



TUGAS AKHIR – SM141501

**KLASIFIKASI ULASAN BUKU MENGGUNAKAN
ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* –
*LONG SHORT TERM MEMORY***

ANSHAR ZAMRUDILLAH ARHAM
NRP. 06111340000118

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

DEPARTEMEN MATEMATIKA
Fakultas Matematika, Komputasi dan Sains Data
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT – SM141501

BOOK REVIEWS CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK – LONG SHORT TERM MEMORY ALGORITHM

ANSHAR ZAMRUDILLAH ARHAM
NRP. 06111340000118

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

DEPARTEMENT OF MATHEMATICS
Faculty of Mathematics, Computing and Data Sciences
Institut Teknologi Sepuluh Nopember
Surabaya 2018

LEMBAR PENGESAHAN

KLASIFIKASI ULASAN BUKU MENGGUNAKAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK – LONG SHORT TERM MEMORY*

BOOK REVIEWS CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK – LONG SHORT TERM MEMORY ALGORITHM

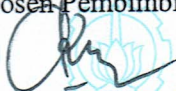
TUGAS AKHIR

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

Oleh:

ANSHAR ZAMRUDILLAH ARHAM

Menyetujui,
Dosen Pembimbing,



Dr. Imam Mukhlash, S.Si, MT.

NIP. 19700831 199403 1 003

Mengetahui
Kepala Departemen Matematika
EMKSD ITS



Dr. Imam Mukhlash, S.Si, MT

NIP. 19700831 199403 1 003

KLASIFIKASI ULASAN BUKU MENGGUNAKAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* – *LONG SHORT TERM MEMORY*

Nama Mahasiswa : Anshar Zamrudillah Arham
NRP : 06111340000118
Departemen : Matematika
Dosen Pembimbing : Dr. Imam Mukhlash, S.Si, MT.

ABSTRAK

Salah satu parameter untuk mengukur kualitas suatu produk adalah ulasan konsumen terhadap produk tersebut, apakah negatif (mengecewakan) atau positif (memuaskan). Ulasan dari konsumen berbeda dengan ulasan dari pedagang atau distributor yang mengandung unsur promosi. Ulasan konsumen cenderung bersifat jujur berdasarkan opini masing-masing. Pada zaman meluasnya perdagangan hingga ke dunia maya ini, diperlukan proses ekstraksi opini dari ulasan konsumen yang biasa disebut *opinion mining*. Proses ini dilakukan untuk mengetahui kecenderungan *reviewer* terhadap objek yang diulasnya. Salah satu model yang digunakan untuk melakukan *opinion mining* yang sedang berkembang saat ini adalah *deep learning*. Pada Tugas Akhir ini dilakukan klasifikasi opini konsumen menggunakan kombinasi algoritma *Convolutional Neural Network* (CNN) dan *Long Short Term Memory* (LSTM). Metode ini diimplementasikan untuk data ulasan buku yang diperoleh dari Amazon.com. Hasil dari metode tersebut menunjukkan performansi yang lebih baik daripada kombinasi algoritma CNN - L2-SVM.

Kata kunci— Ulasan, Opinion Mining, Natural Language Processing, Deep Learning

BOOK REVIEWS CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK – LONG SHORT TERM MEMORY ALGORITHM

Name of Student : Anshar Zamrudillah Arham
NRP : 06111340000118
Department : Mathematics
Supervisor : Dr. Imam Mukhlash, S.Si, MT.

ABSTRACT

One of parameter to measure the quality of a product is the consumer reviews of the product, whether negative (disappointing) or positive (satisfactory). The customer reviews are different with merchants or distributors reviews that contain promotional value. Consumer reviews tend to be honest based on their respective opinions. In this era of widespread trade to the virtual world, it takes the process of opinion extraction from consumer reviews which commonly called opinion mining. The goal of this process is to determine the reviewer tendency towards the reviewed object. One of the evolve models of opinion mining is deep learning. This Final Project is classifying book using a combination of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) algorithms. This method is implemented for book review data obtained from Amazon.com. The results of this method show a better performance than the combination CNN-L2-SVM algorithm.

***Keywords— Reviews, Opinion Mining, Natural Language
Processing, Deep Learning***

KATA PENGANTAR

Segala puji syukur penulis panjatkan kehadiran Alloh SWT, karena dengan ridlo-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

KLASIFIKASI ULASAN BUKU MENGGUNAKAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* – *LONG SHORT TERM MEMORY*

yang merupakan salah satu prasyarat akademis dalam menyelesaikan Program Sarjana (S1) Departemen Matematika di Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan baik berkat kerja sama, bantuan, dukungan dan doa dari banyak pihak. Sehubungan dengan hal tersebut, penulis ingin mengucapkan terimakasih kepada:

- 1.Dr. Imam Mukhlas, S.Si, MT selaku Kepala Departemen Matematika ITS, pembimbing, penguji penelitian Tugas Akhir ini dan dosen wali yang memberi arahan dan motivasi selama proses penelitian dan perkuliahan.
- 2.Kedua Orang tua penulis, yang selalu mendukung, mendoakan dan memotivasi penulis.
- 3.Dr. Didik Khusnul Arif, S.Si, M.Si selaku Kepala Program Studi S1 Departemen Matematika ITS
- 4.Drs. Iis Herisman, M.Si selaku Sekretaris Program Studi S1 Departemen Matematika ITS yang selalu memberi arahan dan pelayanan administrasi selama mengerjakan Tugas Akhir.
- 5.Seluruh jajaran dosen dan staf Departemen Matematika ITS yang telah memberikan ilmu pengetahuan kepada penulis selama diperkuliahan.
- 6.Teman-teman Angkatan 2013 yang selalu mendukung dan memotivasi. Terutama Rozi yang membantu pengerjaan Tugas Akhir ini secara langsung.
- 7.Asna, Firdo, Adit, Alwi, Yudha, Romli, Bayu yang tanpa sengaja selalu ada di sekitar penulis semasa pengerjaan Tugas Akhir.
- 8.Semua pihak yang belum 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.

Surabaya, 11 Januari 2018

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 DAN DIAGRAM	xvii
BAB I PENDAHULUAN	
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	4
BAB II TINJAUAN PUSTAKA	
2.1 Penelitian Yang Terdahulu	6
2.2 <i>Opinion Mining</i>	8
2.3 <i>Natural Language Processing</i>	9
2.4 <i>Deep Learning</i>	9
2.5 Word2vec	10
2.6 <i>Convolutional Neural Network</i>	15
2.6.1 <i>Convolution Layer</i>	17
2.6.2 <i>Pooling Layer</i>	19
2.7 <i>Recurrent Neural Network</i>	20
2.8 <i>Long Short Term Memory</i>	21
2.9 CNN-LSTM	26
BAB III METODOLOGI PENELITIAN	
3.1 Pengumpulan Data	31
3.2 Praproses Data	31
3.3 Ekstraksi Fitur	32
3.4 Implementasi Algoritma CNN-LSTM	32
3.5 Evaluasi	32

3.6 Penyusunan Lapoan.....	33
3.7 Lingkungan <i>Hardware</i> dan <i>Software</i>	33
3.8 Diagram Alir Metodologi Penelitian	33
BAB IV PERANCANGAN DAN IMPLEMENTASI	
4.1 Pengumpulan Data	35
4.2 Praproses Data.....	38
4.3 Ekstraksi Fitur	42
4.4 Arsitektur CNN-LSTM	43
4.5 Implementasi CNN-LSTM.....	45
4.6 <i>Library</i>	48
4.7 Algoritma	49
4.8 Implementasi Sistem	50
BAB V UJI COBA DAN EVALUASI SISTEM	
5.1 Data Uji Coba.....	55
5.2 Hasil Uji Coba Sistem	56
5.3 Evaluasi Model.....	59
5.4 Perbandingan Hasil Penelitian Terdahulu	62
BAB VI KESIMPULAN DAN SARAN	
6.1 Kesimpulan.....	64
6.2 Saran.....	64
DAFTAR PUSTAKA.....	65
Biodata Penulis.....	68

