



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IS184853

**ANALISIS SENTIMEN BERDASARKAN ASPEK PADA
PENDAPAT KONSUMEN PT. TELEKOMUNIKASI
SELULAR DAN PT. INDOSAT OOREDOO DENGAN
MENGUNAKAN PEMODELAN GATED
CONVOLUTIONAL NETWORK**

***ASPECT BASED SENTIMENT ANALYSIS IN
CUSTOMER OPINIONS PT. TELEKOMUNIKASI
SELULAR AND PT. INDOSAT OOREDOO USING
GATED CONVOLUTIONAL NETWORK***

**WENY KINANTI PUTRI
NRP 0521154000009**

**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 - IS184853

**ANALISIS SENTIMEN BERDASARKAN ASPEK
PADA PENDAPAT KONSUMEN PT.
TELEKOMUNIKASI SELULAR DAN PT. INDOSAT
OOREDOO DENGAN MENGGUNAKAN
PEMODELAN GATED CONVOLUTIONAL
NETWORK**

**WENY KINANTI PUTRI
NRP 0521154000009**

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

**DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember**

UNDERGRADUATE THESIS - IS184853

***ASPECT BASED SENTIMENT ANALYSIS IN
CUSTOMER OPINIONS PT. TELEKOMUNIKASI
SELULAR AND PT. INDOSAT OOREDOO USING
GATED CONVOLUTIONAL NETWORK***

**WENY KINANTI PUTRI
NRP 05211540000009**

**Supervisor
Renny Pradina K., ST, MT, SCJP**

**INFORMATION SYSTEM DEPARTMENT
Information Technology and Communication Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2019**

LEMBAR PENGESAHAN

**ANALISIS SENTIMEN BERDASARKAN ASPEK PADA
PENDAPAT KONSUMEN PT. TELEKOMUNIKASI
SELULAR DAN PT. INDOSAT OOREDOO DENGAN
MENGUNAKAN PEMODELAN GATED
CONVOLUTIONAL 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:

WENY KINANTI PUTRI
NRP 0521154000009

Surabaya, Juni 2019

KEPALA
DEPARTEMEN SISTEM INFORMASI



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

LEMBAR PERSETUJUAN

**ANALISIS SENTIMEN BERDASARKAN ASPEK PADA
PENDAPAT KONSUMEN PT. TELEKOMUNIKASI
SELULAR DAN PT. INDOSAT OOREDOO DENGAN
MENGUNAKAN PEMODELAN GATED
CONVOLUTIONAL 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:

WENY KINANTI PUTRI
NRP 0521154000009

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

Renny Pradina K., ST, MT, SCJP

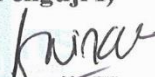
Nur Aini Rakhmawati S.Kom, M.Sc.Eng

Faisal Johan Atletiko, S.Kom, M.Kom




(Pembimbing I)


(Penguji I)


(Penguji II)

Halaman ini sengaja dikosongkan

**ANALISIS SENTIMEN BERDASARKAN ASPEK PADA
PENDAPAT KONSUMEN PT. TELEKOMUNIKASI
SELULAR DAN PT. INDOSAT OOREDOO DENGAN
MENGUNAKAN PEMODELAN GATED
CONVOLUTIONAL NETWORK**

Nama Mahasiswa : Weny Kinanti Putri
NRP : 0521154000009
Departemen : Sistem Informasi FTIK-ITS
Pembimbing I : Renny Pradina K., ST, MT, SCJP

ABSTRAK

Operator telekomunikasi saat ini semakin berlomba untuk menjaga dan meningkatkan jumlah pelanggannya dengan terus meningkatkan kualitas layanannya. Tidak sedikit pelanggan operator telekomunikasi menyampaikan pendapatnya mengenai layanan operator yang digunakannya melalui sosial media, salah satunya melalui Twitter. Karena saat ini sosial media telah menjadi salah satu media utama yang selalu digunakan masyarakat khususnya di Indonesia untuk saling berbagi informasi secara cepat. Informasi yang tersebar di sosial media sebagian besarnya merupakan opini pribadi dari pengguna terhadap suatu objek. Hal tersebut bisa dimanfaatkan perusahaan telekomunikasi untuk mengetahui pandangan pelanggan mengenai layanannya dan menjadi bahan decision making dalam memperbaiki dan meningkatkan layanannya.

Menanggapi hal tersebut, analisis aspek berdasarkan sentimen dapat dilakukan untuk mengetahui hal menjadi konsen keluhan pelanggan dan seberapa pembagian

sentimennya. Sentimen adalah pandangan individu pada suatu objek. Macam sentimen yang digunakan pada penelitian ini ada 4 jenis yaitu sentimen positif, negatif, netral, dan konflik. Analisis aspek berdasarkan sentimen merupakan pendalaman dari teknis analisis sentimen yang telah banyak dilakukan pada pemrosesan bahasa alami.

Gated Convolutional Network merupakan salah algoritma untuk melakukan analisis sentimen berdasarkan aspek. Model GCNN dapat menghasilkan akurasi hingga 82.64% dalam memprediksi sentimen berdasarkan aspek. Algoritma GCNN dibandingkan dengan algoritma CNN masih menghasilkan akurasi yang lebih tinggi 1.6% dibandingkan akurasi model CNN yang hanya mencapai 81.04% dalam melakukan sentimen analisis. Dengan nilai akurasi yang lebih tinggi, GCNN memiliki waktu komputasi yang lebih lama yaitu 182.85 detik untuk setiap percobaan training dan testing data dibandingkan rata-rata waktu komputasi yang dihabiskan oleh model CNN yang hanya 54.93 detik. Perbedaan dari kedua algoritma ini ialah GCNN dapat mencapai akurasi yang lebih tinggi namun memiliki waktu komputasi yang lebih lama dibanding CNN yang waktu komputasinya lebih cepat namun mencapai akurasi yang tidak setinggi GCNN.

Kata Kunci: Analisis sentimen berdasarkan aspek, Twitter, Operator Telekomunikasi, Gated Convolutional Network, Convolutional Neural Network

**ASPECT BASED SENTIMENT ANALYSIS IN
CUSTOMER OPINIONS PT. TELEKOMUNIKASI
SELULAR AND PT. INDOSAT OOREDOO USING
GATED CONVOLUTIONAL NETWORK**

Name : Weny Kinanti Putri
NRP : 0521154000009
Department : Information System FTIK-ITS
Supervisor : Renny Pradina K., ST, MT, SCJP

ABSTRACT

Telecommunications operators are star to increase the number of customers by increas the quality of their services. Not a few customer operators who express their opinions regarding service operators that they use through social media, one of them through Twitter. Because now social media has become one of the main media that has always been used by special communities in Indonesia to share information quickly. Information that is spread on social media is partly a personal opinion of the user towards an object. This can utilize telecommunication companies to find out the views of customers about their services and become material for making decisions in improving and improving their services.

Responding to this, analysis of aspects based on sentiments can be done to find out the things that are considered by customers and regulate the distribution of sentiments. Sentiment is the individual view of an object. There are 4 types of sentiments used in this study, namely positive, negative, neutral, and conflict sentiments. Sentiment based analysis is a deepening of sentiment technical analysis that has been carried out on natural language approvals.

Gated Convolutional Network is an algorithm for analyzing sentiment based on aspects. The GCNN model can produce up to 82.64% in predicting aspects based on sentiments. The GCNN algorithm compared to the CNN algorithm still produces 1.6% higher accuracy than the accuracy of the CNN model which only reached 81.04% in conducting sentiment analysis. With a higher testing value, GCNN has a longer computation time of 182.85 seconds for each training experiment and test data compared to the average computing time spent by the CNN model which is only 54.93 seconds. The difference between the two algorithms is that GCNN can achieve higher accuracy but has a longer computation time than CNN whose computation time is faster but achieves accuracy that is not as high as the GCNN.

Keywords: Aspect Based Sentiment analysis, Twitter, Telecommunication Operators, Gated Convolutional Network, Convolutional Neural Network

KATA PENGANTAR

Dengan mengucapkan rasa syukur kepada Tuhan Yang Maha Pengasih dan Maha Penyayang atas izin-Nya penulis dapat menyelesaikan buku yang sederhana ini dengan judul Analisis Sentimen Berdasarkan Aspek pada Pendapat Konsumen PT. Telekomunikasi Selular dan PT. Indosat Ooredoo dengan Menggunakan Pemodelan Gated Convolutional Network. Dalam penyelesaian Tugas Akhir ini, penulis diiringi oleh pihak-pihak yang selalu memberi dukungan, saran, dan doa sehingga penelitian berlangsung dengan lancar. Secara khusus penulis mengucapkan terima kasih dari lubuk hati terdalam kepada:

1. Tuhan, yang selalu menemani dan membimbing penulis dalam segala aspek kehidupan.
2. Ibu Mahendrawathi ER. S.T., M.Sc., Ph.D. selaku Ketua Departemen Sistem Informasi ITS Surabaya.
3. Ibu Renny Pradina K., ST, MT, SCJP selaku dosen pembimbing yang telah mencurahkan segenap tenaga, waktu dan pikiran dalam penelitian ini, serta memberikan motivasi yang membangun.
4. Ibu Nur Aini Rakhmawati S.Kom, M.Sc.Eng dan Bapak Faisal Johan Atletiko, S.Kom, M.Kom selaku dosen penguji yang telah memberikan kritik dan saran yang membuat kualitas penelitian ini lebih baik lagi.
5. Segenap dosen dan karyawan Departemen Sistem Informasi ITS.
6. Orang tua penulis, yang tiada hentinya mendoakan dan memberikan dukungan kepada penulis.
7. Achmad Yazid Maulana Wachid yang selalu mendoakan dan memberi dukungan penuh pada penulis dalam menjalani suka dan duka menyelesaikan penelitian ini.
8. Serta pihak lainnya yang berkontribusi dalam tugas akhir yang belum dapat penulis sebutkan satu per satu.

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, Juni 2019

Penulis

DAFTAR ISI

| | |
|--|-------------------------------------|
| LEMBAR PENGESAHAN..... | Error! Bookmark not defined. |
| LEMBAR PERSETUJUAN..... | iii |
| ABSTRAK..... | v |
| ABSTRACT..... | vii |
| KATA PENGANTAR..... | ix |
| DAFTAR ISI..... | xi |
| DAFTAR GAMBAR..... | xv |
| DAFTAR TABEL..... | xvii |
| DAFTAR KODE..... | xix |
| 1 BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 2 |
| 1.3 Batasan Permasalahan..... | 3 |
| 1.4 Tujuan..... | 3 |
| 1.5 Manfaat..... | 3 |
| 1.6 Relevansi..... | 4 |
| 2 BAB II TINJAUAN PUSTAKA..... | 5 |
| 2.1 Penelitian Sebelumnya..... | 5 |
| 2.2 Dasar Teori..... | 7 |
| 2.2.1 Machine Learning..... | 7 |
| 2.2.2 Natural Language Processing (NLP)..... | 8 |
| 2.2.3 Word Embeddings..... | 8 |
| 2.2.3.1 GloVe..... | 9 |
| 2.2.4 Sentiment Analysis..... | 9 |
| 2.2.5 Aspect-Based Sentiment Analysis..... | 10 |
| 2.2.5.1 Aspect-category Sentiment Analysis.. | 11 |
| 2.2.6 Gated Convolutional Network..... | 12 |
| 2.2.7 Social Media..... | 14 |
| 2.2.7.1 Twitter..... | 14 |
| 2.2.8 Operator Telekomunikasi..... | 14 |
| 3 BAB III METODOLOGI..... | 17 |
| 3.1 Diagram Metodologi..... | 17 |
| 3.2 Arsitektur Penelitian..... | 17 |
| 3.3 Uraian Metodologi..... | 19 |

| | | |
|---------|--|----|
| 3.3.1 | Studi Literatur | 19 |
| 3.3.2 | Pengumpulan Data | 19 |
| 3.3.3 | Pre-Processing Data | 19 |
| 3.3.3.1 | Pelabelan Tweet | 20 |
| 3.3.3.2 | Word Embedding menggunakan Word2Vec | 21 |
| 3.3.4 | Pelatihan dan Pengujian Data..... | 21 |
| 3.3.5 | Analisis Model | 21 |
| 3.3.6 | Penyusunan Laporan Tugas Akhir | 23 |
| 4 | BAB IV PERANCANGAN | 25 |
| 4.1 | Perancangan Pengumpulan Data..... | 25 |
| 4.2 | Perancangan Pengolahan Data | 25 |
| 4.2.1 | Pemberian Label Dataset..... | 25 |
| 4.2.2 | Pra-proses Data | 30 |
| 4.2.2.1 | Mempersiapkan Data | 30 |
| 4.2.2.2 | Word Embedding | 31 |
| 4.2.3 | Perancangan Pembuatan Model GCNN | 32 |
| 4.2.3.1 | Perancangan <i>Training</i> Model GCNN . | 32 |
| 4.2.3.2 | Perancangan <i>Testing</i> Model GCNN.... | 33 |
| 5 | BAB V IMPLEMENTASI | 35 |
| 5.1 | Lingkungan Implementasi..... | 35 |
| 5.2 | Pra-Proses Data | 36 |
| 5.2.1 | Mempersiapkan Data | 36 |
| 5.2.2 | Word Embedding | 38 |
| 5.3 | Pembuatan Model GCNN | 40 |
| 5.3.1 | Pembuatan Training Model GCNN..... | 40 |
| 5.3.2 | Pembuatan Testing Model GCNN | 49 |
| 6 | BAB VI HASIL DAN PEMBAHASAN..... | 53 |
| 6.1 | Hasil Pelabelan Tweet..... | 53 |
| 6.2 | Hasil Word Embedding..... | 57 |
| 6.3 | Hasil Pembuatan Model GCNN..... | 58 |
| 6.3.1 | Konfigurasi Parameter Awal..... | 58 |
| 6.3.2 | Pengaruh Optimizer dan Filter Region Size..... | 58 |
| 6.3.2.1 | Hasil Percobaan Optimizer Adagrad .. | 58 |
| 6.3.2.2 | Hasil Percobaan Optimizer Adadelata.. | 61 |
| 6.3.2.3 | Hasil Percobaan Optimizer Adam | 62 |

| | |
|--|----|
| 6.3.2.4 Hasil Perbandingan Akurasi per Optimizer | 64 |
| 6.3.3 Hasil Perbandingan Dengan Algoritma Lain | 64 |
| 6.3.4 Hasil Performa Tiap Sentimen dengan Model GCNN | 65 |
| 6.3.5 Grafik ROC/AUC | 66 |
| 7 BAB VII KESIMPULAN DAN SARAN | 67 |
| 7.1 Kesimpulan | 67 |
| 7.2 Saran | 67 |
| 8 DAFTAR PUSTAKA | 69 |
| 9 BIODATA PENULIS | 71 |
| LAMPIRAN A. DATA BERLABEL | 73 |
| LAMPIRAN B. HASIL AKURASI..... | 77 |

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Arsitektur GCN oleh Dauphin [11] | 13 |
| Gambar 3.1 Metode Penelitian..... | 17 |
| Gambar 3.2 Arsitektur Penelitian..... | 18 |
| Gambar 5.1 Contoh 2 Objek Embedding | 45 |
| Gambar 5.2 Hasil Parameter Model GCNN..... | 46 |
| Gambar 6.1 Hasil Embedding | 57 |
| Gambar 6.2 Rata-rata Akurasi Tiap Epoch | 59 |
| Gambar 6.3 Akurasi Rata-rata tiap epoch | 60 |
| Gambar 6.4 Akurasi Tertinggi dari Percobaan..... | 60 |
| Gambar 6.5 Akurasi Tertinggi dari Percobaan..... | 62 |
| Gambar 6.6 Akurasi Tertinggi dari Percobaan..... | 63 |
| Gambar 6.7 Perbandingan Akurasi Tiap Optimizer | 64 |
| Gambar 6.8 Confusion Matrix GCNN | 65 |
| Gambar 6.9 Grafik ROC/AUC..... | 66 |

