



TUGAS AKHIR - SM 141501

## **PENGGALIAN OPINI PADA ULASAN BUKU MENGUNAKAN ALGORITMA CNN - L2-SVM**

MUHAMMAD FAKHRUR ROZI  
NRP 1213 100 100

Dosen Pembimbing  
Dr. Imam Mukhlash, S.Si, MT  
Drs. Soetrisno, Ml.Komp

DEPARTEMEN MATEMATIKA  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





FINAL PROJECT - SM 141501

## **OPINION MINING ON BOOK REVIEW USING CNN - L2-SVM ALGORITHM**

MUHAMMAD FAKHRUR ROZI  
NRP 1213 100 100

Supervisor  
Dr. Imam Mukhlash, S.Si, MT  
Drs. Soetrisno, MI.Komp

DEPARTMENT OF MATHEMATICS  
Faculty of Mathematics and Natural Sciences  
Sepuluh Nopember Institute of Technology  
Surabaya 2017



**LEMBAR PENGESAHAN**

**PENGGALIAN OPINI PADA ULASAN BUKU  
MENGUNAKAN ALGORITMA CNN - L2-SVM**

***OPINION MINING ON BOOK REVIEW  
USING CNN-L2-SVM ALGORITHM***

**TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat  
Untuk memperoleh gelar Sarjana Sains  
Pada bidang studi Ilmu Komputer  
Program Studi S-1 Departemen Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember Surabaya

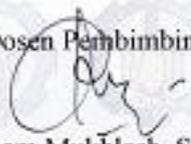
Oleh:  
**MUHAMMAD FAKHRUR ROZI**  
NRP. 1213 100 100

Dosen Pembimbing II

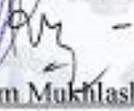
Menyetujui,

Dosen Pembimbing I

  
Drs. Soetrisno, M.I. Komp.  
NIP. 19571103 198603 1 003

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

Mengetahui,  
Kepala Departemen Matematika,

  
Dr. Imam Mukhlash, S.Si, MT  
NIP. 19700831 199403 1 003  
Surabaya, Agustus 2017



# **PENGGALIAN OPINI PADA ULASAN BUKU MENGUNAKAN ALGORITMA CNN-L2-SVM**

**Nama** : Muhammad Fakhurur Rozi  
**NRP** : 1213 100 100  
**Departemen** : Matematika  
**Dosen Pembimbing** : 1. Dr. Imam Mukhlash, S.Si, MT  
2. Drs. Soetrisno, MI.Komp

## **ABSTRAK**

Ulasan suatu produk dapat merepresentasikan kualitas dari produk tersebut. Suatu ekstraksi terhadap ulasan tersebut dapat digunakan untuk mengetahui sentimen dari opini yang diutarakan. Proses untuk mengekstraksi informasi yang berguna dari ulasan pengguna disebut *Opinion mining*. Model ekstraksi ulasan yang berkembang sekarang yaitu model *Deep Learning*. Model tersebut telah banyak digunakan untuk mendapatkan pencapaian performansi pada *Natural Language Processing*. Pada Tugas Akhir ini digunakan salah satu metode *deep learning* yaitu *Convolutional Neural Network* (CNN) sebagai ekstraksi fitur ulasan dan dilakukan klasifikasi dengan menggunakan L2 *Support Vector Machine* (SVM). Metode diimplementasikan untuk dapat mengetahui sentimen dari data ulasan buku. Hasil dari metode tersebut menunjukkan performansi pembelajaran sekitar 83.23% dan performansi pengujian sekitar 64.6 %.

**Kata kunci:** *Ulasan, Opinion Mining, Natural Language Processing, Deep Learning*



## ***OPINION MINING ON BOOK REVIEW USING CNN-L2-SVM ALGORITHM***

***Name*** : ***Muhammad Fakhrrur Rozi***  
***NRP*** : ***1213 100 100***  
***Department*** : ***Mathematics***  
***Supervisor*** : ***1. Dr. Imam Mukhlash, S.Si, MT***  
***2. Drs. Soetrisno, MI.Komp***

### ***ABSTRACT***

*Review of a product can represent quality of a product itself. An extraction to that review can be used to know sentiment of that opinion. Process to extract usefull information of user review is called Opinion Mining. Review extraction model that is enhacing nowadays is Deep Learning model. This Model has been used by many researchers to obtain excelent performance on Natural Language Processing. In this final project, one of deep learning model, Convolutional Neural Network (CNN) is used for feature extraction and L2 Support Vector Machine (SVM) as classifier. These methods are implemented to know the sentiment of book review data. The result of this method shows state-of-the art performance in 83.23% for training phase and 64.6% for testing phase.*

***Keywords: Review, Opinion Mining, Natural Language Processing, Deep Learning***



## KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillahirobbil'aalamiin, segala puji dan syukur bagi Allah SWT yang memiliki apa yang ada di langit dan di bumi dan yang telah memberikan limpahan rahmat, petunjuk serta hidayahNya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul

“PENGALIAN OPINI PADA ULASAN BUKU MENGGUNAKAN ALGORITMA CNN - L2-SVM”

sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika FMIPA Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas Akhir ini dapat terselesaikan dengan baik berkat bantuan dari berbagai pihak. Suatu kebahagiaan dan kewajiban bagi penulis untuk menyampaikan terima kasih kepada semua pihak yang telah membantu dan mendukung atas terselesainya Tugas Akhir:

1. Bapak Dr. Imam Mukhlash, S.Si, MT selaku Kepala Departemen Matematika ITS.
2. Bapak Dr. Imam Mukhlash, S.Si, MT dan Bapak Drs. Soetrisno, MI.Komp selaku pembimbing Tugas Akhir yang telah banyak memberikan bimbingan, arahan serta motivasi sehingga Tugas Akhir ini dapat terselesaikan.
3. Ibu Dian Winda Setyawati, S.Si, M.Si selaku Dosen Wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Deprtemen Matematika ITS.
4. Bapak Drs. Daryono Budi Utomo, M.Si, Ibu Alvida Mustika Rukmi, S.Si, M.Si, dan Bapak Drs. Kamiran,

M.Si selaku Dosen Penguji yang telah memberikan saran demi perbaikan Tugas Akhir ini.

5. Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi S1 Departemen Matematika ITS.
6. Seluruh jajaran dosen dan staf Jurusan Matematika ITS.
7. Keluarga tercinta yang senantiasa memberikan dukungan dan do'a dengan ikhlash.
8. Keluarga Himpunan Mahasiswa Matematika ITS yang telah berusaha memfasilitasi saya untuk berkembang.
9. Teman-teman angkatan 2013 yang saling mendukung dan memotivasi.
10. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Wassalamu'alaikum Wr. Wb

Surabaya, Agustus 2017

Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
<i>ABSTRACT</i> .....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan.....	4
1.5. Manfaat.....	4
1.6. Sistematika Penulisan Tugas Akhir.....	4
BAB II TINJAUAN PUSTAKA.....	7
2.1. Penelitian Terdahulu.....	7
2.2. <i>Opinion Mining</i> .....	8
2.3. <i>Natural Language Processing</i> .....	9
2.4. <i>Deep Learning</i> .....	10
2.5. <i>Word2vec</i> .....	10
2.6. <i>Convolutional Neural Network</i> .....	15
2.7. <i>Softmax</i> .....	21
2.8. <i>L2-Support Vector Machine</i> .....	22
2.9. CNN – L2-SVM.....	24
2.9.1. Tahap Umpan Maju.....	25
2.9.2. Tahap Umpan Mundur.....	25
2.9.3. Perhitungan Nilai <i>Loss</i> (Eror).....	29
BAB III METODOLOGI PENELITIAN.....	31
3.1. Pengumpulan Data.....	31
3.2. Praproses Data.....	31
3.3. Ekstraksi Fitur.....	32

3.4.	Implementasi Algoritma CNN-L2-SVM.....	32
3.5.	Evaluasi .....	33
3.6.	Penyusunan Laporan .....	33
3.7.	Diagram Alir Metodologi Penelitian .....	33
<b>BAB IV PERANCANGAN DAN IMPLEMENTASI .....</b>		<b>37</b>
4.1.	Pengumpulan Data .....	37
4.2.	Praproses Data .....	41
4.3.	Ekstraksi Fitur .....	44
4.4.	Desain Arsitektur CNN - L2-SVM.....	46
4.5.	Implementasi CNN – L2-SVM .....	48
4.6.	Analisis Implementasi Sistem .....	53
4.6.1.	Use Case Diagram .....	53
4.6.2.	Activity Diagram .....	53
4.7.	Lingkungan <i>Hardware</i> dan <i>Software</i> .....	54
4.8.	<i>Library</i> .....	55
4.9.	Implementasi Graphic User Interface (GUI) .....	55
4.10.	Algoritma.....	56
4.11.	Implementasi Sistem .....	57
<b>BAB V UJI COBA DAN EVALUASI SISTEM .....</b>		<b>63</b>
5.1.	Data Uji Coba.....	63
5.2.	Hasil Uji Coba Sistem .....	64
5.3.	Evaluasi Model.....	67
5.4.	Perbandingan Hasil Penelitian Terdahulu .....	70
<b>BAB VI KESIMPULAN DAN SARAN.....</b>		<b>73</b>
6.1.	Kesimpulan.....	73
6.2.	Saran.....	73
Daftar Pustaka .....		75
Biodata Penulis.....		79

## DAFTAR GAMBAR

Gambar 2.1.	Arsitektur model <i>Skip-gram</i> .....	12
Gambar 2.2.	Arsitektur MLP sederhana.....	15
Gambar 2.3.	Arsitektur model dengan dua channel 4untuk sebuah kalimat .....	16
Gambar 2.4.	Jaringan dengan 3 convolution layer .....	17
Gambar 2.5.	Contoh proses konvolusi pada suatu contoh kalimat .....	17
Gambar 2.6.	Prinsip Support Vector Machine .....	23
Gambar 2.7.	Diagram alir metodologi penelitian (1) .....	34
Gambar 2.8.	Diagram alir metodologi penelitian (2) .....	35
Gambar 4.1.	Diagram penyebaran data berdasarkan skor ulasan.....	40
Gambar 4.2.	Diagram alir proses filterisasi.....	42
Gambar 4.3.	Contoh tokenisasi pada kalimat ulasan.....	42
Gambar 4.4.	Diagram alir proses filterisasi.....	44
Gambar 4.5.	Mekanisme pembentukan vektor kata .....	45
Gambar 4.6.	Arsitektur jaringan Convolutional Neural Network .....	47
Gambar 4.7.	Contoh mekanisme pada convolution layer dengan 2 window dimana terdapat 2 filter tiap windownya .....	49
Gambar 4.8.	Diagram alir convolution layer.....	50
Gambar 4.9.	Contoh mekanisme <i>pooling</i> .....	51
Gambar 4.10.	Contoh gambaran jaringan pada output layer .....	52
Gambar 4.11.	Use case diagram sistem.....	53
Gambar 4.12.	Activity Diagram Sistem .....	54
Gambar 4.13.	Tampilan antar muka sistem.....	56
Gambar 5.1.	Grafik pergerakan akurasi data latih.....	65
Gambar 5.2.	Grafik pergerakan akurasi pada data uji .....	66
Gambar 5.3.	Hasil evaluasi model.....	69



## DAFTAR TABEL

Tabel 4. 1. Contoh ulasan buku positif dan negatif .....	38
Tabel 4. 2. Rincian jumlah tiap ulasan buku .....	39
Tabel 4. 3. Contoh data inputan .....	41
Tabel 4. 4. Tabel array kata ulasan hasil tokenisasi .....	43
Tabel 4. 5. Array kata ulasan tanpa stopwords.....	44
Tabel 4. 6. Matriks hasil ekstraksi fitur.....	45
Tabel 4. 7. Hyperparameter Convolution Layer.....	48
Tabel 4. 8. Lingkungan Hardware .....	55
Tabel 5. 1. Rincian Data.....	63
Tabel 5. 2. Data yang digunakan dalam proses komputasi .....	64
Tabel 5. 3. Data ulasan yang salah kelas .....	67
Tabel 5. 4. Hasil penelitian Kim 2014.....	71



# BAB I

## PENDAHULUAN

Pada bab ini akan dijelaskan mengenai hal-hal yang melatarbelakangi munculnya permasalahan dari Tugas Akhir ini. Selanjutnya permasalahan tersebut disusun menjadi rumusan masalah yang akan dikerjakan. Kemudian dibuat beberapa batasan masalah untuk memperoleh tujuan yang diinginkan, serta manfaat yang akan didapatkan. Adapun sistematika penulisan Tugas Akhir ini akan dijabarkan pada akhir bab ini.

### 1.1.Latar Belakang

Pengguna internet di dunia semakin meningkat setiap tahunnya. Menurut situs riset *Internet World Stats* Indonesia berada di posisi ke empat dalam penggunaan internet di Asia mengungguli Korea Selatan dan Thailand [1]. Hal ini memiliki dampak munculnya *E-Commerce* di kalangan masyarakat. *E-Commerce* mengacu pada penggunaan sarana elektronika dan teknologi untuk mengadakan perdagangan (menjual, membeli, transfer, atau bertukar produk, jasa, dan/atau informasi), termasuk dalam bisnis, interaksi bisnis antar bisnis, dan pelanggan dan pebisnis [2]. Pada pelaksanaannya terdapat banyak permasalahan dan perkembangan, sehingga muncul pendapat, saran, dan kritik dalam pembelian produk. *Review* (Ulasan) pada suatu produk mendefinisikan dari kualitas produk yang dipakai. Ulasan tersebut berupa opini-opini yang merepresentasikan baik atau tidaknya produk, tetapi dalam penerapannya sulit untuk mengartikan makna sebenarnya dari opini tersebut. Hal ini menunjukkan perlu dilakukan suatu ekstraksi terhadap ulasan tersebut. Proses untuk mengekstraksi informasi yang berguna dari ulasan pengguna disebut *Opinion mining*. Proses ini dilakukan dengan

menggunakan metode *Natural Language Processing* dan metode analisis teks [3]. Suatu ulasan atau opini diolah agar mengetahui secara akurat tentang emosi yang telah tertulis pada opini tersebut, sehingga didapatkan informasi kualitas dari sebuah produk yang dapat digolongkan menjadi baik, netral, atau buruk. Ulasan pada suatu produk memiliki peranan yang sangat penting dalam berbisnis. Pelanggan akan lebih percaya dengan ulasan orang lain dibandingkan dari pengembang. Hal ini mengakibatkan banyak dari penjual yang membutuhkan ulasan positif dari tiap produk agar dapat terjual dengan baik.