DAFTAR GAMBAR

Gambar 2.1 Contoh pembentukan matriks dari korpus	14
Gambar 2.2 Ilustrasi <i>context window</i>	14
Gambar 2.3 Arsitektur model <i>Skip-gram</i>	15
Gambar 2.4 Arsitektur MLP sederhana	19
Gambar 2.5 Arsitektur model CNN untuk kalimat.....	20
Gambar 2.6 Jaringan dengan 3 <i>convolution layer</i>	21
Gambar 2.7 Contoh proses konvolusi pada kalimat	22
Gambar 2.8 Arsitektur RNN.....	26
Gambar 2.9 <i>Looping</i> pada arsitektur RNN.....	26
Gambar 2.10 Modul pengulang RNN yang berisi satu <i>layer</i>	27
Gambar 2.11 Modul pengulang dalam LSTM berisi empat <i>layer</i> ..	28
Gambar 2.12 Ilustrasi <i>cell state</i>	28
Gambar 2.13 Ilustrasi langkah pertama LSTM	29
Gambar 2.14 Ilustrasi langkah kedua LSTM.....	30
Gambar 2.15 Ilustrasi langkah ketiga LSTM	30
Gambar 2.16 Ilustrasi langkah keempat LSTM.....	31
Gambar 2.8 Arsitektur RNN.....	26
Gambar 2.8 Arsitektur RNN.....	26
Gambar 3.1 Diagram alir metodologi penelitian	40
Gambar 4.1 Diagram penyebaran skor ulasan data	44
Gambar 4.2 Diagram alir proses tokenisasi	45
Gambar 4.3 Contoh proses tokenisasi pada teks	46
Gambar 4.4 Diagram alir proses filterisasi	47
Gambar 4.5 Mekanisme pembuatan vektor kata	48
Gambar 4.6 Diagram alir jaringan CNN	50
Gambar 4.7 Arsitektur jaringan CNN.....	52
Gambar 4.8 Operasi konvolusi pertama	53
Gambar 4.9 Operasi konvolusi kedua.....	55
Gambar 4.10 Operasi konvolusi ketiga	57
Gambar 4.11 Operasi konvolusi keempat.....	59
Gambar 4.12 ilustrasi jaringan pada <i>pooling layer</i>	61
Gambar 4.13 Ilustrasi jaringan pada <i>output layer</i>	62
Gambar 4.14 Diagram alir algoritma CNN-LSTM secara umum ..	53

Gambar 4.15 Diagram alir algoritma CNN-LSTM 53

Gambar 5.1 Ilustrasi pengujian model..... 73

Gambar 5.2 Grafik pergerakan akurasi proses pembelajaran 74

Gambar 5.3 Grafik pergerakan akurasi proses pengujian..... 75

Gambar 5.4 Diagram hasil evaluasi model..... 78

DAFTAR TABEL

Tabel 3.1 Lingkungan <i>hardware</i> dan <i>software</i>	27
Tabel 4.1 Contoh ulasan positif dan negatif	31
Tabel 4.2 Rincian jumlah ulasan tiap buku	67
Tabel 4.3 Tabel array kata ulasan hasil tokenisasi	72
Tabel 4.4 Array kata ulasan tanpa kata hubung atau <i>stopwrods</i>	73
Tabel 4.5 Matriks hasil ekstraksi fitur	73
Tabel 5.1 Rincian jumlah data.....	74
Tabel 5.2 Rincian data uji dan latih.....	75
Tabel 5.3 Contoh data ulasan yang salah label.....	76

BAB I

PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang yang mendasari penulisan Tugas Akhir ini. Di dalamnya mencakup identifikasi permasalahan pada topik Tugas Akhir kemudian dirumuskan menjadi permasalahan yang diberikan batasan-batasan dalam pembahasan pada Tugas Akhir ini.

1.1 Latar Belakang Masalah

Pesatnya peningkatan pengguna internet di dunia berdampak pada maraknya aktifitas jual-beli online yang juga populer disebut *e-commerce*. Pada penggunaannya terdapat banyak perkembangan dan permasalahan, jauhnya jarak antara penjual dan pembeli membuat pentingnya testimoni konsumen terdahulu untuk meyakinkan calon pembeli.

Ulasan atau testimoni konsumen pada sebuah produk adalah representasi kualitas produk tersebut. Ulasan tersebut berupa opini-opini konsumen tentang baik buruknya produk yang mereka beli, namun dalam penerapannya, sulit untuk membedakan secara digital maksud atau kecendrungan dari opini. Maka dari itu perlu dilakukannya ekstraksi terhadap ulasan tersebut. Proses untuk mengekstraksi informasi yang berguna dari ulasan pengguna disebut *Opinion mining*. Proses ini dilakukan sebagai bentuk penerapan dari *Natural Language Processing* dalam menganalisa teks [3]. Opini diolah untuk

mengetahui secara akurat emosi yang disampaikan konsumen pada ulasan tersebut, sehingga didapat informasi tentang kualitas produk yang akan digolongkan menjadi baik, netral, dan buruk.

Dewasa ini, dengan berkembangnya ilmu pengetahuan, berkembang pula metode-metode pada opinion mining. Muncul model yang lebih modern dari metode *Naïve Bayes* dan *Support Vector Machine* yang digunakan Pang dan Lee pada penelitiannya dengan objek ulasan film [4], yaitu model *Deep Learning* (DL). Salah satu metode yang merupakan model DL yaitu *Convolutional Neural Network* (CNN). Pada umumnya CNN diaplikasikan pada pengolahan citra digital untuk klasifikasi maupun klaster. Sedangkan pada penelitian yang dilakukan oleh Jin Wang bersama timnya mengusulkan kombinasi Regional CNN dan LSTM yang diuji pada *dimensional sentiment analysis* [18]. Sementara Long Chen, Yuhang He dan Lei Fan melakukan penelitian tentang pendeskripsian gambar mobil dengan menggunakan algoritma LSTM untuk klasifikasinya [17]. 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.

Maka dari itu, metode yang akan digunakan pada tugas akhir kali ini adalah kombinasi dari metode CNN dan LSTM untuk klasifikasi opini. Opini dari ulasan akan diklasifikasikan menjadi dua kelas, yaitu kelas positif dan kelas negatif. Metode CNN yang digunakan sama seperti pada penelitian Kim [6] akan tetapi fungsi aktivasi yang digunakan adalah LSTM seperti pada penelitian Long Chen [16] dan Jin Wang [17]. Pengujian metode ini akan dilakukan pada data ulasan buku yang didapatkan dari situs online Amazon.com. Performansi yang didapat dari metode ini akan dijadikan bahan pertimbangan untuk penelitian selanjutnya.

1.2 Rumusan Masalah

Rumusan masalah berdasarkan latar belakang dari Tugas Akhir ini yaitu bagaimana cara mengklasifikasikan ulasan buku menggunakan algoritma CNN - LSTM dan mengukur performansinya.

1.3 Batasan Masalah

Batasan masalah dari Tugas Akhir ini antara lain:

1. Data yang digunakan merupakan data ulasan buku di situs Amazon.com
2. Data yang digunakan adalah ulasan dari satu buku yang ulasannya tidak dominan positif atau negatif saja
3. Ulasan berupa kalimat berbahasa inggris
4. Klasifikasi tidak mempertimbangkan susunan kata dan frase kata

5. Data ulasan sudah berlabel (*rating*) dari pengguna Amazon.com

1.4 Tujuan

Adapun tujuan dari Tugas Akhir ini yaitu mengklasifikasi ulasan buku menggunakan algoritma CNN – LSTM untuk menghasilkan model atau *classifier* dari data ulasan yang telah dipelajari dan mengukur performansi model tersebut.

1.5 Manfaat

Manfaat Tugas Akhir kali ini adalah sebagai bentuk rekomendasi kepada netizen 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

Sistematika dari penulisan Tugas Akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi tentang gambaran umum dari penulisan Tugas Akhir ini yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini berisi tentang teori-teori keilmuan untuk menunjang proses penelitian. Mencakup teori-teori keilmuan dan penelitian yang pernah dilakukan.

BAB III METODOLOGI PENELITIAN

Pada bab ini dibahas tentang langkah – langkah dan metode yang digunakan untuk menyelesaikan Tugas Akhir ini.

BAB IV PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dijelaskan tentang model dan desain dari sistem yang akan dibentuk. Hal -hal tersebut meliputi visualisasi data, tahap pra-proses data, transformasi data dengan Word2vec, pembuatan model CNN dan LSTM sebagai acuan dalam implementasi sistem.

BAB V UJI COBA DAN EVALUASI SISTEM

Pada bab ini akan dibahas tentang pengujian sistem yang telah terimplementasi dengan melakukan proses verifikasi dan validasi beserta pengujian kinerja dari sistem yang dibuat.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh dari pembahasan masalah sebelumnya serta saran yang diberikan untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini menjelaskan teori-teori penunjang dalam melakukan penelitian Tugas Akhir ini. Berisi teori-teori yang menjelaskan data yang digunakan, metode untuk ekstraksi *feature*, metode-metode dalam mengklasifikasikan data dan juga bahasa pemrograman yang digunakan untuk mengolah data serta penelitian yang terkait sebelumnya.

