5.12 <i>Training</i> model	41
Kode 5.13 Penghitungan akurasi training	41
Kode 5.14 Testing model	43
Kode 5.15 Menyimpan model	43
Kode 5.16 Menyimpan hasil ke csv	44
Kode 5.17 Menyimpan grafik.....	45
Kode 5.18 Memuat model tertinggi	45
Kode 5.19 <i>Preprocess</i> prediksi	46
Kode 5.20 Prediksi kata	47

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Pada bab ini, akan dijelaskan mengenai proses identifikasi masalah meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan relevansi tugas akhir. Berdasarkan uraian pada bab ini diharapkan gambaran umum permasalahan dan pemecahan masalah masalah pada penelitian ini dapat dipahami.

1.1 Latar Belakang

Indonesia merupakan negara dengan penduduk jumlah yang besar. Berdasarkan data dari proyeksi Badan Perencanaan Pembangunan Nasional (Bappenas) 2013 jumlah penduduk indonesia mencapai 265 juta jiwa pada tahun 2018 [1]. Sedangkan menurut Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) dari hasil survey penetrasi pengguna internet di Indonesia meningkat menjadi 143,7 juta jiwa atau setara 54,7 persen dari jumlah penduduk indonesia saat itu [2]. Tak lepas dari itu penduduk indonesia juga menggunakan internet untuk mengakses media sosial. Berdasarkan penelitian yang dilakukan oleh We Are Social yang bekerjasama dengan Hootsuite, menyebutkan bahwa ada 130 juta orang Indonesia yang terbilang aktif di media sosial salah satunya adalah *Twitter*.

Twitter merupakan salah satu media sosial yang populer digunakan oleh penduduk indonesia yang termasuk dalam kategori *microblogging* atau blog singkat, yang mana penggunaanya bisa menuliskan maksimal 280 karakter dalam 1 kali postingan. Dengan adanya media sosial saat ini akan sangat mudah menyampaikan informasi dan juga pendapat. Karena sekali melakukan *posting* di *Twitter* maka siapapun dapat mengakses postingan tersebut. *Twitter* juga memberikan fasilitas *trending topic* untuk mengetahui apa yang sedang dan

paling banyak di bicarakan orang-orang melalui *Twitter* pada saat itu [3].

Telekomunikasi merupakan interaksi yang dilakukan oleh dua orang atau lebih pada jarak jauh. Di Indonesia sendiri ada banyak penyedia layanan telekomunikasi, menurut data dari katatdata.co.id pada September 2018 terdapat 2 pengguna operator telekomunikasi paling banyak yaitu Telkomsel 167,8 juta pengguna dan Indosat 61,4 juta pengguna dengan pengguna [4]. Dengan banyaknya pengguna yang dimiliki maka banyak pula orang yang berpendapat tentang dua operator telekomunikasi tersebut. Pendapat yang diberikan bisa berupa pendapat yang positif maupun negatif. Pendapat kebanyakan orang saat ini dituangkan kedalam sosial media salah satunya adalah *Twitter*.

Analisis sentimen merupakan salah satu jenis *Natural Language Processing* yang digunakan untuk mengetahui opini publik terhadap suatu produk atau topik tertentu [5]. Sumber data tersebut bisa diambil melalui ulasan atau pendapat pada *twitter*. Dengan ulasan-ulasan dan pendapat publik mengenai operator telekomunikasi yang ada di Indonesia dapat dilakukan klasifikasi sentimen menggunakan *Multi-Grained Attention Network* (MGAN). MGAN sendiri merupakan cara yang digunakan untuk menangkap interaksi antar aspek kata yang terdiri dari 2 proses yaitu *Coarse-Grained Attention* dan *Fine-Grained Attention* yang mana keduanya digabungkan untuk memperkuat perbedaan attention antara aspek dengan konteks dan polaritas sentimen yang berbeda [6].

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang yang telah disampaikan, berikut merupakan perumusan masalah pada tugas akhir ini,

1. Bagaimana menentukan aspek berdasarkan *terms* yang digunakan untuk memberikan label pada dataset operator telekomunikasi Telkomsel dan Indosat?

2. Apa jenis sentimen yang akan diterapkan pada setiap aspek yang di analisis?
3. Bagaimana membuat model dengan *Bidirectional Long-Short Term Memory Network*?
4. Bagaimana mengukur performa sentimen aspek-level pada dataset operator telekomunikasi menggunakan *Multi-Grained Attention Network*?

1.3 Batasan Masalah

Berikut ini Batasan masalah tugas akhir ini, berdasarkan latar belakang dan perumusan masalah yang telah dijelaskan,

1. Sumber data berasal dari *Twitter* yang menggunakan bahasa Indonesia
2. Data yang digunakan terkait dengan topik terkait penyedia layanan telekomunikasi Telkomsel dan Indosat yang ada di Indonesia
3. Model MGAN dan LSTM hanya melakukan training untuk mendapatkan model setimen terbaik berdasarkan kalimat input dan aspeknya.

1.4 Tujuan Penelitian

Berdasarkan perumusan dan Batasan masalah yang telah diuraikan sebelumnya, maka tujuan yang ingin dicapai adalah:

1. Mendapatkan aspek berdasarkan terms pada dataset operator telekomunikasi untuk digunakan sebagai label.
2. Mengetahui jenis sentimen yang akan diberikan berdasarkan 4 kategori yaitu positif, negatif netral dan konflik berdasarkan aspek yang telah di tentukan
3. Membuat model untuk menganalisis klasifikasi sentimen aspek level pada data vektor tentang layanan operator telekomunikasi Telkomsel dan Indosat dengan menggunakan algoritma *Bidirectional Long-Short Term Memory Network* (BiLSTM)

4. Melakukan uji coba training dan testing terhadap dataset vektor untuk menghasilkan model dengan performa terbaik menggunakan *Multi-Grained Attention Network*

1.5 Manfaat Penelitian

Berikut merupakan manfaat yang diharapkan dari penelitian yang dilakukan:

Bagi Akademisi:

1. Memahami *Multi-Grained Attention Network* sebagai metode untuk menangkap interaksi kata pada data berdasarkan aspek yang dimiliki khususnya pada data operator seluler
2. Memahami *Bidirectional Long-Short Term Memory* sebagai metode untuk menangkap interaksi antar kata dengan pengerjaan yang terpisah berdasarkan jarak waktu

Bagi Organisasi:

Perusahaan Telkomsel dan Indosat dapat mengetahui pendapat masyarakat sebagai pengguna dari layanan yang diberikan untuk digunakan sebagai evaluasi dan masukan bagi perusahaan

1.6 Relevansi

Relevansi tugas akhir ini sesuai dengan *roadmap* dari laboratorium Akuisisi Data dan Diseminasi Informasi (ADDI) Departemen Sistem Informasi karena tugas akhir ini berkaitan dengan penerapan matakuliah keilmuan laboratorium ADDI. Mata kuliah tersebut antara lain Sistem Cerdas, Pengolahan Bahasa Alami dan Penggalian Data

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai penelitian sebelumnya dan dasar teori yang menjadi acuan dalam pengerjaan tugas akhir ini. Penelitian sebelumnya memberikan gambaran mengenai apa saja yang telah dilakukan dan dasar teori memberikan gambaran secara umum dari tugas ini.

2.1 Penelitian Sebelumnya

Pada penelitian ini, digunakan beberapa penelitian terdahulu yang berguna sebagai pedoman dan referensi dalam pelaksanaan prosesnya. Informasi yang disampaikan berisi tentang penelitian sebelumnya, hasil penelitian, dan hubungan penelitian yang akan dilakukan terhadap penelitian sebelumnya dalam rangka tugas akhir ini. Berikut penelitian sebelumnya yang akan dijelaskan pada Tabel 2.1 – Tabel 2.3.

Tabel 2.1 Literatur 1

Judul	Multi-grained Attention Network for Aspect-Level Sentiment Classification
Nama, Tahun	Feifan Fan, Yansong Feng, Dongyan Zhao (2018)
Metodologi	<i>Bidirectional Long-Short Term Memory Network</i> <i>Multi-Grained Attention Layer</i>
Relevansi Penelitian	Penelitian ini berfokus pada klasifikasi sentimen aspek level berbasis text dengan menggunakan aspek level attention dengan 6 tahapan, yaitu: 1. Pendefinisian task 2. <i>Input embedding layer</i> 3. <i>Contextual layer</i> 4. <i>Multi-grained attention layer</i> 5. <i>Output layer</i> 6. <i>Model training</i> [6]

Tabel 2.2 Literatur 2

Judul	IARM: Inter-Aspect Relation Modelling with Memory Network in Aspect-Based Sentiment Analysis
Nama, Tahun	Navonil Majumder, Soujanya Poria, Alexander Gelbukh, Md. Shad Akhtar, Erik Cambria, Asif Ekbal (2018)
Metodologi	<i>Long-Short Term Memory</i> <i>Two Different Long-Short Term Memory</i>
Relevansi Penelitian	Penelitian ini berfokus kepada sentimen aspek pada produk dari perusahaan besar seperti Samsung dan Apple dari ulasan pengguna yang terdapat pada berbagai situs web e-commerce dan sosial media. Penyajian yang dilakukan menggunakan pendekatan baru dengan menggabungkan aspek informasi terkait dari aspek target menggunakan <i>Memory Network</i> [7]

Tabel 2.3 Literatur 3

Judul	Recurrent Attention Networkkon Memory for Aspect Sentiment Analysis
Nama, Tahun	Peng Chen, Zhongqian Sun, Lidong Bing, Wei Yang (2017)
Metodologi	<i>Bidirectional Long-Short Term Memory</i> <i>Recurrent Attention Network</i>
Relevansi Penelitian	Penelitian ini mengusulkan kerangka kerja baru berdasarkan <i>neural network</i> untuk mengidentifikasi sentimen dari opini yang ada pada kometar atau ulasan. <i>Framework</i> yang digunakan diadopsi dari <i>multi-attention mechanism</i> , yaitu sebagai mekanisme untuk menangkap fitur sentimen yang dipisahkan oleh jarak yang jauh. Mekanisme pembobotan memori dapat digunakan untuk menyediakan memori yang dibuat khusus untuk target opini yang berbeda dalam sebuah kalimat.

2.2 Dasar Teori

Dasar teori berisi teori-teori yang digunakan dalam pengerjaan penelitian tugas akhir. Dalam landasan teori, acuan yang digunakan adalah berdasarkan penelitian dan buku.

2.2.1 Natural Language Processing

Natural Language Processing adalah serangkaian teknik komputasi secara teoritis untuk menganalisis dan menggambarkan teks yang muncul secara alami pada satu atau lebih tingkat analisis linguistik untuk tujuan mencapai pemrosesan bahasa yang mirip manusia untuk berbagai *tasks* atau aplikasi [8]. Tujuan dari NLP adalah untuk menyelesaikan pemrosesan bahasa yang mirip manusia. Awalnya bidang NLP disebut juga *Natural Language Understanding* (NLU) pada awal kemunculannya, hingga akhirnya ada kesepakatan bahwa NLU merupakan tujuan dari NLP. Sistem NLU adalah sebagai berikut:

1. Parafrase teks input
2. Menterjemahkan ke bahasa lain
3. Menjawab pertanyaan tentang isi teks
4. Menarik Kesimpulan dari teks

Metode yang paling jelas untuk menyajikan apa yang sebenarnya terjadi dalam sistem NLP adalah dengan menggunakan pendekatan 'level bahasa'. Ini juga disebut sebagai model bahasa sinkronis dan dibedakan dari model sekuensial sebelumnya, yang berhipotesis bahwa tingkat pemrosesan bahasa manusia mengikuti satu sama lain dengan cara yang sangat berurutan. Penelitian psikolinguistik menunjukkan bahwa pemrosesan bahasa jauh lebih dinamis, karena levelnya dapat berinteraksi dalam berbagai urutan. Berikut ini adalah level yang terdapat pada NLP:

1. **Phonology** : Tingkat ini berkaitan dengan interpretasi bunyi wicara di dalam dan di seluruh kata-kata
2. **Morphology** : Tingkat ini berkaitan dengan sifat komposisional kata-kata, yang terdiri dari morfem - unit makna terkecil
3. **Lexical** : Pada tingkat ini, manusia, serta sistem NLP, menafsirkan makna kata-kata individual.

4. **Syntactic**: Tingkat ini berfokus pada menganalisis kata-kata dalam kalimat sehingga dapat mengungkap struktur tata bahasa kalimat.
5. **Semantic** : Tingkat pemrosesan ini dapat mencakup disambiguasi kata semantik dengan beberapa indera
6. **Discourse** : tingkat ini berfokus pada sifat-sifat teks secara keseluruhan yang menyampaikan makna dengan membuat hubungan antara kalimat-kalimat komponen
7. **Pragmatic** : Tingkat ini berkaitan dengan penggunaan bahasa yang digunakan dalam situasi dan memanfaatkan konteks di atas dan di atas isi teks untuk pemahaman [8].

2.2.2 Sentimen Analisis

Analisis sentimen merupakan salah satu jenis *Natural Language Processing* yang digunakan untuk mengetahui opini publik terhadap suatu produk atau topik tertentu. Analisis sentimen atau yang juga disebut *opinion mining*, membutuhkan sistem untuk mengumpulkan dan memeriksa opini terhadap produk atau topik yang bersumber dari postingan, komentar, ulasan dan tweet. Analisis sentimen dapat bermanfaat dalam beberapa hal. Misalnya, dalam pemasaran itu membantu menghalangi keberhasilan kampanye iklan atau peluncuran produk baru, menentukan versi produk atau layanan yang populer bahkan mengidentifikasi demografi yang suka atau tidak suka terhadap fitur tertentu [5]. Secara umum sentimen analisis dibagi menjadi 2 kategori:

1. **Coarse-grained sentiment analysis**, yaitu sentimen analisis yang dilakukan pada level dokumen.
2. **Fine-grained sentiment analysis**, yaitu sentimen analisis yang dilakukan pada sebuah kalimat.

Pengklasifikasian yang berorientasi pada sebuah dokumen secara keseluruhan dibagi menjadi 4 macam yaitu positif, negatif, netral dan konflik. Berikut adalah masing-masing penjelasan mengenai sentimen tersebut.

1. **Positif** : Yang dimaksud dengan sentimen positif adalah pendapat yang memiliki makna yang baik yang

didalamnya memuat kata-kata seperti “cepat”, “lancar”, “mudah”, “suka”, “senang” dan sebagainya.

- Contoh** : indosat mantap djaja koneksinya , ngebut euy
2. **Negatif** : Sedangkan sentimen negatif adalah pendapat yang memiliki makna buruk yang memuat kata-kata seperti “lelet”, “jelek”, “sampah” dan lain sebagainya.
Contoh : sinyal jelek bgt sih, kecewa nih pakai simpati daerah buaran, serpong
 3. **Netral** : Lalu yang dimaksud dengan sentimen netral yaitu sebuah pendapat yang tidak memiliki makna baik ataupun buruk yang didalamnya tidak mengandung kata-kata yang positif dan negatif
Contoh : saya dpt sms dari indosat mengenai perubahan paket yellow ttg perubahan harga dan benefit pulsa safe. mohon diinfokan lbh detil. terima kasih
 4. **Konflik** : Sedangkan konflik adalah label yang akan diberikan terhadap opini ketika di dalamnya terdapat 2 maksud yang berlawanan yaitu positif dan negatif.
Contoh : telkomsel nih kmrn2 sinyal ancur bgt T_T sampe gabisa donlot album good luck. skrng sih lumayan cepat, walaupun labil

2.2.3 Word Embeddings

Word embedding adalah Penyisipan kata adalah nama kolektif untuk seperangkat pemodelan bahasa dan teknik pembelajaran fitur dalam pemrosesan bahasa alami (NLP) di mana kata atau frasa dari kosakata dipetakan ke vektor bilangan real. Untuk membandingkan kata, maka perlu mewakili kata-kata dalam bentuk [9]vektor. Kita bisa mewakilinya dalam bentuk simbol-simbol tersendiri. Sehingga dimensi vektor akan sama dengan jumlah total kata dalam kosakata.

2.2.4 Word2vec

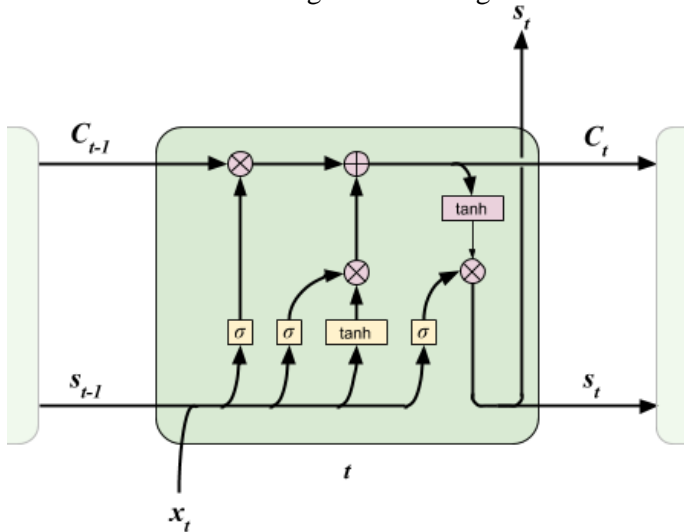
Word2Vec adalah salah satu teknik paling populer untuk mempelajari *word embeddings* menggunakan *Neural Network*. *Word2vec* sendiri dibuat oleh Thomas Micolov pada tahun 2013 di Google [10]. *Word2vec* dapat digunakan dengan 2 metode yang mana keduanya melibatkan neural Network yaitu, Skip

Gram dan Common Bag of Word (CBOW). Metode ini mengambil konteks pada setiap kata sebagai input, kemudian mencoba prediksi kata konteks yang sesuai terhadap kata tersebut. Word2vec dapat merepresentasikan ketergantungan antara satu kata terhadap kata lain. Sehingga ketika sebuah kata akan dicari maka akan di rekomendasikan kata yang memiliki nilai paling dekat terhadap kata input.