Terdapat banyak metode yang telah digunakan dalam penggalan suatu opini. Penelitian dari Pang dan Lee menggali opini dari ulasan film dengan menggunakan metode *Naïve Bayes* dan *Support Vector Machine* (SVM) serta dengan seleksi fitur berupa *Based on Minimum Cut* [4]. Penelitian tersebut menghasilkan performansi sebesar 86.4% [4]. Selain itu, seiring dengan perkembangan ilmu pengetahuan muncul bentuk model yang lebih modern dan sedang berkembang saat ini yaitu model *Deep Learning* (DL). Salah satu metode yang menerapkan model DL yaitu *Convolutional Neural Network* (CNN). Pada umumnya CNN digunakan pada pengolahan citra digital sebagai alat untuk klasifikasi maupun kluster. Sedangkan, pada penelitian yang dilakukan oleh Tang mengusulkan kombinasi dari CNN dan L2-SVM yang diuji pada data citra [5]. Penelitian tersebut mengganti fungsi aktivasi dari CNN yang umumnya menggunakan *softmax* menjadi L2-SVM. Setelah itu, pada tahun 2014 Kim menyampaikan inovasinya berupa penerapan model CNN pada NLP khususnya dalam klasifikasi kalimat [6]. Kim mengusulkan konsep baru dalam penggunaan CNN pada pengolahan teks yang menunjukkan bahwa CNN merupakan metode yang unggul dalam pengolahan teks.

Oleh karena itu, pada tugas akhir kali ini digunakan sebuah kombinasi metode CNN dan L2-SVM dalam pengklasifikasian suatu opini. Metode ini menghasilkan jenis kelas dari suatu opini menjadi dua kelas utama yaitu positif dan negatif. Metode CNN yang digunakan sama seperti pada penelitian Kim [6] akan tetapi fungsi aktivasi yang digunakan adalah L2-SVM seperti pada penelitian Tang [5]. Pengujian metode ini akan dilakukan pada data ulasan buku yang didapatkan dari situs online Amazon.com. Selain itu didapatkan performansi dari metode tersebut sebagai bahan pertimbangan dalam penelitian selanjutnya.

## **1.2.Rumusan Masalah**

Berdasarkan latar belakang dari Tugas akhir ini, maka dapat dirumuskan suatu permasalahan yaitu bagaimana cara mengklasifikasikan ulasan buku menggunakan algoritma CNN - L2-SVM dan mengukur performansinya.

## **1.3.Batasan Masalah**

Permasalahan dalam dunia nyata sangat luas oleh karena itu diberikan beberapa batasan masalah pada Tugas Akhir ini antara lain:

1. Data yang digunakan merupakan data ulasan buku di situs Amazon.com.
2. Ulasan berupa kalimat berbahasa Inggris.
3. Proses klasifikasi tidak mempertimbangkan susunan kata dan frase kata.
4. Data ulasan sudah berlabel oleh pengguna di Amazon.com

#### **1.4.Tujuan**

Adapun tujuan dari tugas akhir ini yaitu mengembangkan perangkat lunak yang mampu mengklasifikasikan ulasan buku menggunakan algoritma CNN - L2-SVM dan mengukur performansinya.

#### **1.5.Manfaat**

Manfaat Tugas Akhir kali ini adalah memberi gambaran unjuk kerja algoritma CNN-L2-SVM, dalam jangka panjang mem berikan rekomendasi kepada masyarakat terhadap kualitas produk melalui ulasan, dan sebagai bahan referensi baru untuk penelitian berikutnya mengenai *Opinion Mining* dan *Deep Learning*.

#### **1.6.Sistematika Penulisan Tugas Akhir**

Tugas akhir ini mempunyai susunan dalam penulisan agar tersusun secara sistematis dan memudahkan pembaca untuk mempelajarinya. Berikut sistematika penulisan Tugas Akhir ini:

##### **1. BAB I PENDAHULUAN**

Pada bab ini menjelaskan gambaran umum dari penulisan Tugas Akhir yang terdiri atas latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

##### **2. BAB II TINJAUAN PUSTAKA**

Pada bab ini dijelaskan beberapa teori dasar yang mendukung dalam pengerjaan Tugas Akhir ini yang meliputi penelitian terdahulu, *Opinion Mining*, *Natural Language Processing*, *Deep Learning*, *Convolutional Neural Network*, *Softmax*, *Support Vector Machine*, dan data uji coba.

### 3. BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan tahap-tahap pengerjaan dalam menyelesaikan Tugas Akhir ini sehingga penelitian dapat dirancang secara sistematis dan diatur dengan sebaik-baiknya.

### 4. BAB IV PERANCANGAN DAN IMPLEMENTASI

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

### 5. BAB V UJI COBA DAN EVALUASI SISTEM

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

### 6. BAB VI KESIMPULAN DAN SARAN

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



## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini dijelaskan beberapa teori dasar yang mendukung dalam pengerjaan Tugas Akhir ini yang meliputi penelitian terdahulu, *Opinion Mining*, *Natural Language Processing*, *Deep Learning*, *Convolutional Neural Network*, *Softmax*, dan *Support Vector Machine*. Adapun data uji coba yang akan digunakan pada Tugas Akhir ini dijelaskan pada akhir bab ini.

#### **2.1. Penelitian Terdahulu**

Terdapat metode yang telah digunakan dalam penggalian suatu opini. Berawal dari penelitian dari Pang dan Lee yang mencoba menggali opini dari ulasan film dengan menggunakan metode *Naïve Bayes* dan *Support Vector Machine* (SVM) serta dengan seleksi fitur berupa *Based on Minimum Cut*. Penelitian tersebut menghasilkan performansi sebesar 86.4% [4]. Selanjutnya penelitian tentang penggalian opini berlanjut dengan berbagai metode dan modifikasi seperti *Maximum Entropy*, *Backpropagation*, *K-means*, dengan performansi berturut-turut 85.4%, 86%, dan 78% [4]. Selain itu seiring dengan perkembangan ilmu pengetahuan muncul bentuk model yang lebih modern dan sedang berkembang saat ini yaitu model *Deep Learning* (DL).

Pada tahun 2013 Tang [5] melakukan penelitian tentang implementasi model DL menggunakan SVM. Metode DL yang digunakan pada penelitian tersebut yaitu metode CNN. Penelitian tersebut mencoba membandingkan penggunaan SVM dalam DL dengan DL pada umumnya. Tang mengganti fungsi aktivasi dari CNN yang umumnya menggunakan *softmax* menjadi L2-SVM.

Hal ini dilakukan karena peneliti melihat persamaan yang sama antara *softmax* dan L2-SVM dengan eror yang dihasilkan lebih kecil dibandingkan dengan menggunakan *softmax*. Penelitian ini menyimpulkan bahwa dengan mengganti *softmax* dengan L2-SVM secara mudah dan membantu dalam tugas pengklasifikasian [5].

Selain itu pada tahun 2014 Kim [6] memperkenalkan penerapan metode CNN dalam klasifikasi suatu kalimat. Penelitian tersebut memperkenalkan model baru untuk diterapkan pada NLP. Kalimat dan kata sebagai input terlebih dahulu dibuat kedalam bentuk vektor. Kim menyimpulkan pada penelitian ini bahwa CNN sederhana dengan satu *layer* konvolusi menghasilkan performansi yang baik [6].

## **2.2. Opinion Mining**

*Opinion Mining* (Penggalian Opini) merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini [7]. Penggalian frase ulasan dan ekstrasi penilaian juga dilakukan agar komputer dapat mengenali dan mengekspresikan emosi dari teks [8]. *Opinion mining* dapat disebut juga dengan analisis sentimen [9]. Analisis sentimen ini dibagi menjadi 2 kategori besar:

### *1. Coarse-grained sentiment analysis*

Pada analisa kategori ini objek yang diamati merupakan sebuah dokumen yang mengandung sebuah opini. Dokumen tersebut dikategorikan berdasarkan kecenderungan dari opini yang berada di dokumen. Kategorinya dapat berupa negatif, netral, dan positif.

## 2. *Fined-grained sentiment analysis*

Pada jenis ini objek yang diamati berupa kalimat suatu opini. Analisa ini lebih sulit dibandingkan dokumen karena informasi yang tersedia hanya terbatas pada satu kalimat. Kalimat tersebut dapat dikategorikan menjadi negatif dan positif.

Beberapa sumber yang sering digunakan untuk analisis sentimen adalah *SentiWordNet* dan *WordNet*. Analisis sentimen terdiri dari 3 subproses besar [10]. Masing-masing subproses tersebut antara lain:

- a. *Subjectivity Classification.*
- b. *Orientation Detection.*
- c. *Opinion Holder and Target Detection.*

## **2.3. *Natural Language Processing***

*Natural Language Processing* (NLP) merupakan sebuah area penelitian dan aplikasi yang menyelidiki bagaimana komputer dapat digunakan untuk memahami dan memanipulasi teks bahasa alami atau pembicaraan untuk melakukan hal yang berguna. Peneliti NLP bertujuan untuk mengumpulkan pengetahuan pada bagaimana manusia mengerti dan menggunakan bahasa sehingga teknik dan alat yang sesuai dapat dikembangkan untuk membuat sistem komputer mengerti dan memanipulasi bahasa alami supaya bekerja dengan tugas yang diinginkan. Dasar dari NLP berada dalam beberapa disiplin ilmu antara lain ilmu komputer dan informasi, ilmu bahasa, matematika, teknik elektro dan elektronika, kecerdasan buatan dan robotika, dan psikologi. Penerapan NLP termasuk dalam

beberapa bidang studi seperti mesin terjemahan, pengolahan teks bahasa alami dan rangkuman, tampilan *user*, pengambilan informasi lintas bahasa dan multibahasa, pengenalan suara, kecerdasan buatan, dan sistem pakar [11].

## **2.4. Deep Learning**

*Deep Learning* merupakan suatu sub-bidang machine learning yang mencoba untuk belajar abstraksi tingkat tinggi dalam data dengan memanfaatkan arsitektur hirarki. Hal ini merupakan pendekatan yang tampak dan telah diterapkan secara luas dalam domain kecerdasan buatan tradisional, seperti *semantic parsing*, *transfer learning*, *natural language processing*, *computer vision*, dan masih banyak lagi. Dalam akhir-akhir tahun ini, *Deep Learning* telah dipelajari secara ekstensif dalam bidang *computer vision*. Pada strukturnya *Deep Learning* memiliki jumlah *Hidden Layer* lebih dari satu. Umumnya, metode ini dapat dibagi menjadi empat kategori antara lain [12]:

1. *Convolutional Neural Networks (CNNs)*
2. *Restricted Boltzmann Machines (RBMs)*
3. *Autocoder*
4. *Sparse Coding*

## **2.5. Word2vec**

Bentuk representasi suatu kata dengan numerik seperti vektor merupakan salah satu cara untuk mendapatkan nilai dari makna yang terkandung suatu kalimat. Pada Tugas Akhir ini digunakan suatu bentuk vektor representasi kata terdistribusi dengan menggunakan algoritma *Word2vec* untuk mendapatkan polaritas sentimen dari ulasan buku.

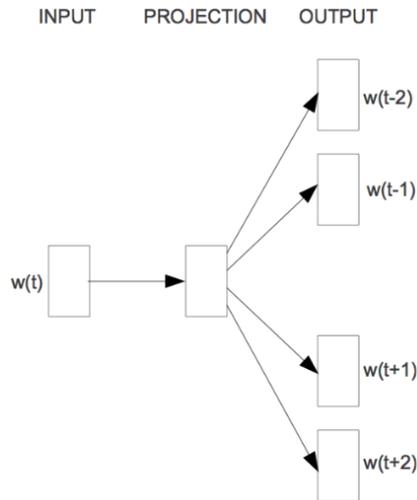
*Word2vec* merupakan suatu cara untuk membuat suatu bentuk representasi terdistribusi kata dalam suatu ruang vektor. Vektor tersebut dapat membantu algoritma pembelajaran untuk

mencapai performansi yang lebih baik dalam NLP dengan mengelompokkan kata-kata yang mirip atau sama. Ada beberapa macam cara dalam membuat vektor kata. Salah satu algoritma yang sedang berkembang saat ini adalah *Global Vectors (Glove)* [16]. *Glove* merupakan suatu algoritma pembelajaran yang tak diawasi untuk mendapatkan representasi vektor kata-kata. Algoritma ini memasukkan peluang munculnya kata dalam suatu *window* (persekitaran kata) ke dalam perhitungannya.

Misalkan matriks dari jumlah kata-kata yang muncul dinotasikan dengan  $X$ , dimana  $X_{ij}$  mengindikasikan jumlah munculnya kata  $j$  yang terjadi dalam konteks kata  $i$ . Dan misalkan  $X_i = \sum_k X_{ik}$  adalah total jumlah kemunculan semua kata terhadap konteks kata  $i$ . Maka dapat diambil probabilitas kata  $j$  muncul dalam konteks kata  $i$  sebagai berikut:

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} \quad (2.1)$$

Pada dasarnya *Glove* merupakan suatu bentuk umum dari algoritma *Skip-gram* yang merupakan salah satu algoritma representasi vektor kata terdistribusi yang telah dikembangkan sebelumnya oleh Mikolov [17]. Ide dasar dari model *Skip-gram* yaitu memprediksi kata-kata disekitar suatu kata. *Skip-gram* dapat digambarkan pada **Gambar 2.1**.



Gambar 2.1. Arsitektur model *Skip-gram*

Pada awalnya kata-kata dalam kosa kata disimbolkan dengan vektor “*one-hot*”. Representasi vektor ini menyimbolkan suatu kata dengan vektor yang nilainya hanya 1 pada posisi tertentu dan 0 untuk yang lain.

$$\underline{x}_d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

$$x_k = 1 ; \underline{x}_{k'} = 0 \text{ untuk } \forall k' \neq k$$

Akan tetapi representasi vektor “*one-hot*” memiliki kelemahan yaitu tidak bisa didapatkan relasi antar vektor kata

dalam kosa kata yang ada. Oleh karena itu model *Skip-gram* mencoba membuat representasi vektor yang mendekati vektor tersebut. Model ini mencoba memaksimalkan probabilitas tiap representasi vektor terhadap vektor “one-hot”. Model ini membagi vektor kata menjadi 2 jenis vektor antara lain vektor kata dan vektor konteks kata yang dapat disimbolkan sebagai  $v$  dan  $u$ . Secara matematis, fungsi objektif dari model *Skip-gram* dapat dituliskan sebagai berikut:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t) \quad (2.2)$$

Ketrangan:

$w$  = kata  
 $c$  = ukuran dari window  
 $p(w_{t+j} | w_t)$  = peluang besyarat dari keakurasian kata  $w_t$  terhadap kata  $w_{t+j}$ .

Peluang tersebut dapat jabarkan sebagai fungsi *softmax* sebagai berikut:

$$p(w_o | w_c) = \frac{\exp(v'_{w_o} u_{w_c})}{\sum_{w=1}^W \exp(v'_{w} u_{w_c})} \quad (2.3)$$

Dengan  $W$  merupakan banyaknya kosa kata.

Inti dari model *Skip-gram* yaitu memaksimalkan peluang logaritmik dari vektor kata dengan konteksnya. Oleh sebab itu proses pelatihan perlu dilakukan dengan data yang sangat besar untuk mencapai titik optimumnya. Sehingga dapat diperoleh

vektor representasi semua kata dalam kosa kata yang ada dan dapat digunakan dalam proses NLP selanjutnya.

Pada sisi yang lain, *Glove* mencoba mengembangkan model *Skip-gram* dengan menambahkan salah satu nilai yaitu peluang kemunculan suatu kata. Hal memiliki keunggulan yaitu kata-kata yang sangat jarang muncul memiliki kesamaan dengan kata-kata yang berkaitan. Sebagai contoh jika dicari kata-kata yang berkaitan dengan kata *frog* (katak) maka hasil yang muncul antara lain: *frogs*, *load*, *liloria*, *leplodaclylidae*, *rana*, *lizard*, *eleutherodaclylus* [18]. Beberapa kata tersebut merupakan spesies dari katak (*frog*).