2.1 Penelitian Terdahulu

Penelitian Jin Wang beserta timnya menyajikan model Regional CNN-LSTM untuk memprediksi peringkat VA teks dengan menangkap informasi local (regional) dalam kalimat dan ketergantungan melintasi kalimat jarak jauh, metode yang diusulkan mengungguli regresi dan metode konvensional berbasis jaringan syaraf tiruan direpresentasikan pada penelitian sebelumnya [18].

Penelitian mengenai *opinion mining* telah dilakukan sejak beberapa tahun silam. Pang dan Lee merintis dengan 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*. Performansi dari penelitian tersebut sebesar 86.4%. Penelitian *opinion mining* selanjutnya dilakukan dengan berbagai metode dan modifikasi seperti *Maximum Entropy*, *Backpropagation*, *K-means*, dengan

performansi masing-masing 85.4%, 86%, dan 78% [4]. Selain dari penelitian tersebut, berkembang pula model yang lebih modern saat ini yaitu model *Deep Learning* (DL).

Tang melakukan penelitian tentang implementasi model DL menggunakan SVM pada tahun 2013. 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 beliau melihat persamaan yang sama antara *softmax* dan L2-SVM dengan *error* 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 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 convolusi menghasilkan performansi yang baik [6].

Long Chen, Yuhang He dan Lei Fan melakukan penelitian tentang pendeskripsian gambar mobil dengan menggunakan algoritma LSTM untuk klasifikasinya. Mereka mengambil

proposal wilayah mobil dengan *Region Convolutional Convolutional Neural Networks* dan memasukkannya ke dalam vektor berukuran tetap. Setiap kata dalam sebuah kalimat juga dimasukkan ke dalam vektor bernilai real dengan ukuran gambar yang sama melalui konteks global jaringan syaraf tiruan. LSTM, memasukkan pasangan gambar-kalimat secara berurutan di tahap pelatihan, dilatih untuk memaksimalkan probabilitas gabungan kata target dalam setiap langkah waktu. Pada tahap uji coba, LSTM yang belum dilatih menerima gambar mobil dan memprediksi deskripsi kata bahasa secara alami [17].

Pada tahun 2017 Fakhur Rozi melakukan penelitian tentang penggalian opini pada ulasan buku menggunakan kombinasi algoritma CNN-L2-SVM. Rozi menggunakan algoritma *Convolutional Neural Network* (CNN) untuk ekstraksi fitur ulasan dan *L2 Support Vector Machine* (SVM) untuk klasifikasi. Hasil dari metode tersebut menunjukkan performansi pembelajaran 83.23% dan performansi pengujian 64.6% [16].

Pada Tugas Akhir ini, penulis akan melakukan pengujian algoritma CNN-LSTM dengan permasalahan dan data yang sama dengan penelitian Rozi [16].

2.2 Opinion Mining

Proses mengekstrak, memahami dan mengolah data tekstual secara otomatis untuk mendapatkan kelas sentimen pada sebuah opini adalah *opinion mining* [7]. Supaya komputer dapat

mengenal dan mengekspresikan emosi dari data teks, dilakukan penggalan frase ulasan dan ekstraksi penilaian [8]. Sebutan lain untuk *opinion mining* adalah analisis sentimen [9]. Terdapat 2 kategori umum pada analisis sentiment, yaitu:

2.2.1 *Coarse-grained sentiment analysis*

Objek yang diamati pada analisa kategori ini merupakan sebuah dokumen yang mengandung opini. Pembagian kategori pada dokumen tersebut dilakukan berdasarkan kecendrungan dari opini yang ada pada dokumen. Kategori dapat berupa positif, negatif dan netral [16].

2.2.2 *Fined-grained sentiment analysis*

Objek yang diamati pada jenis ini berupa kalimat suatu opini. Analisa ini lebih sulit dibandingkan dokumen karena informasi yang tersedia hanya terbatas pada satu kalimat. Kalimat tersebut dikategorikan menjadi positif dan negatif [16].

Sumber-sumber yang kerap digunakan untuk analisis sentimen adalah SentiWordNet dan WordNet. Analisis sentimen terdiri dari empat subproses besar [10]. Masing-masing subproses tersebut antara lain:

a. *Subjectivity Classification*

Subjectivity classification adalah proses menentukan atau membedakan kalimat yang merupakan opini dan kalimat yang bukan opini.

Contohnya kalimat “*this book has 4 chapters*” dan kalimat “*the 4th chapter of this book is touching*”.

b. Orientation Detection

Setelah klasifikasi kategori opini adalah tahap penentuan kelas positif, negatif dan netral.

c. Opinion Holder and Target Detction

Tahap ini adalah penentuan bagian yang merupakan *opinion holder* (pemberi opini) dan bagian yang merupakan *target* (objek yang diulas). Contohnya kalimat “*I really really love this book*”. Dari contoh tersebut didapat bahwa *opinion holder*-nya adalah “*I*” dan *target*-nya adalah “*this book*”.

2.3 Natural Language Processing

Penelitian atau aplikasi yang menyelidiki cara computer memahami dan memanipulasi teks bahasa alami atau pembicaraan untuk melakukan hal yang berguna. Maksud dari penelitian NLP adalah untuk mengumpulkan ilmu tentang bagaimana manusia mengerti dan menggunakan bahasa sehingga teknik dan alat yang sesuai dapat dikembangkan untuk membuat perangkat lunak yang mengerti dan dapat memanipulasi bahasa alami lalu dapat mengerjakan tugas sesuai yang diharapkan. Prinsip dari NLP juga ditemukan di beberapa disiplin ilmu antara lain ilmu komputer dan

informasi, ilmu bahasa, matematika, teknik elektro dan elektronika, kecerdasan buatan, robotika dan psikologi. Penerapan NLP termasuk dalam beberapa bidang studi seperti mesin terjemahan, pengolahan teks bahasa alami, pengolahan teks rangkuman, tampilan user, sistem pakar, pengambilan informasi kecerdasan buatan, lintas bahasa, multibahasa dan pengenalan suara [11].

2.4 Deep Learning

Machine learning memiliki sub-bidang yang disebut *deep learning*. *deep learning* adalah *machine learning* yang menggunakan metode *artificial neural networks* dalam mengerjakan *task*-nya. *Deep learning* mempelajari data dengan abstraksi yang tinggi dengan memanfaatkan arsitektur hirarki. Di penghujung tahun 2016, *deep learning* telah dipelajari secara ekstensif dalam bidang *computer vision*. Pada strukturnya *deep learning* identik dengan jumlah *hidden layer* lebih dari satu. Secara umum, metode ini dapat dibagi menjadi empat kategori antara lain [12]:

1. *Convolutional Neural Networks*
2. *Restricted Boltzmann Machines*
3. *Autocoder*
4. *Sparse Coding*

2.5 Word2vec

Ulasan yang masih berupa teks tidak dapat dikenali dan diproses secara langsung oleh komputer. Diperlukan konversi data ulasan dari bentuk teks ke bentuk numerik seperti vektor. Representasi kata dalam bentuk vektor adalah salah satu cara untuk mendapatkan nilai dari makna atau kecenderungan yang terkandung dalam suatu kalimat. Vektor representasi kata yang digunakan pada Tugas Akhir ini adalah *Word2vec* untuk mendapatkan data vektor berdasarkan data ulasan buku.

Salah satu cara untuk membuat suatu bentuk representasi kata yang terdistribusi dalam suatu ruang vektor adalah *Word2vec*. Untuk membantu algoritma pembelajaran mencapai performa yang lebih baik dalam NLP diperlukan vektor representasi kata. Algoritma untuk membuat vektor kata yang sedang berkembang saat ini adalah *Global Vectors (Glove)* [16]. *Glove* adalah suatu algoritma *unsupervised machine learning* untuk mendapatkan vektor representasi kata. Perhitungan pada algoritma ini adalah memasukkan peluang munculnya kata dalam suatu *window* atau persekitaran kata.

Asumsikan X adalah matriks dari jumlah kata yang muncul, dimana X_{ij} adalah jumlah kemunculan kata j di sekitar kata i . Misalkan $X_i = \sum_k X_{ik}$ adalah total jumlah munculnya semua kata terhadap konteks kata i . Maka dapat diambil probabilitas kata j muncul dalam konteks kata i sebagai berikut:

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

Berikut ini adalah contoh dengan menggunakan korpus “*He is not lazy. He is intelligent. He is smart*”. Pada kalimat tersebut terdapat kata yang ditunjukkan pada Gambar 2.1.