2.2.5 Long-Short Term Memory

Long Short-Term Memory Network (LSTM) merupakan jenis khusus dari *Recurrent Neural Network* (RNN), yang mampu mempelajari depensi *long-term*. Metode tersebut diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997, yang kemudian disempurnakan dan dipopulerkan oleh banyak orang. LSTM dirancang secara eksplisit untuk menghindari masalah ketergantungan jangka panjang. Mengingat informasi untuk jangka waktu yang lama adalah perilaku standar mereka, bukan sesuatu yang mereka perjuangkan untuk pelajari [9].

Berikut ini adalah diagram blok bangunan LSTM:



Gambar 2.1 Diagram blok LSTM oleh Benny Prijono [9]

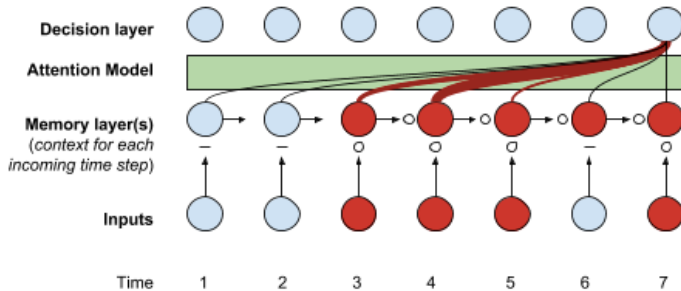


Gambar 2.2 Keterangan diagram blok LSTM oleh Benny Prijono [9]

Pada gambar 2.1 dan gambar 2.2 digambarkan bahwa terdapat tiga *input*. X_t adalah *input* dari langkah waktu saat ini. S_{t-1} adalah *output* dari unit LSTM sebelumnya dan C_{t-1} adalah "memori" dari unit sebelumnya, yang merupakan *input* paling penting. Adapun *output*, S_t adalah *output* dari jaringan saat ini. C_t adalah memori dari unit saat ini [11].

2.2.6 Attention Network

Mekanismenya *attention* dalam *Neural network* berfungsi untuk mengarahkan persepsi serta akses memori (baik yang tersimpan sebelumnya maupun yang tersimpan pada memori jangka pendek). *Attention* akan mem-filter persepsi yang disimpan didalam memori yang mendapat perhatian pada saat itu dan yang sebelumnya telah tersimpan seperti yang ada pada gambar 2.3.

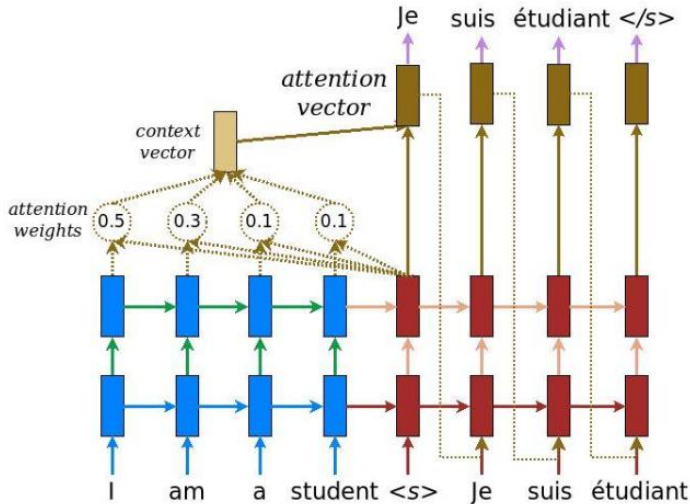


Gambar 2.3 Cara kerja attention oleh Skimind [12]

Mekanisme *attention* digunakan untuk dua tujuan, yang pertama adalah mengurangi beban komputasi dari pemrosesan *input* yang memiliki dimensi tinggi dengan cara hanya memilih subset proses pada input. Yang kedua adalah untuk

memungkinkan sistem fokus pada aspek-aspek berbeda dari input dan dengan demikian akan meningkatkan kemampuannya untuk mengekstrak informasi yang paling relevan untuk setiap bagian dari output, sehingga dapat menghasilkan peningkatan dalam kualitas *output* yang dihasilkan.

Output dari *Attention* merupakan sebuah vektor. Sebelum adanya mekanisme *Attention*, proses dalam menerjemahkan kalimat bergantung pada pembacaan lengkap pada kalimat dan mengompres semua informasi menjadi vektor dengan panjang yang tetap, karena kalimat dengan ratusan kata yang diwakili oleh beberapa kata pasti akan menyebabkan hilangnya informasi, terjemahan yang tidak memadai atau sebagainya.



Gambar 2.4 Arsitektur attention oleh Synced [13]

Mirip dengan arsitektur *encoder-decoder* dasar, mekanisme pada gambar 2.4 memberikan masukkan vektor konteks ke celah antara *encoder* dan *decoder*. Menurut skema di atas, warna biru mewakili *encoder* dan warna merah mewakili *decoder*; dan kita dapat melihat bahwa vektor konteks mengambil semua sel output sebagai input untuk menghitung distribusi probabilitas dari sumber kata-kata dalam sebuah bahasa untuk setiap kata yang ingin dihasilkan oleh *decoder*.

Dengan memanfaatkan mekanisme ini, *decoder* dimungkinkan untuk menangkap informasi yang sedikit global daripada menyimpulkan berdasarkan suatu kondisi tersembunyi [13].

2.2.7 Operator Telekomunikasi

Telekomunikasi merupakan interaksi yang dilakukan oleh dua orang atau lebih pada jarak jauh. Sedangkan operator telekomunikasi merupakan sebuah industri yang menyediakan jasa di bidang telekomunikasi. Di Indonesia sendiri ada banyak penyedia layanan telekomunikasi. Dalam penelitian ini operator telekomunikasi yang akan di gunakan sebagai topik dalam sentimen adalah PT Telekomunikasi Selular (Telkomsel) dan PT Indosat Tbk. (Indosat).

2.2.8 Twitter

Twitter merupakan salah satu media sosial yang populer digunakan oleh penduduk indonesia yang termasuk dalam kategori *microblogging* atau blog singkat, yang mana penggunaanya bisa menuliskan maksimal 280 karakter dalam 1 kali postingan. *Twitter* sendiri merupakan perusahaan yang dirintis oleh Jack Dorsey pada bulan maret 2006 [3]. Pada twitter dapat ditemukan pendapat atau komentar publik terhadap suatu hal, oleh karena itu dataset yang digunakan dalam penelitian ini diambil dari *twitter*.

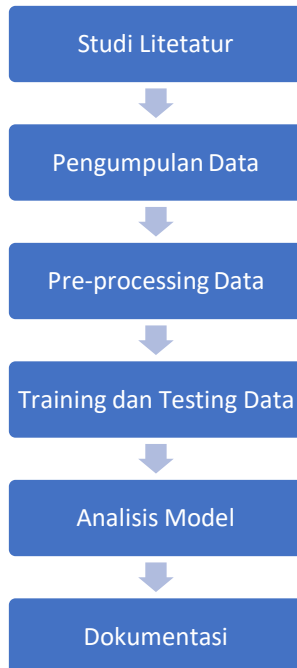
Halaman ini sengaja dikosongkan

BAB III METODOLOGI

Pada bab ini dijelaskan metodologi yang digunakan dalam penelitian dan juga jadwal pengerjaan tugas akhir

3.1 Diagram Metodologi

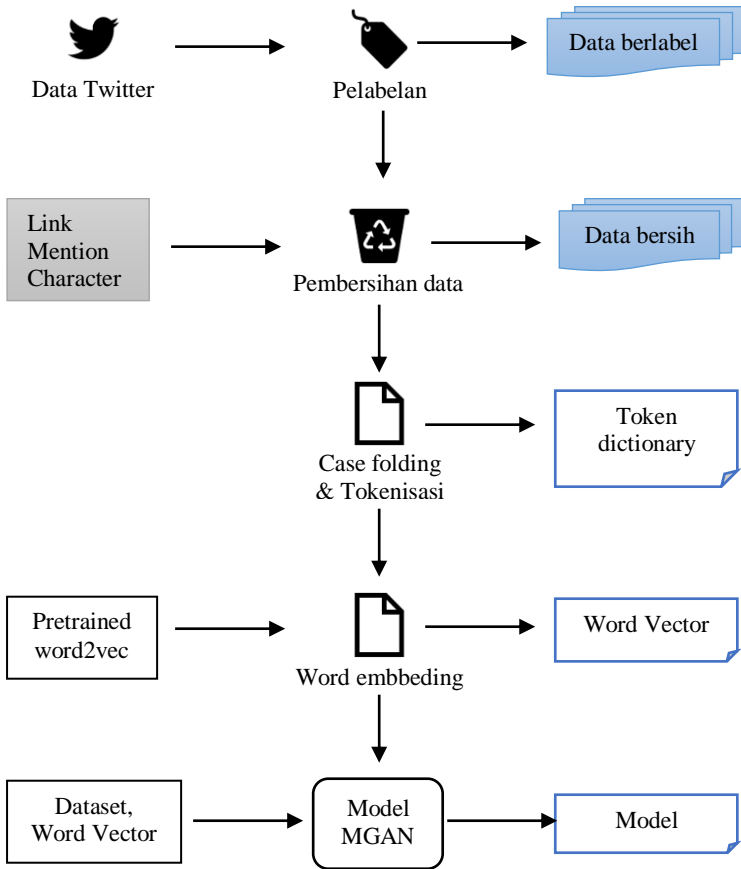
Pada bagian ini akan dijelaskan metodologi dari penelitian yang akan digunakan sebagai panduan agar pengerjaan dari tugas akhir menjadi terarah dan sistematis sehingga dapat berjalan sesuai rencana. Berikut ini merupakan metodologi yang digunakan penulis:



Gambar 3.1 Metodologi

3.2 Arsitektur Penelitian

Pada gambar 3.2 akan dijelaskan mengenai arsitektur penelitian yang digunakan untuk menyusun tugas akhir ini.



Gambar 3.2 Arsitektur Penelitian

3.3 Uraian Metodologi

Berikut merupakan penjelasan dari setiap tahapan yang ada pada metodologi yang digunakan, yaitu:

3.2.1. Studi Literatur

Pada tahap ini dilakukan pembelajaran terhadap ilmu-ilmu yang berkaitan dengan penelitian yang akan dilakukan. Tahap ini dilakukan dengan tujuan untuk memahami konsep metode dan algoritma yang sesuai dengan topik yang akan dilakukan dalam tugas akhir. Adapun literatur utama yang digunakan sebagai pedoman dalam penyusunan tugas akhir yaitu *Multi-grained Attention Network for Aspect-Level Sentiment Classification* [6], *IARM: Inter-Aspect Relation Modelling with Memory Network in Aspect-Based Sentiment Analysis* [7] dan *Recurrent Attention Network Memory for Aspect Sentiment Analysis* [14]

3.2.2. Pengumpulan Data

Data yang akan di gunakan dalam penelitian ini merupakan data yang diambil dari twitter, sehingga pada tahap ini dilakukan pengumpulan data yang diambil/di-*crawl* dari twitter. Dalam melakukan *crawling* data yang sesuai dengan topik yaitu operator telekomunikasi yang mana ada dua operator telekomunikasi yang akan di bandingkan yaitu Telkomsel dan Indosat. Kata kunci yang digunakan untuk pengumpulan data juga harus sesuai yaitu Telkomsel dan Indosat.

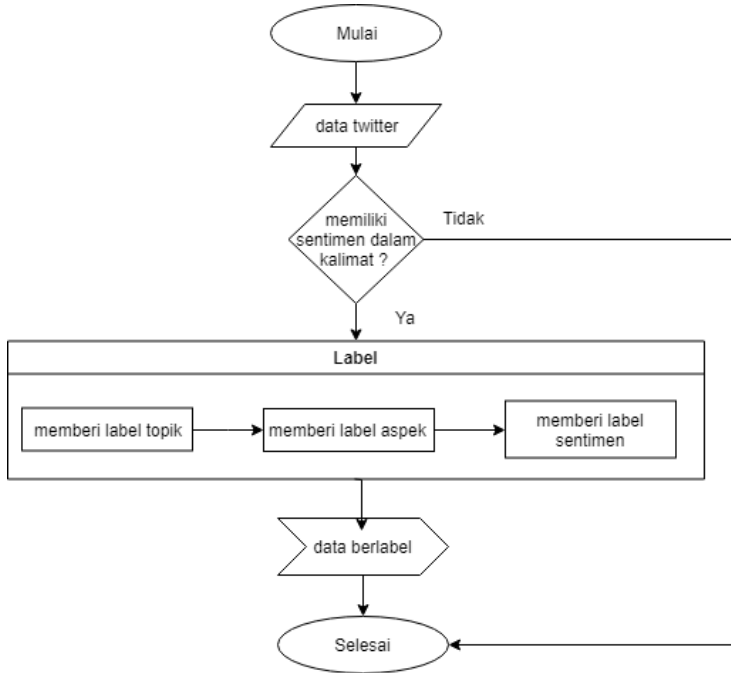
3.2.3. Pre-processing Data

Tahap selanjutnya adalah *pre-processing* data, pada tahap ini dilakukan pemrosesan dari data mentah yang telah didapatkan pada saat *crawling* di tahap sebelumnya sehingga data bisa digunakan sebagai data latih untuk tahap selanjutnya. Berikut adalah tahap tahap *pre-processing* data:

3.2.3.1 Data Tagging

Pada tahap ini dilakukan *data tagging* atau pelabelan data, yaitu memberikan label pada setiap tweet sesuai dengan *terms*. Ada 3 jenis pelabelan yang harus dilakukan yaitu pelabelan topik, pelabelan aspek dan pelabelan sentimen kalimat berdasarkan aspeknya. Label sentimen yang diberikan harus sesuai dengan opini yang ada di dalam kata. Setelah itu pada setiap label diberikan sentimen terhadap topik, sentimen di

bagi menjadi 4 yaitu positif, netral, negatif dan konflik. Proses melakukan pelabelan data dapat dilihat pada gambar 3.3.



Gambar 3.3 Flowchart pelabelan data

3.2.3.2 Pembersihan Data

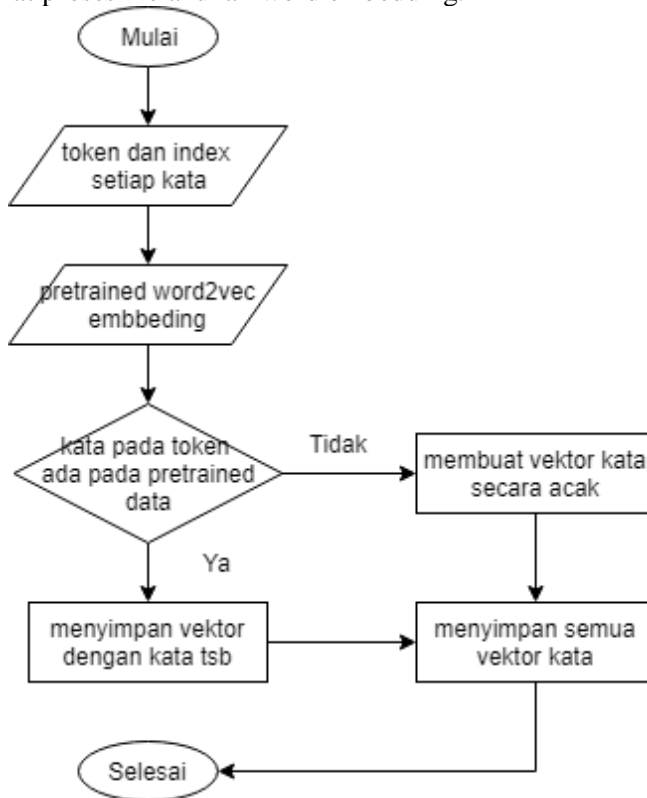
Pada tahap ini dilakukan pembersihan data teks dari link, mention, juga simbol dan karakter. Tujuan dilakukannya pembersihan data adalah agar data yang di input tidak perlu mempertimbangkan hal-hal yang tidak seharusnya diinputkan karena tidak memberikan makna lebih untuk pembuatam model. Seperti link-link url yang mengarah ke situs atau ke halaman lain, mention akun lain dan juga karakter seperti (!@#\$\$%^&*) dan sebagainya.

3.2.3.3 Case folding dan Tokenisasi

Pada tahap ini dilakukan *case folding* atau mengubah huruf menjadi huruf kecil atau *lower case*. Selain itu dilakukan pula tokenisasi atau mengambil setiap kata untuk disimpan di indek yang berbeda setiap katanya.

3.2.3.4 Word Embedding menggunakan Word2vec

Pada tahap ini dilakukan *word embedding* menggunakan library data pretrained word2vec yang gunanya adalah untuk menerjemahkan kata yang akan dimasukkan ke dalam model baik pada saat pelatihan data ataupun pada saat melakukan tes sehingga dapat menghasilkan vektor. Pada Gambar 3.4 dapat dilihat proses melakukan word embedding.



Gambar 3.4 Flowchart word embedding

3.2.4. Training dan Testing Data

Training data dilakukan dengan input data vektor yang dihasilkan dari twitter dengan topik operator telekomunikasi oleh Glove yang kemudian di training menggunakan BiLSTM dan Multi-Grained Attention Network. Kemudian dilakukan pengujian untuk mengetahui *performance* yang dihasilkan oleh model tersebut.

3.2.5. Analisis Model

Tahap selanjutnya adalah Analisis Model, pada tahap ini dilakukan analisis untuk menghitung performa berdasarkan *precision* dan *recall* yang dihasilkan oleh model yang telah dibuat. Pengukuran performa akan dilakukan menggunakan *Accuracy* dan *F1-Score* yang mana nilai yang paling tinggi adalah 1 dan yang paling rendah adalah 0. Berikut ini adalah penjelasan mengenai metode yang akan digunakan dalam pengukuran performa:

- **Relevant** : adalah kondisi ketika hasil yang didapatkan sesuai dengan apa yang sebenarnya, seperti True Positive (TP) dan True Negative(TN)
- **Not Relevant** : adalah kondisi ketika hasil yang diharapkan tidak sesuai dengan apa yang sebenarnya, seperti False Negative(FN) dan False Positive(FP)

Dari pernyataan tersebut berikut adalah perhitungan untuk mendapatkan hasil akurasi dan F1-Score:

- **Precision** : $Precision = \frac{TP}{TP+FP}$

Precision merupakan rasio prediksi benar positif dibandingkan dengan dengan keseluruhan hasil yang di prediksi positif.