Halaman ini sengaja dikosongkan

DAFTAR TABEL

| | |
|---|----|
| Tabel 2.1 Literatur 1 | 5 |
| Tabel 2.2 Literatur 2 | 6 |
| Tabel 2.3 Literatur 3 | 7 |
| Tabel 2.4 Perbedaan Kategori Sentimen | 9 |
| Tabel 2.5 Pemetaan Quality of Service Telekomunikasi pada Kategori Aspek..... | 11 |
| Tabel 3.1 Pelabelan Tweet | 20 |
| Tabel 4.1 Label Aspek pada Dataset | 25 |
| Tabel 4.2 Label Sentimen pada Dataset | 28 |
| Tabel 4.3 Pelabelan Aspek dan Sentimen pada Dataset..... | 29 |
| Tabel 4.4 Parameter Model GCNN | 32 |
| Tabel 4.5 Parameter <i>Training</i> Model GCNN..... | 33 |
| Tabel 5.1 Spesifikasi Hardware | 35 |
| Tabel 5.2 Teknologi yang Digunakan untuk Mengembangkan Model | 35 |
| Tabel 6.1 Distribusi Label Aspek dan Sentimen Data Train.. | 53 |
| Tabel 6.2 Distribusi Label Aspek dan Sentimen Data Tes..... | 55 |
| Tabel 6.3 Jumlah Vocab..... | 57 |
| Tabel 6.4 Konfigurasi Awal Model..... | 58 |
| Tabel 6.5 Hasil Akurasi Model | 59 |
| Tabel 6.6 Hasil Akurasi Model | 61 |
| Tabel 6.7 Hasil Akurasi Model | 62 |
| Tabel 6.8 Perbandingan Akurasi dengan Algoritma Lain | 64 |
| Tabel 6.9 Performa Sentimen..... | 65 |

Halaman ini sengaja dikosongkan

DAFTAR KODE

| | |
|--|----|
| Kode 5.1 Persiapan Data | 37 |
| Kode 5.2 Word Embedding..... | 39 |
| Kode 5.3 Pembaca Dataset..... | 41 |
| Kode 5.4 Getdata dan Pemanggilan Fungsi Cleaning | 42 |
| Kode 5.5 Pemanggilan Kelas DataIndosel | 43 |
| Kode 5.6 Potongan Kode Parameter Model GCNN..... | 44 |
| Kode 5.7 Potongan Kode Perubahan State Model Train..... | 46 |
| Kode 5.8 Potongan Kode Perulangan Epoch pada Train | 47 |
| Kode 5.9 Potongan Kode Algoritma Training | 47 |
| Kode 5.10 Potongan Kode Method Forward()..... | 48 |
| Kode 5.11 Potongan Kode Prediksi Kalimat dalam Train | 49 |
| Kode 5.12 Potongan Kode Mengakses Method Eval..... | 49 |
| Kode 5.13 Potongan Kode Testing Data..... | 50 |

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

Dewasa kini, pengguna smartphone di Indonesia mengalami pertumbuhan yang pesat. Berdasarkan laporan Lembaga Riset Digital Marketing Emarketer, pengguna aktif smartphone di Indonesia bertumbuh dari 55 juta orang pada tahun 2015 menjadi lebih dari 100 juta orang pada 2018. Jumlah tersebut akan menjadikan Indonesia sebagai negara dengan pengguna aktif smartphone terbesar keempat di dunia setelah Cina, India, dan Amerika [1]. Untuk mengoperasikan smartphone, pemerintah kini telah memberlakukan peraturan registrasi kartu prabayar dengan validasi data dukcapil. Hingga batas akhir registrasi ulang pada 30 April 2018, tercatat 254.792.159 nomor pelanggan yang telah terdaftar. Sebuah sumber menyebutkan, dari total nomor yang terdaftar, jumlah pelanggan Telkomsel sebanyak 150 juta, XL Axiata 45 juta pelanggan, Indosat Ooredoo 34 juta pelanggan, Tri Indonesia 17 juta dan Smartfren 7 juta [2]. Dari angka tersebut, *We Are Social* mengungkapkan pengguna internet indonesia telah menyentuh angka 132,7 juta dengan 130 juta diantaranya merupakan pengguna aktif media sosial. Platform media sosial yang digemari masyarakat Indonesia diantaranya adalah Youtube sebanyak 43%, facebook 41%, instagram 38%, twitter 27%, dan lain sebagainya [3].

Kemudahan penulisan pesan / status menjadikan media sosial digemari penggunaanya dari seluruh lapisan masyarakat tidak tergantung usia maupun pekerjaan. Media sosial dapat menjadi sarana penyebaran informasi ataupun opini kepada sesama pengguna media sosial. Pesan yang ada di media sosial

dapat berupa saran, kritik, maupun opini lainnya. Banyaknya macam pesan diposting pengguna pada media sosial dapat menjadi bahan pengamatan untuk mendapatkan informasi aspek yang dibahas pengguna pada media sosial. Hal ini dapat menjadi bahan analisis bagi penyedia layanan telekomunikasi dalam meningkatkan kualitas layanannya untuk tetap bertahan pada persaingan ketat untuk mempertahankan dan menambah jumlah pelanggannya. *Aspect based Sentiment Analysis* merupakan metode untuk mendeteksi aspek yang sering dibahas pada suatu kumpulan opini serta memperkirakan seberapa positif atau negatifnya aspek tersebut. Metode ini akan sangat membantu penyedia layanan telekomunikasi dalam mendeteksi review pelanggan terhadap layanannya.

Oleh karena itu, penelitian ini menawarkan solusi dalam melakukan *aspect-based sentiment analysis* terhadap *review* pelanggan operator telekomunikasi Telkomsel dan Indosat yang didapatkan melalui sosial media twitter. Deteksi aspek dan polaritas sentimen pada *review* pelanggan dapat membantu perusahaan dalam melakukan *decision making* menanggapi kekurangan, keluhan, dan permasalahan yang dihadapi pelanggannya. Penelitian ini menggunakan metode *Gated Convolutional Network* yang merupakan salah satu bentuk metode *text mining* yang menghasilkan sentimen dari aspek-aspek yang terdeteksi pada suatu teks [4].

1.2 Rumusan Masalah

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

1. Bagaimana menentukan aspek berdasarkan kategori yang digunakan untuk memberi label pada dataset operator telekomunikasi Telkomsel dan Indosat?
2. Bagaimana menentukan jenis sentimen yang akan diterapkan pada setiap aspek yang dianalisis?
3. Bagaimana mengetahui performa metode *Gated Convolutional Network* dalam menganalisis aspek dan

polaritas sentimen dataset operator telekomunikasi Telkomsel dan Indosat?

1.3 Batasan Permasalahan

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

1. Sumber data berasal dari Twitter dengan bahasa Indonesia.
2. Data yang digunakan terkait dengan topik terkait penyedia layanan telekomunikasi Telkomsel dan Indosat yang ada di Indonesia.
3. Akurasi yang dihasilkan ialah akurasi model dalam menganalisis sentimen.

1.4 Tujuan

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

1. Mendapatkan aspek berdasarkan kategori-kategori yang termasuk layanan operator telekomunikasi yang ditentukan untuk digunakan sebagai label pada dataset.
2. Menentukan sentimen berdasarkan 4 jenis sentimen positif, negatif, netral, dan konflik pada aspek yang dihasilkan.
3. Melakukan uji coba *training* dan *test* pada *dataset* untuk menghasilkan model dengan performa terbaik menggunakan metode *Gated Convolutional Network*.

1.5 Manfaat

Berikut merupakan manfaat yang diharapkan dari penelitian yang dilakukan:

Bagi Akademisi:

1. Memahami *aspect based sentiment* analisis dengan menggunakan model *Gated Convolutional Network*
2. Memahami cara kerja metode *Gated Convolutional Network*

Bagi Organisasi:

1. Sebagai bahan *decision making* perusahaan untuk terus memperbaiki layanannya

1.6 Relevansi

Proposal tugas akhir ini dikatakan layak menjadi TA dari penulis karena hasil dari TA ini akan membantu PT Telkomsel dan PT Indosat Ooredoo dalam mengetahui aspek layanannya yang sering dibahas pengguna untuk bahan *decision making* peningkatan layanannya. Mata kuliah yang digunakan dalam mengembangkan TA ini adalah mata kuliah Sistem Cerdas, Sistem Pendukung Keputusan, Kecerdasan Bisnis, Penggalian Data dan Analitika Bisnis, dan Pemrosesan Bahasa Alami yang merupakan mata kuliah bidang keilmuan Laboratorium Akuisisi Data dan Diseminasi Informasi.

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 melaksanakan proses-proses dalam penelitian, seperti yang terdapat pada tabel dibawah. 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 | A System for Aspect-based Opinion Mining of Hotel Reviews |
| Nama Peneliti, Tahun Penelitian | Isidoros Perikos, Konstantinos Kovas, Foteini Grivokostopoulou and Ioannis Hatzilygeroudis (2018) |
| Metodologi | <ol style="list-style-type: none">1. Indexing and Preprocessing of Reviews2. Topic Modelling using LDA3. Textual Analysis of Reviews4. Polarity Recognition using Naïve Bayes5. Graphic Visualization |
| Keterkaitan penelitian | Penelitian berfokus pada anlisis aspek yang ada pada review pelanggan hotel. Analisis dilakukan dengan memodelkan topik menggunakan LDA, kemudian menganalisis dependensi kata, lalu mengklasifikasikan aspek ke kategori polaritas |

| | |
|--|--|
| | yang tepat, dan mevisualkan hasil analisis yang telah dilakukan. |
|--|--|

Tabel 2.2 Literatur 2

| | |
|---------------------------------|--|
| Judul | Aspect Based Sentiment Analysis with Gated Convolutional Network |
| Nama Peneliti, Tahun Penelitian | Wei Xue and Tao Li (2018) |
| Metodologi | <ol style="list-style-type: none"> 1. Labeling dataset 2. Word Embedding (GloVe) 3. Training Data 4. Analysis |
| Relevansi Penelitian | <p>Penelitian berfokus pada analisis aspek dan klasifikasi polaritas dari review Restaurant dan Laptop dari data SemEval 2014. Analisis dilakukan dengan membandingkan model Gated Convolutional Network with Aspect Embedding dengan model Neural Network lain yang telah populer untuk analisis sentimen atau klasifikasi kalimat.</p> |

Tabel 2.3 Literatur 3

| | |
|---------------------------------|---|
| Judul | Joint Aspect and Polarity Classification for Aspect-based Sentiment Analysis with End-to-End Neural Networks |
| Nama Peneliti, Tahun Penelitian | Martin Schmitt, Simon Steinheber, Konrad Schreiber (2018) |
| Metodologi | <ol style="list-style-type: none"> 1. Labeling dataset 2. Word Embedding (Word2vec, FastText, GloVe) 3. Training Data 5. Analisis |
| Relevansi Penelitian | Penelitian berfokus pada analisis aspek dan klasifikasi polaritas dari feedback customer pada social media dari data GermEval 2017. Analisis dilakukan dengan membandingkan pendekatan pipeline dan end-to-end pada model LSTM dan CNN. |

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 Machine Learning

Machine learning merupakan pendekatan dalam artificial intelligence (AI) yang banyak digunakan untuk menggantikan

atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi. Sesuai namanya, machine learning mencoba menirukan bagaimana proses manusia atau makhluk cerdas belajar dan menggeneralisasi. Setidaknya ada dua aplikasi utama dalam machine learning yaitu, klasifikasi dan prediksi. Ciri khas dari machine learning adalah adanya proses pelatihan, pembelajaran, atau training. Oleh karena itu, machine learning membutuhkan data untuk dipelajari yang disebut data training. Klasifikasi adalah metode dalam machine learning yang digunakan oleh mesin untuk memilah atau mengklasifikasikan obyek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain. Sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka keluaran dari suatu data masukan berdasarkan data yang sudah dipelajari dalam training. Metode machine learning yang paling populer yaitu sistem pengambilan keputusan, support vector machine (svm), dan neural network [5].

2.2.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah analisis data linguistik, umumnya menggunakan data tekstual seperti dokumen atau publikasi, menggunakan metode komputasi. Tujuan pemrosesan bahasa alami umumnya untuk membangun representasi teks yang menambah struktur pada bahasa alami yang tidak terstruktur, dengan mengambil keuntungan dari wawasan linguistik. Struktur ini dapat bersifat sintaksis, menangkap hubungan gramatikal antara konstituen teks atau lebih semantik, menangkap makna yang disampaikan oleh teks. Natural Language Processing digunakan dalam sistem biologi untuk mengembangkan aplikasi yang mengintegrasikan informasi yang diekstraksi dari literatur dengan sumber data biologis lainnya [6].

2.2.3 Word Embeddings

Word Embedding adalah suatu teknik yang bertujuan untuk memetakan kata menjadi suatu nilai vektor tertentu. *Word Embedding* digunakan sebagai representasi dari input sehingga menjadi aset penting untuk berbagai macam

tugas pada pemrosesan bahasa alami (NLP). Terdapat banyak macam teknik untuk mendapatkan *word embedding*, namun sebagian besar menggunakan pemodelan bahasa neural network [7]. *Word Embedding* dapat membantu untuk menentukan kemiripan kata berdasarkan konteks tertentu dilihat dari nilai vektor kata tersebut. Pada penelitian ini, *word embedding* dilakukan untuk mengubah *dataset* menjadi nilai vektor yang dibutuhkan sebagai input untuk *training*.

2.2.3.1 GloVe

Word2vec merupakan salah satu bentuk dari *word embedding* dimana dikembangkan oleh Google. *Word2vec* sendiri termasuk dalam kategori *neural network* yang menggunakan *hidden layer* dan beberapa *non-linier layer* didalam algoritmanya. Terdapat 2 jenis *word2vec* yaitu *Continuous Bag-of-words (CBOW)* dan *Skip-Gram Model* [8]. *Word embedding* yang dilakukan pada penelitian ini menggunakan *pretrained word2vec* dengan algoritma *skipgram*.

2.2.4 Sentiment Analysis

Sentiment Analysis atau analisis sentimen dapat didefinisikan sebagai proses penggalian sikap, pendapat, pandangan, dan emosi dari teks, sumber pidato, tweet, dan basis data melalui pemrosesan bahasa alami. Analisis sentimen dapat disebut juga sebagai analisis subjektif, penggalian opini, dan ekstraksi penilaian. Analisis sentimen melibatkan klasifikasi opini dalam teks ke dalam jenis sentimen seperti “positif” atau “negatif” atau “netral” [9]. Pada penelitian ini menggunakan 4 jenis sentimen yaitu positif, negatif, netral, dan konflik. Berikut penjelasan sentimen dan contoh perbedaannya:

Tabel 2.4 Perbedaan Kategori Sentimen

| Tweet | Sentimen | Penjelasan |
|-------|----------|------------|
|-------|----------|------------|

| | | |
|---|---------|---|
| Mudah mudahan @Telkomsel bisa lebih baik dlm pelayanan pelanggan krn terus terang urusan sinyal dan coverage saya sangat puas dg Telkomsel. | Positif | Sentimen Positif adalah opini yang memiliki kecenderungan makna baik atau positif |
| Kok sinyal indosat gue suka ilang2an gini,tolong segera diperbaiki,Terima Kasih | Negatif | Sentimen Negatif adalah opini yang memiliki kecenderungan makna buruk atau negatif |
| @Telkomsel min saya mau nanya, ngga sengaja saya lewat jatuh tempo tagihan kartu halo, gimana ya mengurusnya? Apakah denda atau bagaimana? | Netral | Sentimen Netral adalah opini yang memiliki tidak kecenderungan makna baik maupun buruk |
| @thasyasyaa indosat bagus kok. Lumayan lah, tpi agak mahal.. Tp yg paket murah ada.. | Konflik | Sentimen Konflik adalah opini yang memiliki kecenderungan makna baik dan buruk pada satu aspek yang sama [10] |

2.2.5 Aspect-Based Sentiment Analysis

Aspect-based Sentiment Analysis (ABSA) bertujuan menganalisis lebih dalam untuk mendeteksi aspek dan

mengklasifikasikan sentimen dari setiap aspek yang ada pada suatu teks [10]. Terdapat pendekatan yang membagi Aspect-based Sentiment Analysis menjadi 2 subtask yang berbeda, yaitu Aspect-category Sentiment Analysis (ACSA) dan Aspect-term Sentiment Analysis (ATSA). Aspect-category Sentiment Analysis (ACSA) bertujuan untuk memprediksi polaritas sentimen pada aspek, dimana aspek merupakan salah satu dari kumpulan kategori aspek yang telah didefinisikan sebelumnya, dari suatu teks yang dianalisis. Aspect-term Sentiment Analysis (ATSA) bertujuan untuk memprediksi polaritas entitas yang diprediksi sebagai aspek, yang dapat berupa satu atau lebih kata yang muncul pada teks [4]. Pada penelitian ini, pendekatan yang digunakan ialah Aspect-category Sentiment Analysis (ACSA).

2.2.5.1 Aspect-category Sentiment Analysis

Aspect-category Sentiment Analysis (ACSA) bertujuan untuk memprediksi polaritas sentimen pada aspek, dimana aspek merupakan salah satu dari kumpulan kategori aspek yang telah didefinisikan sebelumnya, dari suatu teks yang dianalisis [4]. Pendefinisian kategori aspek dilakukan dengan mengacu pada Quality of Service yang digunakan oleh PT. Indosat Ooredoo dalam melaporkan pencapaian standar kualitas pelayanan. Berikut pemetaan quality of service operator telekomunikasi pada kategori aspek yang digunakan pada penelitian ini.