Fungsi objektif dari *Glove* secara matematis adalah sebagai berikut:

$$J(\theta) = \sum_{i,j} X_i (u_i^T v_j - \log X_{ij})^2 \quad (2.4)$$

Keterangan:

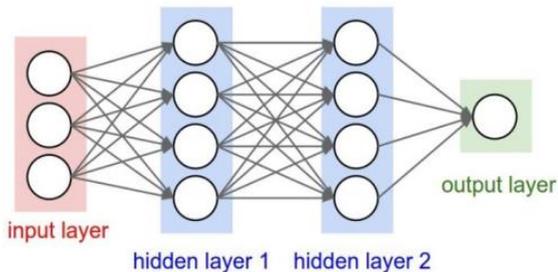
- $J(\theta)$  : fungsi objektif
- $\theta$  : semua parameter yang digunakan
- $X$  : jumlah kata yang muncul
- $u$  : vektor konteks kata
- $v$  : vektor kata

Pada Tugas Akhir ini digunakan vektor representasi kata yang dihasilkan oleh algoritma *Glove*. Data latih yang digunakan untuk membentuk vektor representasi kata yaitu berasal dari data kata Wikipedia pada tahun 2014 dan Gigaword 5. Kata yang digunakan sebanyak 6 miliar kata dengan kosa kata sebanyak 400.000 kata [17]. Representasi vektor yang dihasilkan memiliki beberapa dimensi antara lain 50, 100, 200, dan 300. Agar mempermudah dalam waktu komputasi, Tugas Akhir ini

menggunakan representasi vektor kata dengan dimensi 50 dengan tujuan untuk mempermudah dalam proses komputasi. Vektor tersebut digunakan sebagai referensi dalam pembentukan vektor representasi awal sebelum dilakukan proses pengklasifikasian.

## 2.6. Convolutional Neural Network

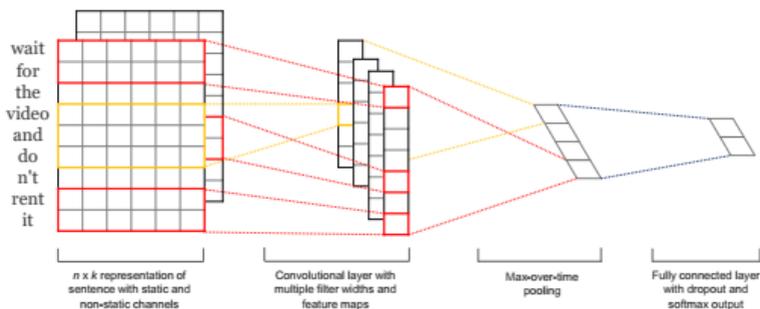
*Convolutional Neural Network* (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi.



Gambar 2.2. Arsitektur MLP sederhana

Sebuah MLP seperti pada **Gambar 2.2.** memiliki  $i$  layer (kotak merah dan biru) dengan masing-masing layer berisi  $ji$  neuron (lingkaran putih). MLP menerima input data satu dimensi dan mempropagasikan data tersebut pada jaringan hingga menghasilkan output. Setiap hubungan antar *neuron* pada dua layer yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Disetiap data input pada layer

dilakukan operasi linear dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi nonlinear yang disebut sebagai fungsi aktivasi [13].

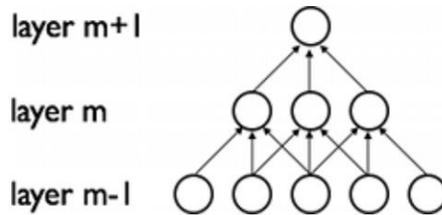


Gambar 2.3. Arsitektur model dengan dua channel untuk sebuah kalimat

Sekarang CNN dapat diterapkan pada dua hal yaitu pada *computer vision* dan NLP. **Gambar 2.3** merupakan arsitektur model dari CNN pada NLP [6]. Bagaimanapun juga terdapat dua komponen utama dari CNN yaitu *convolution layer* dan *pooling layer*.

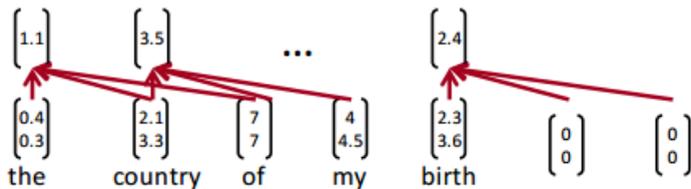
### 1. Convolution Layer

Pada tiap *convolution layer*, filter konvolusi menggunakan korelasi lokal dengan memaksakan pola konektivitas lokal diantara layer yang berdekatan. Layer yang lebih atas  $m$  telah didapat dari sebuah subhimpunan satuan dari layer yang lebih rendah  $m - 1$ . Dibandingkan dengan MLP, keuntungan lainnya dari *convolution layer* adalah jumlah parameter yang berkurang signifikan disebabkan oleh parameter bagi. Konektivitas tersebut digambarkan dalam **Gambar 2.4** [14].



Gambar 2.4. Jaringan dengan 3 convolution layer

Lapisan ini merupakan lapisan pertama dalam arsitektur jaringan CNN – L2-SVM. Pada lapisan ini dilakukan proses konvolusi untuk semua vektor kata yang ada pada ulasan. **Gambar 2.5** menunjukkan contoh dari proses konvolusi pada suatu kalimat dimana untuk kata yang terakhir vektor kata setelahnya yang berada di window dibuat suatu vektor kata yang berupa vektor nol. Akan tetapi pada Tugas Akhir ini konvolusi berakhir sampai vektor kata ke-  $n - h + 1$ .



Gambar 2.5. Contoh proses konvolusi pada suatu contoh kalimat

Jika vektor-vektor kata digabungkan, maka ulasan dapat direpresentasikan sebagai berikut

$$x_{1:n} = x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n \quad (2.5)$$

Keterangan:

$n$  : banyaknya vektor kata

$x_i$  : vektor kata yang berada pada  $\mathbb{R}^{50}$  dengan indeks ke- $i$

Dimana  $\oplus$  merupakan operator penggabungan. Secara umum  $x_{i:i+j}$  berarti hasil gabungan vektor-vektor kata dari indeks ke- $i$  sampai ke- $i + j$  seperti berikut .

$$x_i, x_{i+1}, x_{i+2}, \dots, x_{i+j} \quad (2.6)$$

Operasi konvolusi menggunakan sebuah filter untuk menghasilkan suatu nilai *feature* maka dapat dirumuskan sebagai berikut

$$c_i = f(\text{net}) \quad (2.7)$$

$$\text{net} = w \cdot x_{i:i+h-1} + b \quad (2.8)$$

Keterangan:

$c_i$  : nilai *feature map* pada indeks ke- $i$

$h$  : ukuran *window* kata

$w$  : filter yang berada pada  $\mathbb{R}^{50h}$

$b$  : parameter bias

Pada Tugas akhir ini fungsi non linear yang digunakan merupakan fungsi *Rectified Linear Unit (ReLU)*. Keluaran dari fungsi tersebut memberikan batasan keluaran bernilai positif. Fungsi tersebut dapat dituliskan sebagai berikut:

$$\text{ReLU}(x) = \max(0, x) \quad (2.9)$$

Sehingga persamaan menjadi seperti berikut:

$$c_i = \text{ReLu}(\text{net}) \quad (2.10)$$

$$\text{net} = w \cdot x_{i:i+h-1} + b \quad (2.11)$$

Filter  $w$  di terapkan untuk tiap *window* kata yang mungkin di dalam kalimat ulasan  $\{x_{1:h}, x_{2:h}, x_{3:h}, \dots, x_{n-h+1:n}\}$  sehingga dihasilkan sebuah *feature map* seperti berikut:

$$c = [c_1, c_2, c_3, \dots, x_{n-h+1}] \quad (2.12)$$

$c$  merupakan *feature map*.

Untuk mendapatkan hasil yang maksimal maka filter yang akan digunakan pada Tugas Akhir ini lebih dari satu filter dan begitupun juga dengan *window* kata.

## 2. *Pooling Layer*

Komponen penting kedua CNN adalah *pooling*. Fungsi *layer pooling* sebagai *down-sampling* yang non linear. Algoritma *pooling* hanya fokus pada sub-daerah tertentu dari *feature map*. Tujuan dari *pooling* adalah mencoba mendapatkan informasi yang paling bermanfaat dari daerah tertentu. Algoritma ini sangat penting karena beberapa hal antara lain:

1. Dengan mengeliminasi nilai yang tak maksimal, mengurangi komputasi dari layer yang di atasnya.
2. Menyediakan sebuah bentuk dari translasi invarian.
3. Beberapa kasus, *pooling* membantu untuk merangkum perbedaan panjang fitur menjadi dimensi yang sama [14].

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

$$\hat{c} = \max\{c\} \quad (2.13)$$

$\hat{c}$  merupakan nilai maksimum dari *feature map*  $c$

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

$$z = [\hat{c}_1, \hat{c}_2, \hat{c}_3, \dots, \hat{c}_m] \quad (2.14)$$

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

### 3. *Output layer*

Lapisan terakhir arsitektur jaringan pada Tugas Akhir ini adalah *output layer*. Pada lapisan ini akan digunakan jaringan syaraf tiruan yang bertujuan untuk mengklasifikasikan ulasan ke kelas-kelas yang ditentukan antara ulasan positif atau negatif. Hasil dari *pooling layer* yang berupa vektor *feature* akan melewati hubungan penuh jaringan *Support Vector Machine* (SVM) untuk mendapatkan skor ulasan. Pada Tugas Akhir ini ulasan positif diberi label +1 dan ulasan negatif diberi label -1.

Skor yang bernilai positif digolongkan menjadi ulasan positif, sedangkan skor yang bernilai negatif digolongkan menjadi ulasan negatif.

## 2.7. Softmax

Untuk permasalahan klasifikasi menggunakan teknik DL, suatu hal yang standar untuk menggunakan *softmax* atau pengkodean 1-of-K pada layer paling atas. Fungsi *softmax* tersebut dapat dituliskan sebagai berikut:

$$a_i = \sum_k h_k W_{ki} \quad (2.15)$$

Keterangan:

$a_i$  : total input menuju layer *softmax* untuk kelas ke-i

$h_k$  : inputan layer *softmax*

$W_{ki}$  : bobot layer *softmax*

Kemudian dipunyai

$$p_i = \frac{\exp(a_i)}{\sum_j^n \exp(a_j)} \quad (2.16)$$

Keterangan:

$p_i$  : probabilitas dari kelas ke-i

$n$  : banyaknya kelas

Kelas prediksi  $\hat{t}$  akan menjadi

$$\hat{t} = \underset{t}{\operatorname{argmax}} p_i = \underset{t}{\operatorname{argmax}} a_i \quad (2.17)$$

Dengan  $\hat{t}$  merupakan kelas ke-i.

### 2.8.L2-Support Vector Machine

*Linear Support Vector Machine* pada mulanya diformulasikan untuk klasifikasi biner. Permasalahan klasifikasi pada SVM dapat diformulasikan sebagai berikut:

$$L = \min_{w, \xi_n} \frac{1}{2} w^T w + C \sum_{n=1}^N \xi_n \quad (2.18)$$

Bersyarat pada  $w^T x_n t_n \geq 1 - \xi_n \quad \forall n$

$$\xi_n \geq 0 \quad \forall n$$

Keterangan:

- $w$  : parameter bobot SVM
- $x_n$  : vektor input data
- $\xi_n$  : variabel bebas yang mengkoreksi titik data yang melanggar syarat batas
- $t_n$  : label target data input
- $n$  : banyaknya data pembelajaran

Perlu diperhatikan bahwa dapat dimasukkan bias dengan menambah semua vektor data  $x_n$  dengan nilai skalar 1. Masalah optimasi tak dibatasi yang terhubung sebagai berikut:

$$L = \min_w \frac{1}{2} w^T w + C \sum_{n=1}^N \max\{1 - w^T x_n t_n, 0\} \quad (2.19)$$

$C$  merupakan konstanta sebarang.

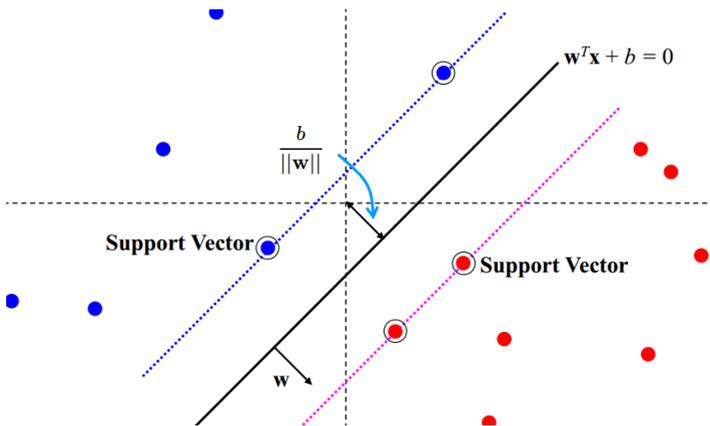
Persamaan objektif (2.5) dikenal sebagai masalah bentuk utama dari L1-SVM, dengan kerugian standar. Karena L1-SVM

tidak terdeferensiasi maka versi yang populer dikenal dengan L2-SVM yang meminimalkan kerugian kuadrat:

$$L = \min_w \frac{1}{2} w^T w + C \sum_{n=1}^N \max\{1 - w^T x_n t_n, 0\}^2 \quad (2.20)$$

L2-SVM dapat diturunkan dan menyebabkan sebuah kerugian lebih besar untuk titik-titik yang melanggar batas. Untuk memprediksi label kelas dari data tes  $x$ :

$$\operatorname{argmax}_t (w^T x) t \quad (2.21)$$



Gambar 2.6. Prinsip Support Vector Machine

Persamaan tersebut merupakan perhitungan untuk multi kelas. Sedangkan pada Tugas Akhir ini hanya menggunakan 2 kelas. Oleh karena itu, untuk mendapatkan skor dari ulasan dapat dirumuskan sebagai berikut:

$$f(z) = W^T z + b_0 \quad (2.22)$$

Keterangan:

$z$  : vektor hasil dari *pooling layer*  
 $W$  : parameter dari SVM  
 $b_0$  : parameter bias dari SVM

$y$  merupakan kelas label dari ulasan buku. Sehingga ketika  $f(z) \geq 0$ , maka nilai  $y = +1$  dan ketika  $f(z) < 0$ , maka nilai  $y = -1$ . Berdasarkan hal tersebut dapat dituliskan sebagai berikut:

$$f(z_i) \begin{cases} \geq 0 & y = +1 \\ < 0 & y = -1 \end{cases} \quad (2.23)$$

Keterangan:

$f(z_i)$  : skor ulasan  
 $y$  : label ulasan

Sehingga dapat diketahui keberhasilan klasifikasi jika  $y_i f(z_i) > 0$ . Pada Tugas Akhir ini inialisasi awal nilai  $W$  dilakukan dengan membangkitkan nilai secara acak antara  $[-1, +1]$ .