- **Recall** : $Recall = \frac{TP}{TP+FN}$

Recall merupakan rasio benar positif yang dibandingkan dengan total data yang benar positif.

- **Accuracy** : $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

Merupakan rasio prediksi Benar (positif dan negatif) dengan keseluruhan data.

- **F1-Score** : $F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$

F1-Score merupakan perbandingan rata-rata presisi dan recall yang dibobotkan [15].

3.2.6. Dokumentasi

Tahap terakhir adalah melakukan dokumentasi sebagai laporan atas terlaksananya penyusunan tugas akhir ini. Dokumentasi dibuat dalam bentuk dokumen yang sudah disesuaikan dengan format yang telah ditentukan. Penyusunan dokumentasi dilakukan secara sistematis sesuai dengan pengerjaan tugas akhir, dari awal hingga berakhirnya proses.

Halaman ini sengaja dikosongkan

BAB IV PERANCANGAN

Pada bab ini diuraikan cara perancangan awal yang diperlukan untuk melakukan pengumpulan data tugas akhir. Bab ini meliputi rancangan penelitian studi kasus dan pembuatan model.

4.1 Perancangan Dataset

Dalam melakukan perancangan yang harus dilakukan pertama kali adalah menyiapkan dataset yang akan digunakan sebagai masukan pada implementasi model *Multi Grained Attention Network*. Dataset yang digunakan merupakan data yang berasal dari twitter yang berkaitan tentang provider Telkomsel dan Indosat. Data tersebut didapatkan dari penelitian sebelumnya yang memiliki jenis data yang sama yaitu tentang operator telekomunikasi.

Dataset Twitter operator telekomunikasi berupa file excel yang nantinya akan diolah terlebih dahulu sehingga dapat digunakan sebagai masukan dengan format csv. Pada tahap ini data dibagi menjadi 2 untuk training dan testing, dengan porsi data training 80% dan data testing 20%.

4.2 Perancangan *Preprocessing Data*

Pada saat *preprocessing* data, dataset akan diolah agar dapat dijadikan sebagai input pada model *Multi Grained Attention Network*. Ada beberapa tahap yang harus dilakukan untuk *preprocessing* data. Berikut ini adalah tahap-tahap *preprocessing*:

4.2.1 Pemberian Label Data

Dataset yang telah didapatkan hanya berisi satu jenis data teks dengan kalimat yang diambil dari *twitter*, yang kemudian data tersebut akan diberi label. Label yang diberikan pada setiap *tweet* bertujuan untuk memberikan pendapat sentimen yang sesuai terhadap suatu aspek pada *tweet*, sehingga label tersebut

dapat dipelajari oleh model pada saat pelatihan data. Label yang akan di berikan akan dibagi menjadi 3 pengelompokan yaitu:

1. Topik

Topik merupakan jenis provider yang terdapat pada kalimat tersebut. Berdasarkan penelitian ini data yang yang diambil hanyalah data yang memiliki topik Telkomsel dan Indosat saja. contoh pelabelan topik dapat dilihat pada Tabel 4.1.

Tabel 4.1 Label Topik

No	Teks	Topik
1	Mantap nih indosat sekarang ngebut internet nya	INDOSAT
2	@lewatmana hush jangan ngeflame. Saya pengguna telkomsel, sangat puas tuh dgn jaringan / service nya.	TELKOMSEL

2. Aspek

Aspek yang akan diterapkan pada penelitian ini menyesuaikan pada aspek kata atau *terms* yang memiliki sentimen pada kalimat. Sehingga untuk aspek tidak dapat ditentukan jumlahnya di awal pengerjaan pemberian label pada dataset.

Cara pemberian label aspek selain diambil berdasarkan *terms*, juga harus memilih objek dalam kalimat yang memiliki sentimen. Jika objek merujuk pada 2 hal contohnya ‘harga paketannya mahal’ maka kata yang diambil adalah kata pertama yaitu harga. Contoh pelabelan aspek pada teks dapat dilihat pada Tabel 4.2.

Tabel 4.2 Label Aspek

No	Teks	Aspek
1	Membalas @aciel @Telkomsel Bikin nangis emang harga paketannya ?	Harga

2	Indosat dh lumayan bagus jaringan internetnya, tpi mSh putus2. Browsingnya, mudah2an bisa lebih baik lagi	Jaringan
3	Membalas @Fauzan_bondos19 @Fauzan_bondos19 tergantung lokasi gan kalo sinyal indosat disana bagus kecepatan lumayan	Sinyal, kecepatan
4	Sekarang udah jamannya 4G LTE, udah ngga ada lagi yang namanya lemot. Kata temen 4G LTEnya Telkomsel dahsyat banget .. Eh pas dites, bener!	4g

3. Sentimen

Pada penelitian ini ada 4 jenis label sentimen yang akan diberikan. Pada Tabel 4.3 yang akan menjelaskan angka-angka yang akan digunakan sebagai label sentimen.

Tabel 4.3 Label Sentimen

Angka	Sentimen	Penjelasan
-1	Negatif	Sentimen negatif adalah pendapat yang memiliki makna yang buruk terhadap aspek
0	Netral	Sentimen netral adalah pendapat yang tidak memiliki makna yang baik maupun buruk terhadap aspek
1	Positif	Sentimen positif adalah pendapat yang memiliki

		makna yang baik terhadap aspek
2	Konflik	Sentimen konflik adalah pendapat yang memiliki makna yang baik sekaligus buruk terhadap aspek yang sama

4.2.2 Pembersihan Dataset

Dataset pada bagian teks *twitter* yang telah diberi label masih banyak memuat karakter dan simbol yang tidak terlalu diperlukan dalam proses pelatihan data, sehingga karakter dan simbol yang masih terdapat pada *tweets* harus dibersihkan. Selain itu ada pula pembersihan yang harus dilakukan untuk mengubah format, seperti menghapus tagar pada *tweet*, penghapusan akun yang di mention dalam *tweet* dan hal lain yang akan di tunjukkan pada poin-poin berikut:

- Menghapus mention akun twitter yang diawali dengan '@' kemudian diikuti kata dibelakangnya.
- Menghapus link URL dalam *tweet* yang diawali dengan 'https://' lalu diikuti kata selanjutnya.
- Menghapus tag atau tanda '#' tagar pada *tweet*
- Menghapus karakter dan simbol pada *tweet*.

4.2.3 Case folding dan Tokenisasi

Dari data yang sudah bersih dari tanda baca atau karakter karakter lain dilakukan *case folding* atau *me-lower case* semua kalimat yang memiliki huruf kapital didalamnya. Setelah itu setiap kalimat tersebut akan ditokenisasi untuk mengambil setiap kata yang berbeda dari semua *tweet*. Kata tersebut akan disimpan dalam dan memiliki index masing-masing. Selanjutnya kata yang telah ditokenisasi akan digunakan untuk *training* model

4.2.4 Word Embedding

Word embedding digunakan untuk mengubah kata pada *tweet* yang telah ditokenisasi yang telah dimiliki menjadi

matriks sehingga data dapat digunakan sebagai masukan dalam model. Untuk penelitian ini menggunakan data *pretrained word embedding* menggunakan metode *word2vec* yang telah di buat pada penelitian sebelumnya. Data *word embedding* tersebut memiliki dimensi yang berukuran 300.

4.3 Perancangan Model MGAN

Pada tahap ini dilakukan perancangan untuk membuat model MGAN. Perancangan model dilakukan dengan 2 tahap yaitu melalui training dan testing. Data yang digunakan sebagai masukan juga berbeda seperti yang sudah dijelaskan sebelumnya, yaitu 80% dari total data sebagai data training dan sisanya merupakan data testing.

4.3.1 Perancangan Training Model MGAN

Perancangan training model merupakan tahap dimana akan menghasilkan model MGAN. Dalam penerapannya terdapat parameter yang harus ditentukan untuk mendapatkan hasil yang terbaik. *Library python* yang digunakan untuk melakukan analisis sentimen menggunakan algoritma MGAN adalah *pytorch*. *Library* tersebut memiliki keunggulan untuk mengemat waktu atau durasi dalam melakukan *training* karena dapat dijalankan di GPU. Input utama yang dibutuhkan untuk menjalankan model yaitu data *pretrained word embedding* dan *dataset train*. Parameter yang akan digunakan pada model MGAN dapat dilihat pada Tabel 4.4.

Tabel 4.4 Parameter Model MGAN

No	Paremeter	Definisi
1	Embed_dim	Ukuran dimensi vektor kata dari word
2	Hidden_dim	Jumlah fitur dalam hidden state h
3	Polarites_dim	Jumlah dimensi pada polaritas sentimen

4	Input_size	Jumlah fitur yang diharapkan dalam input x
5	Num_layer	Jumlah pada recurrent layer
6	Bias	parameter yang memiliki value true atau false, sebagai penentu layer menggunakan bobot bias atau tidak
7	Batch_first	Menyediakan input dan output tensor dalam bentuk (batch, seq, feature)
8	Dropout	memberikan layer dropout pada output dari setiap layer RNN kecuali layer terakhir
9	Bidirectional	Parameter yang memiliki value true atau false, untuk menentukan penggunaan bidirectional pada RNN

Selain parameter model diatas, terdapat pula parameter yang dibutuhkan dalam proses *training*. Parameter *training* yang akan digunakan terdapat pada Tabel 4.5.

Tabel 4.5 Parameter Training Model

No	Parameter	Deskripsi
1	Num_epoch	Jumlah epoch yang dalam training model
2	Batch_size	Jumlah sampel data yang disebarkan ke model MGAN

3	Repeats	Jumlah iterasi training
4	Optimizer	Algoritma optimasi yang digunakan untuk <i>training</i>

4.3.2 Perancangan Testing Model MGAN

Pada tahap ini berfokus pada perancangan metode yang digunakan untuk menguji kualitas model yang akan dihasilkan, sehingga bisa mendapatkan model terbaik. Data yang digunakan adalah *dataset testing*. Pengukuran kualitas pengujian didapatkan melalui perhitungan akurasi dan *F1-score*.

4.4 Perancangan Demo Prediksi

Model yang telah dihasilkan ketika dilakukan training dan testing data akan digunakan untuk memprediksi sentimen yang diberikan terhadap kalimat masukkan. Kalimat tersebut dapat diketikkan secara manual sehingga dapat disesuaikan dengan keinginan. Parameter yang dibutuhkan pada tahap ini adalah sebuah kalimat yang mengandung sentimen dan aspek didalamnya. Selain itu diperlukan juga hasil output dari tokenizer, embedding word2vec dan model tertinggi.

Halaman ini sengaja dikosongkan

BAB V IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai proses implementasi dari perancangan sesuai dengan metode pengembangan yang telah dilakukan sebelumnya.

5.1 Lingkungan Implementasi

Penelitian ini menggunakan perangkat keras dan perangkat lunak guna mendukung dalam proses pengerjaannya. Untuk spesifikasi perangkat keras yang digunakan dapat dilihat pada Tabel 5.1.

Tabel 5.1 Spesifikasi perangkat keras

Nama Perangkat	Komputer
Prosesor	Intel i5 8400
<i>Memory</i>	16 GB DDR4 <i>Memory</i>
GPU	GTX 1070 8 GB
Sistem Operasi	Linux Ubuntu 16.10
Arsitektur Sistem	64 Bit

Selain itu juga terdapat perangkat lunak, library dan bahasa pemrograman yang digunakan dalam penelitian ini. Daftar perangkat lunak dapat dilihat pada Tabel 5.2.

Tabel 5.2 Spesifikasi perangkat lunak

Bahasa Pemrograman	Python 3.6
Editor	<ul style="list-style-type: none">• Notepad++• Atom
Virtual Environment	Anaconda
Library	<ul style="list-style-type: none">• Torch• Numpy

	<ul style="list-style-type: none">• Pickle• Os• re• String• Csv• Tqdm• Math• Sklearn• tensorboardX• argparse• glob• datetime• matplotlib
--	--

5.2 Preprocessing Data

Tahap pertama yang harus dilakukan dari data mentah yang didapatkan dari dataset training csv dan data testing csv adalah mem-preprocess data agar data tersebut dapat masuk kedalam model sesuai dengan input yang dibutuhkan. Sehingga dapat digunakan sesuai kebutuhan.

5.2.1 Pemberian Label Data

Dari dataset yang sudah didapatkan masih berisi teks dari twitter saja. Data tersebut harus diberi label terlebih dahulu sebelum digunakan ke tahap selanjutnya. Proses pelabelan data dilakukan secara manual pada 2000 baris tweet. Dan disimpan dengan format csv agar dapat di masukkan kedalam kode.

5.2.2 Pembersihan Data

Data tweet yang ada pada file csv masih banyak mengandung tanda baca dan link yang kurang diperlukan sehingga dilakukan pembersihan data agar data dapat di gunakan secara efektif dan memberikan hasil model yang optimal.

```

1. text = ''
2. for fname in fnames:
3.     with open(fname, 'r', encoding="utf8") as file:
4.         reader = DictReader(file, delimiter=';')
5.         for row in reader:
6.             if row['ASPECT'] != '':
7.                 text_raw = row['TWEET']
8.                 text_raw = re.sub(r'@[A-Za-z0-9]+', '',
text_raw)
9.                 text_raw = re.sub(r'https?://[A-Za-z0-
9./]+', '', text_raw)
10.                text_raw = re.sub('['+string.punctuation
+' ]', "", text_raw)
11.                text += text_raw + " "

```

Kode 5.1 Pembersihan data

Pada Kode 5.1 menunjukkan proses untuk membersihkan dataset, baris ke 3 *with open(fname, 'r', encoding="utf8")* digunakan untuk membaca data csv yang direktori dan nama filenya sudah tersimpan pada variabel *fname*. Baris ke 4 digunakan untuk mengambil setiap baris data dan kolom yang dipisahkan menggunakan delimiter *semicolon* (;). Karena dalam setiap tweet mungkin untuk tidak memiliki label aspek maka, baris kode *if row['ASPECT'] != ''* pada baris ke 6 digunakan untuk memilih baris data yang telah diberi label saja. Sedangkan fungsi pembersihan berada pada kode baris ke 8 yaitu *r'@[A-Za-z0-9]+'* yang bertujuan untuk menghapus *mention* pada *tweet*, kode baris ke 9, *r'https?://[A-Za-z0-9./]+'* bertujuan untuk menghapus *link url* dan kode baris ke 10 *'[+string.punctuation+]'* bertujuan untuk menghapus semua karakter dan simbol yang ada pada *tweet*.

5.2.3 Case Folding dan Tokenisasi

Setelah data tersebut sudah bersih dari karakter dan *link url* selanjutnya data akan di-*case fold* dan ditokenisasi.

```

1. class Tokenizer(object):
2.     def __init__(self, max_seq_len, lower=True):
3.         self.lower = lower
4.         self.max_seq_len = max_seq_len

```

```

5.     self.word2idx = {}
6.     self.idx2word = {}
7.     self.idx = 1
8.
9.     def fit_on_text(self, text):
10.        if self.lower:
11.            text = text.lower()
12.            words = text.split()
13.            for word in words:
14.                if word not in self.word2idx:
15.                    self.word2idx[word] = self.idx
16.                    self.idx2word[self.idx] = word
17.                    self.idx += 1
18.            return self.word2idx

```

Kode 5.2 *Case folding* dan tokenisasi

Kode 5.2 menunjukkan sebuah *class* yang digunakan untuk melakukan tokenisasi. *Class* tersebut memiliki 2 method yaitu *method* `__init__` pada baris ke 2 dengan dengan 3 parameter didalamnya dan *method* `fit_on_text` pada baris ke 9 dengan 2 parameter. Pada *method* `__init__` terdapat 5 variabel yang mana variable `word2idx` pada baris ke 5 digunakan untuk menyimpan token dengan format *key* berisi teks dan *value*-nya berisi indeks. Sebaliknya variabel `idx2word` pada baris ke 6 digunakan untuk meyimpan token dengan *key* berisi indeks dan *value* berisi teks.

Pada *method* `fit_on_text` digunakan untuk proses *case folding* dan tokenisasi. Pada baris ke 11 `text = text.lower()` berfungsi sebagai *lower case* teks. Kemudian pada baris ke 12 teks tersebut di-*split* dan disimpan setiap kata kedalam variabel `word`. Setelah itu di baris ke 14 setiap kata di cek apakah kata tersebut sudah tersimpan pada variabel `word2idx` jika belum maka kata akan disimpan sebagai token, namun jika kata sudah ada maka akan dilanjutkan ke kata setelahnya sampai data terakhir pada *input*.

```

1. tokenizer = Tokenizer(max_seq_len)
2. tokenizer.fit_on_text(text)
3. pickle.dump(tokenizer, open(dat_fname, 'wb'))

```

Kode 5.3 Menyimpan hasil tokenisasi

Baris ke 1 dan baris ke 2 dari Kode 5.3 digunakan untuk menginisiasi dan menggunakan class tokenizer. Sedangkan pada kode baris ke 3 `pickle.dump(tokenizer, open(dat_fname, 'wb'))` tujuannya adalah untuk menyimpan hasil tokenisasi kedalam sebuah file dengan nama yang sudah dideklarasikan pada variabel `dat_fname`.