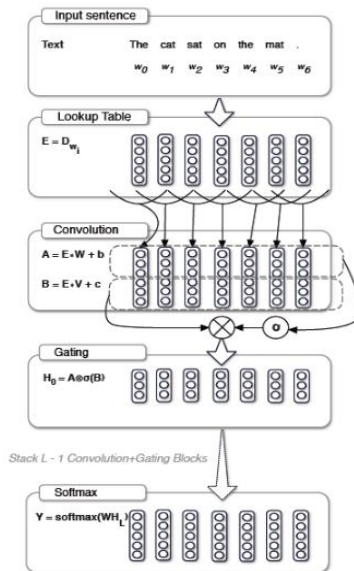
Tabel 2.5 Pemetaan Quality of Service Telekomunikasi pada Kategori Aspek

| No. | Standar Kualitas Layanan | Kategori Aspek |
|-----|---------------------------------------|----------------|
| 1. | <i>Throughput</i> | Kecepatan |
| 2. | <i>Packet Loss</i> | Jaringan |
| 3. | Standar Endpoint Service Availability | Layanan |

| | | |
|----|--|---|
| 4. | Standar Kinerja Layanan Pesan Singkat | Campaign |
| 5. | Standar Kinerja Tagihan | Tagihan |
| 6. | Standar Pemenuhan Permohonan Aktivasi | - Registrasi - Aktivasi |
| 7. | Service Level Call Center Layanan Pengguna | Customer Service |
| 8. | Standar Penanganan Keluhan Umum Pengguna | - Harga - Pengalaman - Promosi - Masa Aktif - Pilihan Paket - Aplikasi - Report |

2.2.6 Gated Convolutional Network

Gated Convolutional Network merupakan metode language modeling yang dibuat untuk memperbaiki metode *recurrent neural network*. Metode ini memungkinkan untuk memproses token yang berurutan secara paralel sehingga meningkatkan keefisienan proses komputasi.



Gambar 2.1 Arsitektur GCN oleh Dauphin [11]

Sama dengan metode CNN, untuk memproses tipe data teks, teks harus dirubah terlebih dahulu menjadi menjadi vektor dengan proses *word embedding*. Namun bedanya *Gated Convolutional Network* menggunakan *multiple layer of gated convolution* untuk memproses hasil dari *word embedding*. Setiap layer dari metode *Gated Convolutional Network* terdiri dari *convolutional block* yang menghasilkan dua *output convolutional* yang terpisah dan *gating block* yang menggunakan satu *output convolutional* untuk menjaga yang lain. Kemudian, langkah softmax berfungsi menghasilkan prediksi seluruh kosakata pada setiap stepnya.

Pada arsitektur diatas, B berisi 'gate' yang mengontrol informasi apa dari A yang diteruskan ke lapisan berikutnya dalam hierarki. Secara konseptual, mekanisme *gating* penting karena memungkinkan pemilihan kata atau fitur yang penting untuk memprediksi kata berikutnya, dan menyediakan mekanisme untuk mempelajari dan menyampaikan informasi

yang relevan saja. Mirip dengan ReLU, mekanisme *gating* juga menyediakan lapisan dengan kemampuan non-linear, sambil memberikan jalur linier untuk gradien selama *backpropagation* (dengan demikian mengurangi masalah gradien menghilang) [11].

2.2.7 Social Media

Media sosial adalah sebuah media online, dengan para penggunanya bisa dengan mudah berpartisipasi, berbagi, dan menciptakan isi meliputi blog, jejaring sosial, wiki, forum dan dunia virtual. Media sosial memudahkan pengguna untuk berinteraksi dengan banyak orang, memperluas pergaulan, jarak dan waktu bukan lagi masalah, lebih mudah dalam mengekspresikan diri, penyebaran informasi dapat berlangsung secara cepat, biaya lebih murah [12]. Media sosial akan menjadi media bagi pelanggan operator telekomunikasi untuk mengekspresikan opini mereka terhadap layanan operator telekomunikasi yang mereka gunakan. Opini pelanggan akan menjadi dataset pada penelitian ini.

2.2.7.1 Twitter

Twitter adalah layanan microblogging dengan jumlah pengguna bulanan mencapai lebih dari 41 juta pengguna pada Juli 2009 dan terus berkembang cepat hingga saat ini. Pengguna Twitter dapat memposting tweet tentang topik apa pun dalam batasan 140 karakter kata setiap tweet dan mengikuti orang lain untuk menerima tweet mereka [13]. Hingga 2018 tercatat twitter memiliki jumlah pengguna bulanan mencapai 336 juta. Karena kemudahannya bagi pengguna untuk berbagi opini serta kemudahannya bagi penulis untuk mendapatkan data, twitter menjadi sosial media yang dipilih penulis untuk mendapatkan dataset pada penelitian ini.

2.2.8 Operator Telekomunikasi

Operator telekomunikasi adalah perusahaan yang menawarkan jasa telekomunikasi meliputi sms, telpon dan akses internet kepada penggunanya. Dua jenis teknologi yang menjadi

pilihan bagi pasar potensial bagi penyedia jasa operator telekomunikasi di Indonesia adalah GSM dan CDMA. Hingga saat ini ada 11 (sebelas) operator telekomunikasi yang beroperasi di Indonesia [15]. Namun pada penelitian ini hanya menggunakan 2 (dua) operator telekomunikasi dengan jumlah pengguna terbanyak yaitu Telkomsel dan Indosat.

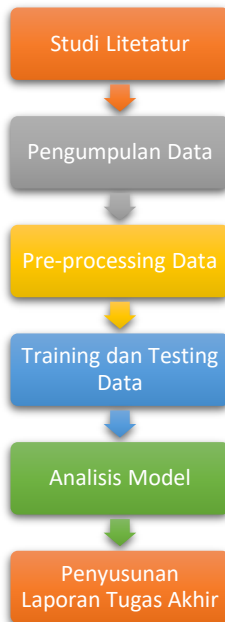
Halaman ini sengaja dikosongkan

BAB III METODOLOGI

Pada bagian ini dijelaskan metodologi yang akan digunakan sebagai panduan dan juga jadwal untuk menyelesaikan penelitian tugas akhir ini.

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:

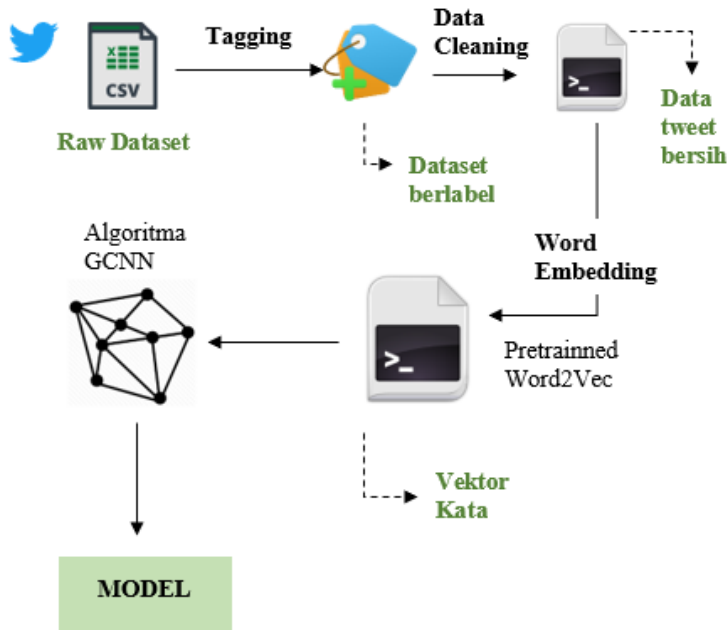


Gambar 3.1 Metode Penelitian

3.2 Arsitektur Penelitian

Pada bagian ini akan dijelaskan arsitektur penelitian yang digunakan pada penelitian ini. Arsitektur ini akan menjelaskan

secara garis besar aktivitas beserta input, output, dan metode disetiap prosesnya. Arsitektur dapat dilihat pada gambar berikut.



Gambar 3.2 Arsitektur Penelitian

Pada gambar 3.2 dijelaskan mengenai arsitektur yang digunakan pada penelitian ini. Input pertama ialah *raw dataset twitter* hasil *crawling* pada penelitian sebelumnya. Kemudian dataset akan ditagging dengan label aspek dan sentimen. Sehingga output dari proses *tagging* data ialah dataset yang telah berlabel. Selanjutnya dataset berlabel akan dilakukan *data cleaning*. Proses *data cleaning ini* akan dilakukan pembersihan dataset dari karakter atau simbol yang kurang berguna, lalu dilakukan *casefolding* untuk membuat seluruh dataset berhuruf kecil semua, dan melakukan tokenisasi pada dataset. Sehingga output dari proses *data cleaning ini* adalah dataset yang sudah bersih dan sudah berupa token.

Proses selanjutnya ialah word embedding dengan menggunakan pretrained word2vec dan outputnya akan berupa vektor kata yang siap masuk dan diolah dalam model. Selanjutnya vektor kata siap masuk untuk diolah dengan menggunakan algoritma GCNN untuk melakukan proses training dan testing data. Hingga outputnya berupa model dengan akurasi terbaik.

3.3 Uraian Metodologi

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

3.3.1 Studi Literatur

Pada tahap ini dilakukan dengan tujuan untuk memahami konsep, metode dan algoritma sesuai bahasan dan permasalahan, sehingga dapat memberi solusi mengenai permasalahan yang akan digunakan dalam penyusunan tugas akhir. Adapun literatur utama yang digunakan sebagai pedoman utama dalam penyusunan tugas akhir yaitu, Aspect Based Sentiment Analysis with Gated Convolutional Networks [4].

3.3.2 Pengumpulan Data

Tahap selanjutnya yaitu tahap pengumpulan data *tweets* yang akan digunakan untuk *training* data ditahap selanjutnya. Untuk mendapatkan *tweets* yang sesuai, pada penelitian ini menggunakan data tweet pelanggan operator telekomunikasi, Telkomsel dan Indosat Ooredoo, dari penelitian sebelumnya. Tweet yang digunakan ialah tweet yang mengandung kata kunci yang terkait dengan kedua operator telekomunikasi. Output dari tahap ini adalah kumpulan *tweet* mentah untuk diproses dulu pada tahap *preprocessing*.

3.3.3 Pre-Processing Data

Tahap selanjutnya yaitu tahap *pre-processing* data, tahap ini bertujuan untuk memproses data mentah hasil *crawling* agar dapat digunakan sebagai data *training* di tahap selanjutnya. Pada tahap *pre-processing* ini melewati beberapa subaktivitas. Output dari tahap ini adalah vektor kata hasil dari aktivitas *word*

embedding sebagai input untuk tahap selanjutnya. Subaktivitas yang dilakukan pada tahap ini adalah:

3.3.3.1 Pelabelan Tweet

Tahap selanjutnya melakukan pelabelan pada tweet dari kedua operator telekomunikasi. Berdasarkan penelitian SemEval-2014 Task 4: *Aspect Based Sentiment Analysis*, proses pelabelan memiliki dua tahap proses. Pada tahap pertama yaitu menentukan kata aspek dan sentimennya. Pada tahap kedua yaitu menentukan kategori aspek dan sentimennya [13]. Berikut perbedaan dari kedua tahap proses tersebut.

Tabel 3.1 Pelabelan Tweet

| Aspek berdasarkan <i>Term</i> dan Sentimennya | |
|--|--|
| <i>Tweet</i> | Indosat parah banget 2 hari ga bisa dipakai karena masalah koneksi. |
| Aspek berdasarkan <i>Term</i> dan Sentimennya | ‘Koneksi’: negatif |
| Aspek berdasarkan Kategori and Sentimennya | |
| <i>Tweet</i> | Sedangkan Telkomsel memang agak mahal harganya dan kuotanya sedikit tapi koneksi masih lebih lumayan cepat |
| Aspect berdasarkan Kategori dan Sentimennya | ‘Harga’: negative, ‘Jaringan’: positif |

Untuk pelabelan *tweet* akan dilakukan oleh 2 anotator. Satu orang bekerja anotator, satunya sebagai validator secara bergantian. Penentuan label berdasarkan tingkat *confident* dari masing-masing anotator dan ditentukan secara bersamaan. Ketika terjadi perbedaan pendapat, akan ditambahkan anotator ketiga untuk mengambil keputusan [16]. Pada penelitian ini pelabelan *tweet* yang dilakukan hanya berdasarkan kategori layanan operator telekomunikasi. Kategori yang digunakan telah didefinisikan sebelumnya oleh peneliti. Jenis sentimen yang digunakan untuk setiap label kategori aspek yang diberikan pada data ada 4, yaitu positif, negatif, netral, dan konflik.

3.3.3.2 Word Embedding menggunakan Word2Vec

Tahap selanjutnya yaitu tahap *word embedding* dengan library *glove-python*. Glove adalah algoritma unsupervised yang dikembangkan oleh Stanford untuk mengubah kata menjadi nilai vektor yang akan di proses dalam algoritma gated convolutional network pada tahap selanjutnya. Pada tahap ini menggunakan skenario data training sehingga dapat memberikan model dengan akurasi tertinggi. *Word embedding* dilakukan pada data yang sesuai dengan dua operator telekomunikasi yang menjadi target, yaitu Telkomsel dan Indosat.

3.3.4 Pelatihan dan Pengujian Data

Tahap selanjutnya yaitu tahap *training* data, dimana input data yang digunakan adalah nilai vektor data tweet operator telekomunikasi Telkomsel dan Indosat. Data *tweets* akan dibagi menjadi data *train* dan data *test* dengan ratio 80:20. *Training* data dilakukan dengan menggunakan metode *Gated Convolutional Network*.

3.3.5 Analisis Model

Tahap selanjutnya yaitu tahap analisis model yang berasal dari tahap sebelumnya. Analisis model dilakukan untuk melihat

performa model berdasarkan nilai *Accuracy* dan *F1-Score*. Pengukuran nilai accuracy dan f1-score dilihat dari kesesuaian prediksi model dalam mengklasifikasikan data menjadi dua kategori, yaitu positif dan negatif. Berikut notasi yang digunakan untuk melihat kesesuaian prediksi:

- *True Positive* (TP) : Data yang diklasifikasikan benar sebagai positif
- *True Negative* (TN) : Data yang diklasifikasikan benar sebagai negatif
- *False Positive* (FP) : Data yang diklasifikasikan salah sebagai positif. Data seharusnya diklasifikasikan sebagai negatif.
- *False Negative* (FN) : Data yang diklasifikasikan salah sebagai negatif. Data seharusnya diklasifikasikan sebagai positif.

Hasil klasifikasi data yang sesuai dengan sebenarnya disebut *Relevant*. Sehingga notasi klasifikasi *True Positive* (TP) dan *True Negative* (TN) termasuk pada kondisi *relevant*. Sedangkan klasifikasi data yang dihasilkan tidak sesuai kenyataan disebut *Not Relevant*. Sehingga notasi klasifikasi *False Positive* (FP) dan *False Negative* (FN) termasuk pada kondisi *not relevant*. Dari notasi klasifikasi data dapat dihitung nilai Precision, Recall, Accuracy dan F1-Score dengan rumus sebagai berikut:

$$Precision (P) = \frac{TP}{TP + FP}$$

$$Recall (P) = \frac{TP}{TP + FN}$$

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

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

3.3.6 Penyusunan Laporan Tugas Akhir

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

Halaman ini sengaja dikosongkan

BAB IV PERANCANGAN

Pada bab ini akan dijelaskan mengenai perancangan penelitian tugas akhir. Perancangan ini diperlukan sebagai panduan dalam melakukan penelitian tugas akhir, yang dijelaskan sebagai berikut.

4.1 Perancangan Pengumpulan Data

Dalam pelaksanaan analisis aspek dan sentimen pendapat pelanggan Telkomsel dan Indosat, hal pertama yang dibutuhkan adalah data. Data yang dibutuhkan untuk analisis ini berasal dari Twitter yang didapat dari penelitian sebelumnya. *Tweet* yang digunakan sebagai data adalah *tweet* dari pelanggan Telkomsel dan Indosat yang memiliki ketentuan sebagai berikut:

- Mengandung kata “telkomsel”
- Mengandung kata “indosat”
- Mengandung *review* mengenai telkomsel atau indosat
- Bukan *tweet* dari *official account* Telkomsel dan Indosat

4.2 Perancangan Pengolahan Data

Sebelum data diolah untuk analisis aspek dan sentimen, data harus terlebih dahulu mengalami pra-pemrosesan data. Proses ini akan mempersiapkan data mentah memiliki kondisi yang sesuai untuk proses *training* pada tahap selanjutnya.