## 2.9. CNN – L2-SVM

Algoritma CNN-L2-SVM merupakan penggabungan antara algoritma Convolutional Neural Network dan L2 Support Vector Machine (L2-SVM). Pada konstruksi modelnya, algoritma L2-SVM digunakan pada layer terakhir pada model jaringan CNN sehingga sebelum masuk ke dalam jaringan L2-SVM inputan vektor harus melalui proses dalam CNN dan selanjutnya

masuk ke dalam jaringan L2-SVM. Model CNN-L2-SVM terdiri atas 3 tahapan utama antara lain:

### **2.9.1. Tahap Umpan Maju**

Pada tahap maju ini dilakukan penginputan data yang melewati jaringan dari awal hingga akhir didapatkan skor dari ulasan buku.

#### *a. Convolution layer*

Lapisan pertama yang harus dilewati pada tahapan umpan maju merupakan convolution layer. Pada layer ini data masukan yang sudah berupa vektor-vektor kata akan diolah dengan proses konvolusi sehingga didapatkan nilai fitur dari kumpulan vektor masukan. Persamaan (2.5) digunakan untuk melakukan proses penggabungan dari vektor kata dalam satu window dan bergerak sepanjang banyaknya kata. Setelah itu persamaan (2.10) dan (2.11) digunakan untuk mendapatkan nilai fitur dari data masukan. Hasil dari layer ini merupakan sebuah vektor *feature map*.

#### *b. Pooling layer*

Lapisan kedua pada algoritma CNN-L2-SVM yaitu *pooling layer*. Pada lapisan ini akan diambil nilai maksimum dari dari *feature map* yang telah didapatkan dari convolution layer. Untuk mendapatkan nilai dari *pooling layer* digunakan persamaan (2.13).

#### *c. Output layer*

Untuk mendapatkan skor dari ulasan buku, maka digunakan persamaan (2.22).

### **2.9.2. Tahap Umpan Mundur**

Tujuan akhir pelatihan pada jaringan adalah untuk mencari gradient pada setiap filter dan parameter yang berkenaan dengan keluaran. Sehingga dapat dilakukan pembaharuan filter

secara bertahap menggunakan metode *mini-batch gradient descent*. Prinsip dari metode ini yaitu pembaharuan dilakukan setiap *mini-batch* yang ditentukan. Mini-batch merupakan jumlah data yang dibutuhkan untuk tiap pembelajaran. *Mini batch* yang digunakan pada Tugas Akhir ini yaitu 64. Misalkan  $\theta$  merupakan vektor semua parameter yang digunakan pada algoritma, maka secara matematis pembaharuan dapat dirumuskan sebagai berikut:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} L \quad (2.24)$$

$\eta$  merupakan parameter pembelajaran. Penentuan nilai  $\eta$  berdasarkan aturan *Adam*. Dengan aturan tersebut parameter pembelajaran tidak ditentukan secara langsung. Sehingga persamaan untuk pembaharuan berubah menjadi berikut:

$$\theta_{t+1} = \theta_t + \Delta\theta_t \quad (2.25)$$

Dimana

$$\Delta\theta_t = - \frac{RMS[\Delta\theta]_{t-1}}{RMS[\nabla_{\theta} L]_t} \nabla_{\theta} L$$

*RMS* merupakan *root mean square*.

Untuk mendapatkan hasil yang optimal maka perlu dilakukan pembelajaran dengan terus melakukan pembaharuan terhadap semua parameter yang ada pada Tugas Akhir ini.

#### a. *Output layer*

Pada *output layer* terdapat parameter  $W$  dan  $b_0$  dalam penentuan skor ulasan. Oleh karena itu dibentuk nilai yang optimal dengan cara mengurangi parameter sebelumnya dengan gradiennya. Berikut gradient dari parameter  $W$ . Persamaan (2.26) merupakan penurunan rumus untuk tiap

data ke- $i$  dan bergerak sampai banyaknya data dan konstanta  $C=1$  untuk mempermudah perhitungan

$$\begin{aligned}
 \frac{\partial L}{\partial W} &= \frac{\partial}{\partial W} \left[ \frac{1}{2} \|W\|^2 + \max(0, 1 - y_i f(z_i))^2 \right] \\
 &= \frac{\partial}{\partial W} \left[ \frac{1}{2} \|W\|^2 \right] + \frac{\partial}{\partial W} \left[ \max(0, 1 - y_i f(z_i))^2 \right] \\
 &= W - 2y_i \max(0, 1 - y_i f(z_i)) \frac{\partial f}{\partial W} \\
 &= W - 2y_i z_i \max(0, 1 - y_i f(z_i)) \tag{2.26}
 \end{aligned}$$

Sedangkan untuk gradien biasanya didapatkan dengan menurunkan fungsi *loss* terhadap bias  $b_0$ . Persamaan (2.27) merupakan penurunan rumus untuk tiap data ke- $i$  dan bergerak sampai banyaknya data dan konstanta  $C=1$  untuk mempermudah perhitungan.

$$\begin{aligned}
 \frac{\partial L}{\partial b_0} &= \frac{\partial}{\partial b_0} \left[ \frac{1}{2} \|W\|^2 + \max(0, 1 - y_i f(z_i))^2 \right] \\
 &= \frac{\partial}{\partial b_0} \left[ \frac{1}{2} \|W\|^2 \right] + \frac{\partial}{\partial b_0} \left[ \max(0, 1 - y_i f(z_i))^2 \right] \\
 &= -2y_i \max(0, 1 - y_i f(z_i)) \frac{\partial f}{\partial b_0} \\
 &= -2y_i \max(0, 1 - y_i f(z_i)) \tag{2.27}
 \end{aligned}$$

$\frac{\partial L}{\partial W}$  dan  $\frac{\partial L}{\partial b_0}$  digunakan selanjutnya untuk memperbaharui parameter  $W$  dan  $b_0$ .

#### b. Convolution layer

Pada *convolution layer* terdapat banyak parameter  $w$  dan  $b$ . Dikarenakan setiap *window* dikenakan parameter yang

berbeda. Misalkan  $w_k$  dan  $b_k$  merupakan parameter yang digunakan untuk mendapatkan *feature map* ke-  $k$  dan konstanta  $C = 1$  untuk mempermudah perhitungan, maka dapat diperoleh turunan gradien dari parameter tersebut sebagai berikut:

$$\begin{aligned}
 \frac{\partial L}{\partial w_k} &= \frac{\partial}{\partial w_k} \left[ \frac{1}{2} \|W\|^2 + \max(0, 1 - y_i f(z_i))^2 \right] \\
 &= \frac{\partial}{\partial w_k} \left[ \frac{1}{2} \|W\|^2 \right] + \frac{\partial}{\partial w_k} \left[ \max(0, 1 - y_i f(z_i))^2 \right] \\
 &= -2y_i \max(0, 1 - y_i f(z_i)) \frac{\partial f}{\partial w_k} \\
 &= -2y_i W_k \max(0, 1 - y_i f(z_i)) \frac{\partial z_i}{\partial w_k} \\
 &= -2y_i W_k \max(0, 1 - y_i f(z_i)) \left(1 - (z_i^k)^2\right) x_{j:j+h-1} \quad (2.28)
 \end{aligned}$$

Dimana

$$j = \operatorname{argmax}_j c_j^k$$

Sedangkan untuk parameter biasnya dapat diperoleh sebagai berikut:

$$\begin{aligned}
 \frac{\partial L}{\partial b_k} &= \frac{\partial}{\partial b_k} \left[ \frac{1}{2} \|W\|^2 + \max(0, 1 - y_i f(z_i))^2 \right] \\
 &= \frac{\partial}{\partial b_k} \left[ \frac{1}{2} \|W\|^2 \right] + \frac{\partial}{\partial b_k} \left[ \max(0, 1 - y_i f(z_i))^2 \right] \\
 &= -2y_i \max(0, 1 - y_i f(z_i)) \frac{\partial f}{\partial b_k} \\
 &= -2y_i W_k \max(0, 1 - y_i f(z_i)) \frac{\partial z_i}{\partial b_k}
 \end{aligned}$$

$$= -2y_i W_k \max(0, 1 - y_i f(z_i)) (1 - (z_i^k)^2) \quad (2.29)$$

Indeks  $k$  bergerak sampai dengan dimensi dari vektor  $z$  yang merepresentasikan banyaknya filter yang digunakan sedangkan indeks ke- $i$  bergerak sampai banyaknya data pembelajaran.

### 2.9.3. Perhitungan Nilai *Loss* (Error)

Pada Tugas Akhir ini, terdapat beberapa parameter yang digunakan untuk mendapatkan skor pada ulasan. Parameter tersebut menentukan keakurasian dalam pengklasifikasian. Oleh karena itu, dilakukan evaluasi terhadap hasil skor yang didapatkan.

Dikarenakan pada Tugas Akhir ini digunakan L2-SVM, maka perhitungan nilai *loss* atau error dapat dihasilkan dari persamaan (2.20). Berdasarkan error yang diperoleh, dapat dilihat seberapa jauh parameter atau bobot yang digunakan dapat mengklasifikasikan data masukkan kedalam kelas yang diinginkan. Nilai error tersendiri yang menentukan apakah bobot atau parameter yang digunakan harus diperbaruhi atau tidak.



## **BAB III METODOLOGI PENELITIAN**

Bab ini menjelaskan tentang tahap pengerjaan dalam menyelesaikan Tugas Akhir ini sehingga penelitian ini dapat dirancang sistematis dan diatur dengan sebaik-baiknya.

### **3.1. Pengumpulan Data**

Pada tahap awal dilakukan pengumpulan data sebagai bahan masukan untuk tugas akhir ini. Data yang dicari merupakan data ulasan suatu produk dengan tipe data berupa teks. Data akan diambil dari salah satu situs *e-commerce* dunia yaitu Amazon.com [15]. Data yang didapat mencakup daftar ulasan buku beserta skor dari ulasan tersebut. Terdapat 8 jenis buku antara lain berjudul *Gone Girl*, *The Girl on The Train*, *The Fault in our Stars*, *Fifty Shades of Grey*, *Unbroken*, *The Hunger Games*, *The Gold finch*, dan *The Martian*.

### **3.2. Praproses Data**

Selanjutnya data yang telah didapatkan terlebih dahulu dilakukan pra-proses data. Tahap ini bertujuan untuk mendapatkan bentuk data yang diinginkan sebelum masuk ketahap implementasi. Terdapat beberapa proses pada tahapan ini antara lain:

a. Tokenisasi

Tahap ini dilakukan pemebentukan *array* dari kata-kata yang ada di dalam ulasan buku. Hal ini bertujuan agar data dapat dijadikan sebagai masukan pada algoritma yang akan digunakan pada Tugas Akhir ini.

b. Filterisasi

Data ulasan buku pasti memiliki kata-kata yang tidak baku, kata henti, dan kata penghubung. Hal tersebut membuat ulasan buku sulit untuk diolah lebih lanjut. Oleh karena itu pada tahap ini dilakukan penghilangan kata-kata tersebut untuk memperingkas ulasan. Sehingga didapatkan kata-kata penting dari ulasan tersebut.

### **3.3. Ekstraksi Fitur**

Data yang sudah dilakukan tahap pra-proses data, selanjutnya dilakukan tahapan ekstraksi fitur. Ekstraksi fitur dilakukan untuk mendapatkan ciri dari suatu data atau kata. Pada tahap ini setiap kata dirubah menjadi representasi vektor yang mewakili polaritas suatu kata. Terdapat beberapa macam metode untuk menjadikan kata menjadi suatu vektor. Salah satu metode yang sedang berkembang sekarang yaitu metode *Word2vec*. Metode ini bertugas untuk merubah sebuah kata menjadi suatu vektor. Vektor yang dihasilkan merupakan konteks dari kata tersebut yang memperhatikan peluang kata yang muncul disekitarnya. Pada perkembangannya *Word2vec* telah dijadikan sebagai dasar dari penelitian NLP. *Word2vec* ini juga memiliki beberapa jenis antara lain *Bag of Word (BOW)*, *Continuous Bag of Word (CBOW)*, *Skip gram negative sampling*, atau yang terbaru yaitu *Global Vector (GloVe)*. Dengan vektor ini, dapat dapat diolah dengan mudah untuk proses selanjutnya.

### **3.4. Implementasi Algoritma CNN-L2-SVM**

Pada tahapan ini dilakukan proses klasifikasi data yang telah melalui tahap ekstraksi fitur. Output dari ekstraksi fitur nantinya berupa matriks yang berupa kumpulan vektor-vektor kata dalam satu kalimat. Matriks tersebut akan masuk ke tahapan selanjutnya. Terdapat dua tahapan utama dalam proses ini yaitu

tahap CNN dan L2-SVM. Pada tahap CNN terdiri dari dua tahapan lagi yaitu proses konvolusi dan proses *pooling*. Hasil dari CNN selanjutnya digunakan sebagai inputan pada L2-SVM. Pada tahap ini pula data dibagi menjadi dua yaitu data latih dan data uji. Data latih akan digunakan untuk proses pelatihan (*training*) untuk mendapatkan parameter yang sesuai dan optimal, sedangkan data uji digunakan untuk melihat keberhasilan dari parameter yang telah didapatkan setelah pelatihan. Data akan diklasifikasikan menjadi dua kelas utama antara lain positif dan negatif dengan menggunakan L2-SVM.

### **3.5.Evaluasi**

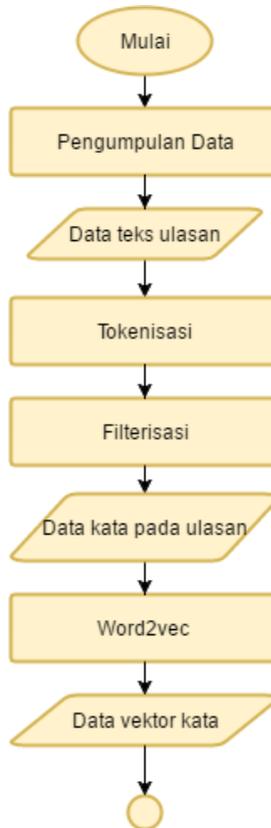
Hasil dari implementasi selanjutnya dilakukan proses pengujian. Akan dilihat perbandingan keberhasilan algoritma untuk mengklasifikasikan ulasan sesuai dengan kelasnya dengan total data uji yang dilakukan. Dari hal tersebut dapat dilihat keakurasian atau performansi dari algoritma yang digunakan terhadap data yang ada.