	He	is	not	lazy	intelligent	smart
He	0	4	2	1	2	1
is	4	0	1	2	2	1
not	2	1	0	1	0	0
lazy	1	2	1	0	0	0
intelligent	2	2	0	0	0	0
smart	1	1	0	0	0	0

Gambar 2.1. Contoh pembentukan matriks dari korpus

Dapat dilihat bahwa kotak merah pada Gambar 2.1 adalah jumlah kata ‘he’ dan ‘is’ yang muncul pada *context window* 2 dan dapat dilihat pada gambar di bawah bahwa jumlah kemunculan kata “he” dan “is” sebanyak 4. Sementara kata ‘lazy’ tidak pernah muncul bersama kata ‘intelligent’ pada *context window* maka dari itu muncul 0 pada kotak biru. Gambar 2.2 adalah ilustrasi dari *context window*.

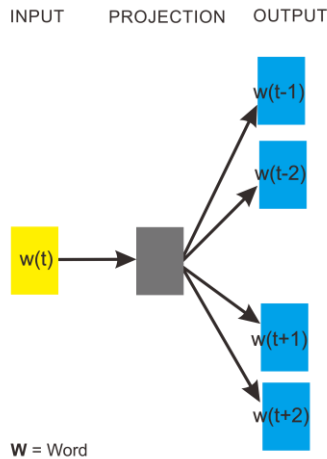
He	is	not	lazy	He	is	intelligent	He	is	smart
He	is	not	lazy	He	is	intelligent	He	is	smart
He	is	not	lazy	He	is	intelligent	He	is	smart
He	is	not	lazy	He	is	intelligent	He	is	smart

Gambar 2.2. Ilustrasi *context window*

Pada contoh korpus di Gambar 2.2, dimisalkan X_{ij} mengindikasikan jumlah munculnya kata j ="is" dalam konteks kata i ="he". Dari korpus di atas didapat $X_i = 0 + 4 + 2 + 1 + 2 + 1 = 10$ dan $X_{ij} = 4$ maka probabilitas kata "is" muncul dalam konteks kata "he" adalah sebagai berikut:

$$P_{ij} = P(j|i) = \frac{4}{10} = 0.4$$

Pada dasarnya Glove adalah bentuk umum algoritma *Skip-gram* yang merupakan salah satu algoritma representasi vektor kata terdistribusi yang telah dikembangkan sebelumnya oleh Mikolov [17]. Konsep model *Skip-gram* adalah prediksi kata di persekitaran suatu kata. Ilustrasi *Skip-gram* ditunjukkan pada Gambar 2.3.



Gambar 2.3. Arsitektur model *Skip-gram*

Menurut sejarahnya kosakata disimbolkan dengan vektor “one-hot”. Vektor ini merepresentasikan perbedaan suatu kata dengan kata lainnya hanya dengan angka 1 pada posisi tertentu dan 0 pada posisi lain.

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

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

Namun kelemahan representasi vektor “one-hot” adalah tidak dapat ditemukannya vektor yang merepresentasikan relasi antar kata dalam kosa kata yang ada. Maka dari itu model *Skip-gram* membuat representasi vektor yang mendekati vektor tersebut. Model ini memaksimalkan probabilitas tiap representasi vektor terhadap vektor “one-hot”. Pada model ini, vektor kata dibagi menjadi dua macam, yaitu vektor kata dan vektor konteks kata yang disimbolkan sebagai v dan u . Fungsi objektif dari model *Skip-gram* dapat dirumuskan 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}

Penjabaran dari peluang tersebut adalah sebagai fungsi *softmax* berikut:

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

Dimana W adalah jumlah kosakata.

Optimalisasi peluang logaritmik dari vektor kata dengan konteks kata tersebut adalah kunci dari model *Skip-gram*. Maka dari itu proses pelatihan harus dilakukan dengan jumlah data yang sangat besar untuk mencapai titik optimumnya. Sehingga pada proses NLP selanjutnya dapat diperoleh vektor representasi semua kata dalam kosakata yang ada dan dapat digunakan.

Pada sisi lain, *Glove* mencoba mengembangkan model *Skip-gram* dengan menambahkan salah satu nilai yaitu peluang kemunculan suatu kata. Hal tersebut 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 dari *Glove* dapat dirumuskan 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
θ	: semua parameter yang digunakan
X	: jumlah kata yang muncul
u	: vektor konteks kata
v	: vektor kata

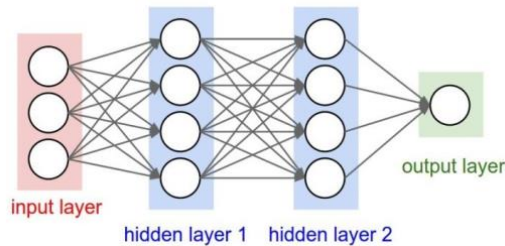
Digunakan data kata dari Wikipedia pada tahun 2014 dan Gigaword 5 sebagai data latih untuk membentuk vektor representasi kata. Terdapat 6 miliar kata yang digunakan dengan kosa kata sebanyak 400.000 kata [17]. Vektor representasi yang dihasilkan memiliki beberapa dimensi antara lain 50, 100, 200, dan 300. Agar mempermudah dalam waktu komputasi, Tugas Akhir ini menggunakan representasi vektor kata dengan dimensi 50 dengan alasan untuk meringankan proses komputasi. Peran dari vektor tersebut adalah sebagai referensi dalam pembentukan vektor representasi sebelum dilakukan tahap klasifikasi.

2.6 Convolutional Neural Network

Pengembangan dari *Multilayer Perceptron* (MLP) yang digunakan untuk mengolah data dua dimensi adalah ide awal dari *Convolutional Neural Network* (CNN). Karena kedalaman jaringan dan memiliki *layer* atau lapisan yang banyak maka CNN termasuk ke dalam jenis *deep neural network*.

CNN memiliki cara kerja yang sama dengan MLP, yang membedakannya adalah setiap neuron pada CNN dilewati dalam

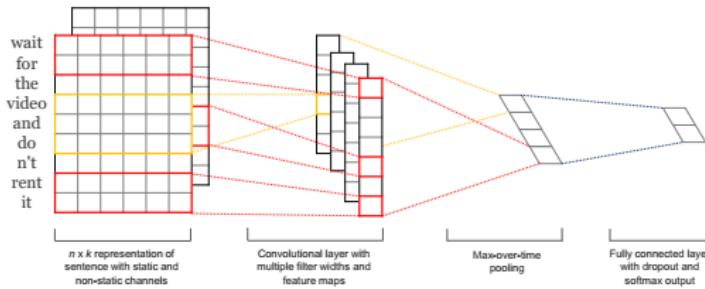
bentuk dua dimensi dan pada MLP ukuran setiap neuronnya hanya satu dimensi. Pada Gambar 2.4. ditunjukkan arsitektur MLP.



Gambar 2.4. Arsitektur MLP sederhana

MLP memiliki i layer yang pada Gambar 2.4 digambarkan dengan kotak merah dan biru dengan setiap layer berisi j_i neuron yang diilustrasikan dengan lingkaran putih. Input data berbentuk satu dimensi yang diterima oleh MLP dan dipropagasikan pada jaringan sebelum MLP menghasilkan output. Setiap hubungan antar neuron pada dua layer yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Di setiap data input pada layer dilakukan operasi linear dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi *non linear* yang disebut sebagai fungsi aktivasi [13].

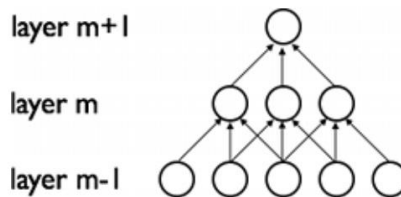
Sekarang CNN dapat diterapkan pada dua hal yaitu pada *computer vision* dan NLP. Gambar 2.5 adalah arsitektur model dari CNN pada NLP [6]. Bagaimanapun juga terdapat dua komponen utama dari CNN yaitu *convolution layer* dan *pooling layer*.