4.2.1 Pemberian Label Dataset

Dataset yang berisi tweet akan melalui proses pelabelan aspek dan sentimen. Label-label aspek yang digunakan telah ditentukan berdasarkan pemetaan quality of service operator telekomunikasi, yaitu:

Tabel 4.1 Label Aspek pada Dataset

| Label Aspek | Alasan |
|--------------------|--|
| Kecepatan | Label kecepatan digunakan ketika kalimat review mengandung pembahasan mengenai kecepatan sebenarnya yang terjadi pada saat transfer data atau aktivitas internet yang digunakan pelanggan. |

| | |
|------------------|--|
| Jaringan | Label jaringan digunakan ketika kalimat review mengandung pembahasan mengenai sinyal, jaringan, maupun koneksi dari operator telekomunikasi yang mereka gunakan. |
| Layanan | Label layanan digunakan ketika kalimat review mengandung pembahasan mengenai layanan utama operator telekomunikasi seperti SMS, telfon, dsb. |
| Campaign | Operator telekomunikasi biasa mengirimkan broadcast sms berisi promo atau event yang sedang berlaku. Label ini digunakan ketika kalimat review membahas opini pelanggan tentang broadcast sms yang mereka terima dari operator telekomunikasi yang mereka gunakan. |
| Tagihan | Label ini digunakan ketika kalimat reiew mengandung pembahasan tentang tagihan, seperti tagihan pengguna layanan pasca bayar atau potongan pulsa akibat penggunaan pelanggan pada suatu promo atau paket tertentu, dll. |
| Registrasi | Label ini digunakan ketika kalimat review mengandung pembahasan mengenai pendaftaran kartu operator telekomunikasi yang digunakan pelanggan. |
| Aktivasi | Label ini digunakan ketika kalimat review mengandung pembahasan mengenai aktivasi paket atau layanan lain yang sedang berlaku. |
| Customer Service | Label ini digunakan ketika kalimat review mengandung pembahasan mengenai pelayanan front desk baik secara offline dari kantor customer service maupun secara online melalui official account operator telekomunikasi. |
| Harga | Label ini digunakan ketika kalimat review pelanggan mengandung pembahasan |

| | |
|------------|--|
| | mengenai harga dari paket atau layanan lain yang disediakan operator telekomunikasi. |
| Pengalaman | Label pengalaman digunakan ketika kalimat review pelanggan mengandung pembahasan tentang pengalaman, kesimpulan pendapat tentang operator telekomunikasi, dan hal-hal yang penilaian yang bersifat general tentang operator berdasarkan pengalaman pelanggan menggunakan jasa operator telekomunikasi. |
| Promosi | Operator telekomunikasi sering memberikan promosi-promosi untuk pelanggannya seperti paket super, atau layanan cashback, sbg. Label ini digunakan ketika kalimat review mengandung pembahasan mengenai pendapat konsumen mengenai promo yang mereka gunakan dari operator telekomunikasi. |
| Masa Aktif | Label ini digunakan ketika pelanggan membahas mengenai masa aktif kartu yang sedang digunakan. Masalah yang biasa dibahas ialah seperti cara perpanjangan masa aktif, pilihan lama masa aktif yang ingin diperpanjang, atau ketidaksesuaian lama masa aktif yang didapat dengan pilihan lama perpanjangan yang telah dipilih, dsb. |
| Paket | Label ini digunakan ketika pelanggan membahas mengenai pendapatan mereka pada pilihan paket yang disediakan operator untuk pelanggannya, baik paket internet, telfon, sms, entertainment, dsb. |
| Aplikasi | Label ini digunakan ketika pelanggan membahas mengenai pendapat dan masalah mereka saat menggunakan aplikasi dari operator telekomunikasi, seperti mytelkomsel, myim3, tcash, dsb. |
| Report | Label report digunakan ketika kalimat review mengandung report pelanggan yang |

| | |
|--|---|
| | biasanya mengenai kartu yang mereka gunakan, seperti kartu terblokir, atau permintaan pemblokiran kartu, dan lain sebagainya. |
|--|---|

Sedangkan label sentimen yang digunakan ada 4, yaitu:

Tabel 4.2 Label Sentimen pada Dataset

| Label Sentimen | Alasan |
|-----------------------|--|
| Positif | Label sentimen positif digunakan ketika kalimat review pelanggan memiliki kecenderungan ke makna baik atau positif, seperti memuji, mengapresiasi, dll. |
| Negatif | Label sentimen negatif digunakan ketika kalimat review pelanggan memiliki kecenderungan ke makna buruk atau negatif, seperti kekecewaan, protes, dll. |
| Netral | Label sentimen netral digunakan ketika kalimat review pelanggan tidak memiliki kecenderungan ke arah negatif maupun positif, biasanya untuk kalimat pertanyaan pelanggan kepada official account operator. |
| Konflik | Label sentimen konflik digunakan ketika kalimat review pelanggan memiliki kecenderungan ke arah negatif dan positif untuk satu aspek yang sama, misalkan setelah memuji suatu aspek |

| | |
|--|---|
| | kemudian menceritakan kekecewaan juga pada aspek tersebut dalam satu kalimat yang sama. |
|--|---|

Setiap tweet dapat memiliki satu atau lebih label aspek dan sentimen sesuai dengan masing-masing pembahasan setiap tweet. Berikut contoh pelabelan aspek dan sentimen pada data *tweet*.

Tabel 4.3 Pelabelan Aspek dan Sentimen pada Dataset

| Tweet | Topik | Kategori | Sentimen |
|---|---------|------------------|----------|
| Membalas @IndosatCare Iya makasih min. Thanks respon cepat nya. Indosat oke | INDOSAT | Customer Service | Positif |
| Damai Itu Saat @Telkomsel menurunkan harga agar pelanggan setianya tdk merasa kecewa atas kemahalannya | TSEL | Harga | Negatif |
| Membalas @bayukrismantoro @Raden_M_I dan 3 lainnya Ehhh btw jaringan telkomsel saya H+ nih gak 4G, gimana cara biar 4G padahal pake internetnya ambil yg 4G ?? Masa iyaa harus minta bantuan istri tetangga ? | TSEL | Jaringan | Netral |
| @IndosatCare dulu di Gresik sinyal Indosat | INDOSAT | Jaringan | Konflik |

| | | | |
|--|--|--|--|
| <p>bagus, 4G dan cukup kencang. Beberapa minggu terakhir muncul di layar sinyal 4G tapi merem melek banyak meremnya. Kirim WA susah, buka Twitter sulit. Kenapa kah?</p> | | | |
|--|--|--|--|

Dataset berisi *tweet review* pelanggan operator telekomunikasi akan diubah menjadi format csv sehingga dapat digunakan sebagai masukan atau *input* pada proses selanjutnya.

4.2.2 Pra-proses Data

Pada tahap ini adalah tahap untuk membuat data memiliki bentuk kondisi yang dibutuhkan saat pemrosesan data sehingga data dapat melakukan tahap proses data dan analisis. Dataset yang telah dilabeli berada dalam format csv dan memiliki 4 field, yaitu *tweet*, topik, *categories*, dan *sentiment*. *Field tweet* berisi kalimat *tweet review* pengguna operator telekomunikasi telkomsel dan indosat. *Field topik* berisi jenis operator telekomunikasi yang dibahas pada *field tweet*. *Field categories* berisi kategori aspek yang terkandung pada *field tweet*. Sedangkan *field sentiment* berisi sentimen yang dari kategori aspek yang terkandung pada *field tweet*.

Dari data berformat csv kemudian dimuat dan dimasukkan ke sebuah *dictionary python*. *Dictionary* ini memiliki 3 key yaitu, *sentence*, operator, dan *aspect_sentiment*. *Key sentence* akan menampung data dari *field tweet* pada dataset. *Key operator* akan menampung data dari *field topik*, sedangkan *key aspect_sentiment* akan menampung pasangan kategori dan sentimen dari *field categories* dan *sentiment*. Setelah dalam bentuk ini, data siap untuk melakukan proses selanjutnya yaitu pembersihan data dan *word embedding*.

4.2.2.1 Mempersiapkan Data

Pada tahap pra-proses data akan dilakukan pembersihan, tokenisasi, dan word embedding pada data

sehingga bisa menjadi masukan pada tahapan selanjutnya. Langkah pembersihan data pada tahap pra-proses ini diperlukan untuk menyeragamkan data dan menghapus karakter kurang bermakna. Karena data *tweet* mentah masih banyak mengandung karakter maupun simbol yang kurang berguna. Sehingga proses pembersihan dataset yang berisi *tweet* mentah dengan karakter maupun simbol harus dibersihkan terlebih dahulu. Kemudian *tweet* juga akan dirubah menjadi huruf kecil. Berikut detail urutan proses pembersihan *tweet*:

- Decoding HTML.
- Menghilangi mention (@ dan user yang disebutkan)
- Pembersihan tautan URL.
- Pembersihan karakter aneh “\ xef \ xbf \ xbd” yang disebut UTF-8 BOM.
- Menghapus tagar dengan tetap membiarkan kata yang mengikutinya.
- Mengubah seleruh huruf menjadi kecil (*lowercase*).

Kemudian langkah selanjutnya adalah tokenisasi. Pada langkah ini, setiap data *tweet* akan ditokenisasi sehingga menjadi suku kata yang unik dari seluruh dataset *tweet*. Kata-kata tersebut akan disimpan dalam suatu *vocab* sehingga kata unik akan memiliki index yang berbeda. Kata-kata yang telah disimpan dalam *vocab* inilah yang akan digunakan dalam *training*.

4.2.2.2 Word Embedding

Dataset yang telah ditokenisasi selanjutnya akan menjalani tahapan *word embedding*. *Word embedding* akan mengubah kata menjadi matriks/vektor. Matriks/vektor ini yang akan menjadi masukan/*input* pada model yang akan dibuat. Pengubahan kata menjadi matrik ini disebut *word embedding*. Untuk proses *word embedding* pada penelitian ini akan menggunakan data *pretrained word embedding* dengan model *word2vec* yang telah dibuat pada penelitiin sebelumnya.

4.2.3 Perancangan Pembuatan Model GCNN

Pada tahap ini akan dilakukan perancangan untuk membuat model Gated Convolutional Network. Data yang digunakan adalah word vector dari hasil word embedding. Keluaran pada tahap ini merupakan model GCNN dengan paramater dan akurasi terbaik.

dfs

4.2.3.1 Perancangan *Training* Model GCNN

Pembuatan *training* model GCNN akan menghasilkan model GCNN. Dalam rangka pembuatan *training* model GCNN, terdapat *hyperparamater* yang harus ditentukan sehingga model dapat memiliki hasil terbaik. Analisis aspek dan sentimen pada penelitian ini menggunakan algoritma GCNN dengan *package pytorch* yang berbasis *Python* dari *Torch*. *Package PyTorch* dapat digunakan untuk komputasi numerik dan telah *support* untuk melakukan operasi di GPU. *PyTorch* juga dilengkapi fungsi-fungsi bawaan yang mempermudah untuk menerapkan model-model *deep learning*. *Package* ini bisa digunakan untuk implementasi model-model *deep learning* atau *conventional machine learning*. *Input* utama dalam proses *training* model GCNN ini adalah output dari proses *word embedding* dan dataset yang telah diolah untuk dapat menjadi input pada *training* model. Berikut adalah paramter yang digunakan pada model GCNN.

Tabel 4.4 Parameter Model GCNN

| Parameter | Definisi |
|------------------|--|
| Embed_num | Jumlah total token yang terdeteksi pada <i>tweet</i> dalam dataset |
| Embed_dim | Ukuran dimensi word embedding |
| Class_num | Jumlah total sentimen yang terdeteksi pada label dataset |

| | |
|--------------|--|
| Aspect_num | Jumlah total aspek yang terdeteksi pada label dataset |
| Kernel_num | Jumlah setiap jenis kernel |
| Kernel_sizes | Ukuran kernel yang dipisahkan oleh koma yang digunakan untuk konvolusi |

Penentuan parameter pada proses *training* juga dibutuhkan sehingga dapat membandingkan model yang dihasilkan dengan skenario *training* yang berbeda. Berikut adalah parameter yang digunakan pada *training* model GCNN.

Tabel 4.5 Parameter *Training* Model GCNN

| Parameter | Definisi |
|------------------|--|
| N_trials | Jumlah iterasi <i>training</i> |
| Epochs | Jumlah epoch untuk <i>training</i> |
| Batch_size | Ukuran batch untuk <i>training</i> model |

4.2.3.2 Perancangan *Testing* Model GCNN

Tahap *testing* ini adalah tentang pengujian kualitas model yang dihasilkan. *Testing* model dipanggil didalam *training* model untuk mengukur model *training* yang terbentuk. Pengukuran *testing* model yang dicatat pada setiap *training* model berupa akurasi.

Halaman ini sengaja dikosongkan

BAB V IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai implementasi dari perancangan penelitian yang telah dibuat pada bab sebelumnya. Hal yang akan dibahas pada bab ini adalah mengenai lingkungan implementasi.

5.1 Lingkungan Implementasi

Pengerjaan penelitian ini menggunakan komputer dengan spesifikasi berikut.

Tabel 5.1 Spesifikasi Hardware

| | |
|----------------------------|---|
| <i>Processor</i> | Intel® Core™ i5-5200U CPU @ 2.20GHz 2.19GHz |
| <i>Memory</i> | 8192MB RAM |
| <i>Graphic Card</i> | Intel® HD Graphics 5500 – NVIDIA GeForce 930M |
| Sistem Operasi | <i>Windows 10 Education 64-bit</i> |
| Arsitektur Sistem | <i>64-bit operating system, x64-based processor</i> |

Dalam pelaksanaan penelitian ini terdapat beberapa teknologi seperti editor, bahasa pemrograman, dan *library* yang akan dicantumkan dalam Tabel 5.2 berikut.

Tabel 5.2 Teknologi yang Digunakan untuk Mengembangkan Model

| | |
|---|---------------------|
| Bahasa Pemrograman | Python 3.6.5 |
| <i>Computational Environment</i> | Jupyter Notebook |
| <i>Virtual Environment</i> | Anaconda |
| <i>Software/Tools</i> | Microsoft Excel 365 |

| | |
|----------------|--|
| Library | <ul style="list-style-type: none"> • Csv • Torchtext • Beautifulsoup4 0.0.1 • RegEx • NLTK • Simplejson • Collections • Numpy • Linecache • Torch • Time • Sys • Os • Gensim |
|----------------|--|

5.2 Pra-Proses Data

Agar data dapat diproses, maka perlu dilakukan persiapan data untuk membuat data memiliki kondisi yang siap untuk menjadi input pada tahap pemrosesan data. Dataset *tweet review* pengguna telkomsel dan indosat akan dimuat dari file csv. Kemudian akan mengalami langkah persiapan data dan word embedding. Output dari pre-proses bertujuan untuk merubah dataset mentah menjadi kondisi yang siap untuk melakukan proses training dan testing.

5.2.1 Mempersiapkan Data

Dataset berasal dari *tweet* yang masih mengandung banyak karakter atau simbol-simbol yang kurang berguna untuk proses penelitian. Maka sebelum data diolah, pada langkah ini akan dilakukan pembersihan data, tokenisasi, serta menyamakan data menjadi huruf kecil semua.

```

1. def tweet_cleaner(text):
2.     pat1 = r'@[A-Za-z0-9]+'
3.     pat2 = r'https?://[A-Za-z0-9./]+'
4.     combined_pat = r'|'.join((pat1, pat2))

```

```

5. soup = BeautifulSoup(text, 'xml')
6. souped = soup.get_text()
7. stripped = re.sub(combined_pat, '', souped)
8. try:
9. clean = stripped.decode("utf-8-
  sig").replace(u"\ufffd", "?")
10. except:
11. clean = stripped
12. letters_only = re.sub("[^a-zA-Z]", " ", clean)
13. lower_case = letters_only.lower()
14. words = tok.tokenize(lower_case)
15. return " ".join(words).strip()

```

Kode 5.1 Persiapan Data

Kode persiapan data dimulai dengan langkah membersihkan dataset dari karakter yang kurang berguna. Pembersihan dataset yang dilakukan adalah:

- Mengubah karakter html menjadi *text* umum sehingga dapat data diolah pada tahap selanjutnya. Hal ini dilakukan dengan menggunakan *BeautifulSoup* pada kode baris 5 - 6.
- Menghilangkan *mention user* pada *tweet* dengan kode pada baris 2 dan 7. Karena penyebutan nama pengguna pada *tweet* tidak menambah *value* untuk proses analisis pada penelitian ini.
- Pembersihan *url link* pada *tweet* dengan kode pada baris 3 dan 7. Sama seperti *mention user*, pembersihan *url link* karena *link* yang tercantum pada *tweet* tidak menambah *value* apapun pada analisis ini.
- Menghilangkan karakter UTF-8 BOM (Byte Order Mark) yang beberapa muncul pada *tweet* dataset dengan kode pada baris 9.
- Menghilangkan *hashtag* tanpa menghapus kata setelahnya dengan kode pada baris 12. Karena *hashtag* sering kali membawa informasi penting mengenai suatu *tweet*. Contoh *tweet* “yang lagi mau liburan di *The Jungle Bogor Indosat Super WIFI* juga ada di sana looh, ngenet asik sambil liburan #Mantap @indosatbogor”. Kata #Mantap memiliki informasi

positif mengenai kondisi yang dibahas pada *tweet* yaitu jaringan super wifi. Maka hanya tanda hashtag yang dihilangkan sedangkan kata setelahnya tidak dihapus.