### **3.6.Penyusunan Laporan**

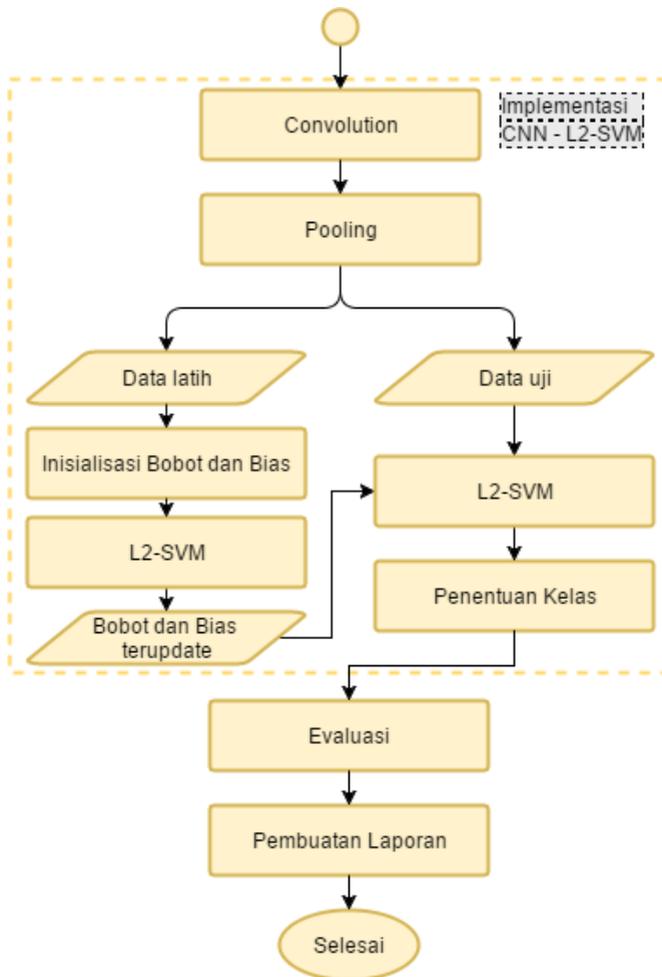
Tahap akhir setelah semua tahap dijalankan adalah penulisan laporan tugas akhir. Konten laporan tugas akhir sesuai dengan hasil yang telah didapatkan dari proses awal pengumpulan data sampai dengan penarikan kesimpulan.

### **3.7.Diagram Alir Metodologi Penelitian**

Gambaran tahap-tahap dalam penelitian pada Tugas Akhir ini disajikan sebagai diagram alir pada **Gambar 2.7** dan **Gambar 2.8**.



Gambar 2.7. Diagram alir metodologi penelitian (1)



Gambar 2.8. Diagram alir metodologi penelitian (2)



## **BAB IV**

### **PERANCANGAN DAN IMPLEMENTASI**

Bab ini menjelaskan perancangan desain sistem yang digunakan sebagai acuan untuk implementasi sistem. Perancangan sistem menggambarkan proses rancang bangun secara terperinci dari awal tahap pengumpulan data hingga pembuatan model CNN-L2-SVM. Langkah-langkah dalam pengimplementasian sistem berdasarkan desain sistem yang sudah dibentuk.

#### **4.1. Pengumpulan Data**

Dalam penerapan Tugas Akhir ini dibutuhkan data sebagai bahan masukan dari algoritma yang akan dijalankan. Data yang digunakan dalam tugas akhir kali ini menggunakan dataset dari situs Amazon.com. Dataset terdiri dari ulasan buku yang diberikan oleh pelanggan terhadap buku yang sudah dibaca sebelumnya.

Tiap masukan pada dataset dipisah dengan karakter (' $\backslash n$ '). Setiap masukan mengandung empat atribut yang dipisah menggunakan sebuah spasi (' ') antara lain:

1. Skor ulasan.
2. Url terakhir dari ulasan.
3. Judul ulasan.
4. HTML dari teks ulasan.

Pengumpulan data ulasan buku dilakukan dengan cara mencari sumber *respository dataset*. Data yang diperoleh merupakan data ulasan dari 8 jenis buku secara acak dari situs penjualan *online* dunia yaitu *www.amazon.com*. Dataset ini disimpan dalam bentuk *file csv* untuk mempermudah penyimpanan dan penggunaan. Supaya dataset dapat digunakan

secara optimal, maka dilakukan pengurangan atribut data dan menghilangkannya. Data skor dari ulasan dipisahkan dari dataset kemudian disimpan di lain dokumen, sedangkan untuk *url* dari halaman ulasan dihapus karena data yang dibutuhkan hanya ulasan dan skor dari ulasan tersebut. Selanjutnya, dikarenakan data skor yang didapatkan merupakan skor dari 1 sampai 5, maka data dirubah menjadi 2 kelas yaitu positif dan negatif. Data yang memiliki skor 1 dan 2 tergolong menjadi ulasan negative dan skor 4 dan 5 tergolong ulasan positif. Untuk skor 3 dihilangkan karena nilai tengah dari skor sulit ditentukan ulasan positif atau ulasan negatif. Hal ini juga mengantisipasi adanya kerancuan dalam mengklasifikasi ulasan. Berikut contoh potongan dari ulasan *dataset* dari Tugas Akhir ini dapat dilihat pada **Tabel 4.1**.

Tabel 4. 1. Contoh ulasan buku positif dan negatif

1	Ulasan Positif	<i>“Patrick.M-The Hunger Games is a great book. Suzanne Collins has written the best book ever to me. The Hunger Games (book 1) and Catching Fire (book 2) are full of action romance and adventure. Other authors have read this book and they loved it. ... ”</i>
2	Ulasan Negatif	<i>“Holy Cow! Triple crap! My inner goddess just kicked the stuffing out of me for finishing ""50 Shades"" of the worst book ever written; my psyche may never recover. I'm sure this must be some type of hoax perpetrated by a 10th grader with unsupervised access to the internet. ... ”</i>

Total data ulasan buku yang diperoleh yaitu 213.335 yang terdiri dari 8 jenis buku berbeda dengan pengambilan secara acak. Rincian jumlah data ulasan tiap buku dapat dilihat pada **Tabel 4.2**.

Tabel 4. 2. Rincian jumlah tiap ulasan buku

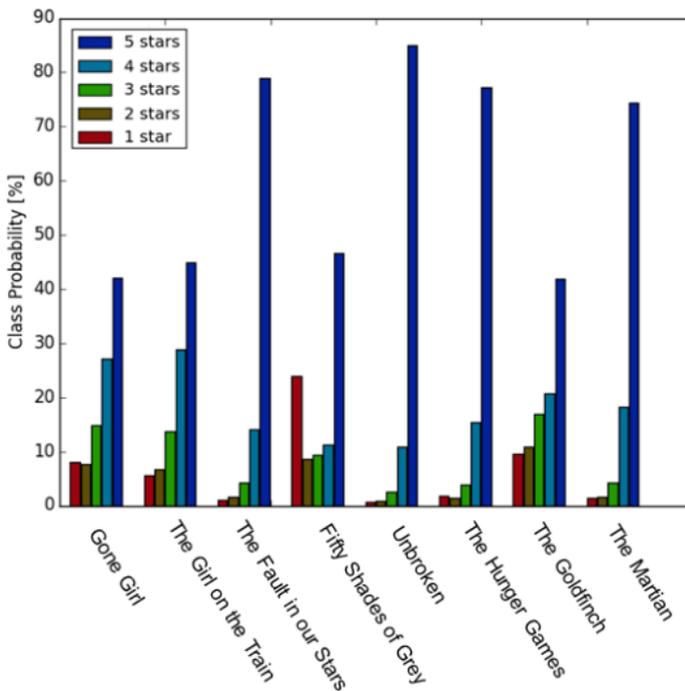
Judul Buku	Jumlah ulasan
<i>Gone Girl</i>	41.974
<i>The Girl on the Train</i>	37.139
<i>The Fault in our Stars</i>	35.844
<i>Fifty Shades of Grey</i>	32.977
<i>Unbroken</i>	25.876
<i>The Hunger Games</i>	24.027
<i>The Goldfinch</i>	22.862
<i>The Martian</i>	22.571

Untuk menghindari ketidakseimbangan data, maka data yang digunakan harus berdistribusi secara seragam agar diperoleh hasil yang maksimal. Oleh karena itu, dipilih salah satu dataset yang memiliki sebaran data yang paling baik dari 8 dataset ulasan buku tersebut.

Pada **Gambar 4.1**, dapat dilihat penyebaran data pada tiap jenis ulasan buku. Terdapat beberapa jenis data ulasan buku yang memiliki perbandingan sangat tidak seimbang. Sebagai contoh pada data ulasan buku yang berjudul “*Unbroken*”, dapat dilihat bahwa ulasan buku dengan skor 1 sangatlah sedikit jika dibandingkan dengan ulasan buku dengan skor 5. Sedangkan, data ulasan buku yang berjudul “*Gone Girl*” memiliki penyebaran yang paling baik dengan perbandingan antara data ulasan positif dan ulasan negatif relatif tidak jauh. Sehingga, pada Tugas Akhir

ini data ulasan buku tersebut yang digunakan sebagai input dari sistem yang akan dibuat.

Selain itu, data akan dibagi menjadi 2 berdasarkan prinsip *10-fold cross validation* yaitu data latih dan data uji dengan perbandingan 90% untuk data latih dan 10% untuk data uji. Tidak ada perhitungan khusus dalam penetapan proporsi penggunaan data tersebut.



Gambar 4.1. Diagram penyebaran data berdasarkan skor ulasan

Berikut contoh data yang diinputkan untuk dilakukan proses pembelajaran untuk mendapatkan model.

Tabel 4. 3. Contoh data inputan

Ulasan	Label
That book brilliant. That ending vile. Wow. Just finished the last page. As obviously brilliant and insightful as the author is about the mind of men and women...	Positif
worst ending ever written I wish I had never read this book. There is not a single likable character in it and the ending was infuriating ...	Negatif
Horrible book I was stunned by how bad this book was considering how popular it was. The two main characters are really twisted. As I kept reading, the book got worse ...	Negatif

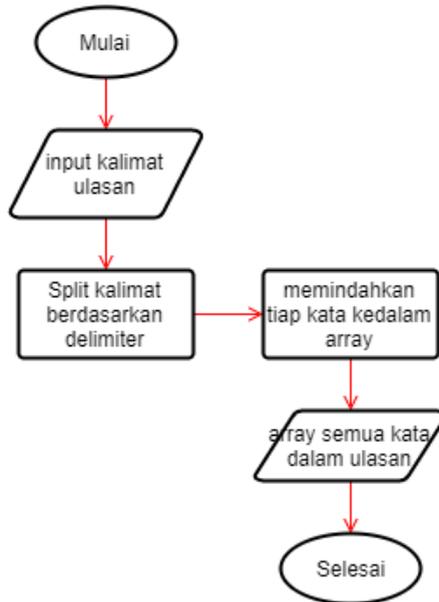
## 4.2.Praproses Data

Data ulasan buku yang didapatkan masih belum bisa digunakan secara langsung. Hal ini dikarenakan data yang didapatkan masih berbentuk suatu kalimat utuh sehingga sulit untuk bisa melakukan ekstraksi sentimen pada ulasan tersebut. Oleh sebab itu, data terlebih dahulu harus melewati tahap praproses dengan tujuan didapatkan bentuk data yang lebih mudah diolah pada proses selanjutnya. Pada Tugas Akhir ini dilakukan 2 tahap praproses data yaitu tahap tokenisasi dan filterisasi.

### a. Tokenisasi

Pada tahap tokenisasi dilakukan suatu proses pemecahan tiap kalimat ulasan menjadi bentuk kata-kata yang terpisah satu sama lain atau disebut juga dengan token. Hal ini bertujuan agar dapat diamati makna setiap kata dalam ulasan sehingga dapat ditentukan apakah ulasan tersebut termasuk ulasan yang positif

atau negatif. **Gambar 4.2** menunjukkan diagram alir proses tokenisasi pada suatu kalimat ulasan.



Gambar 4.2. Diagram alir proses filterisasi

Sedangkan pada **Gambar 4.3** merupakan gambaran hasil dari tokenisasi pada suatu kalimat ulasan.

“The Hunger Games is a great book.”



{‘The’, ‘Hunger’, ‘Games’, ‘is’, ‘a’, ‘great’, ‘book’}

Gambar 4.3. Contoh tokenisasi pada kalimat ulasan

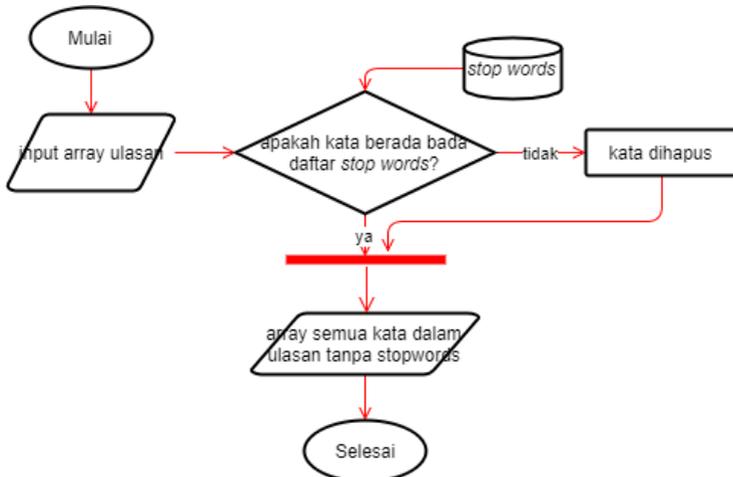
Berdasarkan proses filterisasi yang dilakukan pada setiap kalimat ulasan buku, maka kalimat ulasan pada **Tabel 4.3** dapat dihasilkan sebagai berikut:

Tabel 4. 4. Tabel array kata ulasan hasil tokenisasi

No	Array kata ulasan
1	["That" "book" "brilliant" "That" "ending" "vile" "Wow" "Just" "finished" "the" "last" "page" ...]
2	["worst" "ending" "ever" "written" "I" "wish" "I" "had" "never" "read" "this" "book" "There" ...]
3	["Horrible" "book" "I" "was" "stunned" "by" "how" "bad" "this" "book" "was" ...]

b. Filterisasi

Proses selanjutnya pada praproses data yaitu tahap filterisasi. Pada tahap ini kata-kata atau karakter yang tidak memiliki arti seperti kata penghubung, kata henti, imbuhan dan lain sebagainya dihilangkan dari ulasan. Selain itu kata-kata pada ulasan buku dilakukan proses *case folding* (perubahan huruf capital menjadi huruf kecil) agar dapat. Hal ini dilakukan agar sistem dapat memproses data dengan lebih efektif dan efisien. Diagram alir tentang proses filterisasi seperti **Gambar 4.4**.



Gambar 4.4. Diagram alir proses filterisasi

Berdasarkan proses filterisasi yang diterapkan maka ulasan pada **Tabel 4.4** maka dapat dihasilkan array ulasan kata yang sudah bebas dari *stop words*.

Tabel 4. 5. Array kata ulasan tanpa stopwords

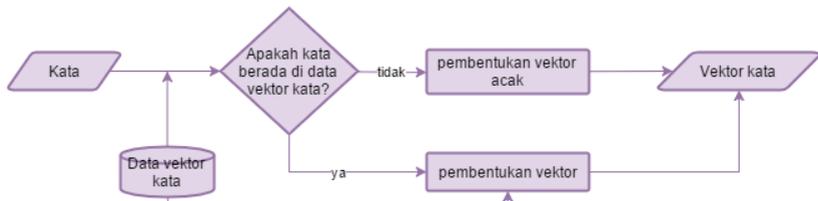
No	Array kata ulasan
1	["book" "brilliant" "ending" "vile" "Wow" "finished" "last" "page" "obviously" "brilliant" ...]
2	["worst" "ending" "ever" "written" "wish" "never" "read" "book" "not" "single" ...]
3	["Horrible" "book" "stunned" "how" "bad" "book" "two" "main" "characters" ....]