Gambar 2.5. Arsitektur model CNN untuk kalimat

2.6.1 Convolution layer

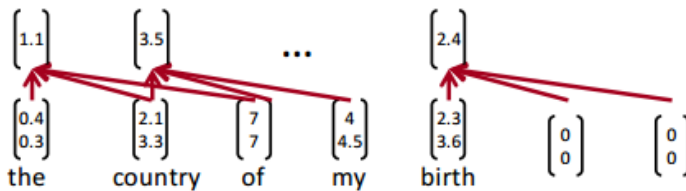
Filter konvolusi menggunakan korelasi lokal dengan memaksakan pola konektivitas lokal di antara *layer* yang berdekatan pada setiap *convolution layer*. *Layer* yang lebih atas, m didapat dari sebuah subhimpunan satuan dari *layer* yang lebih rendah, $m - 1$. Keuntungan dari *convolutional layer* dibandingkan dengan MLP adalah jumlah parameter yang berkurang secara signifikan disebabkan oleh parameter bagi. Konektivitas tersebut digambarkan dalam Gambar 2.6. [14].



Gambar 2.6. Jaringan dengan 3 convolution layer

Proses ini adalah *layer* pertama pada arsitektur jaringan CNN-LSTM. Sesuai dengan namanya, pada *layer* ini dilakukan proses konvolusi untuk semua vektor kata yang ada pada ulasan.

Pada Gambar 2.7 akan ditunjukkan contoh proses konvolusi pada suatu kalimat dari penelitian terdahulu, dimana untuk vektor setelah kata terakhir yang masih berada pada *window* bernilai nol. Namun pada Tugas Akhir ini, konvolusi berakhir hingga vektor kata ke- $n - h + 1$.



Gambar 2.7. Contoh proses konvolusi pada kalimat

Pada dasarnya proses konvolusi adalah proses mengoperasikan setiap input dengan setiap bobot menggunakan operasi dot produk lalu menjumlahkan hasil operasi tersebut dengan nilai bias pada masing-masing bobot. Hasil dari operasi dot produk dengan nilai bobot dan penjumlahan dengan nilai bias akan diproses kedalam *activation function*.

Terlihat pada Gambar 2.7 bahwa contoh tersebut menggunakan matriks nilai bobot berukuran 2×3 dan *stride* sebanyak 1. Ukuran matriks nilai bobot ditunjukkan dengan bersatunya 3 vektor input yang masing-masing vektornya memiliki 2 elemen. Pada Gambar 2.7 terlihat juga bahwa pergeseran operasi konvolusi sebanyak 1 vektor, hal tersebut menunjukkan jumlah *stride*. Pembahasan contoh operasi konvolusi yang lebih detail akan dibahas pada subbab 4.5 Implementasi CNN-LSTM.

Setelah beberapa vektor kata digabung, ulasan direpresentasikan seperti pada persamaan (2.5)

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

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 antara beberapa vektor kata dari indeks ke- i sampai ke- $i + j$ seperti berikut:

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

Filter (bobot) pada operasi konvilusi dibutuhkan untuk mendapatkan nilai *feature map*, berikut adalah hubungan antara *feature map* dan *filter*:

$$net = w \cdot x_{i:i+h-1} + b \quad (2.7)$$

$$c_i = f(net) \quad (2.8)$$

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

Activation function atau fungsi non linear yang digambarkan sebagai $f(net)$ pada persamaan (2.7) pada Tugas Akhir ini adalah

fungsi *Rectified Linear Unit (ReLU)*. Hasil dari fungsi tersebut memiliki batasan output bernilai positif. Berikut adalah rumus dari *Rectified Linear Unit (ReLU) function*:

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

Jadi, persamaan *feature map* menjadi seperti berikut:

$$net = w \cdot x_{i:i+h-1} + b \quad (2.10)$$

$$c_i = ReLu(net) \quad (2.11)$$

Untuk setiap *window* kata yang ada dalam kalimat ulasan diterapkan *filter* w . Kata-kata pada kalimat ulasan digambarkan sebagai $\{x_{1:h}, x_{2:h}, x_{3:h}, \dots, x_{n-h+1:n}\}$ sehingga didapatkan *feature map* sebagai berikut:

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

c adalah *feature map*.

Pada Tugas Akhir ini akan digunakan lebih dari satu filter dan *window* kata untuk mendapatkan hasil yang maksimal.

2.6.2 Pooling layer

Pooling layer adalah komponen penting kedua pada CNN setelah *convolution layer*. *Pooling layer* memiliki fungsi sebagai *down-sampling* yang bersifat *non-linear*. *Pooling layer* bertujuan untuk menyeleksi informasi yang penting dari *feature map*. Seleksi tersebut dilakukan dengan mengeliminasi nilai yang kurang maksimal untuk mengurangi proses komputasi. Selain itu *pooling* juga dilakukan untuk menyediakan bentuk translasi invarian.

Pooling juga mengatasi masalah yang ditimbulkan dari perbedaan panjang *feature map* sehingga didapatkan dimensi yang sama.

Operasi *pooling* dilakukan setelah didapatkan *feature map*. *Feature map* yang telah didapat dari *convolution layer* akan diproses di *pooling layer*. Inti dari lapisan ini adalah pengambilan nilai-nilai penting dari setiap *feature map* dengan mengambil nilai yang maksimum. Terdapat beberapa metode untuk proses *pooling* yaitu *max pooling*, *average pooling*, dan lain-lain. Pada Tugas Akhir ini metode yang digunakan adalah metode *pooling* yang sering digunakan untuk *task NLP* yaitu *max pooling*. Operasi *pooling* dirumuskan sebagai berikut:

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

Hasil dari *pooling layer* merupakan suatu vektor yang terdiri atas nilai maksimum tiap *feature map* dan vektor tersebut memiliki jumlah elemen sebanyak m . Hal ini dikarenakan banyaknya *filter* berjumlah m pula. Vektor tersebut digambarkan seperti persamaan (2.14).

$$z = [\hat{c}1, \hat{c}2, \hat{c}3, \dots, \hat{c}m] \quad (2.14)$$

2.6.3 Output layer

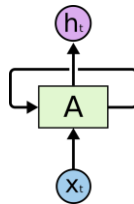
Peran *output layer* adalah klasifikasi ulasan yang telah diekstrak fiturnya pada *layer-layer* sebelumnya menggunakan jaringan syaraf tiruan dengan kelas-kelas tujuan yang ditentukan antara kelas positif dan negatif. Vektor *feature* yang merupakan output dari *pooling layer* akan melewati hubungan penuh jaringan *Long Short Term Memory (LSTM)* untuk mendapatkan skor

ulasan. Label untuk ulasan positif dan ulasan negatif pada tugas akhir ini adalah $+1$ dan -1 .

2.7 Recurrent Neural Network

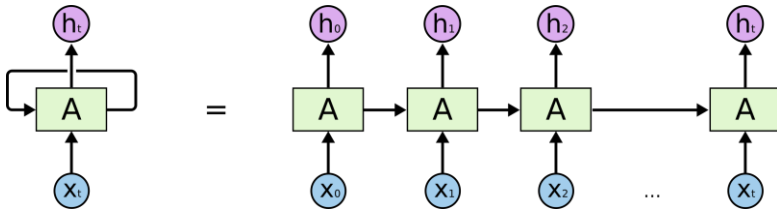
Dasar dari pengembangan *Recurrent Neural Network* (RNN) adalah cara berfikir manusia pada umumnya yang tidak setiap saat membuat keputusan secara tunggal. Manusia cenderung menimbang atau memperhitungkan masa lalu dalam pengambilan keputusan. Seperti analogi tersebut, RNN tidak membuang begitu saja informasi dari masa lalu. Hal itulah yang membuat RNN berbeda dari *Artificial Neural Network* biasa.

RNN adalah salah satu *neural network* yang diperuntukkan memproses data bersambung (*sequential data*). RNN dapat menyimpan memori terdahulu karena proses *looping* pada arsitekturnya.



Gambar 2.8. Arsitektur RNN

Gambar 2.8 menunjukkan arsitektur dari algoritma RNN yang berulang. Pada Gambar 2.9 akan ditunjukkan arsitektur RNN yang mengilustrasikan *looping* pada arsitektur RNN.