- Mengubah kalimat *tweet* pada dataset menjadi huruf kecil (*case fold*) dengan kode pada baris 13. Hal ini dilakukan dengan pertimbangan bahwa saat word embedding pada tahap selanjutnya, besar kecil huruf dapat mempengaruhi nilai matriks yang dihasilkan.

Lalu langkah terakhir adalah tokenisasi dataset dengan kode pada baris 14. Seluruh kalimat *tweet* pada dataset akan dirubah menjadi kata-kata unik untuk menjalani proses selanjutnya

5.2.2 Word Embedding

Setelah data telah melalui proses pembersihan, langkah selanjutnya adalah *word embedding*. *Word embedding* ini akan mengubah kata menjadi matriks. Proses *word embedding* pada penelitian ini menggunakan *pre-trained word2vec* yang dihasilkan dari *training word embedding* data twitter dengan menggunakan bahasa indonesia pada penelitian sebelumnya. Hal yang pertama dilakukan adalah memuat *pretrained embedding* tersebut dengan menggunakan *library gensim*.

```

1. embeddings_google = '../Embedding_w2v_indo/modelapik
  .bin'
2. embeddings_path = embeddings_google
3. word2vec_model = KeyedVectors.load_word2vec_format(e
  mbeddings_path, binary=True, unicode_errors='ignore'
  )
4. emb_init_values = []
5. unk = []
6. a = 0
7. b = 0
8. for i in range(len(text_field.vocab.stoi)): # Untuk
  memastikan bahwa urut
9. word = text_field.vocab.itos[i]
10. if word == '<unk>':
    a. emb_init_values.append(np.random.uniform(-
      0.25, 0.25, 300).astype('float32'))

```



```

11. elif word == '<pad>':
a. emb_init_values.append(np.zeros(300).astype('float32'))

12. elif word in word2vec_model.wv.vocab:
a. emb_init_values.append(word2vec_model.wv.word_vec(word))
b. b = b+1

13. else:
a. emb_init_values.append(np.random.uniform(-0.25, 0.25, 300).astype('float32'))
b. a = a+1
c. unk.append(word)
1. # print(word)
14. known_word = b
15. unknown_word = a

16. print(known_word, unknown_word)

```

Kode 5.2 Word Embedding

Pada kode baris 3, hasil memuat *pretrained word2vec* disimpan dalam variabel yang diberi nama `word2vec_model`. Lalu pada kode baris 4 – 7, diinisiasi list bernama `emb_init_values` dan `unk` yang akan menampung matriks dari dataset yang merupakan hasil proses word embedding. Dan juga diinisiasi variabel `a` yang menghitung kata dari dataset yang dapat dicari matriksnya pada *pretrained word2vec* dan variabel `b` untuk menghitung kata dari data set yang tidak ditemukan matriksnya pada proses word embedding ini.

Pada proses word embedding ini akan mengambil satu persatu token dari dataset yang telah disimpan pada vocab bernama `text_field` dengan kode baris 9. Jika token yang diambil berupa '`<unk>`' maka akan dihasilkan random matriks sebagai hasil word embeddingnya dengan kode 10a dan jika token yang diambil berupa '`<pad>`' maka matriks zero yang akan dijadikan hasil word embeddingnya dengan kode 11a. Jika pada kode baris 12a, token yang diambil dapat ditemukan pada vocab `word2vec_model` maka outputnya berupa matriks dari vocab yang sesuai dengan token. Hasil embedding token '`<unk>`', '`<pad>`', token yang ditemukan pada vocab `word2vec` disimpan


```

18.     else:
19.         example['aspect_sentiment'].append((row
w['CATEGORIES'], row['SENTIMENT']))
20.         dataset.append(example)
21.     else:
22.         pass
23. return dataset

```

Kode 5.3 Pembaca Dataset

Method diatas berfungsi untuk memuat dataset dari file berformat csv. Method ini membutuhkan parameter berupa path lokasi file dataset berada. Karena menggunakan format csv, untuk memuat data digunakan delimiter semi-colon (;) pada kode baris 4. Dataset pada file csv memiliki 4 field yaitu tweet, topik, categories, dan sentiment. Setelah dimuat, datanya dimasukkan pada list dictionary dengan 3 key, yaitu sentence, operator, dan aspect_sentiment pada kode baris 8 - 10. Pada kode baris ke 8, key sentence untuk menampung data dari field tweet yang berisi kalimat review dari twitter, sedangkan pada kode baris ke 9 key operator menampung jenis operator dari field topik yang dibahas pada kalimat review, dan pada baris ke 17 dan 19 key aspect_sentiment memuat aspek dan sentimen dari field categories dan sentiment.

```

1. def getdata(path1, path2):
2.     datatrain = readfile(path1)
3.     for i in datatrain:
4.         trainbersih = tweet_cleaner(i['sentence'])
5.         i['sentence'] = trainbersih
6.     datatrain = list(datatrain)
7.     indosel_train = []
8.     train_total = []
9.     train_total += datatrain
10.    indosel_train = train_total
11.    print("# Train: {}".format(len(indosel_train)))
12.    datatest = readfile(path2)

```

```

13. for i in datatest:
14.     testbersih = tweet_cleaner(i['sentence'])
15.     i['sentence'] = testbersih

16. datatest = list(datatest)
17. indosel_test = []
18. test_total = []
19. test_total += datatest
20. indosel_test = test_total
21. print("# Test: {}".format(len(indosel_test)))

22. return indosel_train, indosel_test

```

Kode 5.4 Getdata dan Pemanggilan Fungsi Cleaning

Method ini memiliki tujuan memanggil method readfile pada baris ke untuk membaca pada baris 2 dan 12 dan memanggil method tweet_cleaner untuk membersihkan dataset pada kode baris 3 – 4 sama seperti yang telah dijelaskan diatas. Pembersihan tweet hanya dilakukan pada key sentence dari list dictionary hasil method readfile untuk membaca dataset.

```

1. def load_indosel_data(text_field, as_field, sm_field
, **kargs):
2.     train_data, test_data = getdata(train_path, test_
path)

3.     predict_test = [{"aspect": "layanan",
"sentence": "sinyal di tmp2 saya sering mati2 uda
h minggu ini ga bisa nelpon ditelpon internet lemot"},
{"aspect": "kecepatan",
"sentence": "sinyal di tmp2 saya sering mati2 uda
h minggu ini ga bisa nelpon ditelpon internet lemot"}]

4.     predict_data = DataIndosel(text_field, as_field,
sm_field, predict_test)

5.     train_data, dev_data, mixed_data = DataIndosel.sp
lits_train_test(text_field, as_field, sm_field, t
rain_data, test_data)

6.     text_field.build_vocab(train_data, dev_data)
7.     as_field.build_vocab(train_data, dev_data)
8.     sm_field.build_vocab(train_data, dev_data)

```

```

9.  train_iter, test_iter, mixed_test_iter, predict_iter = data.Iterator.splits((train_data, dev_data, mixed_data, predict_data), batch_sizes=(arg.get("batch_size"), len(dev_data), len(mixed_data), len(predict_data)), **kargs)
10. return train_iter, test_iter, mixed_test_iter, predict_iter

```

Kode 5.5 Pemanggilan Kelas DataIndosel

Pada method ini juga diinisiasi 3 vocab dari 3 variabel bertipe field dari library torchtext dan bernama `text_field`, `as_field`, dan `sm_field` yang akan menampung data dari data train dan test pada kode baris 6 - 8. Vocab `text_field` menampung token dari key sentence pada dictionary dataset train dan test. Vocab `as_field` menampung token key aspek dan vocab `sm_field` menampung key sentiment dari dictionary dataset train dan test. Method ini bertujuan untuk memanggil kelas `DataIndosel` yang memiliki 2 method yaitu `unroll` dan `split_train_test` pada baris ke 5. Fungsi `unroll` bertujuan untuk memisahkan data review yang mengandung satu aspek dan sentimen dengan yang memiliki lebih dari satu aspek dan sentiment. Fungsi `unroll` ini dipanggil di dalam fungsi `split_train_test` untuk memisahkan data *train*, data *test*, data *test* yang memiliki aspek dan sentimen lebih dari satu, dan data *predict test*.

Kemudian vocab `text_field` dan `as_field` akan mengalami proses word embedding seperti yang telah dijelaskan diatas hingga keluar output matriks embedding dari token `text_field` dan `as_field`. Selanjutnya mulai dilakukan inisiasi model yang mengakses kelas `CNN_Gate_Aspect_Text`. Model GCNN ini juga mengacu pada penelitian Wei Xue di tahun 2018 dengan menggunakan proses konvolusi 1 dimensi.

```

1. class CNN_Gate_Aspect_Text(nn.Module):
2.     def __init__(self, args):
3.         super(CNN_Gate_Aspect_Text, self).__init__()
4.         self.args = args
5.         V = arg.get("embed_num")
6.         D = arg.get("embed_dim")

```

```

7.     C = arg.get("class_num")
8.     A = arg.get("aspect_num")

9.     Co = arg.get("kernel_num")
10.    Ks = arg.get("kernel_sizes")

11.    self.embed = nn.Embedding(V, D)
12.    print("self embed atas",self.embed)
13.    self.embed.weight = nn.Parameter(arg.get("embeddi
ng"), requires_grad=True)

14.    self.aspect_embed = nn.Embedding(A, arg.get("aspe
ct_embed_dim"))
15.    self.aspect_embed.weight = nn.Parameter(arg.get("
aspect_embedding"), requires_grad=True)

16.    self.convs1 = nn.ModuleList([nn.Conv1d(D, Co, K)
for K in Ks])
17.    self.convs2 = nn.ModuleList([nn.Conv1d(D, Co, K)
for K in Ks])

18.    self.fc1 = nn.Linear(len(Ks)*Co, C)
19.    self.fc_aspect = nn.Linear(arg.get("aspect_embed_
dim"), Co)

20.    print("def init")
...

```

Kode 5.6 Potongan Kode Parameter Model GCNN

Pada kelas model GCNN membutuhkan parameter seperti panjang vocabulary text, panjang vocab aspek, panjang vocab sentimen, lalu dimensi vektor kata, hasil embedding kata dan aspek, serta jumlah dan ukuran kernel yang digunakan untuk proses konvolusi seperti pada kode baris 5 – 10. Setelah itu juga membuat 2 objek embedding bernama embed pada baris 11 untuk kalimat review dan aspect_embed pada baris 14 untuk aspek dengan method `nn.Embedding()` dari library `torch.nn`. Method ini membutuhkan dua parameter utama, yaitu jumlah voab dan dimensi vektor kata. Untuk objek embedding bernama embed, ia menggunakan parameter jumlah vocab teks yang diproses pada penelitian ini dan dimensi vektor sebesar 300. Sedangkan objek embedding bernama aspect_embed

menggunakan jumlah vocab dari aspek yang diproses pada penelitian ini dengan dimensi vektor sebesar 300 juga.

```
self.embed = Embedding(5306, 300)
self.aspect_embed = Embedding(22, 300)
```

Gambar 5.1 Contoh 2 Objek Embedding

Setelah itu yaitu pemberian bobot awal setiap kata pada objek embed dan aspect_embed dengan menggunakan hasil word embedding pada teks dan aspek pada kode baris ke 13 dan 15. Pemberian bobot ini dilakukan menggunakan method `weight()` dengan method `nn.Parameter()` yang berisi paramater hasil word embedding yang telah diubah format menjadi array numpy dan menggunakan `gradient`.

Proses selanjutnya adalah konvolusi menggunakan method `nn.Conv1d()` dari library `torch.nn` pada baris 16 dan 17. Untuk proses konvolusi membutuhkan tiga parameter utama yaitu, dimensi vektor, jumlah kernel yang digunakan, dan ukuran kernel yang digunakan. Parameter terakhir adalah ukuran `stride`, jarak perpindahan disetiap pergerakan filter, yang bernilai 1. Hasil konvolusi ini disimpan dalam bentuk array dan ditransformasikan ke dalam bentuk daftar modul dengan menggunakan method `nn.ModuleList()`.

Kemudian dibuat 2 objek linear bernama `fc1` dan `fc_aspect` menggunakan method `nn.Linear()` pada baris ke 18 dan 19. Objek `fc1` menggunakan parameter banyaknya ukuran kernel yang digunakan dikali dengan jumlah kernel yang digunakan dan jumlah sentimen yang digunakan. Sedangkan untuk objek `fc_aspect` menggunakan parameter dimensi vektor kata untuk aspek dan jumlah kernel yang digunakan.

```

CNN_Gate_Aspect_Text(
  (embed): Embedding(5306, 300)
  (aspect_embed): Embedding(22, 300)
  (convs1): ModuleList(
    (0): Conv1d(300, 100, kernel_size=(3,), stride=(1,))
    (1): Conv1d(300, 100, kernel_size=(4,), stride=(1,))
    (2): Conv1d(300, 100, kernel_size=(5,), stride=(1,))
  )
  (convs2): ModuleList(
    (0): Conv1d(300, 100, kernel_size=(3,), stride=(1,))
    (1): Conv1d(300, 100, kernel_size=(4,), stride=(1,))
    (2): Conv1d(300, 100, kernel_size=(5,), stride=(1,))
  )
  (fc1): Linear(in_features=300, out_features=4, bias=True)
  (fc_aspect): Linear(in_features=300, out_features=100, bias=True)
)

```

Gambar 5.2 Hasil Parameter Model GCNN

Mulai memasuki proses training, method `train()` digunakan untuk proses training penelitian ini.

```

1. def train(train_iter, dev_iter, mixed_test_iter, model, args, text_field, aspect_field, sm_field, predict_iter):
2.     time_stamps = []
3.
4.     optimizer = torch.optim.Adagrad(model.parameters(), lr=args.get("lr"), weight_decay=args.get("l2"), lr_decay=args.get("lr_decay"))
5.
6.     steps = 0
7.     model.train()
8.     start_time = time.time()

```

Kode 5.7 Potongan Kode Perubahan State Model Train

Pada baris ke 4, variabel `optimizer` menyimpan hasil optimisasi menggunakan method `torch.optim.Adagrad()` dengan parameter method berupa parameter inisiasi awal model, inisiasi learning rate, weight decay, dan learning rate decay. Kemudian membuat proses training akan dilaksanakan di GPU dengan method `model.cuda()`. Serta membuat model berada pada state training dengan menggunakan method `model.train()` dari library `pytorch` pada kode baris ke 7.


```

1. for epoch in range(1, arg.get("epochs")+1):
2.     steps = 0
3.     print("Epoch : ",epoch)
4.     for batch in train_iter:
5.         feature, aspect, target = batch.text, batch.aspe
           ct, batch.sentiment

6.         feature.data.t_()
7.         if len(feature) < 2:
8.             continue
9.         if not arg.get("aspect_phrase"):
10.            aspect.data.unsqueeze_(0)
11.            aspect.data.t_()
12.            target.data.sub_(1)# batch first, index align

13.        if arg.get("cuda"):
14.            feature, aspect, target = feature.cuda(), asp
           ect.cuda(), target.cuda()

```

Kode 5.8 Potongan Kode Perulangan Epoch pada Train

Baris pertama didalam method train(), terdapat perulangan epoch sebanyak nilai epoch yang ditentukan. Didalam perulangan epoch tersebut terdapat perulangan sebanyak jumlah mini batch yang ditentukan. Lalu dilakukan ekstraksi kalimat review, aspek, dan label masing-masing dari dataset pada kode baris 5.

```

1. optimizer.zero_grad()
2. logit, _, _ = model(feature, aspect)

```

Kode 5.9 Potongan Kode Algoritma Training

Selanjutnya pada potongan kode 5.9 baris pertama, dilakukan penghilangan gradien pada optimizer. Kemudian pada baris kedua, mengakses method forward() yang ada dalam model GCNN dengan parameter kalimat dan aspek dari hasil ekstraksi pada awal training.

```

1. def forward(self, feature, aspect):
2.     feature = self.embed(feature) # (N, L, D)
3.     aspect_v = self.aspect_embed(aspect) # (N, L', D)
4.     aspect_v = aspect_v.sum(1) / aspect_v.size(1)

```

```

5. x = [F.tanh(conv(feature.transpose(1, 2))) for conv
      in self.convs1] # [(N,Co,L), ...]*len(Ks)
6. y = [F.relu(conv(feature.transpose(1, 2)) + self.fc
      _aspect(aspect_v).unsqueeze(2)) for conv in self.co
      nvs2]
7.     x = [i*j for i, j in zip(x, y)]

8. # pooling method
9. x0 = [F.max_pool1d(i, i.size(2)).squeeze(2) for i i
      n x] # [(N,Co), ...]*len(Ks)
10. x0 = [i.view(i.size(0), -1) for i in x0]

11. x0 = torch.cat(x0, 1)
12. logit = self.fc1(x0) # (N,C)
13. return logit, x, y

```

Kode 5.10 Potongan Kode Method Forward()

Pada baris 2 dan 3 method forward() ini dilakukan proses embedding dengan method nn.Embedding() yang telah diinisiasi pada awal pembuatan model GCNN. Hasil embedding dari kalimat review ini disimpan pada variabel bernama feature, sedangkan hasil embediing dari aspek disimpan pada variabel bernama aspect_v. Lalu pada baris ke 5, variabel feature masuk ke proses konvolusi filter yang pertama (self.convs1) yang telah diinisiasi sebelumnya. Lalu konvolusi filter pertama ini dilakukan proses aktivasi dengan fungsi ReLU untuk mengubah nilai vektor negatif menjadi 0 tanpa mengubah bentuk vektor. Ada kemungkinan terjadi kesamaan antara nilai output dari fungsi ReLU (bertanggungjawab pada ekstraksi aspek) dengan nilai aspek yang diberikan. Jika nilai ini adalah nol, maka sentiment feature akan memblok gate. Namun jika tidak, maka besaran dari gate Tanh (ekstraksi sentimen) akan diperbesar. Proses aktivasi fungsi ReLU dan Tanh menggunakan library nn.Functional(). Dan pada baris ke 7, hasil output dari kedua fungsi aktivasi ini dikalikan dan disimpan pada variabel.