### 4.3.Ekstraksi Fitur

Setelah diperoleh array kata-kata pada tiap ulasan yang bebas dengan *stopwords*, maka dilakukan perubahan kata-kata

tersebut menjadi representasi vektor kata. Vektor yang digunakan merupakan vektor *Word2vec* berupa vektor *Glove*.

Untuk kata yang tidak termuat dalam vektor *Glove*, maka vektor kata tersebut akan dibentuk secara acak. **Gambar 4.5** menunjukkan gambaran umum mekanisme pembentukan vektor kata pada ulasan.



Gambar 4.5. Mekanisme pembentukan vektor kata

Berdasarkan proses ekstraksi fitur yang dilakukan, maka hasil dari array kata ulasan tanpa *stopwords* menjadi sebuah matriks representasi dari kalimat ulasan. Ukuran dari matriks yang terbentuk akan didapatkan sebesar  $n \times 50$  dengan  $n$  merupakan panjang kata dalam sebuah ulasan dan vektor kata yang sepanjang 50. **Tabel 4.6** menjelaskan hasil dari proses ekstraksi fitur.

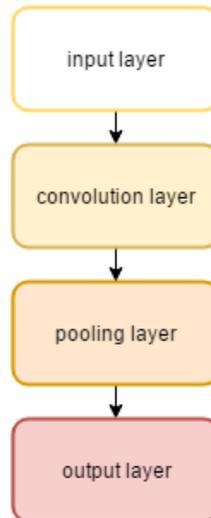
Tabel 4. 6. Matriks hasil ekstraksi fitur

No	Array kata ulasan
1	$\begin{bmatrix} -0.0077 & 0.9346 & -0.7319 & \dots \\ -0.562 & 0.2454 & -0.7003 & \dots \\ -0.1733 & 0.2825 & -0.6046 & \dots \\ -0.2862 & -1.2141 & -1.0587 & \dots \\ -0.3636 & 0.2676 & 0.5763 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$

2	$\begin{bmatrix} -0.1497 & -0.4225 & 0.1674 & \dots \\ -0.1733 & 0.2825 & -0.7003 & \dots \\ -0.0918 & 0.0458 & -0.0146 & \dots \\ -0.1027 & 0.4431 & -0.629 & \dots \\ 0.3887 & 0.6805 & -0.1175 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$
3	$\begin{bmatrix} 0.3614 & -0.5514 & -0.7005 & \dots \\ -0.0077 & 0.9346 & -0.7319 & \dots \\ 0.0806 & -0.6435 & 0.1234 & \dots \\ 0.6894 & -0.1064 & 0.1708 & \dots \\ -0.1798 & -0.404 & -0.1653 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$

#### 4.4.Desain Arsitektur CNN - L2-SVM

Pada penerapan algoritma *Deep Learning*, kedalaman arsitektur merupakan hal yang membedakan algoritma tersebut dengan yang lainnya. Hal ini juga mempengaruhi tingkat kemampuan suatu jaringan tersebut. Pada tugas akhir ini arsitektur jaringan model *Convolutional Neural Network* (CNN) terdiri atas 4 layer utama antara lain *input layer*, *convolutional layer*, *pooling layer*, dan *output layer*. Dengan inputan berupa matriks dari ulasan dan suatu vektor kelas berdimensi dua. Pada Tugas Akhir ini hanya menggunakan masing-masing satu *convolutional layer* dan *pooling layer*. Hal ini dikarenakan data yang akan diolah merupakan data teks yang tidak memiliki dimensi yang tinggi seperti data citra sehingga arsitektur ini sudah mampu mengklasifikasikan data teks ulasan dengan baik. Pada **Gambar 4.6** memperlihatkan arsitektur dari jaringan yang dibentuk pada Tugas Akhir ini.



Gambar 4.6. Arsitektur jaringan Convolutional Neural Network

Pada lapisan pertama adalah *input layer* yang merupakan lapisan masukan. Pada lapisan ini terdapat matriks dari gabungan vektor-vektor representasi kata dalam satu ulasan. Pada Tugas Akhir ini dimensi yang digunakan pada vektor kata yaitu 50, sehingga pada *input layer* matriks masukan berukuran  $k \times 50$  dengan  $k$  merupakan banyaknya kata dalam satu ulasan.

Setelah itu, matriks pada input layer masuk ke *convolution layer*. Pada lapisan ini dibentuk suatu vektor *feature map* sebanyak filter yang digunakan pada proses konvolusi. Tiap filter digunakan untuk semua *window* yang mungkin sehingga tiap *feature map* terdiri dari berbagai jenis *window*.

Hasil pada *convolution layer* selanjutnya masuk ke lapisan selanjutnya yaitu *pooling layer*. Pada lapisan ini diambil nilai *feature map* terbaik dari tiap filter yang digunakan. Sehingga didapatkan *feature* yang paling penting dari ulasan. Hasil dari *pooling layer* berupa vektor berukuran banyaknya filter yang

digunakan. Vektor tersebut yang akan digunakan untuk mendapatkan *output layer* dengan jaringan terhubung penuh. *Support Vector Machine* (SVM) digunakan untuk mendapatkan skor tiap kelas sehingga dapat diklasifikasikan dalam kategori positif atau negatif. Metode pelatihan yang digunakan pada Tugas Akhir ini yaitu *mini-batch gradient descent* dengan aturan pembaharuan *Adam* (*Adaptive Moment Estimation*).

#### 4.5.Implementasi CNN – L2-SVM

Konstruksi model jaringan mengacu pada arsitektur CNN yang sudah dibangun sebelumnya. Model berupa persamaan antara pola fitur dengan pola fitur lainnya. Berikut penjabaran model pada setiap lapisan arsitektur jaringan

##### a. *Convolution layer*

Pada layer pertama pada algoritma *convolutional neural network* dilakukan proses konvolusi untuk mendapatkan *feature map* dari inputan awal. Pada **Tabel 4.7** dijabarkan parameter-parameter yang digunakan pada Tugas Akhir ini:

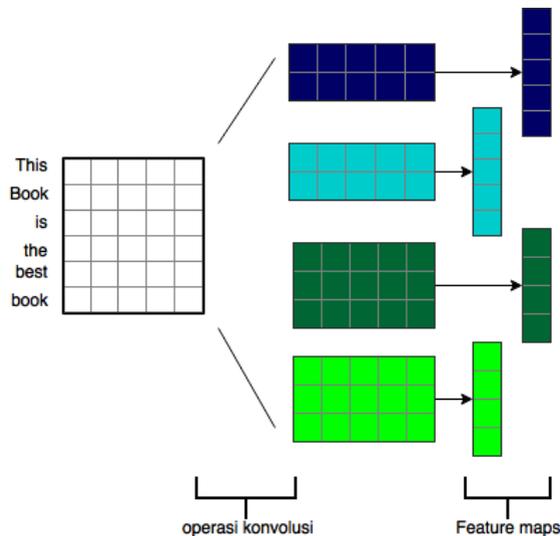
Tabel 4. 7. Hyperparameter Convolution Layer

No	Hyperparameter	Ukuran
1	Banyaknya filter ( $m$ )	10/Window
2	<i>Window</i> ( $h$ )	3, 4, dan 5

Pemilihan ukuran window dan filter tidak ada aturan khusus, akan tetapi nilai tersebut sudah umum digunakan pada penelitian sebelumnya.

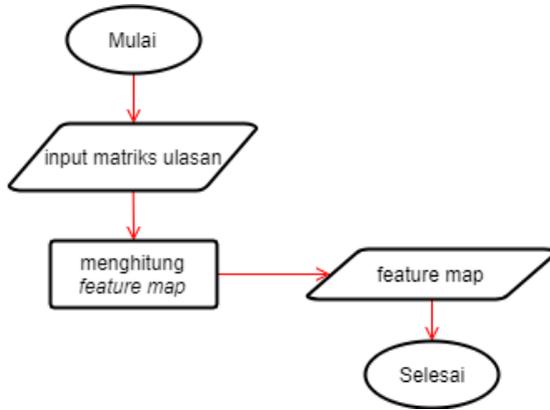
**Gambar 4.7** menjelaskan contoh mekanisme pada *convolution layer*. Pada contoh tersebut dimisalkan satu kalimat ulasan terdiri dari 6 kata dengan tiap kata

direpresentasikan sebagai vektor kata. Dimensi dari vektor kata tersebut adalah 5. Vektor-vektor kata tersebut selanjutnya dilakukan proses konvolusi dengan ukuran *window* 2 dan 3. Tiap *window* diberlakukan 2 filter untuk mendapatkan *feature maps*, sehingga terdapat 4 filter yang berbeda yang digunakan. Oleh karena itu terbentuk *feature map* sebanyak 4 dimana tiap *window* menghasilkan 2 *feature map*.



Gambar 4.7. Contoh mekanisme pada convolution layer dengan 2 window dimana terdapat 2 filter tiap windownya

Berdasarkan gambaran mekanisme dari convolution layer, maka dapat dibuat diagram alir dari convolution layer seperti pada **Gambar 4.8**.



Gambar 4.8. Diagram alir convolution layer

Misalkan  $w$  diinisialisasi secara acak sehingga diperoleh sebagai berikut:

$$w = [0.123 \quad 0.001 \quad -0.05 \quad 0.005 \quad \dots]$$

Dari filter tersebut dapat diperoleh nilai dengan persamaan (2.6) – (2.10) menjadi sebagai berikut:

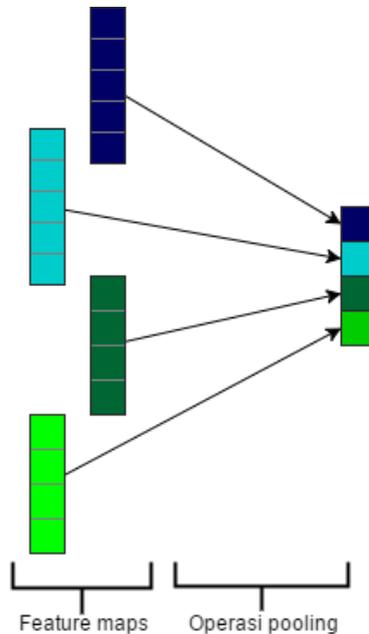
$$c = [0.1687834 \quad 0.1865951 \quad -0.8144682 \quad 0.111807 \quad \dots]$$

Setelah dikenakan fungsi linear *ReLU*, *feature map* menjadi seperti berikut:

$$c = [0.1687834 \quad 0.1865951 \quad 0 \quad 0.111807 \quad \dots]$$

#### b. Pooling layer

Selanjutnya akan diambil fitur yang paling penting dari vektor ulasan yang diinputkan. Gambaran dari *pooling layer* dapat dilihat pada **Gambar 4.9**.



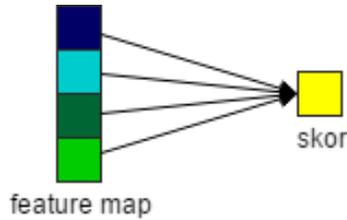
Gambar 4.9. Contoh mekanisme *pooling*

Berdasarkan *feature map* yang telah di bentuk, maka dapat diperoleh nilai maksimum dari tiap-tiap filter. Hasil setelah dilakukan proses *pooling layer* dengan persamaan (2.12) adalah sebagai berikut:

$$z = [0.98476242 \quad 0.893842 \quad 0.999934 \quad \dots ]$$

c. *Output layer*

Hasil dari *pooling* layer akan masuk ke dalam jaringan *output layer* ini dapat digambarkan seperti pada **Gambar 4.10**.



Gambar 4.10. Contoh gambaran jaringan pada output layer

Berdasarkan persamaan (2.21) maka dapat diperoleh skor dari kelas ulasan sebagai berikut dengan nilai dari parameter bias  $b_0 = 0.001$  dan  $W = [0.1 \ 0.002 \ -0.013 \ -0.0087 \ \dots]$ .

$$f(z) = W^T z + b_0 = [0.1 \ 0.002 \ -0.013 \ -0.0087 \ \dots]^T \\ [0.98476242 \ 0.893842 \ 0.999934 \ \dots] + 0.001 = 0.3789$$

Skor yang didapatkan dapat menentukan kelas dari ulasan tersebut. Karena hasil bernilai positif maka ulasan diprediksi berada di kelas positif.

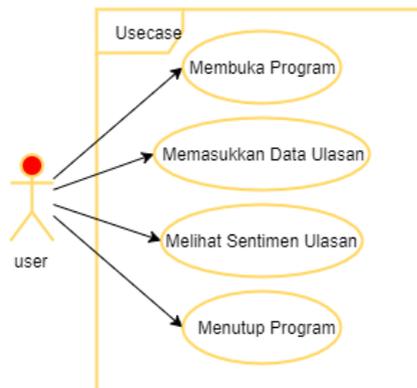
Setelah melalui lapisan-lapisan pada jaringan CNN-L2-SVM, dilakukan penghitungan error dengan persamaan (2.20) selanjutnya dilakukan tahapan umpan mundur dengan persamaan (2.26), (2.27), (2.28) dan (2.29). Setelah itu dilakukan pembaharuan bobot dan parameter dengan menggunakan persamaan (2.25).

## 4.6. Analisis Implementasi Sistem

Dalam proses implementasi suatu sistem diperlukan suatu analisis terhadap sistem sehingga sistem dapat dibuat secara optimal. Ada banyak cara dalam menganalisis suatu sistem yaitu menggunakan *usecase diagram* dan *activity diagram*.

### 4.6.1. Use Case Diagram

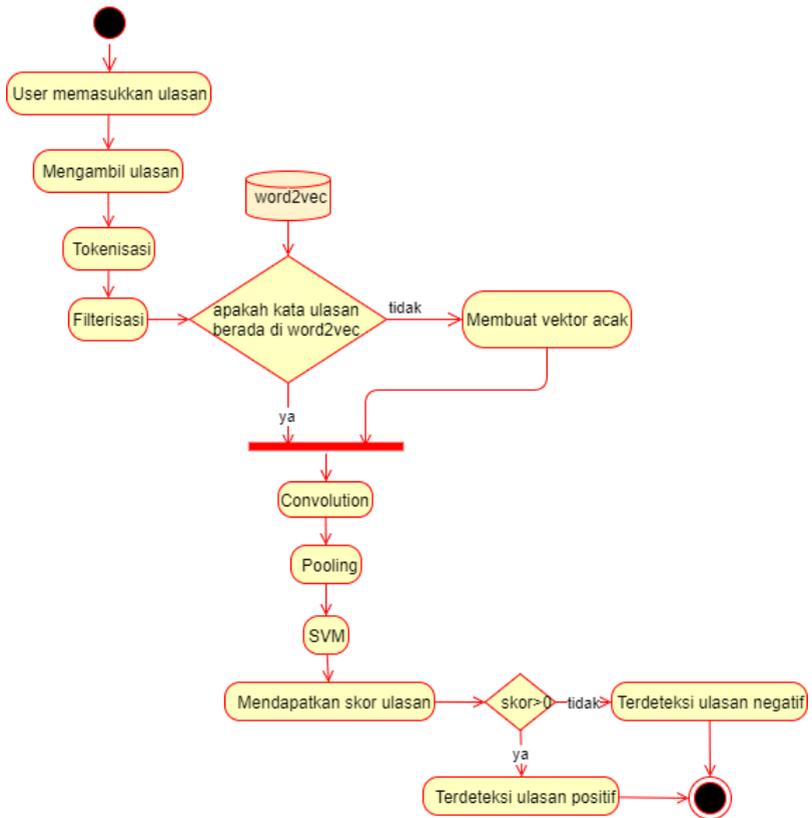
*Use case diagram* merupakan suatu diagram yang menjelaskan interaksi yang terjadi pada suatu sistem dengan aktor sistem. Adapun *use case diagram* dari sistem Tugas Akhir ini seperti pada **Gambar 4.11**.