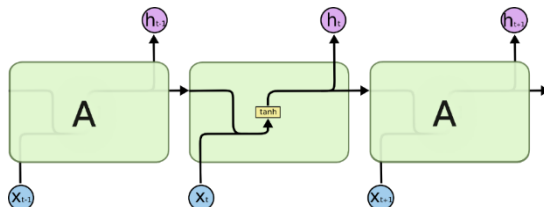


Gambar 2.9. *Looping* pada arsitektur RNN

2.8 Long Short Term Memory

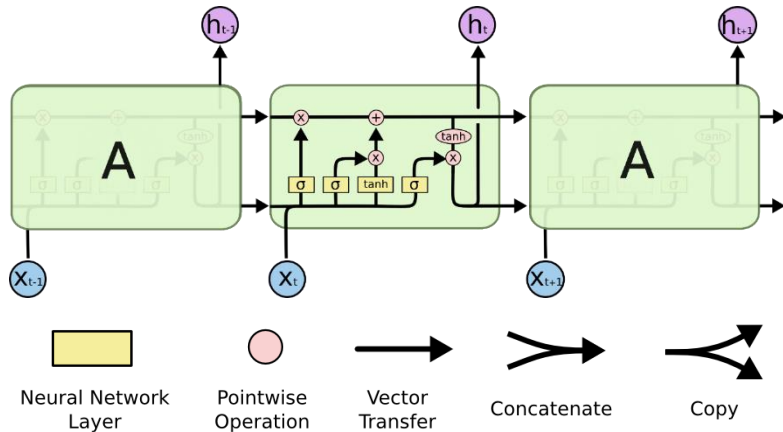
Long Short Term Memory (LSTM) adalah sebuah jenis khusus dari algoritma *Recurrent Neural Network* (RNN) yang mampu mempelajari ketergantungan jangka panjang. LSTM cocok digunakan untuk masalah yang memiliki ketergantungan jangka panjang. Mengingat informasi jangka panjang adalah perilaku bawaan LSTM. LSTM juga memiliki struktur rantai seperti struktur RNN, perbedaannya terletak pada struktur modul pengulangannya.

Seluruh *Recurrent Neural Network* memiliki bentuk rantai modul pengulangan jaringan syaraf tiruan. Dalam RNN standar, modul pengulangan ini akan memiliki struktur yang sangat sederhana, seperti lapisan *tanh* tunggal.



Gambar 2.10. Modul pengulang RNN yang berisi satu *layer*

LSTM juga memiliki struktur seperti rantai, namun modul pengulangan memiliki struktur yang berbeda. Sebagai gantinya, LSTM memiliki empat lapisan jaringan syaraf tunggal yang berinteraksi dengan cara yang berbeda dengan RNN.

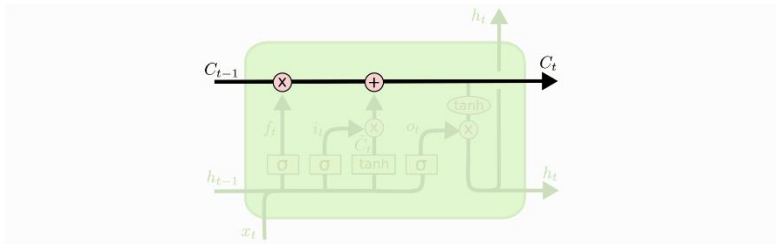


Gambar 2.11. Modul pengulang dalam LSTM berisi empat *layer*

Pada Gambar 2.11 setiap baris membawa keseluruhan vektor, dari output satu simpul ke input lain. Lingkaran merah muda mewakili operasi *pointwise*, seperti penambahan vektor, sedangkan kotak kuning adalah lapisan jaringan syaraf tiruan yang dipelajari. Penggabungan garis menunjukkan rangkaian (*concatenation*), sementara garis bercabang menunjukkan kontennya disalin dan salinannya masuk ke lokasi yang berbeda [15].

Kunci pada algoritma LSTM adalah *cell state* atau garis horizontal yang melewati bagian atas diagram. *Cell state* bekerja

seperti *conveyor belt*. Dia berjalan ke seluruh rantai hanya dengan sedikit interaksi linear. Sangat mudah bagi informasi untuk mengalir begitu saja tanpa perubahan.



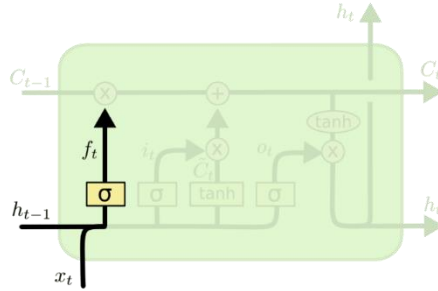
Gambar 2.12. Ilustrasi *cell state*

LSTM memiliki kemampuan untuk menambah dan menghapus informasi ke *cell state*, diatur secara teliti oleh struktur yang disebut *gates*. *Gates* adalah cara untuk melepas informasi yang lewat. Mereka terdiri dari lapisan jaringan saraf sigmoid dan operasi perkalian *pointwise*.

Berikut adalah langkah-langkah pada LSTM:

1. Menentukan informasi yang akan dibuang dari *cell state*

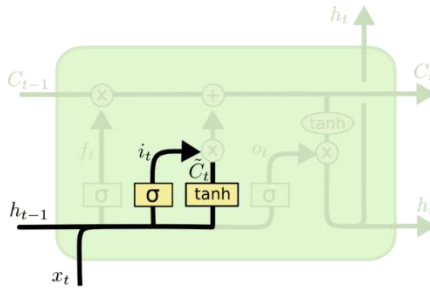
Inti dari tahap ini terletak pada *layer* sigmoid atau yang juga disebut *forget gate*. Sesuai dengan namanya, *layer* ini berperan untuk menyeleksi informasi yang akan dilupakan. *Forget gate* menghasilkan angka 0 dan 1 untuk *cell state* C_{t-1} . Angka 1 merepresentasikan untuk menjaga memori dan 0 representasi untuk melupakan memori.



Gambar 2.13. Ilustrasi langkah pertama LSTM

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.15)$$

2. Menentukan informasi yang akan dimasukkan ke *cell state*



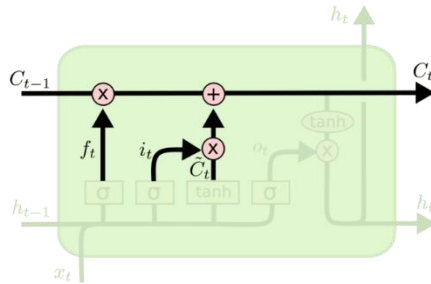
Gambar 2.14. Ilustrasi langkah kedua LSTM

Pada tahap ini terdapat dua *layer* yaitu *layer* sigmoid dan *layer tanh*. *Layer* sigmoid dinamakan *input layer* yang memutuskan nilai mana yang akan diperbarui. Selanjutnya *layer tanh* berperan untuk membentuk vektor dari nilai kandidat baru \tilde{C}_t yang dapat ditambahkan ke *cell state*. Lalu *cell state* yang lama, C_{t-1} di-update menjadi *cell state baru*, C_t .

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.16)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.17)$$

3. Menambahkan informasi baru

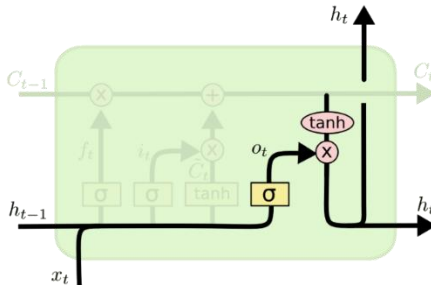


Gambar 2.15. Ilustrasi langkah ketiga LSTM

Selanjutnya akan kita eksekusi apa yang sudah diputuskan pada tahap sebelumnya. Persamaan (14) adalah cara meng-*update* informasi atau memori baru:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.18)$$

4. Menemukan output



Gambar 2.16. Ilustrasi langkah keempat LSTM

Akan dijalankan *layer* sigmoid yang menentukan bagian dari *cell state* mana yang akan dijadikan output. Lalu masukkan *cell state* melewati *tanh* (untuk memaksa nilainya menjadi antara

-1 dan 1) dan mengalikannya dengan hasil dari *layer* sigmoid agar kita hanya menghasilkan bagian yang telah ditentukan.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.19)$$

$$h_t = o_t * \tanh(C_t) \quad (2.20)$$

2.9 CNN-LSTM

CNN-LSTM adalah kombinasi antara algoritma *Convolutional Neural Network* (CNN) dan algoritma *Long Short Term Memory* (LSTM). Sesuai dengan urutan namanya, konstruksi model CNN-LSTM meletakkan algoritma LSTM pada layer terakhir di model jaringan CNN sehingga sebelum memasuki jaringan LSTM vektor input harus melalui proses algoritma CNN setelah itu jaringan LSTM. Tahapan pada model CNN-LSTM terdiri atas 3 tahapan:

1. Tahap Umpan Maju

Tahap umpan maju adalah proses masuknya input data yang melewati jaringan dari awal hingga 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-LSTM yaitu *pooling layer*. Pada lapisan ini akan diambil nilai maksimum 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 (2.20).