Kemudian masuk ke pooling layer max pooling 1 dimensi, hasil perkalian dua fungsi aktivasi sebelumnya akan diambil nilai terbesarnya menggunakan method max_pool1d() dari library nn.Functional() seperti yang ditunjukkan pada kode baris ke 9. Dari pooling layer ini akan menghasilkan fixed-sized

vektor. Lalu pada layer final fully-connected dengan fungsi softmax function akan menggunakan vektor tersebut untuk memprediksi polaritas sentimen.

```

1. loss = F.cross_entropy(logit, target)
2. loss.backward()
3. optimizer.step()

4. steps += 1
5. if steps % arg.get("log_interval") == 0:
6.     corrects = (torch.max(logit, 1)[1].view(target.size()).data == target.data).sum()
7.     accuracy = 100.0 * corrects/batch.batch_size

```

Kode 5.11 Potongan Kode Prediksi Kalimat dalam Train

Langkah selanjutnya melakukan proses cross entropy untuk setiap hasil prediksi sentimen pada semua kalimat review pada dataset dengan method `cross_entropy()` dari library `nn.Functional()`. Karena ukuran data sebesar 2000 dan ukuran batch adalah 32, maka akan ada 50 batch pada setiap epochnya. Pada baris ke 5-7, setiap jumlah batch yang dapat habis dibagi 10 (`log_interval`) akan dilakukan perhitungan kebenaran prediksi yang dilakukan oleh model dengan menggunakan method `torch.max()`. Lalu akan dihitung akurasi dari prosentasi jumlah prediksi yang benar dari satu ukuran batch data yang diproses. Ukuran akurasi ini dihitung dari data pada mini batch tersebut (bukan data testing).

```

1. dev_acc_epoch, _, _ = eval(dev_iter, model)
2. if mixed_test_iter:
3.     #print(steps, "step mixed_test_iter")
4.     mixed_acc_epoch, _, _ = eval(mixed_test_iter, mode
5.     1)
6. else:
7.     mixed_acc = 0.0

```

Kode 5.12 Potongan Kode Mengakses Method Eval

5.3.2 Pembuatan Testing Model GCNN

Testing dilakukan pada setiap epoch dan setiap selesai menjalankan sejumlah epoch yang ditentukan pada setiap trial/repeat. Pada penelitian ini menggunakan trial sejumlah 5 dan epoch yang nilainya 5, 10, 15, 20, dan 25 untuk memeriksa

hasil terbaik model ada pada epoch yang beberapa. Testing ini mengakses method `eval()` yang membutuhkan parameter berupa data tes dan model yang telah diinisiasi.

```

1. def eval(data_iter, model):
2.     model.eval()
3.     corrects, avg_loss = 0, 0
4.     loss = None
5.     for batch in data_iter:
6.         feature, aspect, target = batch.text, batch.aspec
7.         t, batch.sentiment
8.         feature.data.t_()
9.         if not arg.get("aspect_phrase"):
10.            aspect.data.unsqueeze_(0)
11.            aspect.data.t_()
12.            target.data.sub_(1) # batch first, index align
13.
14.        if arg.get("cuda"):
15.            feature, aspect, target = feature.cuda(), aspec
16.            t.cuda(), target.cuda()
17.
18.        logit, pooling_input, relu_weights = model(featur
19.        e, aspect)
20.        loss = F.cross_entropy(logit, target, size_averag
21.        e=False)
22.        avg_loss += loss.data
23.        corrects += (torch.max(logit, 1)
24.        [1].view(target.size()).data == target.data).sum(
25.        )
26.
27.    size = len(data_iter.dataset)
28.    avg_loss = loss.data/size
29.    accuracy = 100.0 * corrects/size
30.    model.train()
31.    if arg.get("verbose") > 1:
32.        print('\nEvaluation - loss: {:.6f} acc: {:.4f
33.        }%({}/{})'.format(avg_loss, accuracy, corrects
34.        , size))
35.    return accuracy, pooling_input, relu_weights

```

Kode 5.13 Potongan Kode Testing Data

Untuk melakukan testing data, state model akan dirubah menjadi `model.eval()` seperti pada potongan kode 5.13 baris ke 2. Lalu dilakukan ekstraksi kalimat review, aspek, dan label

masing-masing dari data tes pada baris 5 dan 6. Dan kembali dilakukan prediksi sentimen dengan menggunakan method `forward()` pada baris 14 untuk data tes ini. Lalu juga masih sama dilakukan `cross entropy`, mengambil nilai max hasil prediksi, dan kemudian akan mengembalikan nilai akurasi tes pada baris 15 - 21.

Halaman ini sengaja dikosongkan

BAB VI HASIL DAN PEMBAHASAN

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

6.1 Hasil Pelabelan Tweet

Pada tahap pelabelan ini ada dua langkah yang dilakukan, yaitu pelabelan aspek dan pelabelan sentimen. Berikut hasil pemberian label aspek dan sentimen yang dilakukan pada data *train*.

Tabel 6.1 Distribusi Label Aspek dan Sentimen Data Train

| Data Train : 1600 | | |
|--------------------------|-----------------------|---------------------|
| Label Aspek | Label Sentimen | Jumlah Tweet |
| Jaringan | Positif | 128 |
| | Negatif | 314 |
| | Netral | 12 |
| | Konflik | 29 |
| Kecepatan | Positif | 106 |
| | Negatif | 169 |
| | Netral | 5 |
| | Konflik | 20 |
| Pengalaman | Positif | 93 |
| | Negatif | 102 |
| | Netral | 6 |
| | Konflik | 1 |
| Harga | Positif | 69 |

| | | |
|------------------|---------|-----|
| | Negatif | 119 |
| | Netral | 10 |
| | Konflik | 3 |
| Layanan | Positif | 24 |
| | Negatif | 66 |
| | Netral | 6 |
| | Konflik | 4 |
| Customer Service | Positif | 61 |
| | Negatif | 81 |
| | Netral | 26 |
| | Konflik | 3 |
| Tagihan | Positif | 2 |
| | Negatif | 132 |
| | Netral | 7 |
| | Konflik | 2 |
| Paket | Positif | 50 |
| | Negatif | 27 |
| | Netral | 25 |
| | Konflik | 3 |
| Promosi | Positif | 52 |
| | Negatif | 10 |
| | Netral | 4 |
| | Konflik | 2 |
| Aktivasi | Negatif | 22 |
| | Netral | 14 |

| | | |
|----------|---------|----|
| | Konflik | 3 |
| Campaign | Positif | 1 |
| | Negatif | 20 |
| | Netral | 1 |
| | Konflik | 1 |
| Aplikasi | Positif | 3 |
| | Negatif | 14 |
| | Netral | 9 |
| | Konflik | 1 |
| Report | Negatif | 2 |
| | Netral | 13 |

Pelabelan aspek dan sentimen juga dilakukan untuk data tes. Berikut distribusinya.

Tabel 6.2 Distribusi Label Aspek dan Sentimen Data Tes

| Data Train : 400 | | |
|-------------------------|-----------------------|---------------------|
| Label Aspek | Label Sentimen | Jumlah Tweet |
| Jaringan | Positif | 17+6+7 |
| | Negatif | 76+20 |
| | Netral | 1 |
| | Konflik | 11+2 |
| Kecepatan | Positif | 23+4+3+6 |
| | Negatif | 27+12+5 |
| | Konflik | 24+2 |
| Pengalaman | Positif | 10 |

| | | |
|------------------|---------|----------|
| | Negatif | 38 |
| Harga | Positif | 3+2+4 |
| | Negatif | 9+14+7+5 |
| | Netral | 10 |
| | Konflik | 1 |
| Layanan | Positif | 3+4 |
| | Negatif | 5+6 |
| | Konflik | 2 |
| Customer Service | Positif | 15 |
| | Negatif | 19 |
| | Netral | 1 |
| | Konflik | 1 |
| Tagihan | Negatif | 36 |
| Paket | Positif | 14 |
| | Negatif | 4 |
| | Netral | 2 |
| Promosi | Positif | 6 |
| | Negatif | 3 |
| Aktivasi | Negatif | 4 |
| Campaign | Negatif | 9 |
| | Netral | 6 |
| Aplikasi | Positif | 1 |
| | Negatif | 3 |
| | Netral | 1 |
| Report | Netral | 1 |

6.2 Hasil Word Embedding

Dari hasil pembersihan data, data ditokenisasi dan dimasukkan kedalam vocab. Terdapat 3 vocab yang diinisiasi untuk menampung kalimat, aspek, dan sentimen dari dataset. Vocab `text_field` untuk menampung kalimat review tweet, vocab `as_field` untuk menampung label sentimen pada dataset, dan vocab `sm_field` untuk menampung label setimen pada dataset. Berikut jumlah token yang disimpan pada vocab.

Tabel 6.3 Jumlah Vocab

| Vocab | Jumlah |
|------------|--------|
| Text_field | 5309 |
| As_field | 14 |
| Sm_field | 4 |

Vocab `text_fiel` dan `as_field` akan mengalami tahap word embedding. Word embedding yang dilakukan menggunakan pretrained word2vec embedding dari penelitian sebelumnya. Untuk melakukan embedding tersebut dibutuhkan 3 parameter, yaitu vocabulary yang akan diproses word embedding, batas inialisasi random (uniform) untuk embedding sebesar 0.25, dan dimensi embedding yang digunakan ialah 300.

Hasil dari embedding akan dirubah menjadi tensor dengan library torch. Berikut hasil embedding yang siap masuk kedalam model.

```
tensor([[ 0.0539,  0.1231, -0.0868, ..., -0.1874, -0.2151,  0.2173],
        [ 0.0000,  0.0000,  0.0000, ...,  0.0000,  0.0000,  0.0000],
        [-0.2389, -0.5745,  0.1374, ..., -0.2886,  0.5011,  0.0728],
        ...,
        [ 0.0645, -0.1538,  0.0094, ..., -0.0700,  0.1980, -0.2127],
        [ 0.0651,  0.5013, -0.4988, ...,  0.4987,  0.1850, -0.1261],
        [ 0.1362, -0.2552, -0.2862, ..., -0.0238,  0.1208,  0.1776]])
```

Gambar 6.1 Hasil Embedding

6.3 Hasil Pembuatan Model GCNN

Dari pembuatan model GCNN akan dihasilkan nilai akurasi train dan test. Berikut hasil training dan akurasi yang didapatkan dari model

6.3.1 Konfigurasi Parameter Awal

Sebelum melakukan training dan testing data, hyperparameter akan ditentukan terlebih dahulu sehingga hasilnya dapat dilihat dengan jelas. Berikut tabel konfigurasi awal.

Tabel 6.4 Konfigurasi Awal Model

| Parameter | Nilai |
|----------------------|---------------------|
| Trial | 5 |
| Epoch | 25 |
| Batch Size | 32 |
| Word Embedding Model | Pretrained Word2vec |
| Activation Function | Tanh ReLU |
| Pooling | Max_pool1d |

6.3.2 Pengaruh Optimizer dan Filter Region Size

Proses training dan testing dilakukan sebanyak 5 kali perulangan. Setiap perulangan tersebut digunakan epoch senilai 25.

6.3.2.1 Hasil Percobaan Optimizer Adagrad

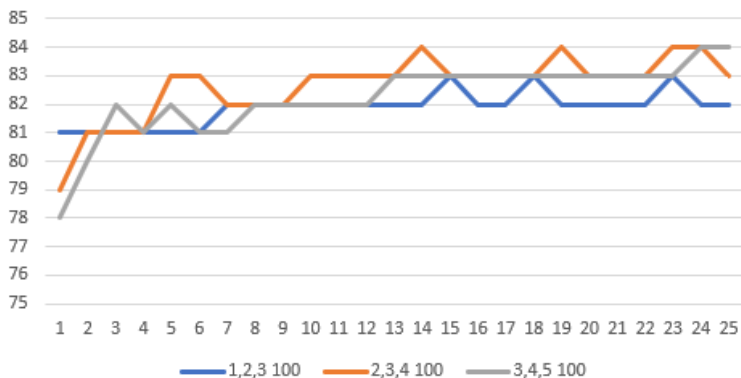
Pada percobaan ini akan diuji efek penggunaan optimizer Adagrad dengan perubahan multi-region size dan perubahan feature maps terhadap akurasi model. Adagrad adalah algoritma untuk optimasi berbasis gradien. Optimizer ini memiliki learning rate pada parameternya. Dalam performanya, optimizer ini nilai update yang besar ketika pembaruan parameter jarang terjadi dan akan memiliki nilai update yang kecil ketika pembaruan parameter sering terjadi [17]. Pengukuran yang digunakan

pada percobaan ini adalah akurasi, recall, precision, F-measure, dan waktu. Berikut hasil skenario ini.

Tabel 6.5 Hasil Akurasi Model

| LR | Region Size | Feature Maps | Acc | FM | Precision | Recall | Time |
|------|-------------|--------------|-------|-------|-----------|--------|--------|
| 0.01 | 1,2,3 | 100 | 81.88 | 55.43 | 67.28 | 53.50 | 113.58 |
| 0.01 | 2,3,4 | 100 | 82.48 | 55.42 | 72.72 | 53.39 | 119.82 |
| 0.01 | 3,4,5 | 100 | 82.24 | 53.67 | 69.80 | 52.05 | 126.87 |
| 0.01 | 1,2,3 | 200 | 82.44 | 55.44 | 70.45 | 53.15 | 147.81 |
| 0.01 | 2,3,4 | 200 | 82.64 | 55.72 | 75.32 | 53.34 | 182.85 |
| 0.01 | 3,4,5 | 200 | 82.28 | 54.45 | 74.37 | 52.74 | 182.85 |

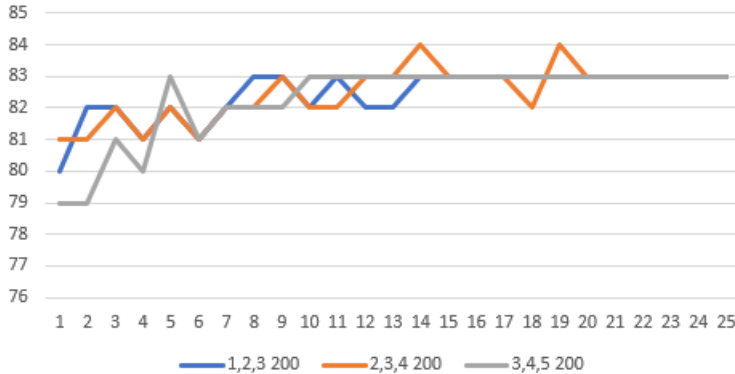
Dari tabel 6.7, dengan nilai region size 2,3,4 selalu memiliki nilai tertinggi untuk akurasi, f-measure, precision, dan recall baik pada feature maps 100 dan 200. Parameter terbaik tetap berdasarkan akurasi tertinggi yaitu pada akurasi tertinggi yang ada pada skenario dengan region size 2,3,4 dan feature maps 200.