Gambar 4.11. Use case diagram sistem

### 4.6.2. Activity Diagram

Untuk menjelaskan proses berjalannya suatu program dari awal sampai akhir, dapat digunakan activity diagram. Adapun activity diagram pada Tugas Akhir ini seperti pada **Gambar 4.12** berikut.



Gambar 4.12. Activity Diagram Sistem

#### 4.7. Lingkungan *Hardware* dan *Software*

Lingkungan perancangan sistem dibangun dari dua lingkungan *software* dan lingkungan *hardware*. Spesifikasi lingkungan perancangan sistem secara lengkap dapat dilihat di **Tabel 4.8**

Tabel 4. 8. Lingkungan Hardware

Lingkungan	Spesifikasi	
Hardware	Processor	AMD A10-5750M 2.5 GHz
	RAM	4 GB
Software	Sistem Operasi	Windows 8.1 Pro 64 bit
	Tools	Python 3.5 64-bit

#### 4.8. Library

Pada tugas akhir ini digunakan *library* yang ada pada Python sebagai pembentuk model *deep learning* yang akan dibangun. Terdapat tiga *library* utama yang digunakan pada Tugas Akhir ini, antara lain *Keras*, *Sckit-learn*, dan *Gensim*. *Library* tersebut menyediakan fungsi-fungsi untuk dapat membangun model *deep learning* maupun SVM dengan mudah.

*Library Keras* merupakan salah satu *library* yang khusus dibuat untuk mengolah data dengan menggunakan model *deep learning*. Selain itu untuk membentuk model jaringan SVM digunakan *library* bernama *Sckit-learn*. Sama halnya dengan *Keras*, *library* ini dibuat khusus untuk membentuk model jaringan syaraf tiruan dengan salah satu modelnya yaitu SVM. Untuk dapat merubah kata menjadi vektor maka digunakan *library Gensim*. *Library* tersebut menyediakan model *Word2vec* sehingga dapat digunakan dengan mudah untuk merubah kata menjadi representasi vektor kata.

#### 4.9. Implementasi Graphic User Interface (GUI)

Desain GUI dirancang sejelas mungkin untuk kemudahan dalam penggunaannya. Pada Tugas Akhir ini terdapat GUI yang digunakan dalam pengimplementasian sistem. GUI tampilan

antar muka dengan pengguna untuk bisa menggunakan sistem dengan mudah. Dengan implementasi tersebut pengguna dapat mengetahui sentiment dari suatu ulasan dengan cara memasukkan ulasan yang diinginkan. Tampilan tersebut dapat dilihat pada **Gambar 4.13**.



Gambar 4.13. Tampilan antar muka sistem

Pada gambar tersebut dapat dilihat tampilan antar muka dari sistem yang diimplementasikan. Pengguna akan mengisi ulasan pada kotak yang telah disediakan setelah itu menekan tombol untuk mengetahui sentimen dari ulasan tersebut.

#### 4.10. Algoritma

Pada Tugas Akhir ini digunakan algoritma CNN dan SVM dalam penggalian opini suatu ulasan. Algoritma tersebut dapat dijelaskan pada *pseudocode* berikut:

```

window = {h1, h2, h3, ..., hn},
z = ∅
data = {X1, X2, ..., Xm}
for each h in window:
    w ← initializeFilter(h, m)
    for X in data:
        s ← size(X)
        c = ∅
        for i in 0: s - n + 1
            x ← concatenate(Xi:i+h-1)
            temp ← nonLinear(wTx + b)
            c ← c ∪ temp
        end
        ĉ ← max(c)
        z ← z ∪ ĉ
    end
end
w ← initializeWeight(size(z))
skor ← svm(wTz + b)
if skor > 0
    return "positive"
else
    return "negative"

```

#### 4.11. Implementasi Sistem

Agar dapat dengan mudah dibentuk implementasi sistem ulasan, maka dalam prosesnya dibentuk 3 modul/kelas antara lain modul utama, modul data, dan modul *Word2vec*.

## 1. Modul data

Modul data dibentuk bertujuan untuk melakukan proses pengambilan data, pembersihan data, tokenisasi data, dan pembentukan data. Pada awalnya data ulasan buku dipisah menjadi 3 kumpulan data yaitu data positif, data negatif, dan data yang memiliki skor 3. Hal ini bertujuan untuk memudahkan sistem untuk mengolah data sehingga dapat digunakan oleh sistem dengan baik. Data disimpan di direktori dengan nama “data\_positif.pos” untuk data positif dan “data\_negatif.pos” untuk data negatif. Adapun cara untuk mengambil data dari direktori sebagai berikut:

```
# Load data dari direktori
positive_examples =
list(open("./data/data_positif.pos", encoding="utf8", errors='ignore').readlines())
positive_examples = [s.strip() for s in
positive_examples]
negative_examples =
list(open("./data/data_negatif.pos", encoding="utf8", errors='ignore').readlines())
negative_examples = [s.strip() for s in
negative_examples]
```

Hasil dari algoritma tersebut merupakan kumpulan kalimat ulasan yang akan diolah lebih lanjut.

Selanjutnya implementasi dilakukan proses tokenisasi dan filterisasi. Hal tersebut dilakukan bertujuan untuk membuat algoritma lebih efisien dalam proses perulangannya.

Bentuk implementasi dari proses tersebut dapat dibuat sebagai berikut:

```
# membersihkan data
x_text = positive_examples +
negative_examples
x_text = [clean_str(sent) for sent in
x_text]
x_text = [s.split(" ") for s in x_text]
```

Selain itu dibuat label untuk tiap data yang dapat dibentuk sebagai berikut:

```
# Membangkitkan label data
positive_labels = [[0, 1] for _ in
positive_examples]
negative_labels = [[1, 0] for _ in
negative_examples]
y = np.concatenate([positive_labels,
negative_labels], 0)
```

## 2. Modul *Word2vec*

Pada tahap ini dibuat algoritma untuk merubah kata menjadi vektor-vektor kata. Sumber basis data vektor kata yang digunakan merupakan hasil dari model *Glove (Global Vector)* dengan ukuran dimensi kata 50 tiap kata. Implementasi tahap ini dapat dibuat sebagai berikut:

```
# Inisialisasi model Word2vec
print('Training Word2vec model...')
sentences = [[vocabulary_inv[w] for w in
s] for s in sentence_matrix]
```

```
embedding_model =
    Word2vec.Word2vec(sentences,
workers=num_workers, size=num_features,
min_count=min_word_count, window=context,
sample=downsampling)
```

Untuk kata yang tidak berada pada *Word2vec* data, maka kata dibentuk secara acak. Bentuk implementasi dari tahap tersebut sebagai berikut:

```
# penambahan vektor acak
embedding_weights = {key:
embedding_model[word] if word in
embedding_model else
np.random.uniform(-0.25, 0.25,
embedding_model.vector_size)
for key, word in vocabulary_inv.items() }
```

### 3. Modul Utama

Pada modul utama, terdapat proses pembentukan model dan tahap pembelajaran dengan dilakukan validasi untuk tiap iterasinya. Iterasi yang digunakan sebanyak 20 iterasi. Pembentukan model jaringan dapat diimplementasikan sebagai berikut:

```
# membentuk model
model_input = Input(shape=input_shape)
z = Embedding(len(vocabulary_inv),
embedding_dim, input_length=sequence_length,
name="embedding")(model_input)
z = Dropout(dropout_prob[0])(z)
```

```

# Convolutional
conv_blocks = []
for sz in filter_sizes:
conv = Convolution1D(filters=num_filters,
kernel_size=sz,padding="valid",activation
="relu", strides=1)(z)
conv = MaxPooling1D(pool_size=2)(conv)
conv = Flatten()(conv)
conv_blocks.append(conv)
z = Concatenate()(conv_blocks)
if len(conv_blocks) > 1 else
conv_blocks[0]
z = Dropout(dropout_prob[1])(z)
z = Dense(hidden_dims,
activation="tanh")(z)
model_output = Dense(1,
activation="sigmoid")(z)

model = Model(model_input, model_output)
model.compile(loss="binary_crossentropy",
optimizer="adam", metrics=["accuracy"])
model_svm =
Model(model1.input, [model1.layers[11].out
put])
x_svm_test = model_svm.predict(x_train)
clf = svm.LinearSVC()

```

Selanjutnya dilakukan proses pembelajaran dengan menggunakan data yang sudah dibentuk. Implementasi tersebut dapat buat sebagai berikut:

```
# proses pembelajaran
model.fit(x_train, y_train,
batch_size=batch_size, epochs=num_epochs,
validation_data=(x_test, y_test),
verbose=2)
clf.fit(x_svm_test,y_train)
x_svm_test = model_svm.predict(x_test)
clf.score(x_svm_test,y_test)
```

## BAB V

### UJI COBA DAN EVALUASI SISTEM

Pada bab ini dijelaskan tahap-tahap uji coba berdasarkan implementasi sistem yang telah dibuat. Hasil uji coba akan dianalisis melalui proses verifikasi dan validasi sehingga dapat melakukan evaluasi pada sistem.

#### 5.1.Data Uji Coba

Setelah proses pembelajaran maka dilakukan proses pengujian. Untuk melihat validasi dari model maka pada Tugas Akhir ini digunakan metode *10-fold cross validation* untuk melihat keberhasilan model yang dibentuk. Data ulasan yang buku "*Gone Girl*" memiliki jumlah ulasan sebanyak 41.974 data. Dari data tersebut dapat dibagi menjadi tiga jenis data yaitu data positif, data negatif dan data netral. Data positif didasarkan pada skor ulasan yang bernilai 4 dan 5 sedangkan data negatif didasarkan pada skor ulasan yang bernilai 2 dan 1. Untuk data ulasan yang bernilai 3 tidak digunakan pada Tugas Akhir ini. Rincian data dapat dilihat pada **Tabel.5.1** berikut:

Tabel 5. 1. Rincian Data

Jenis Data	Jumlah Data
Data ulasan positif	27454
Data ulasan negatif	9494
Data ulasan netral	5026
Total data	41974

Untuk mempercepat proses komputasi maka data yang digunakan hanya sebanyak 10.662 data ulasan dengan rincian

5331 data ulasan positif dan 5331 data ulasan negatif. Hal ini dilakukan agar penyebaran data positif dan negatif mejadi sama rata. Selanjutnya, dari total data yang digunakan data dibagi lagi menjadi dua jenis dengan proporsi 90% untuk data latih dan 10% untuk data uji berdasarkan prinsip *10-fold cross validation*. Penjelasan rinci tentang pembagian data dapat dilihat pada **Tabel 5.2** berikut:

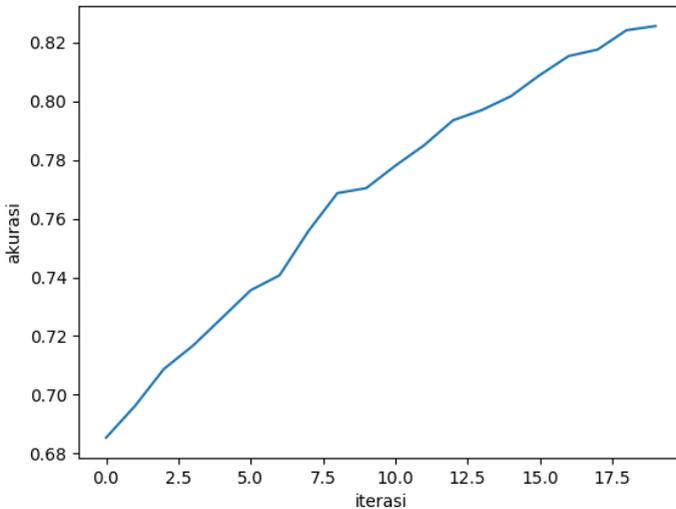
Tabel 5. 2. Data yag digunakan dalam proses komputasi

<b>Jenis Data</b>	<b>Jumlah Data</b>
Data latih	9.595
Data Uji	1.067
Total Data	10.662

## **5.2.Hasil Uji Coba Sistem**

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

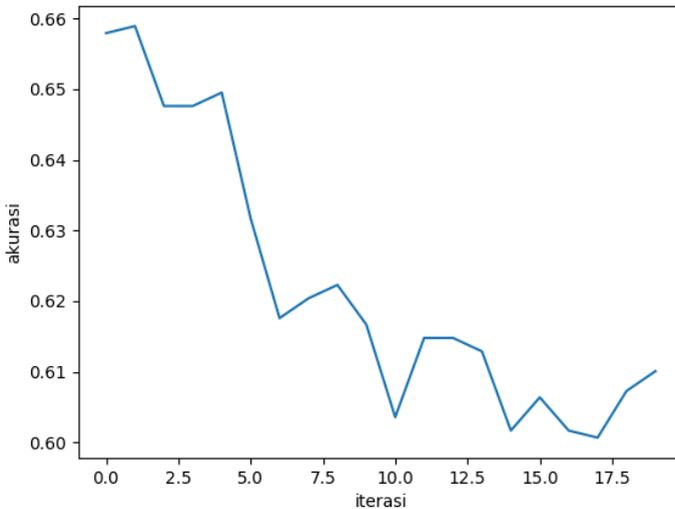
Pengujian/validasi dilakukan dengan menggunakan data uji yang telah dikelompokkan sebelumnya. Data uji dan data latih merupakan data yang saling independen sehingga dapat dilihat kemampuan model untuk menangani data yang baru. Berdasarkan hasil implementasi yang dilakukan didapatkan keakurasian tiap iterasi untuk tiap jenis data. Pada **Gambar 5.1** menunjukkan keakurasian model selama proses pembelajaran yang dilakukan sebanyak 20 kali iterasi.



Gambar 5.1. Grafik pergerakan akurasi data latih

Akurasi model yang dihasilkan pada data latih sangatlah baik karena hanya dengan 20 kali iterasi akurasi dapat mencapai 83.23% dan akan meningkat untuk iterasi yang selanjutnya. Akan tetapi untuk mendapatkan hasil tersebut membutuhkan waktu cukup banyak yaitu sekitar 76 menit.

Sedangkan akurasi model yang dihasilkan pada data uji memiliki hasil yang lebih rendah dibandingkan dengan data latih. Hal ini disebabkan data uji tidak dilakukan pengawasan tentang kebenaran dalam mengklasifikasi data. Pada **Gambar 5.2** menggambarkan akurasi dari data uji selama 20 kali iterasi.



Gambar 5.2. Grafik pergerakan akurasi pada data uji

Hasil dari pengujian terjadi perubahan akurasi yang cukup fluktuatif dibandingkan dengan akurasi saat proses pembelajaran. Akurasi model dengan data uji menghasilkan akurasi akhir sebesar 61.94%. Dapat dilihat dari grafik akurasi bahwa akurasi untuk data uji mengalami penurunan akurasi dari angka 66% hingga bertahan pada nilai 60 – 61% pada akhir iterasi.