5.2.4 Word Embedding

Pada tahap ini dilakukan *word embedding* yaitu mengubah kata menjadi matriks. Data *pretrained word2vec* harus sudah disiapkan sebelumnya, data tersebut adalah `modelw2v.bin` hasil *training word embedding* pada penelitian sebelumnya yang sudah menggunakan data berbahasa indonesia. Untuk menggunakan data hasil *training* tersebut harus menggunakan *library gensim*.

```

1. from gensim.models.keyedvectors import KeyedVectors
2. def build_w2v_embedding_matrix(word2idx, embed_dim,
   dat_fname):
3.     print('loading word vectors...')
4.     embedding_matrix = np.zeros((len(word2idx) + 2, em
   bed_dim))
5.     embeddings_path = 'modelw2v.bin'
6.     word2vec_model = KeyedVectors.load_word2vec_format
   (embeddings_path, binary=True, unicode_errors = 'ign
   ore')
```

Kode 5.4 Loading file word2vec embedding

Pada Kode 5.4 merupakan sebuah *method* yang bertujuan untuk me-load hasil *word embedding* dari *file* yang sudah di deklarasikan. Pada baris ke 2 menjelaskan bahwa parameter yang dibutuhkan ada 3 yaitu `word2idx`, memuat token yang didapatkan di tahap sebelumnya. Kemudian `embed_dim` yaitu ukuran dimensi dari *embedding*. Terakhir `dat_fname` yang memiliki value string nama file yang nanti akan digunakan untuk meyimpan hasil *embedding*. Kode pada baris ke 6 `KeyedVectors.load_word2vec_format(embeddings_path, binary=True, unicode_errors='ignore')` berfungsi untuk membaca file *word embedding*.

```

1. word_vec = {}
2.   known_word=0
3.   unknown_word=0
4.   for word, i in tqdm(word2idx.items()):
5.       if word in word2vec_model.wv.vocab:
6.           embedding_matrix[i] = word2vec_model.wv.word_vec(word)
7.           known_word+=1
8.       else :
9.           vec_buatan = np.random.uniform(-0.25, 0.25,
10.            embed_dim).astype('float32')
11.           embedding_matrix[i] = vec_buatan
12.           unknown_word+=1
13.   print(known_word, unknown_word)
14.   pickle.dump(embedding_matrix, open(dat_fname, 'wb'))
15.   return embedding_matrix

```

Kode 5.5 *Embedding* token

Baris *word in word2vec_model.wv.vocab* pada Kode 5.5 digunakan untuk memastikan setiap kata pada token ada pada *vocab* di *word embedding*, jika ada maka matriks pada *word* akan disimpan kedalam variabel *embedding_matrix*. Jika *word* tidak ditemukan pada *vocab* maka *embedding_matrix* akan diisi dengan *random matrix*. Setiap kata akan dihitung total token yang sudah digunakan dan dimasukkan pada variabel *known_word* dan *unknown_word* yang ada pada baris kode ke 2 dan 11 guna menghitung berapa jumlah kata yang dapat diketahui dan yang tidak diketahui.

5.3 Pembuatan Model MGAN

Pada tahap ini semua data hasil dari tokenisasi dan *embedding* akan digunakan baik untuk *training* maupun *testing* dalam pembuatan model *Multi Graind Attention Network*.

```

1. trainset = ABSADataset(opt.dataset_file['train'], self.lf.tokenizer)
2. testset = ABSADataset(opt.dataset_file['test'], self.lf.tokenizer)

```

```

3. self.train_data_loader = DataLoader(dataset=trainset
   , batch_size=opt.batch_size, shuffle=True)
4. self.test_data_loader = DataLoader(dataset=testset,
   batch_size=opt.batch_size, shuffle=False)

```

Kode 5.6 Persiapan data

Pada Kode 5.6 baris kode pertama dan kedua digunakan untuk mempersiapkan data *train* dan data tes untuk di tokenisasi setiap *field* data dan disimpan masing masing teks, aspek dan polaritasnya ke dalam *list*. Sedangkan *DataLoader* pada baris kode ke 3 dan 4 bertujuan untuk mengambil data dalam *batch* agar proses pembacaan data bisa lebih cepat. *Batch_size* yang didefinisikan pada kode tersebut adalah 50.

```

1. opt.device = torch.device('cuda' if torch.cuda.is_av
   ailable() else 'cpu')
2. self.model = opt.model_class(self.embedding_matrix,
   opt).to(opt.device)

```

Kode 5.7 Inisiasi model

Kode 5.7 baris pertama menjelaskan bahwa perangkat yang digunakan sudah ter-*install cuda* maka *device* yang digunakan adalah *cuda* namun jika tidak maka *device* akan menggunakan *cpu*. Baris kode kedua *opt.model_class(self.embedding_matrix, opt)* merupakan inisiasi dari model MGAN dengan parameter *embedding_matrix* dan *device*.

5.3.1 Inisiasi Argumen

Dalam pembuatan model akan banyak dibutuhkan argumen yang perlu di inisiasi agar mudah dalam menggunakan argumen tersebut dan mudah untuk dilakukan perubahan jika dibutuhkan. Argumen tersebut dapat dilihat pada Kode 5.8.

```

1. if __name__ == '__main__':
2.     parser = argparse.ArgumentParser()
3.     parser.add_argument('--model_name',
   default='mgan', type=str)

```

```

4.     parser.add_argument('--dataset',
5.     default='twitter', type=str, help='twitter')
6.     parser.add_argument('--optimizer',
7.     default='adam', type=str)
8.     parser.add_argument('--initializer',
9.     default='xavier_uniform_', type=str)
10.    parser.add_argument('--learning_rate',
11.    default=0.001, type=float)
12.    parser.add_argument('--dropout', default=0,
13.    type=float)
14.    parser.add_argument('--l2reg', default=0.00001,
15.    type=float)
16.    parser.add_argument('--num_epoch', default=20,
17.    type=int)
18.    parser.add_argument('--batch_size', default=50,
19.    type=int)
20.    parser.add_argument('--log_step', default=5,
21.    type=int)
22.    parser.add_argument('--logdir', default='log',
23.    type=str)
24.    parser.add_argument('--embed_dim', default=300,
25.    type=int)
26.    parser.add_argument('--hidden_dim', default=300,
27.    type=int)
28.    parser.add_argument('--max_seq_len', default=80,
29.    type=int)
30.    parser.add_argument('--polarities_dim',
31.    default=4, type=int)
32.    parser.add_argument('--hops', default=3,
33.    type=int)
34.    parser.add_argument('--device', default=None,
35.    type=str)
36.    opt = parser.parse_args()

```

Kode 5.8 Inisiasi argumen

Argumen yang di inisiasi pada kode 5.8 baris ke 3 adalah *model_name* yaitu nama *class* dari model yang akan digunakan, pada baris ke 4 argumen *dataset* berisi alamat direktori dari data *train* dan data tes, pada baris ke 5 *optimizer* yang digunakan adalah *adam*, *initializer*, *learning_rate*, *dropout*, *l2reg*, *num_epoch* sebanyak 20, *batch_size* sebanyak 50, *log_step* 5, *logdir*, *embed_dim* 300, *hidden_dim* 300 dan argumen lain beserta *value*-nya.

5.3.2 Pembuatan *Class* MGAN

MGAN ini merupakan penelitian yang sebelumnya telah dilakukan oleh Feifan Fan dkk. Hasil dari embedding dan dataset dimasukkan ke dalam class ini. Pembuatan model MGAN ditunjukkan pada Kode 5.9. Kode tersebut digunakan untuk menginisiasi variabel variabel yang akan digunakan pada class.

```

1. class MGAN(nn.Module):
2.     def __init__(self, embedding_matrix, opt):
3.         super(MGAN, self).__init__()
4.         self.opt = opt
5.         self.embed = nn.Embedding.from_pretrained(torch.
        tensor(embedding_matrix, dtype=torch.float))
6.         self.ctx_lstm = DynamicLSTM(opt.embed_dim, opt.h
        idden_dim, num_layers=1, batch_first=True, bidirecti
        onal=True)
7.         self.asp_lstm = DynamicLSTM(opt.embed_dim, opt.h
        idden_dim, num_layers=1, batch_first=True, bidirecti
        onal=True)
8.         self.location = LocationEncoding(opt)
9.         self.w_a2c = nn.Parameter(torch.Tensor(2*opt.hid
        den_dim, 2*opt.hidden_dim))
10.        self.w_c2a = nn.Parameter(torch.Tensor(2*opt.hid
        den_dim, 2*opt.hidden_dim))
11.        self.alignment = AlignmentMatrix(opt)
12.        self.dense = nn.Linear(8*opt.hidden_dim, opt.pol
        arities_dim)

```

Kode 5.9 Inisialisasi model MGAN

Untuk forward dari *input text* yang sebelumnya sudah dipisah pada index list yang berbeda dimasukkan kedalam variabel. Seperti pada Kode 5.10

```

1. def forward(self, inputs):
2.     text_raw_indices = inputs[0]
3.     aspect_indices = inputs[1]
4.     text_left_indices= inputs[2]
5.     batch_size = text_raw_indices.size(0)
6.     ctx_len = torch.sum(text_raw_indices != 0, dim=1
    )

```

```

7.     asp_len = torch.sum(aspect_indices != 0, dim=1)
8.     left_len = torch.sum(text_left_indices != 0, dim
    =-1)
9.     aspect_in_text = torch.cat([left_len.unsqueeze(-
    1), (left_len+asp_len-1).unsqueeze(-1)], dim=-1)
10.
11.     ctx = self.embed(text_raw_indices)
12.     asp = self.embed(aspect_indices)
13.
14.     ctx_out, (_, _) = self.ctx_lstm(ctx, ctx_len)
15.     ctx_out = self.location(ctx_out, aspect_in_text)
16.     ctx_pool = torch.sum(ctx_out, dim=1)
17.     ctx_pool = torch.div(ctx_pool, ctx_len.float().u
    nsqueeze(-1)).unsqueeze(-1)
18.     asp_out, (_, _) = self.asp_lstm(asp, asp_len)
19.     asp_pool = torch.sum(asp_out, dim=1)
20.     asp_pool = torch.div(asp_pool, asp_len.float().u
    nsqueeze(-1)).unsqueeze(-1)

```

Kode 5.10 Foward MGAN

5.3.3 Pembuatan *Training* model MGAN

Model mulai dijalankan ketika *CrossEntropyLoss()* pada baris ke pertama dijalankan untuk mendapatkan parameter. Pada Kode 5.11 baris ke 3 yaitu *optimizer* digunakan sebagai *input* pada *method train*.

```

1. criterion = nn.CrossEntropyLoss()
2. _params = filter(lambda p: p.requires_grad, self.mod
    el.parameters())
3. optimizer = self.opt.optimizer(_params, lr=self.opt.
    learning_rate, weight_decay=self.opt.l2reg)

```

Kode 5.11 Inisiasi CrossEntropyLoss

Pada *method train()*, *training* diulang sebanyak jumlah *epoch* yang sudah di inisialisasikan menggunakan kode pada baris pertama. Parameter yang dimasukkan pada model yaitu *train_data_loader* seerti yang ada pada baris kode ke 5. Kode

5.12 berikut merupakan kode ketika *input* disesuaikan dan dimasukkan kedalam model.

```

1. for epoch in range(self.opt.num_epoch):
2.     print('>' * 100)
3.     print('epoch: ', epoch)
4.     n_correct, n_total = 0, 0
5.     for i_batch, sample_batched in enumerate(self.t
rain_data_loader):
6.         global_step += 1
7.         self.model.train()
8.         optimizer.zero_grad()
9.
10.        inputs = [sample_batched[col].to(self.opt.dev
ice) for col in self.opt.inputs_cols]
11.        outputs = self.model(inputs)
12.        targets = sample_batched['polarity'].to(self.
opt.device)
13.
14.        loss = criterion(outputs, targets)
15.        loss.backward()
16.        optimizer.step()

```

Kode 5.12 Training model

Pada baris kode ke 7 *self.model.train()* dipanggil agar model berjalan dengan mode *train*. Variabel *inputs* pada baris ke 10 berisi *list* dari *train_data_loader* yang dipisahkan sesuai dengan pemecahan data pada saat preproses yaitu *text_raw_indices*, *aspect_indices* dan *text_left_indices*. Dari baris kode ke 11 *outputs = self.model(inputs)* akan menghasilkan nilai dari model. Sedangkan variabel *targets* di baris ke 12 digunakan untuk menyimpan polaritas sentimen yang benar. Sehingga didapatkan nilai *loss* dari *outputs* dan *targets* berdasarkan baris kode ke 14.

```

1. n_correct += (torch.argmax(outputs, -1) == targets
).sum().item()
2. n_total += len(outputs)
3. train_acc = n_correct / n_total

```

Kode 5.13 Penghitungan akurasi training

Setelah mendapatkan nilai *outputs*, nilai tersebut di translasikan agar menjadi sentimen menggunakan baris kode pertama `torch.argmax(outputs, -1)` seperti yang dapat dilihat pada Kode 5.13. Hasilnya dibandingkan dengan *targets* apakah nilainya benar atau tidak. Nilai benar akan dijumlah pada variabel *n_correct* yang akan digunakan sebagai nilai dalam menghitung akurasi *train*. Pada baris kode ke 2 akan dihitung jumlah total data yang di *train*. Kemudian di baris ke 3 memnunjukkan perhitungan akurasi berdasarkan jumlah benar dibandingkan jumlah data.

5.3.4 Pembuatan *Testing* model MGAN

Model yang didapatkan dari hasil training digunakan untuk testing dengan menggunakan kode berikut:

```

1. self.model.eval()
2. n_test_correct, n_test_total = 0, 0
3. t_targets_all, t_outputs_all = None, None
4. with torch.no_grad():
5.     for t_batch, t_sample_batched in enumerate(self.te
st_data_loader):
6.         t_inputs = [t_sample_batched[col].to(opt.device)
for col in self.opt.inputs_cols]
7.         t_targets = t_sample_batched['polarity'].to(opt.
device)
8.         t_outputs = self.model(t_inputs)
9.
10.        n_test_correct += (torch.argmax(t_outputs, -1)
== t_targets).sum().item()
11.        n_test_total += len(t_outputs)
12.
13.        if t_targets_all is None:
14.            t_targets_all = t_targets
15.            t_outputs_all = t_outputs
16.        else:
17.            t_targets_all = torch.cat((t_targets_all, t_ta
rgets), dim=0)
18.            t_outputs_all = torch.cat((t_outputs_all, t ou
tputs), dim=0)
19.
20. test_acc = n_test_correct / n_test_total

```

```
21. f1 = metrics.f1_score(t_targets_all.cpu(), torch.argmax(
    max(t_outputs_all, -1).cpu(), labels=[0, 1, 2],
    average='macro')
```

Kode 5.14 Testing model

Pada Kode 5.14 baris `self.model.eval()` pada baris pertama digunakan agar model berjalan dengan mode *eval*. Seperti yang ada saat training, di baris kode ke 5 variabel *inputs* berisi *list* dari *test_data_loader* yang dipisahkan sesuai dengan pemecahan data pada saat preproses. Pada baris kode ke 7 *t_targets* digunakan untuk menyimpan polaritas sentimen pada data tes. Sampai dihasilkannya output dari model yang didapatkan. *Output* tersebut akan di akumulasi menjadi akurasi pada baris ke 20. Pada baris kode ke 21, `f1 = metrics.f1_score(t_targets_all.cpu(), torch.argmax(t_outputs_all, -1).cpu(), labels = [0, 1, 2], average='macro')` berfungsi untuk menghitung hasil dari F1_score. Selanjutnya hasil akurasi train, akurasi tes dan hasil F1_score pada setiap batch di cari dan di simpan nilai tertinggi pada sebuah variabel.

```
1. if test_acc > max_test_acc:
2.     max_test_acc = test_acc
3.     if test_acc > max_test_acc_overall:
4.         if not os.path.exists('state_dict'):
5.             os.mkdir('state_dict')
6.             path = 'state_dict/{0}_{1}_acc{2}'.format(self.
pt.model_name, self.opt.dataset, round(test_acc, 4))
7.             torch.save(self.model.state_dict(), path)
8.             print('>> saved: ' + path)
9.         if f1 > max_f1:
10.            max_f1 = f1
```

Kode 5.15 Menyimpan model

Pada Kode 5.15 baris pertama yaitu `if test_acc > max_test_acc` akan mencari nilai paling maksimal setiap kali melakukan test. Model yang menghasilkan nilai paling tinggi kemudian akan disimpan dengan menggunakan kode pada baris ke 7 yang berisi `torch.save(self.model.state_dict(), path)`. Dengan memastikan bahwa model dengan nilai akurasi yang

sama belum ada pada direktori. Model tersebut disimpan agar dapat digunakan untuk demo prediksi.

Selain menyimpan model, hasil akurasi train, tes dan F1_score juga disimpan sebagai teks. Dengan menyimpan hasil akurasi tersebut akan tercatat nilai nilai berapa saja yang dapat di hasilkan oleh model.

```

1. writer.add_scalar('loss', loss, global_step)
2. writer.add_scalar('acc', train_acc, global_step)
3. writer.add_scalar('test_acc', test_acc, global_step)
4. csvData.append([epoch,global_step,loss.item(), train
   _acc, test_acc, f1, ',','])
5.
6. csv.register_dialect('myResult',delimiter = ';', quo
   ting=csv.QUOTE_NONE, skipinitialspace=False)
7. dateTimeObj = datetime.now()
8. timestampStr = dateTimeObj.strftime("%Y%b%d-
   %H.%M.%S")
9. if not os.path.exists('acc_result'):
10. os.mkdir('acc_result')
11. with open('acc_result/hasil'+timestampStr+'.csv', 'w
   ') as csvFile:
12. writer1 = csv.writer(csvFile, dialect='myResult')
13. writer1.writerows(csvData)

```

Kode 5.16 Menyimpan hasil ke csv

Berdasarkan Kode 5.16, hasil akurasi disimpan kedalam sebuah file csv. File tersebut akan di create setiap kali ada repeat *train* dan setiap file memuat akurasi setiap *batch* pada 20 *epoch*. Untuk kode `dateTimeObj = datetime.now()` bertujuan untuk menginisiasi tanggal saat ini. Sedangkan pada baris ke 7 `dateTimeObj.strftime ("%Y%b%d-%H.%M.%S")` bertujuan untuk memformat tanggal dan jam sesuai urutan tahun, bulan, tanggal, jam, menit dan detik kedalam variabel `timestampStr` seperti pada baris kode ke 8. Agar lebih mudah di mengerti maka penyimpanan file diberikan nama menggunakan variabel tersebut.