Gambar 6.2 Rata-rata Akurasi Tiap Epoch

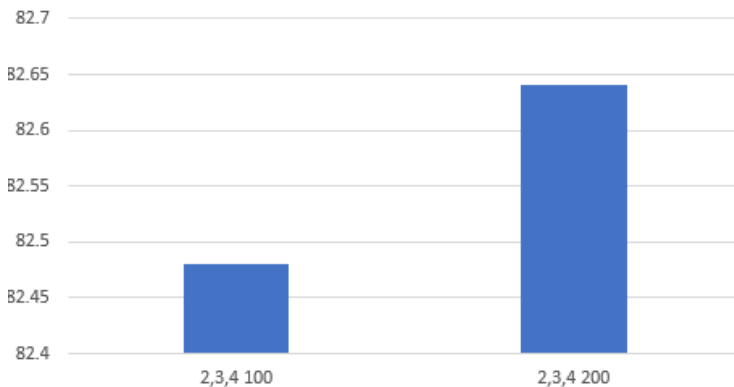
Dari gambar diatas dapat dilihat rata-rata nilai akurasi setiap epoch pada percobaan nilai feature maps 100 dan region

size kombinasi 1,2,3 – 2,3,4 – 3,4,5 pada optimizer Adagrad. Dari grafik dapat diketahui nilai rata-rata akurasi tertinggi ditunjukkan oleh region size 2,3,4 dan feature maps 100 yang mencapai nilai akurasi 84%.



Gambar 6.3 Akurasi Rata-rata tiap epoch

Pada penggunaan nilai feature maps 200, nilai akurasi rata-rata yang tertinggi ada pada kombinasi region size 2,3,4 dibandingkan nilai region size 1,2,3 dan 3,4,5 pada optimizer Adagrad.



Gambar 6.4 Akurasi Tertinggi dari Percobaan

Dari 6 kali kombinasi percobaan yang telah dilakukan diatas, dapat diketahui kombinasi region size 2,3,4 dengan feature maps 100 dan 200 adalah yang menghasilkan nilai rata-rata akurasi tertinggi. Jika keduanya dibandingkan seperti pada gambar 6.4, maka nilai rata-rata akurasi dari kombinasi region size 2,3,4 dan feature maps 200 adalah yang menghasilkan nilai akurasi lebih tinggi selisih 0.16% dari kombinasi region size 2,3,4 dengan nilai feature maps 200.

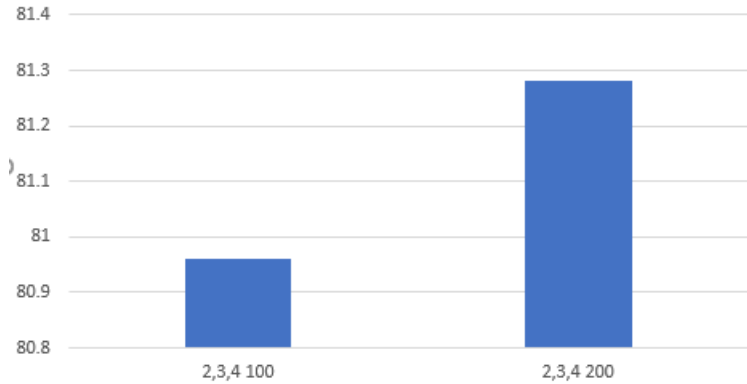
6.3.2.2 Hasil Percobaan Optimizer Adadelata

Pada percobaan ini akan diuji efek penggunaan optimizer Adadelata dengan perubahan multi-region size dan perubahan feature maps terhadap akurasi model. Adadelata adalah turunan dari optimizer adagrad untuk upaya mengurangi sifat agresifnya. Optimizer ini juga secara tetap menurunkan learning rate. Adadelata tidak mengakumulasi semua kuadrat gradien yang lalu, namun hanya mengakumulasi untuk satu ukuran waktu tertentu saja [17]. Pengukuran yang digunakan pada percobaan ini adalah akurasi, recall, precision, F-measure, dan waktu. Berikut hasil skenario ini.

Tabel 6.6 Hasil Akurasi Model

| LR | Region Size | Feature Maps | Acc | FM | Precision | Recall | Time |
|------|-------------|--------------|-------|-------|-----------|--------|--------|
| 0.01 | 1,2,3 | 100 | 80.32 | 53.43 | 66.35 | 52.50 | 142.15 |
| 0.01 | 2,3,4 | 100 | 80.96 | 56.06 | 70.27 | 55.04 | 176.44 |
| 0.01 | 3,4,5 | 100 | 80.84 | 54.90 | 69.64 | 53.66 | 147.49 |
| 0.01 | 1,2,3 | 200 | 80.72 | 55.25 | 68.09 | 54.08 | 164.61 |
| 0.01 | 2,3,4 | 200 | 81.26 | 56.66 | 71.66 | 55.46 | 181.38 |
| 0.01 | 3,4,5 | 200 | 80.76 | 55.11 | 73.44 | 54.26 | 202.11 |

Sama seperti pada percobaan sebelumnya, dari tabel 6.8 nilai tertinggi untuk akurasi, f-measure, precision, dan recall terdapat pada nilai region size 2,3,4 baik pada feature maps 100 dan 200. Parameter terbaik tetap berdasarkan akurasi tertinggi yaitu pada akurasi tertinggi yang ada pada skenario dengan region size 2,3,4 dan feature maps 200.



Gambar 6.5 Akurasi Tertinggi dari Percobaan

Pada penggunaan optimizer Adadelta, nilai rata-rata akurasi tertinggi ada pada kombinasi region size 2,3,4 pada feature maps 100 dan 200. Jika dibandingkan seperti pada gambar 6.5, dapat dilihat jika nilai rata-rata akurasi dari region size 2,3,4 dan feature maps 200 masih lebih tinggi dari feature maps 100. Selisihnya sebesar 0.3%.

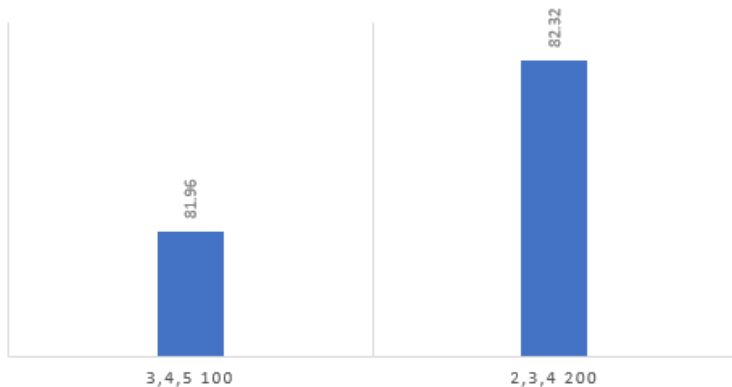
6.3.2.3 Hasil Percobaan Optimizer Adam

Pada percobaan ini akan digunakan optimizer Adam. Optimizer ini adalah metode untuk menghitung adaptive learning rate untuk setiap parameter. Adam menyimpan seluruh kuadrat gradien masa lampau dan juga menyimpan rata-rata gradien lampau dalam kurun waktu tertentu [17]. Pengukuran yang digunakan pada percobaan ini adalah akurasi, recall, precision, F-measure, dan waktu. Berikut hasil skenario ini

Tabel 6.7 Hasil Akurasi Model

| LR | Region Size | Feature Maps | Acc | FM | Precision | Recall | Time |
|-------|-------------|--------------|-------|-------|-----------|--------|--------|
| 0.001 | 1,2,3 | 100 | 81.88 | 57.46 | 74.10 | 55.05 | 120.90 |
| 0.001 | 2,3,4 | 100 | 81.64 | 55.06 | 72.85 | 52.98 | 124.44 |
| 0.001 | 3,4,5 | 100 | 81.96 | 55.17 | 72.56 | 53.14 | 153.94 |
| 0.001 | 1,2,3 | 200 | 82.04 | 57.02 | 73.75 | 54.44 | 185.41 |
| 0.001 | 2,3,4 | 200 | 82.32 | 56.46 | 76.51 | 54.06 | 167.59 |
| 0.001 | 3,4,5 | 200 | 81.96 | 54.06 | 71.51 | 52.04 | 186.57 |

Dari tabel 6.9, dapat diamati terjadi kenaikan akurasi, f-measure, precision, dan recall setiap kali ada kenaikan nilai pada region size dan feature maps. Pada percobaan dengan nilai feature maps 100 nilai akurasi tertinggi ada pada region size 3,4,5. Sedangkan pada percobaan dengan nilai feature maps 200 nilai akurasi tertinggi ada pada region size 2,3,4.

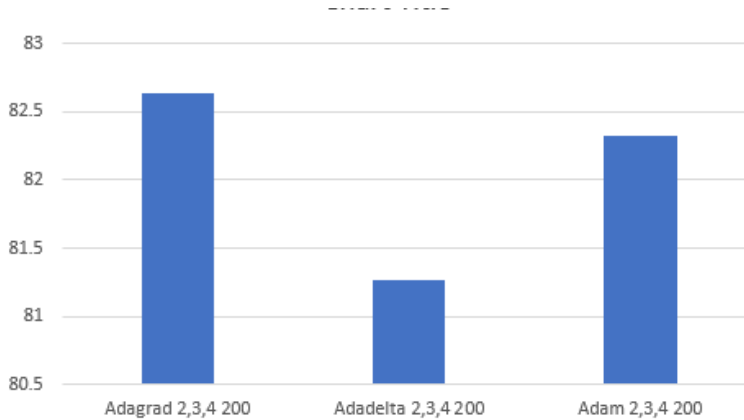


Gambar 6.6 Akurasi Tertinggi dari Percobaan

Dari gambar diatas dapat dilihat jika dibandingkan nilai akurasi dari region size 2,3,4 dan feature maps 200 pada optimizer adam dapat menghasilkan akurasi yang paling tinggi yaitu 82.32%.

6.3.2.4 Hasil Perbandingan Akurasi per Optimizer

Dari setiap nilai rata-rata akurasi tertinggi dari ketiga optimizer yang digunakan untuk percobaan, akan dibandingkan nilainya untuk melihat nilai akurasi yang dapat dicapai masing-masing optimizer.



Gambar 6.7 Perbandingan Akurasi Tiap Optimizer

Gambar 6.7 menunjukkan nilai rata-rata akurasi yang dapat dicapai oleh masing-masing optimizer dengan skenario percobaan yang telah dilakukan. Akurasi tertinggi dicapai oleh optimizer Adagrad dengan region size 2,3,4 dan nilai feature maps 200 yaitu 82.64% dengan selisih 0.32% lebih tinggi dari optimizer Adam dan selisih 1.38% lebih tinggi dari optimizer Adadelta.

6.3.3 Hasil Perbandingan Dengan Algoritma Lain

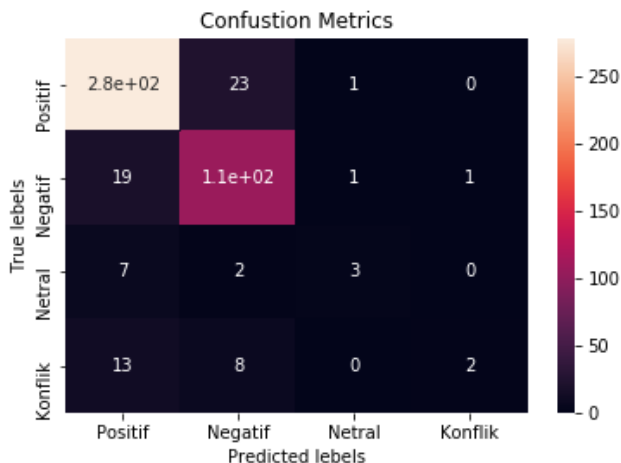
Hasil akurasi pada penelitian ini akan dibandingkan dengan algoritma lain yaitu CNN yang hasilnya dapat dilihat pada tabel berikut ini.

Tabel 6.8 Perbandingan Akurasi dengan Algoritma Lain

| | GCNN | CNN |
|---------|--------|--------|
| Akurasi | 82.64% | 81.04% |
| Time | 182.85 | 54.93 |

Berdasarkan tabel 6.10, akurasi yang dihasilkan dengan algoritma GCNN lebih tinggi dari CNN yaitu 82.64%. Dengan waktu yang lebih lama dari CNN yaitu 182.85 detik setiap trialnya.

6.3.4 Hasil Performa Tiap Sentimen dengan Model GCNN



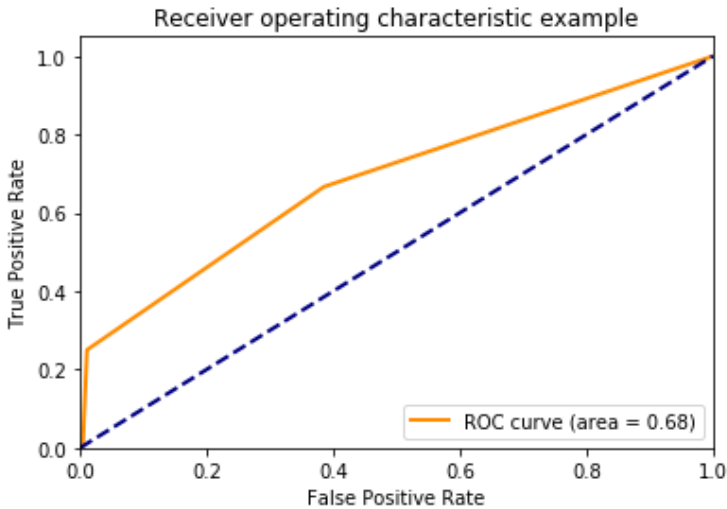
Gambar 6.8 Confusion Matrix GCNN

Tabel 6.9 Performa Sentimen

| | F1-Score | Precision | Recall |
|---------|----------|-----------|--------|
| Positif | 89.85% | 87.73% | 92.07% |
| Negatif | 79.85% | 76.42% | 83.59% |
| Netral | 35.29% | 60% | 25% |
| Konflik | 25% | 66.6% | 15.38% |

Dari gambar 6.8 dan tabel 6.11, performa terbaik ada pada sentimen positif. Hal ini ditunjukkan pula oleh confusion matrix pada gambar 6.8 bahwa kebenaran prediksi sentimen positif ialah yang terbanyak dibandingkan sentimen lainnya.

6.3.5 Grafik ROC/AUC



Gambar 6.9 Grafik ROC/AUC

Grafik Receiver Operating Characteristic (ROC) pada gambar 6.9 menunjukkan kinerja model GCNN dalam melakukan klasifikasi. Grafik ROC ditunjukkan dengan garis berwarna orange pada gambar 6.9. Sedangkan Area Under Curve (AUC) ialah daerah dibawah kurva ROC yang menunjukkan nilai 0.68 untuk model GCNN ini.

BAB VII

KESIMPULAN DAN SARAN

Bab kesimpulan dan saran membahas mengenai kesimpulan proses penelitian yang telah dilakukan dan saran yang diusulkan baik untuk perusahaan maupun untuk penelitian serupa di masa mendatang.

7.1 Kesimpulan

Dari proses penelitian pembuatan model GCNN untuk sentimen analisis menggunakan GCNN yang telah dilakukan adalah sebagai berikut:

1. Setelah melakukan percobaan pada 3 optimizer yang berbeda, penggunaan optimizer yang tepat dapat meningkatkan hasil akurasi.
2. Kombinasi region size yang tepat dapat meningkatkan akurasi yang dihasilkan model.
3. Nilai feature maps juga dapat meningkatkan akurasi model walaupun pengaruhnya tidak terlalu signifikan.
4. Dengan menggunakan parameter terbaik dari percobaan yang telah dilakukan, rata-rata akurasi yang dapat dicapai oleh model GCNN pada penelitian ini ialah 82.64%.
5. Perbandingan performa akurasi antara model GCNN dengan algoritma CNN menghasilkan akurasi yang berselisih 1.6% lebih tinggi untuk model GCNN yang hanya mencapai nilai akurasi setinggi 81.04%.
6. Rata-rata lama training dan testing model dalam satu kali trial pada model GCNN lebih lama dibandingkan dengan model CNN.