2. Tahap Umpan Mundur

Tahap umpan mundur adalah proses pembelajaran atau pelatihan bagi model. Sesuai dengan namanya, proses ini bergerak mundur dari layer output ke layer input. Hal ini bertujuan untuk mengevaluasi error yang dihasilkan. Tujuan akhir pelatihan pada jaringan adalah untuk mencari gradien pada setiap filter dan parameter yang berkenaan dengan keluaran. Sehingga filter dapat diperbarui 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 θ merupakan vektor semua parameter yang digunakan pada algoritma, maka pembaharuan dirumuskan sebagai berikut:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} L \quad (2.21)$$

η adalah parameter pembelajaran. Untuk menentukan nilai η dilakukan berdasarkan aturan *Adam*. Berdasarkan aturan *Adam* persamaan untuk pembaharuan parameter pembelajaran berubah menjadi berikut:

$$\theta_{t+1} = \theta_t + \Delta\theta_t \quad (2.22)$$

Dimana

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

RMS adalah *root mean square*.

Perlu dilakukan pembelajaran dengan terus melakukan pembaharuan terhadap semua parameter yang ada pada Tugas Akhir ini untuk mendapatkan hasil yang optimal.

a. *Output Layer*

Terdapat parameter W dan b_0 pada *output layer* untuk menentukan skor data ulasan. Maka dari itu dibentuk nilai optimal dengan cara mengurangi parameter sebelumnya dengan gradiennya. Persamaan (2.23) merupakan penurunan rumus untuk tiap data ke- i dan bergerak sampai banyaknya data dan konstanta $C=1$ untuk mempermudah perhitungan. Untuk mendapatkan gradien dari parameter W adalah sebagai berikut:

$$\begin{aligned}
\frac{\partial L}{\partial W} &= \frac{\partial}{\partial W} \left[\frac{1}{2} \|W\|^2 + \max(0, 1 - y_i f(z_i))^2 \right] \\
&= \frac{\partial}{\partial W} \left[\frac{1}{2} \|W\|^2 \right] + \frac{\partial}{\partial W} [\max(0, 1 - y_i f(z_i))^2] \\
&= W - 2y_i \max(0, 1 - y_i f(z_i)) \frac{\partial f}{\partial W} \\
&= W - 2y_i z_i \max(0, 1 - y_i f(z_i)) \quad (2.23)
\end{aligned}$$

Gradien bias didapatkan dengan menurunkan fungsi *loss* terhadap bias b_0 . Persamaan (2.24) merupakan penurunan rumus untuk tiap data ke- i dan bergerak sampai banyaknya data dan konstanta $C=1$ untuk mempermudah perhitungan.

$$\begin{aligned}
\frac{\partial L}{\partial b_0} &= \frac{\partial}{\partial b_0} \left[\frac{1}{2} \|W\|^2 + \max(0, 1 - y_i f(z_i))^2 \right] \\
&= \frac{\partial}{\partial b_0} \left[\frac{1}{2} \|W\|^2 \right] + \frac{\partial}{\partial b_0} [\max(0, 1 - y_i f(z_i))^2] \\
&= -2y_i \max(0, 1 - y_i f(z_i)) \frac{\partial f}{\partial b_0} \\
&= -2y_i \max(0, 1 - y_i f(z_i)) \quad (2.24)
\end{aligned}$$

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

b. Convolution Layer

Terdapat banyak parameter w dan b pada *convolution layer* karena setiap *window* dioperasikan parameter yang berbeda. Misal 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:

$$\begin{aligned}
\frac{\partial L}{\partial w_k} &= \frac{\partial}{\partial w_k} \left[\frac{1}{2} \|W\|^2 + \max(0, 1 - y_i f(z_i))^2 \right] \\
&= \frac{\partial}{\partial w_k} \left[\frac{1}{2} \|W\|^2 \right] + \frac{\partial}{\partial w_k} \left[\max(0, 1 - y_i f(z_i))^2 \right] \\
&= -2y_i \max(0, 1 - y_i f(z_i)) \frac{\partial f}{\partial w_k} \\
&= -2y_i W_k \max(0, 1 - y_i f(z_i)) \frac{\partial z_i}{\partial w_k} \\
&= -2y_i W_k \max(0, 1 - y_i f(z_i)) (1 - (z_i^k)^2) x_{j:j+h-1} \\
&\quad (2.25)
\end{aligned}$$

Dimana

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

Parameter bias diperoleh dari persamaan berikut:

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

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

3. Perhitungan Nilai *Loss (Error)*

Terdapat beberapa parameter yang digunakan untuk mendapatkan skor ulasan pada Tugas Akhir ini. Parameter tersebut menentukan akurasi pada klasifikasi. Oleh karena itu, dilakukan evaluasi terhadap hasil skor yang didapatkan.

Perhitungan nilai *loss* atau *error* dapat dihasilkan dari persamaan (2.20) karena Tugas Akhir ini menggunakan LSTM untuk klasifikasi. Dapat dilihat seberapa jauh parameter atau bobot yang digunakan dapat mengklasifikasi data masukkan kedalam kelas yang diinginkan berdasarkan nilai *error* yang diperoleh. Pembaharuan nilai bobot atau parameter yang digunakan ditentukan oleh nilai *error* tersebut.

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan tahap-tahap pengerjaan Tugas Akhir ini agar tersusun secara sistematis.

3.1 Pengumpulan Data

Tahap pertama adalah pengumpulan data sebagai input untuk Tugas Akhir ini. Data yang digunakan merupakan data ulasan suatu buku dengan tipe data berupa teks. Data akan diambil dari salah satu situs *e-commerce* ternama di 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

Setelah mengumpulkan data, data harus melalui pra-proses data terlebih dahulu. Tahap ini bertujuan untuk mendapatkan bentuk data yang diinginkan sebelum masuk ketahap implentasi. Terdapat beberapa proses pada tahapan ini antara lain:

a. Tokenisasi

Tahap ini dilakukan pemebentukan array dari kata-kata yang ada di dalam ulasan buku.

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.

3.3 Ekstraksi Fitur

Data yang sudah dilakukan tahap pra-proses data 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. Metode yang dilakukan yaitu *Word2vec*. Metode ini bertugas untuk merupah sekumpulan kata menjadi suatu vektor.

Implementasi algoritma CNN – LSTM

3.4 Implementasi Algoritma CNN-LSTM

Pada tahapan ini dilakukan proses klasifikasi data yang telah melalui tahap ekstraksi fitur. Terdapat dua tahapan utama dalam proses ini yaitu tahap CNN dan LSTM. Pada tahap CNN diterapkan dua tahapan lagi yaitu proses konvolusi dan proses *pooling*. Hasil dari CNN digunakan sebagai inputan pada LSTM. Pada tahap ini pula data dibagi menjadi dua yaitu data latih dan data uji. Pada tahap tersebut data diklasifikasikan menjadi dua kelas antara lain positif dan negatif.

3.5 Evaluasi

Hasil dari implementasi selanjutnya dilakukan proses pengujian. Pada tahap ini dilihat tiga aspek yaitu *precision*, *recall*, dan *F-measure*. Tiga aspek tersebut yang akan digunakan untuk melihat keakurasian atau performansi dari metode yang digunakan.

3.6 Penyusunan Laporan

Pada tahapan terakhir pada Tugas Akhir ini yaitu penyusunan laporan. Hasil yang didapatkan selanjutnya dilakukan analisa dan penarikan kesimpulan.

3.7 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 3.1

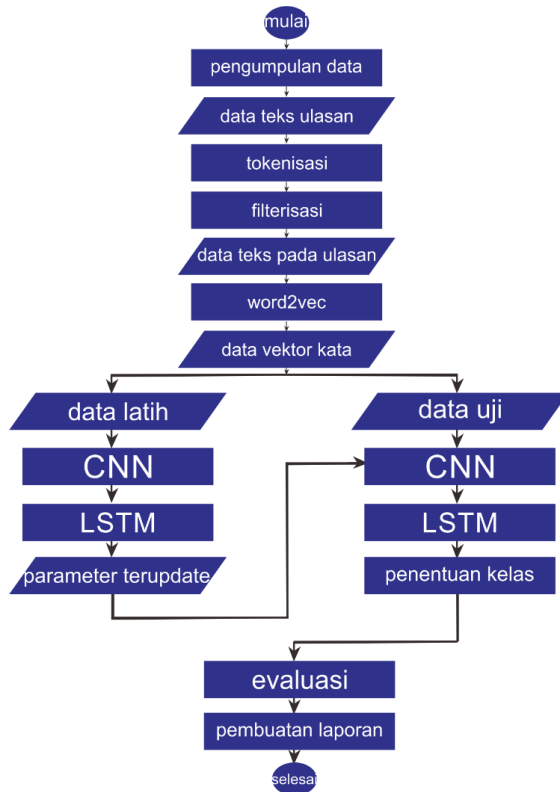
Tabel 3. 1. Lingkungan *hardware* dan *software*

Lingkungan	Spesifikasi	
<i>Hardware</i>	<i>Processor</i>	AMD A8-7410 APU with AMD Radeon R5 Graphics (4CPU's), ~2.2GHz
	RAM	8 GB
<i>Software</i>	Sistem operasi	Windows 10 Pro 64-bit

	<i>Tools</i>	Python 3.5 64-bit using Spyder Anaconda
--	--------------	---

3.8 Diagram Alir Metodologi Penelitian

Berikut adalah ilustrasi dalam bentuk *flowchart* langkah-langkah penelitian pada Tugas Akhir ini.