Selain disimpan dalam bentuk teks, akurasi juga perlu disimpan dan ditampilkan secara visual atau dengan bentuk

grafik agar mudah untuk melihat jika ada peningkatan atau penurunan. Berikut adalah kode untuk membuat grafik:

```
1. plt.plot(range(1,totalprint),list_accuracy,marker=' ',
, color='olive', linewidth=2)
2. plt.plot(range(1,totalprint),list_accuracy_test,mark
er=' ', color='blue', linewidth=2)
3. plt.plot(range(1,totalprint),list_f1,marker=' ', colo
r='yellow', linewidth=2)
4. plt.savefig('grafik/grafik'+timestampStr+'.png')
5. plt.clf()
```

Kode 5.17 Menyimpan grafik

Pada Kode 5.17 baris ke 1 sampai ke 3, *plt.plot* berfungsi untuk menampilkan diagram garis dengan variabel x yang bernilai panjang akurasi dan y yang bernilai isi dari akurasi. Setiap 1 diagram menampilkan 3 garis yaitu, akurasi *train*, akurasi *test* dan *F1_score*.

5.4 Demo Prediksi

Setelah menghasilkan model yang terbaik, model tersebut dapat digunakan untuk melakukan prediksi semtimen. Berikut ini adalah kode untuk mengambil dan memuat model terbaik.

```
1. self.model.eval()
2. list_of_files = glob.glob('state_dict/*')
3. max_model = max(list_of_files, key=os.path.basename)
4. print('\nmodel max :',max_model)
5.
6. model = MGAN(self.embedding_matrix, self.opt).to(opt
.device)
7. model.load_state_dict(torch.load(max_model))
```

Kode 5.18 Memuat model tertinggi

Pada Kode 5.18 baris ke 2, *glob.glob('state_dict/*')* bertujuan untuk melihat semua file yang ada pada folder *state_dict*. Kemudian baris kode ke 3 *max(list_of_files, key=os.path.basename)* digunakan untuk memilih file yang paling maksimal berdasarkan nama file. Untuk memuat model

menggunakan sintaks `model.load_state_dict()` pada baris kode ke 7.

```

1. masukkan = input("masukkan kalimat yang ingin di cek
   : ")
2.   while masukkan != 'exit':
3.     asp = input('masukkan aspek :')
4.     masukkan = re.sub(r'@[A-Za-z0-9]+', '',
   masukkan)
5.     masukkan = re.sub(r'https?://[A-Za-z0-9./]+',
   '', masukkan)
6.     masukkan = re.sub('[\'+string.punctuation+']', ""
   , masukkan)
7.
8.     text_left, _, text_right = [s.lower().strip() fo
   r s in masukkan.partition(asp)]
9.
10.    text_raw_indices = [self.tokenizer.text_to_seque
   nce(text_left + " " + asp + " " + text_right)]
11.    text_left_indices = [self.tokenizer.text_to_sequ
   ence(text_left)]
12.    aspect_indices = [self.tokenizer.text_to_sequenc
   e(asp)]
13.    data=[torch.LongTensor(text_raw_indices).to(opt.
   device),
14.    torch.LongTensor(aspect_indices).to(opt.device),
15.    torch.LongTensor(text_left_indices).to(opt.devic
   e)]

```

Kode 5.19 Preprocess prediksi

Pada kode 5.19 baris ke 4 sampai ke 6 menunjukkan bahwa variabel `masukkan` harus di preproses terlebih dahulu agar tidak ada tanda baca, karakter, *link* atau huruf besar yang ada pada data yang akan di *input*. Kemudian pada baris ke 8 teks di split menjadi 3 bagian dan di tokenisasi. Setelah itu disimpan pada variabel `data` dalam bentuk *tensor* agar dapat digunakan oleh model.

```

1. output_demo = model(data)
2. prediction = int(torch.argmax(output_demo, -1))
3. print(prediction)
4.

```



```
5. if (prediction == 0):
6.     print ('negatif')
7. if (prediction == 1):
8.     print ('netral')
9. if (prediction == 2):
10.    print ('positif')
11. if (prediction == 3):
12.    print ('konflik')
```

Kode 5.20 Prediksi kata

Baris pertama Kode 5.20, menunjukkan bahwa hasil dari *model(data)* disimpan dalam variabel *output_demo*. Kemudian kode di baris kedua *torch.argmax(output_demo, -1)* akan menghitung prediksi. Nilai prediksi tersebut di ubah menjadi kata sesuai dengan aturan pada baris kode ke 5 sampai ke 12. Jika prediksi berniali 0 maka hasilnya adalah negatif, jika prediksi bernilai 1 maka hasilnya adalah netral, jika prediksi bernilai 2 maka hasilnya adalah positif, dan yang terakhir jika prediksi bernilai 3 maka hasilnya adalah konflik.

Halaman ini sengaja dikosongkan

BAB VI

HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil dan analisis terhadap penelitian yang diperoleh dari implementasi penelitian.

6.1 Preprocessing

Dari data tweet yang telah didapatkan, data masih tidak dapat digunakan karena tidak memiliki label. Sebelum data digunakan baik untuk training dan testing harus dilakukan preprocessing terlebih dahulu terhadap data. Berikut adalah hasil dari tahap preprocessing.

6.1.1 Hasil Pemberian Label Dataset

Pemberian label pada data dilakukan secara manual oleh manusia, karena tujuan dari pembuatan model ini adalah untuk bisa melatih model agar dapat memperkirakan apa label yang seharusnya diberikan pada suatu kalimat dengan aspek tertentu. Sampel hasil dari proses pelabelan secara manual akan dilampirkan pada LAMPIRAN A, dengan jumlah 90 baris dari 2000 baris.

6.1.2 Pembersihan Data

Setelah data sudah memiliki label, data dapat di-*import* kedalam kode untuk dibersihkan. Data akan dibersihkan dari tanda baca, simbol, karakter dan juga link. Sampel data hasil pembersihan dapat dilihat pada Tabel 6.1

Tabel 6.1 Sampel data hasil pembersihan

Data Sebelum	Data sesudah
Telkomsel loop :*"@MrObhelix: Apa kartu yg enak buat internetan? Gak lemot dan gak lelet tapi murah meriah?"	Telkomsel loop Apa kartu yg enak buat internetan Gak lemot dan gak lelet tapi murah meriah

@indosatjateng Jaringan Cepat Indosat! Paling stabil...silakan pilih 3GB, 11GB, 13GB, #superinternetindosat #group4	Jaringan Cepat Indosat Paling stabil silakan pilih 3GB 11GB 13GB superinternetindosat group4
Bener bgt "@p_Liziy: Telkomsel makin lemot aja (?-?)?"	Bener bgt Liziy Telkomsel makin lemot aja
lumayan dapet ganti 36 ribu pulsa dari indosat, setelah ditelpon oleh mbak cs indosat yang suaranya renyah banget... http://fb.me/2bK21R8oR	lumayan dapet ganti 36 ribu pulsa dari indosat setelah ditelpon oleh mbak cs indosat yang suaranya renyah banget

6.1.3 Case Folding dan Tokenisasi

Data yang sudah bersih dari tanda baca, karakter, simbol dan link selanjutnya akan di-*case folding* dan tokenisasi. Data akan di *split* setiap kata dan disimpan tanpa ada kata yang duplikat. Sampel data hasil *case folding* dan tokenisasi dapat dilihat pada Tabel 6.2

Tabel 6.2 Sampel hasil case folding dan tokenisasi

Teks	Teks Lower	Token
Ini whatsapp yang ngaco atau telkomsel yg lemot sih Membalas Kalau pascabayar gimana min Membalas Wah seru banget pakai indosat Paling bisa memancing emosi Sinyal sering banget tiba2 jelek	ini whatsapp yang ngaco atau telkomsel yg lemot sih membalas kalau pascabayar gimana min membalas wah seru banget pakai indosat paling bisa memancing emosi sinyal sering banget tiba2 jelek	'ini': 1, 'whatsapp': 2, 'yang': 3, 'ngaco': 4, 'atau': 5, 'telkomsel': 6, 'yg': 7, 'lemot': 8, 'sih': 9, 'membalas': 10, 'kalau': 11, 'pascabayar': 12, 'gimana': 13, 'min': 14, 'wah': 15, 'seru': 16, 'banget': 17, 'pakai': 18, 'indosat': 19,

pulsa kepotong tiap hari tarif nelpon sms ngga jelas	pulsa kepotong tiap hari tarif nelpon sms ngga jelas	'paling': 20, 'bisa': 21, 'memancing': 22, 'emosi': 23, 'sinyal': 24, 'sering': 25, 'tiba2': 26, 'jelek': 27, 'pulsa': 28, 'kepotong': 29, 'tiap': 30, 'hari': 31, 'tarif': 32, 'nelpon': 33, 'sms': 34, 'ngga': 35, 'jelas': 36
---	---	--

Pada kolom token di tabel tersebut setiap kata memiliki angka di sebelahnya, angka tersebut merupakan sebuah index yang di simpan pada sebuah *dictionary*. Token memiliki 2 variabel didalamnya *word2idx* dan *idx2word*. Sehingga pada saat *word2idx* dipanggil maka teks sebagai *key* dan indeks sebagai *value* yang akan tampil.

Pada saat tokenisasi dilakukan juga penghitungan aspek kata pada data training dan data testing. Hasil dari jumlah kata dapat di lihat pada tabel 6.3

Tabel 6.3 Jumlah label

Data	Jenis Label	Jumlah
Data train	Aspek	195
	Sentimen	'0': 648, '2': 365, '1': 84, '3': 42
Data test	Aspek	57
	Sentimen	'0': 128, '2': 85, '1': 24, '3': 13

Pada tabel diatas dapat dilihat bahwa semakin banyak data yang dimiliki maka aspek unik yang dimiliki juga akan bertambah. Sedangkan baik pada aspek maupun pada sentiment jumlah data yang dimiliki sangat tidak seimbang. Contohnya data test sentimen 0 memiliki jumlah 128 sedangkan pada

kosakata pada data yang tidak sesuai dengan bahasa Indonesia yang baik, seperti kata-kata singkatan, kata gaul, dan lain sebagainya. Selain itu pada tahap ini *method* menghasilkan file hasil *embedding* matriks dari token pada data *input*. File tersebut yang nanti akan digunakan pada saat menjalankan *class* MGAN.

6.2 Hasil Pembuatan Model MGAN

Dari model MGAN yang telah dibangun model akan menghasilkan nilai akurasi *train* dan tes. Nilai akurasi tersebut didapatkan berdasarkan masukan dan juga konfigurasi awal yang telah ditentukan dengan berbagai parameter yang sudah dijelaskan pada bab 4.3.1 sebelumnya.

6.2.1 Konfigurasi Parameter Awal

Sebelum melakukan proses training data, terlebih dahulu harus ditentukan konfigurasi parameter di awal agar mendapatkan hasil yang jelas. Berikut tabel 6.4 yang merupakan konfigurasi parameter awal.

Tabel 6.4 Konfigurasi awal model

No	Parameter	Nilai
1	Embed_dim	300
2	Hidden_dim	300
3	Polarities_dim	4
4	Num_layer	1
5	Bias	True
6	Batch_first	True
7	Dropout	0
8	Bidirectional	True

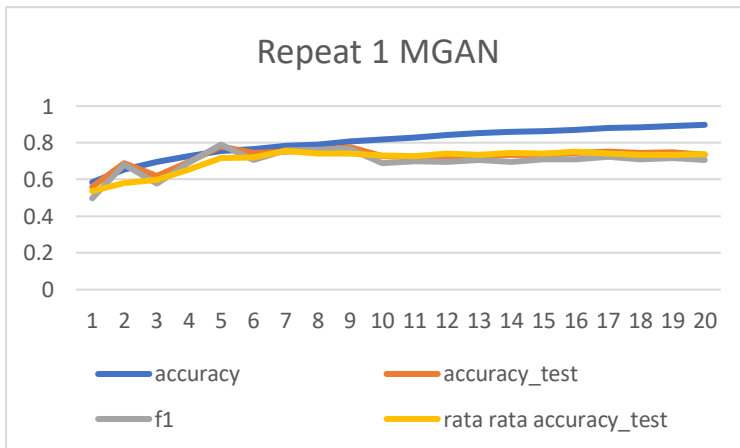
Selain konfigurasi model ada pula konfigurasi training yang harus didefinisikan di awal, berikut Tabel 6.5 yang mendefinisikan konfigurasi awal pada training.

Tabel 6.5 Konfigurasi awal training

No	Parameter	Nilai
1	Num_epoch	20
2	Batch_size	50
3	Repeats	5
4	Optimizer	Adam

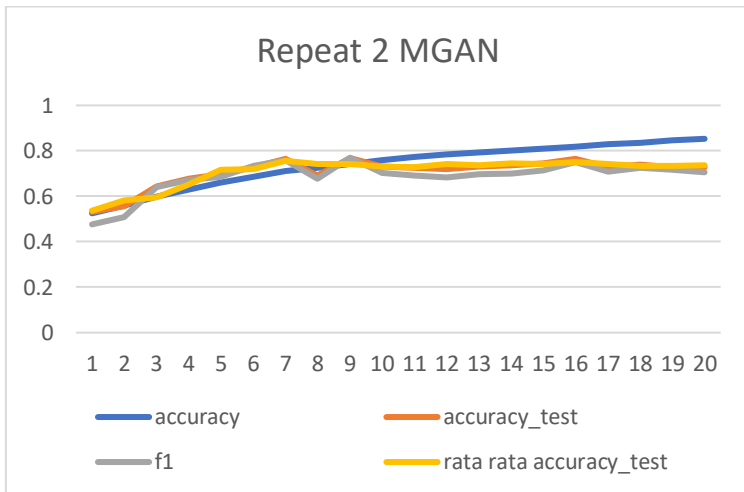
6.2.2 Hasil Training dan Tes

Setelah melakukan preprocessing data hingga data siap digunakan pada model. Tahap selanjutnya adalah menjalankan model dengan melakukan pelatihan dan tes data. Hal itu akan diulang sebanyak 5 kali. Dan setiap perulangan akan terdiri dari 20 epoch. Berikut ini adalah grafik dari hasil akurasi selama 1 kali perulangan, yang dapat dilihat pada gambar 6.2 – 6.6. Grafik tersebut berdasarkan pada data di LAMPIRAN C.

**Gambar 6.2** Percobaan Perulangan ke-0

Pada gambar 6.2 merupakan perulangan ke-0 untuk data yang lebih spesifik dapat dilihat pada LAMPIRAN B. Gambar tersebut menunjukkan bahwa nilai akurasi train semakin meningkat. Sedangkan untuk nilai akurasi test rata-rata berada

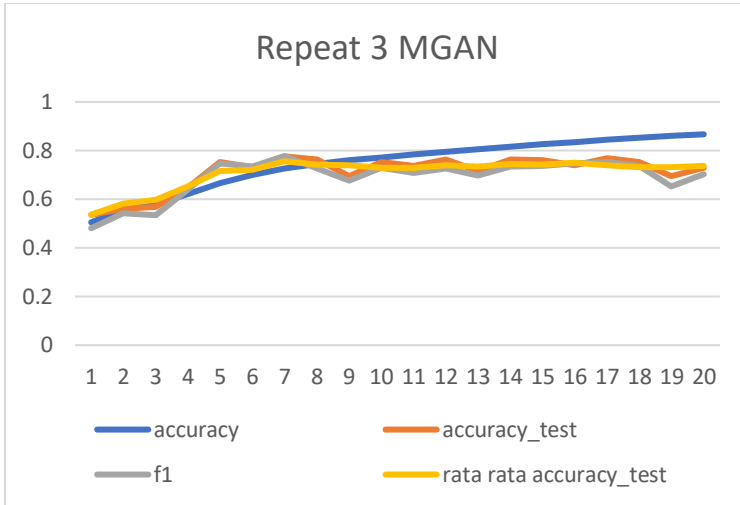
pada nilai 70% - 79%. Kemudian nilai pada F1-Score masih berada dibawah nilai akurasi test.



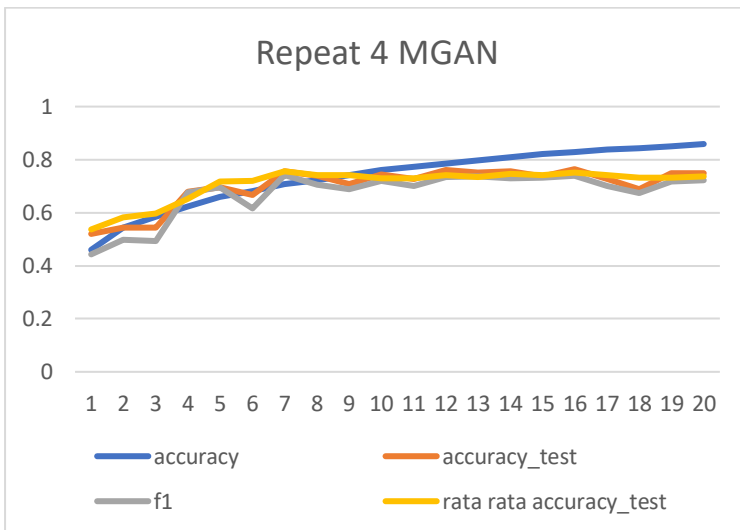
Gambar 6.3 Percobaan Perulangan ke-1

Gambar 6.3 yang merupakan perulangan ke-1 dari training, memiliki data yang sama seperti pada LAMPIRAN C. Berdasarkan gambar tersebut dapat di simpulkan bahwa nilai akurasi train semakin meningkat, namun nilai akurasi test dan F1-score mengalami over fitting. Ketika berada di data ke 20-an, akurasi mencapai titik maksimalnya dan setelah itu hasilnya cenderung menurun dan statis.