7.2 Saran

Dalam pengerjaan tugas akhir, terdapat beberapa saran yang diharapkan dapat bermanfaat bagi perusahaan maupun untuk pengembangan penelitian ke depan, yaitu:

1. Menambah lebih banyak jumlah data untuk mencapai akurasi yang lebih tinggi.
2. Mengkombinasikan nilai region size yang lain sehingga dapat menemukan kombinasi nilai region

size terbaik yang dapat semakin meningkatkan nilai akurasi.

3. Menentukan nilai feature maps terbaik untuk dapat meningkatkan nilai akurasi.
4. Meratakan persebaran sentimen yang seimbang pada data train dan data test sehingga dapat meningkatkan performa model dalam meprediksi sentimen.

DAFTAR PUSTAKA

- [1] I. Rahmayani, “Indonesia Raksasa Teknologi Digital Asia,” KOMINFO, [Online]. Available: https://kominfo.go.id/content/detail/6095/indonesia-raksasa-teknologi-digital-asia/0/sorotan_media. [Diakses 22 Januari 2019].
- [2] D. Hutabarat, “Inilah Rincian Jumlah Pelanggan Prabayar Masing-masing Operator,” KOMINFO, [Online]. Available: https://www.kominfo.go.id/content/detail/13131/inilah-rincian-jumlah-pelanggan-prabayar-masing-masing-operator/0/sorotan_media. [Diakses 21 Januari 2019].
- [3] A. T. Haryanto, “130 Juta Orang Indonesia Tercatat Aktif di Medsos,” DETIKINET, [Online]. Available: <https://inet.detik.com/cyberlife/d-3912429/130-juta-orang-indonesia-tercatat-aktif-di-medsos>. [Diakses 22 Januari 2019].
- [4] W. Xue dan T. L. , “Aspect Based Sentiment Analysis with Gated Convolutional Networks,” *Association for Computational Linguistics*, 2018.
- [5] R. P. Lebrecht, “Word Embeddings for Natural Language Processing,” Swiss, 2016.
- [6] A. A. Hania, “Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning,” *Yayasan Teknologi Indonesia*, 2017.
- [7] K. V. dan K. B. Cohen, “Natural Language Processing,” 2013.
- [8] A. Gelbukh, “International Journal of Computational Linguistics and Applications,” vol. 5, 2014.
- [9] V. A. Kharde dan S. S. Sonawane, “Sentiment Analysis of Twitter Data: A Survey of Techniques,” *International Journal of Computer Applications*, vol. 139, 2016.

- [10] M. P. D. G. J. P. H. P. I. A. dan S. M. , “SemEval-2014 Task 4: Aspect Based Sentimen Analysis,” *Semantic Evaluation*, pp. 27-35, 2014.
- [11] M. S. S. S. K. S. dan B. R. , “Joint Aspect and Polarity Classification for Aspect-based Sentiment Analysis with End-to-End Neural Network,” *Empirical Methods in Natural Language Processing*, 2018.
- [12] Y. N. Dauphin, A. Fan, M. Auli dan D. Grangier, “Language Modeling with Gated Convolutional Networks,” *International Conference on Machine*, 2017.
- [13] A. S. Cahyono, “Pengaruh Media Sosial terhadap Perubahan Sosial Masyarakat di Indonesia,” 2016.
- [14] S. A. H. Basori dan W. B. Surastyo, “Aplikasi Web Crawler untuk Web Content pada Mobile Phone,” 2009.
- [15] H. Noviyarto, “Pengaruh Perilaku Konsumen Mobile Internet,” *Jurnal Telekomunikasi dan Komputer*, vol. 1, 2010.
- [16] S. Ruder, “An overview of gradient descent optimization,” *Insight Centre for Data Analytics*, 2017.
- [17] K. H. L. C. P. H. dan M. S. , “What is Twitter, a Social Network or a News Media?,” 2010.

BIODATA PENULIS



Penulis lahir di Madiun pada tanggal 7 Maret 1997. Merupakan anak kedua dari 2 bersaudara. Penulis telah menempuh beberapa pendidikan formal yaitu, SDN 1 Mangkujayan Ponorogo, SMPN 1 Ponorogo, dan SMAN 1 Ponorogo.

Pada tahun 2015 pasca lulus SMA, penulis melanjutkan pendidikan di perguruan tinggi Institut Teknologi Sepuluh Nopember Surabaya di Departemen Sistem Informasi FTIK melalui jalur SNMPTN dan terdaftar sebagai mahasiswa dengan NRP 0521154000009. Selama menjadi mahasiswa, penulis mengikuti beberapa organisasi dan kepanitian. Penulis telah menjadi staf departemen Hubungan Luar HMSI ITS periode 2016/2017 dan juga sekaligus menjadi staf departemen OSR BEM FTIF pada periode yang sama 2016/2017. Pada tahun ketiga masa perkuliahan, penulis kembali meneruskan menjadi staf ahli pada departemen OSR BEM FTIF 2017/2018. Penulis juga menjadi anggota Pemandu ITS setelah mengikuti serangkaian kegiatan Latihan Ketrampilan Manajemen Mahasiswa (LKMM) Pra-Tingkat Dasar, Tingkat Dasar, dan Pelatihan Pemandu. Penulis juga sempat menjadi staf magang di software house Energeek dan Telkosel Area Jawa Bali.

Pada tahun keempat, karena penulis memiliki ketertarikan pada bidang pengolahan data, maka penulis mengambil bidang minat Akuisisi Data dan Diseminasi Infoemasi (ADDI). Penulis dapat dihubungi melalui email wenykp@gmail.com

Halaman ini sengaja dikosongkan

LAMPIRAN A. DATA BERLABEL

| TWEET | TOPIK | CATEGORIS | ASPECT |
|---|---------|------------------|-----------------|
| hallo indosat yang baik hati, dan provider yang ga ada matinya..terima kasih loh atas pelayanan yang sangat memuaskan #5monthwithindosat | INDOSAT | Pengalaman | Positif |
| Membalas @INDOSAT_M3 @INDOSAT_M3 bnr2 ngak diperhatiin kualitas ISAT BB! sibuk Promo trus tp kualitas sinyal ancurrr!!!! kecewa bneran dh pelayanan INDOSAT!! | INDOSAT | Jaringan Layanan | Negatif Negatif |
| Membalas @IndosatCare Mungkin sekitar 1 tahun aku pake provider Indosat. Sejauh ini cukup puas, nyaman & aman. Karena terbukti tarifnya paling murah & | INDOSAT | Harga Jaringan | Positif Positif |

| | | | |
|---|---------|------------|---------|
| <p>koneksi internet juga bisa diandalkan. Apalagi semenjak ada paket Yellow 1GB Rp 1.000 jadi makin betah deh ? #HariKonsumenNasional</p> | | | |
| <p>Sore min, pulsa IM3 saya berkurang terus padahal ga buat akses internet, telepon atau sms dan tadi sudah ke galeri indosat katanya sudah dilaporkan dan hanya disuruh nunggu saja. terima kasih @IndosatCare</p> | INDOSAT | Tagihan | Negatif |
| <p>@lewatmana hush jangan ngeflame. Saya pengguna telkomsel, sangat puas tuh dgn jaringan / servicenya.</p> | TSEL | Pengalaman | Positif |
| <p>Membalas @acieltelkomsel Bikin nangis emang harga paketannya ?</p> | TSEL | Harga | Negatif |

| | | | |
|---|------|--------------------|---------------------|
| <p>ga enak,3 aje mendinganRT @jvndrirnldi: Ganti telkonsel lah,sinyal kuatt,trs murah RT @RaldyWayne: jadi ga enak anjir axix,mahal pula,lama"g</p> | TSEL | Jaringan Har ga | Positif Posi tif |
|---|------|--------------------|---------------------|

Halaman ini sengaja dikosongkan

LAMPIRAN B. HASIL AKURASI

| Batchstep | Loss | Accuracy |
|------------------|-------------|-----------------|
| Epoch = 1 | | |
| 10 | 0.818516 | 62.00% |
| | | |
| 20 | 0.647292 | 75.00% |
| | | |
| 30 | 0.655068 | 75.00% |
| | | |
| 40 | 0.884376 | 68.00% |
| | | |
| 50 | 0.637318 | 78.00% |
| | | |
| Epoch = 2 | | |
| | | |
| 10 | 0.198749 | 100.00% |
| | | |
| 20 | 0.287622 | 87.00% |
| | | |
| 30 | 0.350233 | 87.00% |
| | | |
| 40 | 0.481854 | 78.00% |
| | | |
| 50 | 0.310766 | 93.00% |

| | | |
|-----------|----------|---------|
| Epoch = 3 | | |
| | | |
| 10 | 0.124354 | 100.00% |
| | | |
| 20 | 0.402991 | 87.00% |
| | | |
| 30 | 0.191909 | 93.00% |
| | | |
| 40 | 0.174943 | 93.00% |
| | | |
| 50 | 0.155455 | 93.00% |
| | | |
| Epoch = 3 | | |
| | | |
| 10 | 0.074333 | 100.00% |
| | | |
| 20 | 0.144051 | 93.00% |
| | | |
| 30 | 0.259015 | 93.00% |
| | | |
| 40 | 0.060959 | 100.00% |
| | | |
| 50 | 0.136821 | 96.00% |
| | | |

| | | |
|-----------|----------|---------|
| Epoch = 4 | | |
| | | |
| 10 | 0.109289 | 96.00% |
| | | |
| 20 | 0.091108 | 96.00% |
| | | |
| 30 | 0.046168 | 100.00% |
| | | |
| 40 | 0.04997 | 100.00% |
| | | |
| 50 | 0.079669 | 93.00% |
| | | |
| Epoch = 5 | | |
| | | |
| 10 | 0.022233 | 100.00% |
| | | |
| 20 | 0.022728 | 100.00% |
| | | |
| 30 | 0.033523 | 100.00% |
| | | |
| 40 | 0.187913 | 96.00% |
| | | |
| 50 | 0.097352 | 96.00% |
| | | |

| | | |
|-----------|----------|---------|
| Epoch = 6 | | |
| | | |
| 10 | 0.113233 | 96.00% |
| | | |
| 20 | 0.047661 | 100.00% |
| | | |
| 30 | 0.040847 | 100.00% |
| | | |
| 40 | 0.085563 | 96.00% |
| | | |
| 50 | 0.045236 | 96.00% |
| | | |
| Epoch = 7 | | |
| | | |
| 10 | 0.170543 | 93.00% |
| | | |
| 20 | 0.095351 | 96.00% |
| | | |
| 30 | 0.111372 | 93.00% |
| | | |
| 40 | 0.074503 | 96.00% |
| | | |
| 50 | 0.017941 | 100.00% |
| | | |

| | | |
|-----------|----------|---------|
| Epoch = 8 | | |
| | | |
| 10 | 0.017857 | 100.00% |
| | | |
| 20 | 0.023205 | 100.00% |
| | | |
| 30 | 0.049373 | 96.00% |
| | | |
| 40 | 0.062192 | 96.00% |
| | | |
| 50 | 0.00753 | 100.00% |
| | | |
| Epoch = 9 | | |
| | | |
| 10 | 0.008098 | 100.00% |
| | | |
| 20 | 0.00539 | 100.00% |
| | | |
| 30 | 0.078107 | 96.00% |
| | | |
| 40 | 0.007581 | 100.00% |
| | | |
| 50 | 0.013022 | 100.00% |
| | | |

| | | |
|------------|----------|---------|
| Epoch = 10 | | |
| | | |
| 10 | 0.052696 | 96.00% |
| | | |
| 20 | 0.013402 | 100.00% |
| | | |
| 30 | 0.209989 | 90.00% |
| | | |
| 40 | 0.019156 | 100.00% |
| | | |
| 50 | 0.151713 | 90.00% |
| | | |
| Epoch = 11 | | |
| | | |
| 10 | 0.084827 | 96.00% |
| | | |
| 20 | 0.068307 | 96.00% |
| | | |
| 30 | 0.036747 | 100.00% |
| | | |
| 40 | 0.004783 | 100.00% |
| | | |
| 50 | 0.00384 | 100.00% |
| | | |

| | | |
|------------|----------|---------|
| Epoch = 12 | | |
| | | |
| 10 | 0.002771 | 100.00% |
| | | |
| 20 | 0.01364 | 100.00% |
| | | |
| 30 | 0.025637 | 100.00% |
| | | |
| 40 | 0.059655 | 96.00% |
| | | |
| 50 | 0.065227 | 96.00% |
| | | |
| Epoch = 13 | | |
| | | |
| 10 | 0.108964 | 96.00% |
| | | |
| 20 | 0.106642 | 96.00% |
| | | |
| 30 | 0.003653 | 100.00% |
| | | |
| 40 | 0.047723 | 100.00% |
| | | |
| 50 | 0.018989 | 100.00% |
| | | |

| | | |
|------------|----------|---------|
| Epoch = 14 | | |
| | | |
| 10 | 0.005457 | 100.00% |
| | | |
| 20 | 0.003764 | 100.00% |
| | | |
| 30 | 0.010724 | 100.00% |
| | | |
| 40 | 0.084591 | 96.00% |
| | | |
| 50 | 0.006597 | 100.00% |
| | | |
| Epoch = 15 | | |
| | | |
| 10 | 0.022405 | 100.00% |
| | | |
| 20 | 0.040903 | 100.00% |
| | | |
| 30 | 0.005189 | 100.00% |
| | | |
| 40 | 0.1083 | 96.00% |
| | | |
| 50 | 0.031568 | 100.00% |
| | | |

| | | |
|------------|----------|---------|
| Epoch = 16 | | |
| | | |
| 10 | 0.005336 | 100.00% |
| | | |
| 20 | 0.008322 | 100.00% |
| | | |
| 30 | 0.038769 | 96.00% |
| | | |
| 40 | 0.005147 | 100.00% |
| | | |
| 50 | 0.089824 | 93.00% |
| | | |
| Epoch = 17 | | |
| | | |
| 10 | 0.004961 | 100.00% |
| | | |
| 20 | 0.059775 | 96.00% |
| | | |
| 30 | 0.005029 | 100.00% |
| | | |
| 40 | 0.003431 | 100.00% |
| | | |
| 50 | 0.011628 | 100.00% |
| | | |

| | | |
|------------|----------|---------|
| Epoch = 18 | | |
| | | |
| 10 | 0.00963 | 100.00% |
| | | |
| 20 | 0.00214 | 100.00% |
| | | |
| 30 | 0.011531 | 100.00% |
| | | |
| 40 | 0.003232 | 100.00% |
| | | |
| 50 | 0.014759 | 100.00% |
| | | |
| Epoch = 19 | | |
| | | |
| 10 | 0.005018 | 100.00% |
| | | |
| 20 | 0.001304 | 100.00% |
| | | |
| 30 | 0.001441 | 100.00% |
| | | |
| 40 | 0.003195 | 100.00% |
| | | |
| 50 | 0.006607 | 100.00% |
| | | |

| | | |
|------------|----------|---------|
| Epoch = 20 | | |
| | | |
| 10 | 0.017483 | 100.00% |
| | | |
| 20 | 0.008638 | 100.00% |
| | | |
| 30 | 0.018173 | 100.00% |
| | | |
| 40 | 0.005092 | 100.00% |
| | | |
| 50 | 0.028076 | 100.00% |
| | | |
| Epoch = 21 | | |
| | | |
| 10 | 0.002793 | 100.00% |
| | | |
| 20 | 0.016035 | 100.00% |
| | | |
| 30 | 0.004055 | 100.00% |
| | | |
| 40 | 0.006853 | 100.00% |
| | | |
| 50 | 0.00353 | 100.00% |
| | | |

| | | |
|------------|----------|---------|
| Epoch = 22 | | |
| | | |
| 10 | 0.005611 | 100.00% |
| | | |
| 20 | 0.00833 | 100.00% |
| | | |
| 30 | 0.008468 | 100.00% |
| | | |
| 40 | 0.008072 | 100.00% |
| | | |
| 50 | 0.003582 | 100.00% |
| | | |
| Epoch = 23 | | |
| | | |
| 10 | 0.009812 | 100.00% |
| | | |
| 20 | 0.014204 | 100.00% |
| | | |
| 30 | 0.006839 | 100.00% |
| | | |
| 40 | 0.018462 | 100.00% |
| | | |
| 50 | 0.006264 | 100.00% |
| | | |

| | | |
|------------|----------|---------|
| Epoch = 25 | | |
| | | |
| 10 | 0.03624 | 96.00% |
| | | |
| 20 | 0.002539 | 100.00% |
| | | |
| 30 | 0.02025 | 100.00% |
| | | |
| 40 | 0.004511 | 100.00% |
| | | |
| 50 | 0.004511 | 100.00% |
| | | |
| DevAcc | Time | |
| 84.0000 | 128.5191 | |
| 84.0000 | 132.5118 | |
| 83.0000 | 121.9105 | |
| 84.0000 | 128.4818 | |
| 83.0000 | 129.4762 | |