Jika dilihat hasil saat pembelajaran dan pengujian, dapat terlihat bahwa grafik memiliki pergerakan yang berkebalikan. Hal tersebut menandakan bahwa model mengalami *overfitting*. *Overfitting* merupakan suatu kejadian dimana model terlalu baik dalam mengklasifikasi data pembelajaran akan tetapi sangat buruk dalam mengklasifikasi data uji. Permasalahan *overfitting* merupakan permasalahan yang umum terjadi pada klasifikasi menggunakan jaringan syaraf tiruan. Terdapat banyak hal yang

menyebabkan *overfitting* antara lain jumlah data yang terlalu sedikit, jumlah variabel input yang terlalu sedikit atau kualitas data yang kurang (*outlier* data, *noise*, konsistensi, korelasi, *input* dan *output*, atau sebaran data). Pada kondisi ini, hal yang menyebabkan *overfitting* yaitu kualitas data yang kurang baik. Ditemukan data ulasan positif yang berlabelkan negatif dan kebalikannya yaitu data ulasan negatif yang berlabelkan positif pada data ulasan buku. Pada **Tabel 5.3** menunjukkan salah satu data yang salah.

Tabel 5. 3. Data ulasan yang salah kelas

Label	Ulasan
Positif	Only worth of two stars Went on and on too long and ending was so flat I was expecting a more powerful ending. Enough said
Negatif	Five Stars Great book - loved the twists and turns in this one. Definitely worth the read

### 5.3. Evaluasi Model

Pada tahap ini dilakukan evaluasi terhadap model yang telah dibentuk. Model tersebut perlu dilihat performansinya dalam melakukan tugas yang diinginkan. Terdapat beberapa cara evaluasi yang dapat dilakukan dalam tugas klasifikasi pada suatu model antara lain dengan melalui akurasi, presisi (*precision*), *recall*, dan F1.

Sebelumnya hasil dari proses klasifikasi data dapat dikategorikan kedalam empat jenis data antara lain *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). *True positive* (TP) memiliki maksud bahwa data yang diklasifikasikan merupakan data positif dan telah benar

diklasifikasikan pada kelas positif. Lain halnya dengan *true negative* (TN) merupakan data yang bernilai negatif dan telah benar diklasifikasikan pada kelas negatif. Sedangkan *false positive* (FP) merupakan data yang bernilai positif akan tetapi diklasifikasikan pada kelas negatif. Selain itu *false negative* (FN) merupakan data yang bernilai negatif akan tetapi diklasifikasikan pada kelas positif.

Akurasi merupakan salah satu evaluasi pada tugas klasifikasi untuk mengetahui berapa kali data dapat diklasifikasi sesuai dengan kelas dengan benar. Akurasi mengukur ketepatan dan kemiripan hasil pada waktu yang sama dengan membandingkannya terhadap nilai absolut. Secara umum perhitungan akurasi dapat dirumuskan sebagai berikut:

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

Presisi merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Sekilas presisi dan akurasi memiliki arti yang sama. Akan tetapi kedua istilah tersebut memiliki perbedaan. Akurasi didefinisikan nilai yang menandakan seberapa dekat nilai pengukuran yang dilakukan (kuantitas) terhadap nilai yang sebenarnya. Sedangkan presisi atau yang dikenal juga dengan istilah reproduktifitas (*reproducibility*) memiliki pengertian seberapa jauh pengulangan pengukuran dalam kondisi yang tidak berubah memberikan suatu hasil yang sama. Secara umum perhitungan presisi dapat dituliskan sebagai berikut:

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

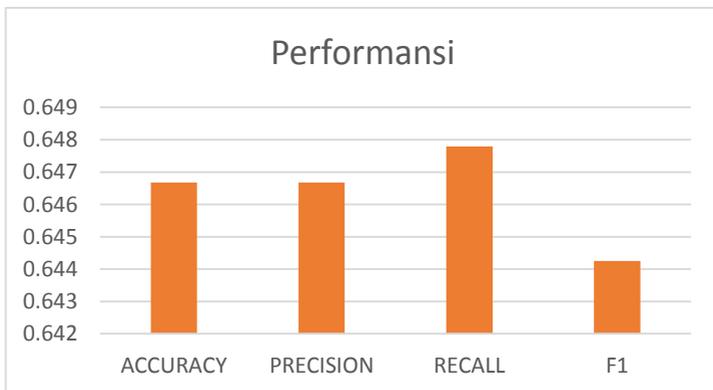
Sementara itu *recall* merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Perhitungan untuk mengetahui *recall* dapat dijabarkan sebagai berikut:

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

Sedangkan F1 dapat diinterpretasikan sebagai rata-rata dari presisi dan *recall*. Persamaan F1 dapat dituliskan sebagai berikut:

$$F1 = 2 \times \frac{(precision \times recall)}{(precision + recall)}$$

Pada Tugas Akhir ini dilakukan keempat cara evaluasi untuk mengetahui performansi dari model yang telah dihasilkan. Grafik nilai akurasi, presisi, *recall*, dan F1 dapat dilihat pada **Gambar 5.3**.



Gambar 5.3. Hasil evaluasi model

Berdasarkan grafik tersebut dapat dilihat bahwa nilai akurasi dari data uji yang dilakukan menghasilkan nilai sekitar 0.646 atau 64.6 %. Untuk nilai presisi didapatkan nilai yang sama dengan akurasi yaitu sekitar 0.646 atau 64.6%. Sedangkan untuk nilai *recall* dihasilkan nilai sekitar 0.647 atau 64.7%. Berbeda halnya dengan nilai F1 yang menghasilkan nilai sekitar 0.644 atau 64.4% yang mana nilainya lebih rendah dibandingkan dengan yang lain.

Hasil tersebut menunjukkan bahwa model memiliki performansi yang cukup baik dalam mengklasifikasikan data ulasan buku dengan tingkat akurasi, presisi, *recall*, dan F1 yang relatif sama berada pada persekitaran 64%.

#### **5.4.Perbandingan Hasil Penelitian Terdahulu**

Pada penelitian sebelumnya mengenai *opinion mining* atau analisi sentiment menghasilkan akurasi yang baik. Peneliti mencoba membandingkan metodenya dengan metode yang telah ada sebelumnya dan mencoba menggunakan beberapa jenis data.

Pada penelitian Kim [6] digunakan model *convolutional neural network* (CNN) untuk melakukan klasifikasi kalimat opini. Data yang digunakan pada penelitian tersebut beberapa macam yang diambil pada penelitian sebelumnya. Rincian hasil dari penelitian Kim dapat dilihat pada **Tabel 6.4**. Kim mencoba beberapa perlakuan untuk mengevaluasi modelnya dan hasil yang didapatkan berbeda-beda berdasarkan data yang digunakan. Seperti pada hasil data SST-1 hanya diperoleh hasil 48 % sedangkan untuk yang lain dapat mencapai lebih dari 80%.

Sementara itu pada penelitian Tang [5] digunakan metode deep learning using support vector machine. Tang mengkombinasikan metode convolutional dengan support vector machine untuk mengklasifikasikan sebuah data citra. Hasilnya

didapatkan error yang lebih kecil dibandingkan dengan tanpa menggunakan support vector machine dengan hasil sebesar 11.9% yang mana lebih kecil dibandingkan dengan metode sebelumnya sebesar 14%.

Selain itu penelitian juga telah dilakukan terhadap data ulasan buku Amazon.com oleh Taspinar [15]. Peneliti melakukan analisis sentimen menggunakan support vector machine dengan Bag-of-Word (BOW) sebagai konstruksi vektor kata. Hasilnya diperoleh akurasi sebesar 60%.

Berdasarkan hasil yang diperoleh pada penelitian sebelumnya bahwa hasil dari Tugas Akhir ini memiliki rentang yang tidak begitu jauh dari penelitian sebelumnya. Jika dibandingkan dengan penelitian Taspinar, hasil dengan algoritma pada Tugas Akhir ini lebih unggul dibandingkan dengan metode sebelumnya.

Tabel 5. 4. Hasil penelitian Kim 2014

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	<b>89.6</b>
CNN-non-static	<b>81.5</b>	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	<b>88.1</b>	93.2	92.2	<b>85.0</b>	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	-	-	-	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	-	-	-	-
RNTN (Socher et al., 2013)	-	45.7	85.4	-	-	-	-
DCNN (Kalchbrenner et al., 2014)	-	48.5	86.8	-	93.0	-	-
Paragraph-Vec (Le and Mikolov, 2014)	-	<b>48.7</b>	87.8	-	-	-	-
CCAE (Hermann and Blunsom, 2013)	77.8	-	-	-	-	-	87.2
Sent-Parser (Dong et al., 2014)	79.5	-	-	-	-	-	86.3
NBSVM (Wang and Manning, 2012)	79.4	-	-	93.2	-	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	-	-	<b>93.6</b>	-	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	-	-	93.4	-	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	-	-	<b>93.6</b>	-	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	-	-	-	-	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	-	-	-	-	-	82.7	-
SVM <sub>S</sub> (Silva et al., 2011)	-	-	-	-	<b>95.0</b>	-	-

Sehingga penggalian sentiment atau opini pada ulasan buku dengan menggunakan algoritma CNN-L2-SVM menghasilkan performa yang cukup baik. Dibutuhkan pembelajaran dan iterasi yang lebih banyak untuk dapat menghasilkan performa yang baik. Selain itu vektor representasi kata sangat membantu dalam menentukan sentimen dari ulasan. Dengan vektor kata tersebut memudahkan dalam proses penentuan ulasan dalam proses pengklasifikasian.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1. Kesimpulan**

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

1. Konstruksi model Convolutional Neural Network (CNN) dan L2 Support Vector Machine (L2-SVM) mampu menentukan sentiment atau opini dari ulasan buku.
2. Performansi yang didapatkan dari model untuk menentukan sentimen ulasan sebesar 83.23% untuk data training dan sebesar 64.4% untuk data uji.

#### **6.2. Saran**

Saran yang diberikan untuk perbaikan pada penelitian selanjutnya antara lain:

1. Menggunakan hardware komputer dengan sistem GPU. Hal ini dikarenakan sistem tersebut sangatlah sesuai untuk menjalankan deep learning seperti CNN.
2. Menambahkan metode untuk menghindari overfitting
3. Melakukan eksplorasi terhadap kedalaman jaringan dan jenis klasifikasi.
4. Menggunakan vektor representasi kata yang lebih baik.
5. Menggunakan dataset primer sendiri yang berupa bahasa Indonesia.



## Daftar Pustaka

- [1] Internet World Stats. (2016). “***Internet users in the Top 20 Countries as of June 30, 2016***”.  
<http://www.internetworldstats.com/top20.htm>. [diakses 30 November 2016]
- [2] Manzoor, Amir. (2010). “***E-Commerce: An Introduction***”. Saarbrucken: Lap Lambert
- [3] Ali, Farman.Kwak, Kyung-Sup. dan Kim, Yong-Gi. (2016). “***Opinion mining based on fuzzy domain ontology and Support Vector Machine: A proposal to automate online review classification***”. Elsevier. 47:235-250
- [4] Singh, V. dan Dubey, S. K. (2014). “***Opinion Mining and Analysis: A Literature Review***”. IEEE 5th International Conference- Confluence The Next Generation Information Technology Summit. Hal 236-237
- [5] Tang, Y. (2013). “***Deep Learning using Linear Support Vector Machines***”. International Conference on Machine Learning 2013: Challenges in Representation Learning Workshop. Atlanta USA
- [6] Kim, Y. (2014). “***Convolutional Neural Networks for Sentence Classification***”. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Hal 1746-1751
- [7] Rozi, I. F. Pramono, S. H. dan Dahlan, E. A. (2012). “***Implementasi Opinion Mining (Analisis Sentimen) untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi***”. EECCIS Vol. 6, No. 1. Hal 37-44

- [8] Pang, B. Lee, L (2008). “*Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval*”. Vol. 2, Nos. 1–2. Hal 1–135
- [9] Pang, B. Lee, L (2002). “*Thumbs up? Sentiment Classification using Machine Learning Techniques*”. Proceedings of EMNLP, pp. 79–86
- [10] Zagibalov, T. dan Carroll, J. (2008). “**Automatic Seed Word Selection for Unsupervised Sentiment Classification of Chinese Text**”. Proceedings of the 22nd International Conference on Computational Linguistics. Hal 1073-1080
- [11] Chowdhury, G. G. (2003), “*Natural language processing*”. Ann. Rev. Info. Sci. Tech., 37: 51–89. doi: 10.1002/aris.1440370103
- [12] Guo Y dkk. (2015), “*Deep learning for visual understanding: A review*”. Neurocomputing 187. Hal 27-48
- [13] Suartika, I. W. Wijaya, A. Y. dan Soelaiman, R. (2016), “*Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101*”. Jurnal Teknik ITS. Vol. 5. Hal 2301-9271
- [14] Ma, M. (2015). “*Convolutional Neural Network for Computer Vision and Natural Language Processing. Graduate Center*”. The City University of New York
- [15] Taspinar, Ahmet. (2016). “*Sentiment Analysis with bag-of-words*”. (Online). (<http://ataspinar.com/2016/01/21/-sentiment-analysis-with-bag-of-words/>, diakses 19 Juli 2017)

- [16] Mikolov, Tomas, et al. "***Distributed representations of words and phrases and their compositionality***". Advances in neural information processing systems. 2013.
- [17] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "***Glove: Global vectors for word representation.***" EMNLP. Vol. 14. 2014.



## Biodata Penulis



Penulis memiliki nama lengkap Muhammad Fakhru Rozi, lahir di Sidoarjo pada tanggal 19 Desember 1995. Penulis berasal dari Kota Sidoarjo, bertempat tinggal di Ds. Wonokasian RT.08/RW.03 Kec. Wonoayu, Kab. Sidoarjo. Pendidikan formal yang pernah ditempuh yaitu MI Roudlatul

Muttaalimin 1 Wonokasian (2002-2008), SMP Negeri 4 Sidoarjo (2008-2011), dan SMA Negeri 3 Sidoarjo (2011-2013). Kemudian penulis melanjutkan studi di jurusan Matematika ITS, dengan bidang minat ilmu komputer. Dalam bidang minat ini penulis mulai mengenal bahasa pemrograman diantaranya adalah C, Java, Python dan MATLAB. Semasa menempuh jenjang pendidikan S-1, penulis juga berorganisasi di KM ITS melalui LDJ MATEMATIKA ITS yang bernama IBNU MUQLAH sebagai Ketua Dep. Dana Usaha (2015-2016) dan aktif di Unit Kegiatan Mahasiswa Koperasi Mahasiswa Dr. Angka ITS. Selain itu, penulis juga melaksanakan Kerja Praktek di PT. Inti Konten Indonesia, Bandung dengan proyek pembuatan permainan berbasis lokasi pada tahun 2016. Selama penulisan Tugas Akhir ini Penulis tidak lepas dari kekurangan, untuk itu penulis mengharapkan kritik, saran, dan pertanyaan mengenai Tugas Akhir ini yang dapat dikirimkan melalui e-mail ke [muhammadfakhrurozi95@gmail.com](mailto:muhammadfakhrurozi95@gmail.com).