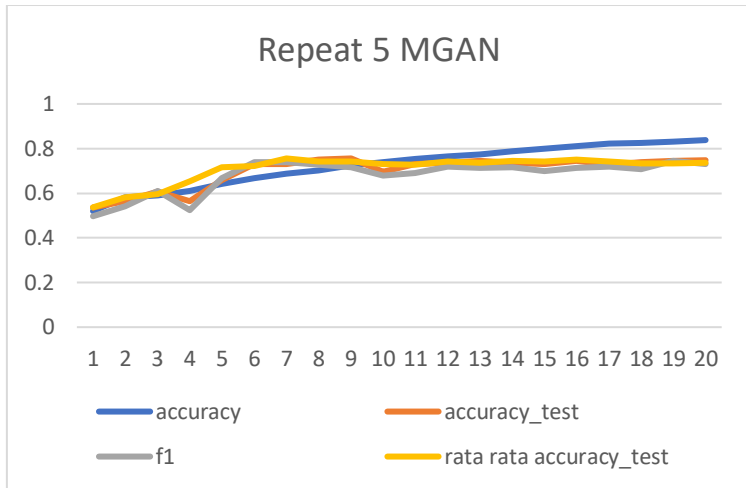
Gambar 6.4 sampai 6.6 menunjukkan hasil dari perulangan ke-2 – ke-4 yang mana detail data dapat dilihat pada LAMPIRAN C. Ketiga gambar tersebut hampir memiliki pola yang sama hasil akurasi test tertinggi dicapai antara epoch ke 5 sampai ke 9, setelah itu nilainya akan cenderung stabil. Sehingga jumlah epoch 20 dirasa cukup optimal untuk digunakan dalam model ini.



Gambar 6.4 Percobaan Perulangan ke-2



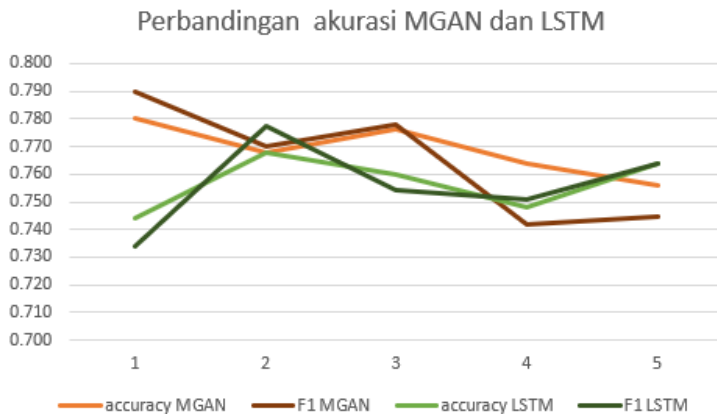
Gambar 6.5 Percobaan Perulangan ke-3



Gambar 6.6 Percobaan Perulangan ke-4

6.3 Perbandingan Model lain

Selain menggunakan model MGAN untuk melakukan training dan testing dilakukan juga training dan testing data menggunakan model lain yaitu Model LSTM. Hasil Testing data dapat dilihat pada gambar 6.7



Gambar 6.7 Perbandingan akurasi MGAN dan LSTM

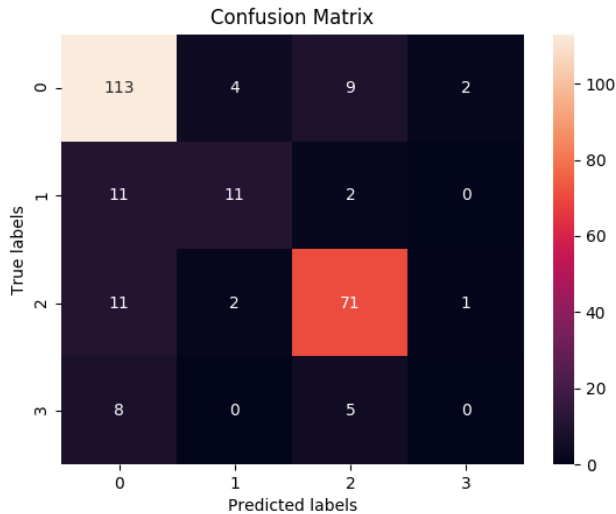
Gambar diatas menunjukkan bahwa dari segi akurasi dan F1 score model MGAN lebih baik walaupun ada suatu titik dimana nilai LSTM lebih tinggi yaitu pada percobaan ke 2. Berikut ini adalah tabel 6.6 perbandingan dari MGAN dan LSTM.

Tabel 6.6 Perbandingan Model MGAN dan LSTM

accuracy	Accuracy test	f1	precision	recall	duration
MGAN					
0.897717	0.780	0.7899	0.7666	0.8228	36.28434
0.852941	0.768	0.7698	0.7476	0.8101	35.62477
0.866769	0.776	0.7777	0.7586	0.8186	35.08795
0.858867	0.764	0.7416	0.7504	0.7975	35.09077
0.838016	0.756	0.7449	0.7292	0.7848	35.55212
LSTM					
0.917032	0.744	0.7340	0.7465	0.7848	7.357495
0.892669	0.768	0.7774	0.7986	0.8101	7.303848
0.899254	0.760	0.7544	0.7861	0.8017	7.389278
0.885426	0.748	0.7508	0.7797	0.7890	7.360339
0.860843	0.764	0.7640	0.7451	0.8487	7.559521

6.4 Pengukuran Performa per Sentimen Model MGAN

Dalam mengukur performa tidak hanya menggunakan akurasi saja namun bisa menggunakan F1 Score, precision dan recall. Gambar 6.8 menunjukkan hasil confusion matrix menggunakan model MGAN.



Gambar 6.8 Confusion Matrix Model MGAN

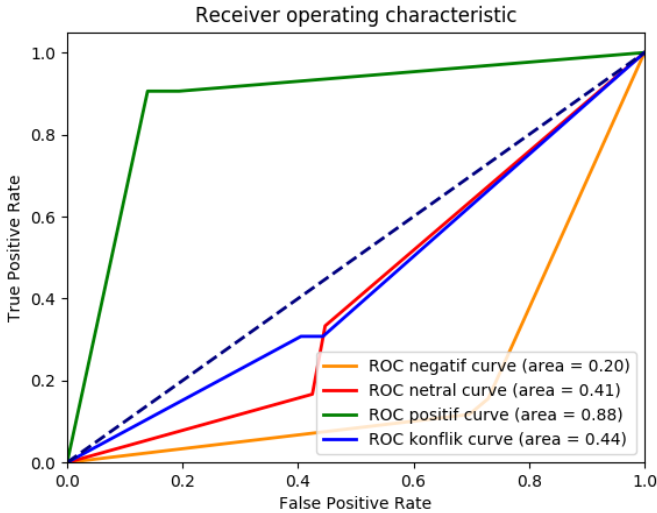
Pengukuran performa per sentimen pada model MGAN dapat dilihat pada tabel 6.7

Tabel 6.7 Performa per sentimen MGAN

Sentimen	precision	recall	f1 Score
Negatif	0.882813	0.79021	0.833948
Netral	0.458333	0.647059	0.536585
positif	3.380952	0.816092	1.314815
konflik	0	0	0

Hasil dari sentimen konflik sangat buruk karena sama sekali tidak ada kata bersentimen konflik yang di prediksi dengan benar. Sedangkan hasil precission dari sentimen netrak dan positif juga tidak bagus karena berada di bawah 0.5.

Selain pengukuran performa menggunakan confusion matrix, performa dapat pula dilihat menggunakan Receiver Operating Characteristic (ROC) dan Area Under the Curve AUC. Grafik ROC AUC dapat dilihat pada gambar 6.9.

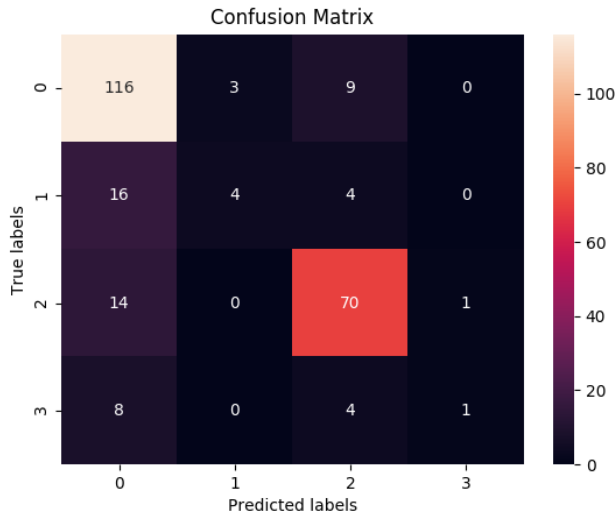


Gambar 6.9 ROC AUC Model MGAN

Grafik tersebut menunjukkan hasil yang baik untuk curva positif dengan nilai AUC 0.88 dan negatif dengan nilai AUC 0.2 karena semakin jauh dari garis baseline atau garis putus-putus berwarna biru. Sedangkan garis netral dan konflik memiliki nilai 0.41 dan 0.44, yang sangat dekat dengan garis baseline. Hal tersebut menunjukkan hasil yang kurang baik untuk performa sentimen netral dan konflik.

6.5 Pengukuran Performa per Sentimen Model LSTM

Pengukuran performa juga dilakukan pada model LSTM untuk menghitung nilai F1 Score, precission dan recall. Gambar 6.10 menunjukkan hasil confusion matrix menggunakan model LSTM.



Gambar 6.10 Confusion Matrix Model LSTM

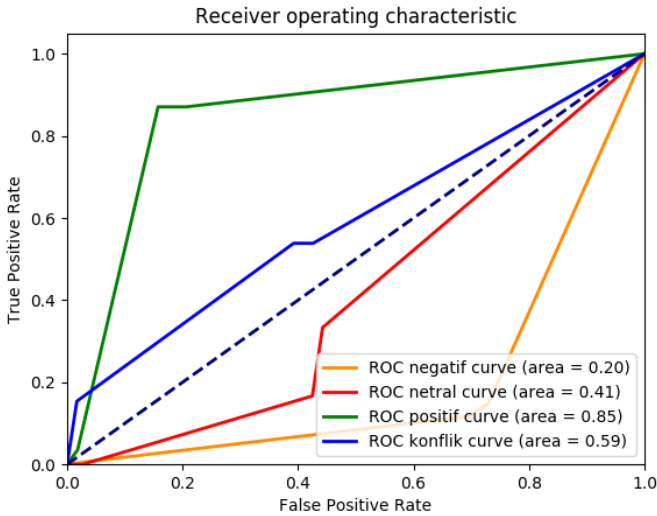
Pengukuran performa per sentimen pada model MGAN dapat dilihat pada tabel 6.7

Tabel 6.8 Performa per sentimen LSTM

Sentimen	precision	recall	f1 score
Negatif	0.90625	0.753247	0.822695
Netral	0.166667	0.571429	0.258065
positif	0.823529	0.804598	0.813953
konflik	0.076923	0.5	0.133333

Hasil precision dari sentimen konflik dan netral dan positif juga tidak bagus karena berada di bawah 0.5.

Selain pengukuran performa menggunakan confusion matrix, performa dapat pula dilihat menggunakan Receiver Operating Characteristic (ROC) dan Area Under the Curve AUC. Grafik ROC AUC dapat dilihat pada gambar 6.11



Gambar 6.11 ROC AUC Model LSTM

Grafik tersebut menunjukkan hasil yang baik untuk curva positif dengan nilai AUC 0.85 dan negatif dengan nilai AUC 0.2 karena semakin jauh dari garis baseline atau garis putus-putus berwarna biru. Sedangkan garis netral dan konflik memiliki nilai 0.41 dan 0.69, yang sangat dekat dengan garis baseline. Hal tersebut menunjukkan hasil yang kurang baik untuk performa sentimen netral dan konflik.

6.6 Hasil Demo Prediksi

Setelah menghasilkan banyak model dengan berbagai akurasi, dari semua model tersebut akan dicari model terbaik untuk digunakan dalam demo prediksi. Kode akan me-load model terbaik, kemudian dimasukkan kalimat dan aspek yang ingin di cek sentimennya. Setelah semua input sudah di masukkan maka kalimat akan di cek di dalam mosel terbaik dan menghasilkan sentimen dari kalimat tersebut berdasarkan aspek. Contoh hasil dari prediksi dapat dilihat pada gambar 6.12.


```

masukkan kalimat yang ingin di cek : koneksi nya lelet banget
masukkan aspek :koneksi
0
negatif
masukkan kalimat yang ingin di cek : suka deh kalo internet lancar jaya gini
masukkan aspek :internet
2
positif

```

Gambar 6.12 Hasil prediksi sentimen

Pada gambar 6.12 dimasukan sebuah kalimat untuk memprediksi kalimat yang memiliki aspek koneksi dan sentimen di prediksi dengan benar yaitu negatif. Contoh ke 2 dimasukan kalimat dengan aspek internet dan sentimennya dapat di presiksi dengan benar yaitu positif.

```

masukkan kalimat yang ingin di cek : aktifin 4g telkomsel gimana ya ?
masukkan aspek :4g
0
negatif
masukkan kalimat yang ingin di cek : kemarin sms baik baik aja, kok sekarang jadi
pending terus ya
masukkan aspek :sms
0
negatif

```

Gambar 6.13 Hasil prediksi sentimen 2

Pada gambar 6.13 dimasukan kalimat dengan aspek 4g yang seharusnya memiliki sentimen netral namun hasil yang ditunjukkan adalah negatif. Begitu pula dengan input yang kedua kalimat dengan aspek sms seharusnya memiliki sentimen konflik namun hasil yang di dapatkan adalah negatif.

Karena nilai performa untuk kalimat netral dan konflik cukup rendah, maka kemungkinan untuk mendapatkan hasil prediksi yang benar juga akan kecil.

Halaman ini sengaja dikosongkan

BAB VII

KESIMPULAN DAN SARAN

Bab kesimpulan dan saran membahas mengenai kesimpulan proses penelitian yang telah dilakukan dan saran yang diusulkan baik untuk untuk penelitian serupa di masa mendatang dan pengembangan yang lebih baik lagi.

Kesimpulan

Kesimpulan dari proses penelitian pembuatan model sentimen analisis menggunakan MGAN yang telah dilakukan adalah sebagai berikut:

1. Pada tahap word embedding dari 4259 kata 642 atau sekitar 14% kata tidak dapat di ditemukan dalam word embedding sehingga kata tersebut harus dibuatkan vector secara acak oleh model.
2. Nilai akurasi training dapat mencapai 100% pada epoch ke 17 sampai 20 dan nilai akurasi train cenderung stabil ketika mencapai epoch 5 – 8, sehingga jumlah epoch yang paling optimal untuk digunakan pada model MGAN adalah 20 epoch.
3. Keseimbangan label atau sentimen pada data training sangat berpengaruh pada hasil performa akurasi, F1 score, precision dan recall baik untuk model MGAN maupun LSTM.
4. Hasil akurasi dan f1-score dari LSTM lebih rendah jika dibandingkan dengan hasil yang di dapatkan ketika menggunakan model MGAN.
5. Waktu dalam menjalankan model LSTM lebih singkat yaitu rata-rata selama 7.3 detik, sedangkan waktu untuk menjalankan model MGAN lebih lama yaitu sekitar 35 detik jika menggunakan CUDA.

Saran

Berdasarkan penelitian tugas akhir ini, adapun beberapa saran untuk pengembangan penelitian ke depan yaitu sebagai berikut:

1. Jenis kata unik yang ada pada model embedding dirasa masih kurang untuk digunakan karena masih ada 14% dari total kata yang tidak dapat ditemukan, sehingga akan lebih baik untuk menambah data pada model *word embedding* atau mencoba *word embedding* lain selain *word2vec*.
2. Dataset yang digunakan harus memiliki label yang seimbang agar memiliki performa yang lebih baik untuk semua sentimen yang ada, tidak hanya pada satu jenis sentimen saja yang memiliki performa baik.
3. Jika dibandingkan dengan model LSTM waktu yang digunakan untuk training model MGAN lebih lama segingga akan lebih baik jika dapat mempersingkat waktu train dan tes menggunakan model MGAN.

DAFTAR PUSTAKA

- [1] “2018, Jumlah Penduduk Indonesia Mencapai 265 Juta Jiwa,” Katadata, 18 Mei 2018. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2018/05/18/2018-jumlah-penduduk-indonesia-mencapai-265-juta-jiwa>. [Diakses 1 February 2019].
- [2] “Penetrasi Internet di Indonesia Capai 143 Juta Jiwa,” APJII, 22 Maret 2018. [Online]. Available: <https://apjii.or.id/content/read/104/348/BULETIN-APJII-EDISI-22---Maret-2018>. [Diakses 1 Februari 2019].
- [3] “Twitter Inc.,” Twitter, [Online]. Available: <https://about.twitter.com/id.html>. [Diakses 1 February 2019].
- [4] “Wajib Registrasi, Pelanggan Seluler Indosat Menyusut 34%,” Katadata, 19 November 2018. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2018/11/19/wajib-registrasi-pelanggan-seluler-indosat-menyusut-34>. [Diakses 11 Februari 2019].
- [5] G.Vinodhini dan RM.Chandrasekaran, “Sentiment Analysis and Opinion Mining: A Survey,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 6, pp. 282-292, 2012.
- [6] F. Fan, Y. Feng dan D. Zhao, “Multi-grained Attention Network for Aspect-Level Sentiment Classification,” dalam *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.
- [7] N. Majumder, S. Poria, A. Gelbukh, M. S. Akhtar, E. Cambria dan A. Ekbal, “IARM: Inter-Aspect Relation Modeling with Memory Networks in Aspect-Based Sentiment Analysis,” dalam *Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.
- [8] E. D. Liddy, *Natural Language Processing, Encyclopedia of Library and Information Science*, 2001.

- [9] B. Prijono, “Belajar Pembelajaran Mesin di Indonesia,” indoML.com, 13 April 2018. [Online]. Available: <https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2/>. [Diakses 11 Maret 2019].
- [10] D. Karani, “Introduction to Word Embedding and Word2Vec,” Toward Science, 1 September 2018. [Online]. Available: <https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa>. [Diakses 2 Juli 2019].
- [11] S. Yan, “Understanding LSTM and its diagrams,” Medium.com, 14 Maret 2016. [Online]. Available: <https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714>. [Diakses 27 Februari 2019].
- [12] Skymind, “A.I. Wiki,” [Online]. Available: <https://skymind.ai/wiki/attention-mechanism-memory-network>. [Diakses 11 Maret 2019].
- [13] Synced, “A Brief Overview of Attention Mechanism,” Medium.com, 26 September 2017. [Online]. Available: <https://medium.com/syncedreview/a-brief-overview-of-attention-mechanism-13c578ba9129>. [Diakses 27 Februari 2019].
- [14] P. Chen, Z. Sun, L. Bing dan W. Yang, “Recurrent Attention Network on Memory for Aspect Sentiment Analysis,” dalam *Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, 2017.
- [15] R. Arthana, “Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning,” Medium.com, 5 April 2019. [Online]. Available: <https://medium.com/@rey1024/mengenal-accuracy-precision-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8>. [Diakses 2 Juli 2019].

LAMPIRAN A. DATA BERLABEL

TWEET	TOPIK	CATEGORIES	SENTIMEN
@indosatcare Kenapa kartu diregistrasi nggak bisa pak? Padahal nomor kk & ktp nya sudah benar? Sampai 5x gagal, tolong di daftarkan segera, demi kenyamanan pengguna kartu indosat. Terima kasih	INDOSAT	registrasi	-1
indosat mantap djaja koneksinya , ngebut euy	INDOSAT	koneksi	1
Membalas @IndosatCare @IndosatCare mau nangis deh rasanya pdahl saya percaya betul dengan indosat soal koneksi di wil Karimun tp sudah 3 hari koneksi kacau balau	INDOSAT	koneksi	2
Sinyal Indosat asu ari musim hujan, hadeuh	INDOSAT	sinyal	-1
Membalas @tsel_malang @tsel_malang iya min. Udah bisa kok. Hihhi. Tengkyu yaaaa :) seneng banget pake telkomsel. Sinyal lancaaar {}	TSEL	sinyal	1
@IndosatCare 30 ribu saya beneran atau bohongan saya coba buat internet lngsung tanpa daftar paket internet dan alhamdulillah bisa buat internet trus saya buka halaman indosat.. Baru kebuka saya dpt sms dari indosat "Kamu internetan dg tarif perKB,pemakai	INDOSAT	paket	2

TWEET	TOPIK	CATEGORIES	SENTIME N
@IndosatCare tolong kasih penjelasan yg bisa saya terima dong.. kenapa 3 hari ini sinyal Indosat sangat mengecewakan di daerah Mampang Depok	INDOSAT	sinyal	-1
yang lagi mau liburan di The Jungle Bogor Indosat Super WIFI juga ada di sana looh, ngenet asik sambil liburan #Mantap @indosatbogor	INDOSAT	wifi	1
Membalas @Telkomsel mantap jiwa nih min harga pakatnya	TSEL	harga paket	1 1
Terima kasih kepada sponsor yg sudah mendukung acara kita, Indosat dan Nafigo. Sehingga acara kita dapat berjalan dengan lancar ?	INDOSAT	sponsor	1
Indosat sinyale koyo NGELEC.... Asu	INDOSAT	sinyal	-1
kecewa berat saiah dengan @XLcare, lebih baik beralih ke @indosat lah...kualitas sinyal nya pasti!!!	INDOSAT	sinyal	1
@IndosatCare min, jaringan indosat lemot bgt dan tolong jangan kasih solusi "update jaringan manual lewat hp" dan sejenisnya. karena yang bermasalah kan bukan hpnya tapi indosatnya. jadi tolong kasih solusi yang bener2 bisa dilakukan dan menyelesaikan mas	INDOSAT	jaringan	-1

TWEET	TOPIK	CATEGORIES	SENTIME N
Kebiasaan pake indosat ngecek pulsa salah,tapi makasih telkomsel,internet mu cepat sekali :*.	TSEL	internet	1
@Telkomsel sinyal jelek bgt sih, kecewa nih pakai simpati daerah buaran, serpong	TSEL	sinyal	-1
Membalas @Telkomsel Min mau nanya ni.. knp sim card telkomsel saya tdk bisa buat panggilan keluar sedangkan kemarin bisa . Sudah dua minggu koit min sdh berhasil registrasi pula gimana saran dan tanggapan anda mewakili telkomsel ?	TSEL	telepon registrasi	-1 1
Sekarang udah jamannya 4G LTE, udah ngga ada lagi yang namanya lemot. Kata temen 4G LTEnya Telkomsel dahsyat banget .. Eh pas dites, bener!	TSEL	4g	1
makasih @indosatcare keluhanku sudah di tangani dengan cepat, i love indosat	INDOSAT	keluhan	1
di buat nangis sama jaringan indosat :/ gara* jaringan nya jelek nyangka nya mz dari dia ga di lz hampura sayank :* @egi_pribadi	INDOSAT	jaringan	-1
Membalas @Telkomsel Saya mau nangis kejer rasanya mas. Kalau saya main mobile legend tuh tiba tiba jaringan terputus tanpa sebab. Tau gak telkomsel	TSEL	jaringan koneksi	-1 -1

TWEET	TOPIK	CATEGORIES	SENTIME N
sering banget kayak gini. Padahal untuk streaming apapun itu lancar lancar aja. Mas saya udah 2x turun credit score mobile legend gara2 koneksi terputus			
mantap..sinyal 3G stabil skrg dirumah...thx telkomsel...	TSEL	sinyal	1
Membalas @Telkomsel Kalo kayak gini gimana coba aktifinnyaaaaa !!! Kasih tau dong caranyaaaaa	TSEL	aktivasi	0
monyet koneksi sampah, telkomsel nepu mulu dan gw mau aja ditipu terus. jadi yang bego siapa?	TSEL	koneksi	-1
Membalas @Telkomsel @Telkomsel anyway, sangat puas dgn pelayanan Grapari Tsel Bogor cs Amel sabtu kmren. Thx	TSEL	pelayanan	1
@Telkomsel min saya mau nanya, ngga sengaja saya lewat jatuh tempo tagihan kartu halo, gimana ya mengurusnya? Apakah denda atau bagaimana?	TSEL	tagihan	0
Kumpul mahadelta dapet rejeki dari indosat lumayan 11gb free lancar abis. Terima kasih indosat	INDOSAT	bonus	1
@Telkomsel anjing lu!kalau gk suka mobile legend jangan dilenotin dong sinyalnya asu! AOV itu asu kyk telkomsel	TSEL	sinyal	-1

TWEET	TOPIK	CATEGORIES	SENTIME N
nih provider @IndosatCare harus banyak belajar sama @Telkomsel keluhan cepat di tanggapi ,mungkin late respon terlalu banyak yg complain ke indosat ,bertahun2 pake im3 sama telkomsel baru indosat yg sering pulsa raib pake tuyul apa sih	TSEL	keluhan	1
apakah ini kemajuan? apa indosat makin maju? dari kemarin-kemarin internetnya cepet terusssss selamat !	INDOSAT	internet	1
CS nya lelet dan lemot, @Telkomsel harus merubah sistem pelayanan, agar tidak ada korban lagi setelah saya. Kecewa dg peleyanannya..	TSEL	pelayanan	-1
Membalas @IndosatCare Anjing indosat goblok fakk maen moba aja leg nya minta di tampil kontrol	INDOSAT	moba	-1
Asu @indosat jaringane kek tai percuma pket banyak tp nge lag gbsa ngapa2in sumpah payah jancuk	INDOSAT	jaringan	-1
Membalas @a_damnthing @InfoTwitwor dan 4 lainnya Iya nih kontrol indosat pulsa gua dipotong 50k anjing babi, pulsa cepe beli paket telpon 20 sisa 30 gajelas assu	INDOSAT	pulsa	-1

TWEET	TOPIK	CATEGORIES	SENTIME N
Pertama kalinya XL dibawa ke luar rumah. Signal internetnya bagus! bye bye Indosat. cukup lah ya bertahan 6 tahun. :)	INDOSAT	sinyal	1
Susah sama @Telkomsel smpai skrg cuma di tanya, di tanggapinya & perbaikannya ga ada padahal setia menggunakan tsel cuma skrg kecewa brt	TSEL	perbaikan	-1
Kecewa sama pelayanan Telkomsel, harga Paket makin naik tapi koneksi abal.	TSEL	pelayanan harga koneksi	-1 -1 -1
Membalas @IndosatCare waktu itu kartu udah hangus karena lupa isi pulsa, alhamdulillah pas di urus ke gallery indosat pas hari itu juga kartu bisa di reaktifasi lagi meskipun harus nunggu 2 bulan, thanks indosat	INDOSAT	kartu aktivasi	-1 1
@IndosatCare halo indosat, saya sudah kirim keluhan via DM. tolong cek dan follow up, ya. mohon bantuannya. terima kasih.	INDOSAT	pelayanan	0
Membalas @IndosatCare saya akui indosat cukup bagus, bahkan kecepatan downloadpun bisa mencapai 400 kb per detik	INDOSAT	kecepatan	1
@Telkomsel ini jaringan kenapa 3g dan ilang2? Cacat banget koneksi	TSEL	3g koneksi	-1 -1

TWEET	TOPIK	CATEGORIES	SENTIME N
Paketan internet indosat bikin nangis... boross yawooh	INDOSAT	paket	-1
Gak boleh nyebut merk.. titik RT @KakaLebah: Internetnya telkomsel lemot...titik	TSEL	internet	-1
Membalas @IndosatCare sudah saya dm. cek segera. itu no hp saya. tolong matikan layanan prime joy yang saya tidak daftar itu. saya tidak terima pulsa saya terpotong tiap bulan hanya demi benefit prime dari indosat yang SAMA SEKALI TIDAK BERGUNA DAN TAK AKAN SAYA GUNAKAN. Thanks.	INDOSAT	pulsa	-1
@IndosatCare Selamat malam, kartu Indosat saya kok pulasanya berkurang terus ya tiba tiba, bisa tolong dicek siapa tau ada aplikasi gak jelas yang teregister, terima kasih	INDOSAT	pulsa	-1
Kecewa dg kualitas Indosat 00.07 kuota masih 14Gban. 00.30 kuota ludes gak jelas @IndosatCare	INDOSAT	kuota	-1
Haha :p tapi da lelet pake telkomsel a @AldyAPekerti: @ndeyunie engga ah udh enak pake yang ini murah ??a??a??a??a	TSEL	intenet harga	-1 -1

TWEET	TOPIK	CATEGORIES	SENTIMEN
@IndosatCare tolong cek no 6285812042746 knp ga ada balasan dr indosat ya Terima kasih Segera ditindak lanjuti	INDOSAT	balasan	-1
@Telkomsel Lurd cek poin gimana yah?	TSEL	poin	0
@Telkomsel baru kali ini saya kecewa sekali dengan pelayanan caroline officer 188 (agent name tari) saya menanyakan ttg proses running paket data saya kenapa tdk berjalan seperti biasanya.	TSEL	pelayanan	-1
delapan tahun make telkomsel baru kali ini tukar poin,, 170 poin = 20ribu itupun udah seneng banget...	TSEL	poin	1
@IndosatCare kenapa ya jaringan indosat belakangan ini suka hilang sinyal.. lama kelamaan bisa kecewa saya... dan 1 lagi, kualitas 4G di Palembang buruk sekali	INDOSAT	sinyal	-1
Sinyal @simPATI di tmpt saya sering mati2 udah 2 minggu ini, ga bisa nelpon & ditelpon, internet lemot @Telkomsel @telkomselflash	TSEL	telepon internet	-1 -1
Membalas @fadhilhrp @Telkomsel Indosat juga lagi bermasalah.. alhamdulillah ternyata provider mahal juga bermasalah	INDOSAT	harga	-1

TWEET	TOPIK	CATEGORIES	SENTIME N
jaringan indosat lumayan bagus :3 , kapan 4G nya hadir ?? @indosatOoredoo	INDOSAT	Jaringan 4g	1 0
@IM3Ooredoo Maksud nya apa ya ? Ini karena sistem indosat yang trouble atukah kesengajaan ? Tolong dikembalikan pulsa saya. Terima ka	INDOSAT	pulsa	-1
@Telkomsel Memang Macam Anjing Sinyalnya, Lemah, Gx Stabil n Yang Pasti Minta Cabut Tu Tower,,,	TSEL	sinyal	-1
ucweb indosat mantap! bisa new tab ! (norak), murah pula!wew siapa yang mau?	INDOSAT	harga	1
hallo indosat yang baik hati, dan provider yang ga ada matinya..terima kasih loh atas pelayanan yang sangat memuaskan #5monthwithindosat	INDOSAT	pelayanan	1
Membalas @ferdy_doki telkomsel murah jg jok RT @ferdy_doki Gedor be RT "@chandra_jmb: belum dtng dy RT @ferdy_doki Duku jg enak RT "@chandra_jmb: lewat jendela	TSEL	harga	1
@Telkomsel jancok banget ni kartu saya udah maketin paket flash eh tapi pulsa tetep kepotong !!!	TSEL	paket pulsa	0 -1
Dunia harus tau kalo skrng di rumahku udah ada sinyal 4g indosat ? Alhamdulillah packet yellowku wkwk	INDOSAT	sinyal paket	1 1

TWEET	TOPIK	CATEGORIES	SENTIME N
@IndosatCare tolong informasinya segera. Saya belum pernah ikut paypro, tapi disuruh masukkan pin. Berapa pin saya. No indosat saya 085854055623. Terima kasih 10.2	INDOSAT	pin	0
sipooo RT @jejenieeee: Oke makasih glo RT @theodregloria: lumayan RT @jejenieeee: eh yang indosat sinyalnya bagus gak?	INDOSAT	sinyal	1
Membalas @Telkomsel @Telkomsel yeee terbukaaa... terima kasih @Telkomsel engkau memang pelayanan terbaik....	TSEL	pelayanan	1
Iya,bg! Kenceng XL dari pd telkomsel ._ RT @Ridhobono: Emang XL juo balik.. Halo Lemot.. Paketnya aja yg mahal.. katanya kecepatan sampai	TSEL	paket kecepatan	-1 -1
Membalas @IndosatCare Kemrin waktu ke pantai parangtritis jogja, jaringan 4G nya masih mantab bngt. Suka pakai indosat ooredoo. Bnyak bonusnya. #HariKonsumenNasional	INDOSAT	jaringan	1
Membalas @Telkomsel Saya mau regis tapi nik tidak ditemukan gimana tuh min	TSEL	registrasi	-1

TWEET	TOPIK	CATEGORIES	SENTIME N
indosat lg ada masalah ya min? nomor saya tidak bisa terima sms masuk dan harus di copot simcardnya dulu baru smsnya masuk tolong di cek @IndosatCare	INDOSAT	sms	2
Membalas @INDOSAT_M3 @INDOSAT_M3 bnr2 ngak diperhatiin kualitas ISAT BB! sibuk Promo trus tp kualitas sinyal ancurrr!!!! kecewa bneran dh pelayanan INDOSAT!!	INDOSAT	sinyal pelayanan	-1 -1
Mudah mudahan @Telkomsel bisa lebih baik dlm pelayanan pelanggan krn terus terang urusan sinyal dan coverage saya sangat puas dg Telkomsel.	TSEL	pelayanan sinyal	0 1
lumayan bagus wan RT @stalkerlove03: Rekomendasi dong... Jaringan indosat khususnya im3 bagus ga sih????	INDOSAT	jaringan	1
@IndosatCare halo indosat. ada pengaturan yang cocok gk buat hp advan i5c, karena saat perpindahan kartu ke hp lain (samsung A5 SM-A500F)internetnya lebih cepat.	INDOSAT	internet	0
pakai telkomsel itu enak yaa gak pernah pending kalau bbman terus kalau telfonan sama smsan murah banget	TSEL	telepon sms	1 1

TWEET	TOPIK	CATEGORIES	SENTIME N
@IndosatCare untuk yg kesekian kalinya pulsa saya hilang tiba-tiba. pulsa 3000 tinggal 800. mantap indosat diem-diem mencuri	INDOSAT	pulsa	-1
Mau sampai kapan gangguan sinyal begini @Telkomsel? Semingguan tidak bisa dipakai buat main mobile legends. Saya kecewa. Beneran.	TSEL	sinyal	-1
tanggapan dari CS indosat cukup bagus lah menurut saya	INDOSAT	tanggapan	1
Membalas @Telkomsel @telkomsel owh itu aj ya. Oh ya makasih ya Paket TAU nya udh 3x ak pake itu. dan ak merasa sangat puas. semga pket TAU di adakn lg.	TSEL	paket	1
@IndosatCare tolong baca DM saya. Saya urgent perlu bantuan. Nomor indosat saya sdh tdk aktif berhari hari. Terima kasih	INDOSAT	nomor	-1
Jadi, kesimpulannya.. Indosat tetap ee", baru jam segini signalnya lumayan bagus. Itu juga lumayan. Sampai lebaran pun ga akan gue maafin!	INDOSAT	sinyal	2
tp katae plng enak telkomsel? RT @CHIZTYT Podo ae se jatuhe kabehRT @codyputri: lek paketan iku murah endi?? R ... "http://bit.ly/jjrRjG"	TSEL	paket	1

TWEET	TOPIK	CATEGORIES	SENTIME N
Membalas @IndosatCare Mungkin sekitar 1 tahun aku pake provider Indosat. Sejauh ini cukup puas, nyaman & aman. Karena terbukti tarifnya paling murah & koneksi internet juga bisa diandalkan. Apalagi semenjak ada paket Yellow 1GB Rp 1.000 jadi makin betah deh ? #HariKonsumenNasional	INDOSAT	harga paket koneksi	1 1 1
Sms blast Telkomsel Sampah banget isinya. Udah bayar pulsa tetep aja di blast. Unethical!	TSEL	sms pulsa	-1 -1
Jaringan bb telkomsel napa ilang mulu.. Lemot cuy.. --"	TSEL	jaringan	-1
Kak...sinyal di daerah saya kok sekarang jelek sejak ada sinyal 4g...hp ahrus di lock 4g agar dapt sinyal 4g...kalo gak di lock cuma dpt sinyal H dan itu lemot...bantuannya kak....@Telkomsel	TSEL	sinyal	-1
Pada heboh paket BIS 3..namun telkomsel tetap juaranya..murah koq njaluk enak ! http://myloc.me/45GVi	TSEL	paket	1
Lagi-lagi TCash telkomsel @Telkomsel @TCASH_ID Mesin EDC nya error.. Kedua kali nya di Centre Point Medan. Ckckck.. Kecewa bngt.	TSEL	tcash	-1

TWEET	TOPIK	CATEGORIES	SENTIME N
Satu kata "telkomsel" ??browsing cepet RT @nezzarerraldin: indosat sialaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaannnnn	TSEL	browsing	1
Sebagai konsumen @Telkomsel saya sangat kecewa dengan iklan ini kemana saya harus mengadu?	TSEL	iklan	-1
dear @Telkomsel kenapa kalo saya nonton film dari viu, kuota yang berkurang malah kuota data reguler ya bukan kuota viu? ini kuota viu saya ada 97 GB gimana caranya agar tidak terbuang percuma karena saya takut dosa menyia2kan hal yg bisa berfaedah, apalagi di bulan puasa ini.	TSEL	kuota	0
@IndosatCare Malam. Nomor saya sudah tidak aktif sejak 27/4, sudah ke Galeri Indosat tapi katanya harus tunggu 180 hari baru bisa diaktifkan. Apa tidak ada cara lain yang lebih cepat, soalnya nomor saya sudah dimasukkan sejak lama untuk beberapa keperluan	INDOSAT	aktivasi	0

LAMPIRAN B. Jumlah Token Aspek

1. Aspek pada Data Train

'sinyal': 143, 'jaringan': 110, 'paket': 90, 'pulsa': 80, 'internet': 53, '4g': 38, 'sms': 38, 'koneksi': 30, 'pelayanan': 29, 'kuota': 28, 'layanan': 22, 'signal': 17, 'dm': 15, 'promo': 14, 'nomor': 12, 'registrasi': 12, 'kartu': 12, 'respon': 12, 'kualitas': 11, '3g': 11, 'harga': 11, 'modem': 11, 'iklan': 9, 'game': 8, 'wifi': 7, 'download': 7, 'bb': 7, 'komplain': 6, 'kecepatan': 6, 'browsing': 6, 'telfon': 6, 'telpon': 6, 'bonus': 6, 'tarif': 5, 'streaming': 5, 'speed': 5, 'twitter': 5, 'cs': 5, 'nelpon': 5, 'service': 4, 'paketan': 4, 'hadiah': 4, 'flash': 4, 'telp': 4, 'akses': 4, 'keluhan': 4, 'bis': 4, 'bbm': 4, 'aplikasi': 4, 'provider': 4, 'poin': 4, 'whatsapp': 3, 'main': 3, 'panggilan': 3, 'operator': 3, 'daftar': 3, 'tcash': 3, 'data': 3, 'tweet': 3, 'inovasi': 3, 'apn': 3, 'puk': 3, 'masa aktif': 2, 'pengguna': 2, 'lte': 2, 'email': 2, 'cepat': 2, 'voucher': 2, 'mahal': 2, 'fb': 2, 'konten': 2, 'trouble': 2, 'ooredoo': 2, 'tlp': 2, 'unlimited': 2, 'roaming': 2, 'pin': 2, 'online': 2, 'tagihan': 2, 'youtube': 2, 'hooq': 2, 'blackberry': 2, 'prime': 2, 'spotify': 2, 'pascabayar': 1, 'grapari': 1, 'bayar': 1, 'broswing': 1, 'telkomsel': 1, 'perdana': 1, 'connect': 1, 'as': 1, 'browser': 1, 'balas': 1, 'polling': 1, 'komunikasi': 1, 'migrasi': 1, 'review': 1, 'murah': 1, 'hotspot': 1, 'payment': 1, 'pes 2018': 1, 'upload': 1, 'iuran': 1, 'server': 1, 'update': 1, 'mycare': 1, 'bantuan': 1, 'bundling': 1, 'produk': 1, 'story': 1, 'inet': 1, 'akun': 1, 'servis': 1, 'nelepon': 1, 'purworejo': 1, 'aov': 1, 'sistem': 1, 'tm': 1, 'probolinggo': 1, '4G': 1, 'website': 1, 'galeri': 1, 'pembayaran': 1, 'kouta': 1, 'myim3': 1, 'koin': 1, 'ujian': 1, 'opera': 1, 'sim card': 1, '2g': 1, 'saldo': 1, 'nokia': 1, 'android': 1, 'tanggap': 1, 'kuis': 1, 'bislite': 1, 'pusla': 1, 'social': 1, 'center': 1, 'gsm': 1, 'bersaing': 1, 'regulasi': 1, 'stock': 1, 'cek': 1, 'telkomselsiaga': 1, 'mifi': 1, 'puas': 1, 'stabil': 1, 'notifikasi': 1, 'socmed': 1, 'call': 1, 'thr': 1, 'petugas': 1, 'visi': 1, 'misi': 1, 'nelp': 1, 'unreg': 1, 'kritik': 1, 'saran': 1, 'blbi': 1, 'komitmen': 1, 'quota': 1, 'jarangan': 1, 'login': 1, 'password': 1, 'kwalitas': 1, 'line': 1, 'iflix': 1, 'aset': 1, 'konsumen': 1, 'pasca': 1, 'prabayar': 1, 'aktivasi': 1, 'register': 1, 'mention': 1, 'perbaikan': 1, 'pemeliharaan': 1, 'iphone': 1, 'complain': 1, 'netflix': 1, 'konfirmasi': 1, 'tanggapan': 1, 'hspda':

1, 'prosedur': 1, 'sim': 1, 'program': 1, 'informasi': 1, 'sdm': 1, 'tlfn': 1, 'info': 1, 'performa': 1, 'logo': 1, 'janji': 1, 'jangkauan': 1

2. Aspek pada Data Tes

'sinyal': 50, 'paket': 24, 'jaringan': 18, 'internet': 15, 'pulsa': 15, 'pelayanan': 14, 'sms': 11, 'koneksi': 10, 'harga': 9, '4g': 8, 'kuota': 7, 'registrasi': 5, 'keluhan': 4, 'aktivasi': 4, 'telepon': 3, 'kecepatan': 3, 'poin': 3, 'wifi': 2, 'bonus': 2, 'intenet': 2, 'nomor': 2, 'browsing': 2, 'iklan': 2, 'promo': 2, 'sponsor': 1, 'tagihan': 1, 'moba': 1, 'perbaikan': 1, 'kartu': 1, '3g': 1, 'balasan': 1, 'pin': 1, 'tanggapan': 1, 'tcash': 1, 'streaming': 1, 'support': 1, 'telfon': 1, 'pake': 1, 'puk': 1, 'app': 1, 'event': 1, 'Tcash': 1, 'email': 1, 'fb': 1, 'bb': 1, 'perdana': 1, 'nelpon': 1, 'akses': 1, 'malem': 1, 'youtube': 1, 'aplikasi': 1, 'tarif': 1, 'kualitas': 1, 'respon': 1, 'galeri': 1, 'aktif': 1, 'jawaban': 1

LAMPIRAN C. Hasil Akurasi MGAN

1. Perulangan ke-0

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
0	23	0.849016	0.585	0.56	0.497182	0.614828	0.590717
1	46	0.864489	0.653333	0.688	0.683007	0.66753	0.725738
2	69	0.51438	0.696923	0.62	0.576644	0.682092	0.654008
3	92	0.327464	0.727778	0.696	0.692082	0.702985	0.734177
4	115	0.609217	0.755048	0.78	0.789924	0.766575	0.822785
5	138	0.6481	0.76699	0.744	0.707867	0.689058	0.744
6	161	0.191935	0.783512	0.752	0.75831	0.740341	0.793249
7	184	0.233498	0.790386	0.76	0.761109	0.747373	0.801688
8	207	0.210579	0.805787	0.776	0.766959	0.764204	0.776
9	230	0.141471	0.817823	0.728	0.689942	0.686514	0.728
10	253	0.108716	0.829701	0.724	0.701008	0.693032	0.724
11	276	0.274268	0.842375	0.724	0.697639	0.704715	0.724
12	299	0.058892	0.852117	0.732	0.706133	0.705179	0.732
13	322	0.198066	0.858402	0.736	0.697356	0.700683	0.736

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
14	345	0.22145	0.864501	0.736	0.708731	0.716557	0.736
15	368	0.020774	0.871717	0.744	0.711454	0.737362	0.744
16	391	0.023951	0.879493	0.752	0.72369	0.755036	0.752
17	414	0.011799	0.885419	0.744	0.708521	0.708842	0.744
18	437	0.021513	0.892055	0.748	0.71667	0.751914	0.748
19	460	0.00559	0.897717	0.736	0.706477	0.717878	0.736
Max		0.00559	0.897717	0.78	0.789924	0.766575	0.822785
Durasi		36.28434					

2. Perulangan ke-1

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
0	23	0.998632	0.525	0.532	0.476302	0.58166	0.624413
1	46	1.301047	0.557778	0.556	0.509127	0.576624	0.586498
2	69	1.023576	0.598462	0.644	0.641733	0.63052	0.679325
3	92	0.703594	0.628889	0.676	0.671199	0.651815	0.71308
4	115	0.744081	0.661106	0.696	0.684432	0.687563	0.734177

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
5	138	0.410362	0.68708	0.728	0.734348	0.71471	0.767932
6	161	0.432897	0.70988	0.764	0.759734	0.743231	0.805907
7	184	0.403799	0.723868	0.692	0.676889	0.711439	0.729958
8	207	0.410886	0.74154	0.768	0.769809	0.746925	0.810127
9	230	0.286466	0.758999	0.732	0.702951	0.683439	0.732
10	253	0.349303	0.772397	0.724	0.690637	0.66839	0.724
11	276	0.243797	0.784824	0.72	0.683128	0.673693	0.72
12	299	0.273873	0.793374	0.732	0.697499	0.687048	0.732
13	322	0.178152	0.802077	0.736	0.699335	0.683595	0.736
14	345	0.292374	0.81036	0.744	0.713075	0.694543	0.744
15	368	0.110121	0.819464	0.764	0.749763	0.747628	0.764
16	391	0.075445	0.828549	0.728	0.709125	0.706835	0.728
17	414	0.064096	0.835997	0.74	0.725636	0.727422	0.74
18	437	0.13255	0.845263	0.728	0.715439	0.727734	0.728
19	460	0.026847	0.852941	0.728	0.705551	0.710522	0.728
Max Acc		0.026847	0.852941	0.768	0.769809	0.747628	0.810127

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
Durasi		35.62477					

3. Perulangan ke-2

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
0	23	1.199685	0.505	0.536	0.480631	0.589408	0.565401
1	46	0.952289	0.557778	0.56	0.542269	0.560678	0.590717
2	69	0.936435	0.58	0.568	0.534498	0.560603	0.599156
3	92	0.701443	0.621111	0.648	0.639194	0.663222	0.683544
4	115	0.441171	0.665496	0.752	0.747665	0.719718	0.793249
5	138	0.42515	0.699029	0.732	0.733668	0.717075	0.772152
6	161	0.42201	0.725614	0.776	0.777737	0.753156	0.818565
7	184	0.326838	0.745109	0.764	0.727362	0.711823	0.764
8	207	0.578632	0.760177	0.696	0.675674	0.678647	0.696
9	230	0.527402	0.772169	0.752	0.728256	0.709856	0.752
10	253	0.163307	0.783293	0.736	0.707068	0.701453	0.736
11	276	0.209656	0.795088	0.764	0.727759	0.732144	0.764

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
12	299	0.25266	0.804986	0.716	0.696482	0.715364	0.716
13	322	0.117948	0.815607	0.764	0.735006	0.730956	0.764
14	345	0.135482	0.826163	0.76	0.737201	0.739997	0.76
15	368	0.129632	0.834946	0.74	0.74789	0.734416	0.780591
16	391	0.048236	0.8451	0.768	0.75249	0.758593	0.768
17	414	0.014051	0.852717	0.752	0.738142	0.735363	0.752
18	437	0.112008	0.861246	0.696	0.65306	0.646643	0.696
19	460	0.084405	0.866769	0.728	0.702114	0.685058	0.728
Max Acc		0.014051	0.866769	0.776	0.777737	0.758593	0.818565
Durasi		35.08795					

4. Perulangan ke-3

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
0	23	1.0733	0.46	0.52	0.442886	0.539633	0.610329
1	46	1.013549	0.544444	0.544	0.497376	0.539502	0.57384
2	69	0.690873	0.584615	0.544	0.492949	0.56158	0.57384

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
3	92	0.717237	0.622222	0.68	0.677423	0.656207	0.7173
4	115	0.500235	0.660228	0.696	0.694861	0.699675	0.734177
5	138	0.563384	0.681105	0.668	0.615544	0.668279	0.668
6	161	0.2529	0.707992	0.756	0.741647	0.727112	0.797468
7	184	0.202693	0.723309	0.74	0.706123	0.697982	0.74
8	207	0.387783	0.742521	0.708	0.6896	0.68744	0.708
9	230	0.413082	0.759877	0.744	0.718855	0.714186	0.744
10	253	0.345684	0.772397	0.728	0.700747	0.700644	0.728
11	276	0.363327	0.785924	0.76	0.733722	0.730242	0.76
12	299	0.15499	0.79679	0.752	0.737785	0.730331	0.752
13	322	0.053151	0.809314	0.756	0.729783	0.716623	0.756
14	345	0.059198	0.820603	0.736	0.731106	0.730166	0.736
15	368	0.008879	0.82914	0.764	0.738241	0.750373	0.764
16	391	0.054676	0.837342	0.728	0.699442	0.682477	0.728
17	414	0.191242	0.844111	0.688	0.673956	0.687467	0.688
18	437	0.101138	0.851286	0.748	0.717896	0.707152	0.748

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
19	460	0.03455	0.858867	0.748	0.723061	0.72036	0.748
Max Text Acc		0.008879	0.858867	0.764	0.741647	0.750373	0.797468
Durasi		35.09077					

5. Perulangan ke-4

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
0	23	1.0407	0.52	0.536	0.496916	0.531807	0.629108
1	46	0.949295	0.582222	0.564	0.540523	0.551632	0.594937
2	69	0.725095	0.590769	0.608	0.610531	0.59593	0.64135
3	92	0.865014	0.608889	0.564	0.525521	0.589894	0.594937
4	115	0.666876	0.642669	0.66	0.667719	0.660427	0.696203
5	138	0.445367	0.668409	0.732	0.738195	0.714387	0.772152
6	161	0.420364	0.686595	0.732	0.738623	0.722775	0.772152
7	184	0.397195	0.703186	0.752	0.72717	0.71158	0.752
8	207	0.266816	0.723884	0.756	0.716872	0.728893	0.756
9	230	0.42207	0.740123	0.696	0.678087	0.678118	0.696

epoch	batch	loss	accuracy	accuracy_test	f1	precision	recall
10	253	0.28407	0.752623	0.732	0.69033	0.684712	0.732
11	276	0.366795	0.766129	0.736	0.717836	0.702165	0.736
12	299	0.099717	0.774249	0.744	0.712789	0.701067	0.744
13	322	0.124633	0.788546	0.736	0.714947	0.709934	0.736
14	345	0.146564	0.799532	0.732	0.698927	0.699462	0.732
15	368	0.01787	0.810064	0.744	0.712431	0.699214	0.744
16	391	0.050701	0.821309	0.728	0.719292	0.715813	0.728
17	414	0.144655	0.824441	0.74	0.70869	0.703441	0.74
18	437	0.209801	0.830669	0.744	0.744905	0.729156	0.78481
19	460	0.054346	0.838016	0.748	0.731202	0.726967	0.748
Max Text Acc		0.01787	0.838016	0.756	0.744905	0.729156	0.78481
Durasi		35.55212					

BIODATA PENULIS



Penulis lahir di Kabupaten Malang pada tanggal 26 April 1995. Merupakan anak kedua dari dua bersaudara. Penulis telah menempuh beberapa pendidikan formal yaitu; SD Negeri Turen 1, SMP Negeri 01 Turen, dan SMK Negeri 4 Malang.

Pada tahun 2014 pasca kelulusan SMK penulis sempat bekerja di sebuah perusahaan percetakan di Kabupaten Gresik sebagai staff MIS hingga kurang lebih 1 tahun. Kemudian pada tahun 2015 penulis melanjutkan pendidikan dengan jalur SBMPTN di Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi – Institut Teknologi Sepuluh Nopember (ITS) Surabaya dan terdaftar sebagai mahasiswa dengan NRP 0521 15 4000 0124. Selama menjadi mahasiswa aktif mengikuti kegiatan kemahasiswaan beberapa kepanitiaan baik di dalam kampus serta di luar, serta pernah menjabat sebagai Staff Departemen Riset dan Teknologi HMSI FTIK ITS, Staff Divisi Sosial Masyarakat dan Kepala Divisi Pelayanan Sosial UKM KSR PMI ITS. Penulis juga pernah mengikuti pelatihan Google Develop Kejar (Mobile Web Specialist) di tahun 2018. Di bidang akademik, penulis aktif menjadi asisten dosen pada beberapa mata kuliah seperti Bahasa Pemrograman dan Algoritma Struktur Data.

Pada tahun keempat, penulis memiliki ketertarikan di bidang pengolahan data serta machine learning maka dari itu penulis mengambil bidang minat Akuisisi Data dan Diseminasi Informasi (ADDI). Penulis dapat dihubungi melalui email di magridprisca@gmail.com