Gambar 3.1. Diagram alir metodologi penelitian

BAB IV

PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dijelaskan proses-proses pada perancangan desain sistem yang menjadi acuan pada implementasi sistem tersebut. Proses rancang bangun dari awal tahap pengumpulan data hingga pembuatan model algoritma CNN-LSTM digambarkan secara terperinci oleh perancangan sistem. Langkah-langkah pada implementasi sistem ditentukan berdasarkan desain sistem yang sudah dibentuk.

4.1 Pengumpulan Data

Pada implementasi Tugas Akhir ini dibutuhkan data sebagai inputan program. Data yang akan digunakan pada Tugas Akhir ini adalah dataset ulasan buku pada situs jual beli amazon.com.

Setiap *input* dipisah dengan karakter ('\\n'). Setiap masukan memiliki empat atribut berbeda yang dipisah menggunakan spasi. Adapun empat karakter tersebut adalah:

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

Data ulasan buku dikumpulkan dengan cara mencari sumber *repository dataset* sehingga diperoleh data ulasan dari delapan buku acak dari situs amazon.com. Untuk mempermudah penyimpanan serta penggunaan maka data disimpan dalam bentuk *file csv*. Dilakukan pengurangan atribut data agar dataset dapat

digunakan secara optimal. Data *score* yang didapat dari data ulasan akan disimpan di dokumen yang berbeda. Karena data yang dibutuhkan hanya data ulasan dan data *score*-nya maka url dari halaman ulasan akan dihapus. Data *score* yang didapat merupakan angka dari 1 sampai 5. Dari data *score* tersebut akan dibagi menjadi dua kelas, yaitu kelas positif dan kelas negatif. Data yang memiliki *score* 1 dan 2 akan diklasifikasikan menjadi data negatif dan data yang memiliki *score* 4 dan 5 akan diklasifikasikan menjadi data positif. Sedangkan data ber-*score* 3 dihapus karena nilai tengah dan sulit mengidentifikasi kecenderungan berdasarkan *score* data tersebut. Hal itu dilakukan juga untuk menghindari kerancuan dalam klasifikasi ulasan. Berikut adalah contoh ulasan dari *dataset* pada Tugas Akhir ini.

Tabel 4.1. Contoh ulasan positif dan negatif

Ulasan positif	<p>“Why I loved the ending of this terrific thriller, I envy those of you who haven't yet had the fun of reading this wonderfully entertaining and original novel.

"Gone Girl" is the kind of book that compels you to read even when you DON'T have time. If you appreciate skillful writing, it will hook you like an addiction. You will resent distractions, ignore the phone, postpone making dinner in order to stay with it. It's that much fun to read, I promise....”</p>
----------------	---

Ulasan negatif	“worst ending ever I don't like spoilers in reviews so won't include one here, but the ending of this book is so bad I can't believe it was ever published. I had mixed feelings about the book -- somewhat clever, somewhat annoying -- but I stuck with it thinking the author would find a great way to wind it up. Instead we get . . . you've got to be kidding me.”
----------------	---

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.

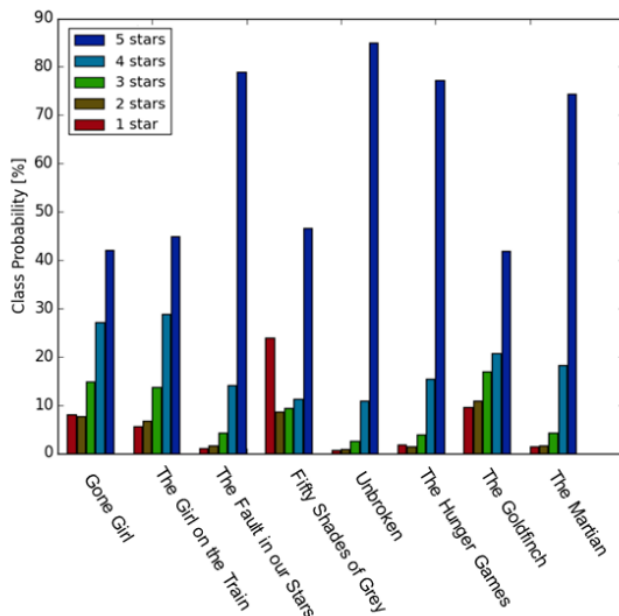
Tabel 4.2. Rincian jumlah ulasan tiap buku

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

Data yang akan digunakan adalah data yang terdistribusi secara merata untuk menghindari ketidakseimbangan. Maka dari

itu akan dipilih satu dataset dari dataset-dataset ulasan delapan buku yang memiliki sebaran data yang paling merata.

Akan ditunjukkan pada Gambar 4.1 penyebaran data pada tiap jenis ulasan buku. Terlihat bahwa ulasan buku *Unbroken* memiliki jumlah *score* 1 jauh lebih sedikit dibanding jumlah *score* 5. Sementara ulasan buku *Gone girl* memiliki penyebaran data yang cukup seimbang dengan kesenjangan antara ulasan positif dan negatif kecil. Oleh karena itu pada Tugas Akhir ini data input yang akan digunakan untuk sistem yang akan dibuat adalah data ulasan dari buku *Gone Girl*.



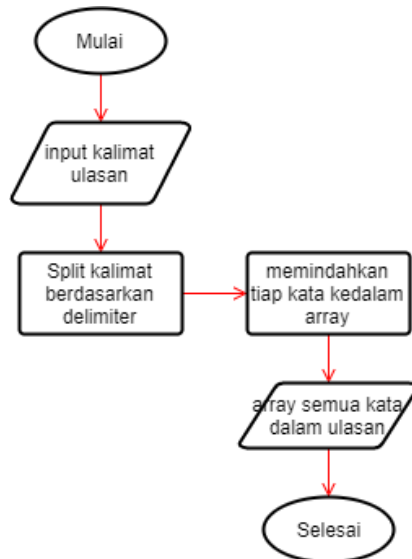
Gambar 4.1. Diagram penyebaran skor ulasan data

4.2 Praproses Data

Sebelum data diproses, data harus melalui tahap praproses data. Sebab data masih berbentuk kalimat atau paragraph utuh. Proses komputasi tidak secara langsung dapat mengenali data berupa teks sehingga sangat sulit dilakukan ekstraksi sentimen pada ulasan. Maka pada Tugas Akhir ini, akan dilakukan dua tahap praproses data yaitu:

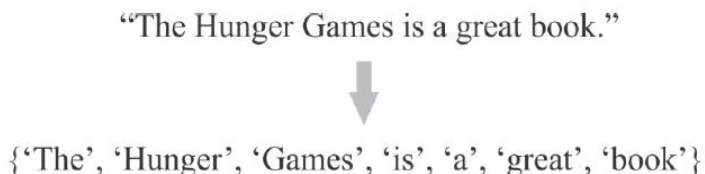
a. Tokenisasi

Tokenisasi merupakan proses pemisahan kata di tiap kalimat pada ulasan. Kata-kata yang sudah saling terpisah disebut token. Tokenisasi dilakukan untuk mempermudah pengamatan makna tiap kata yang berpengaruh dalam menentukan ulasan tersebut positif atau negatif. Akan ditunjukkan diagram alir proses tokenisasi pada Gambar 4.2.



Gambar 4.2. Diagram alir proses tokenisasi

Sementara pada Gambar 4.3 ditunjukkan contoh tokenisasi pada data teks.



Gambar 4.3. Contoh proses tokenisasi pada teks

Pada Tabel 4.3 akan ditunjukkan hasil dari tokenisasi.

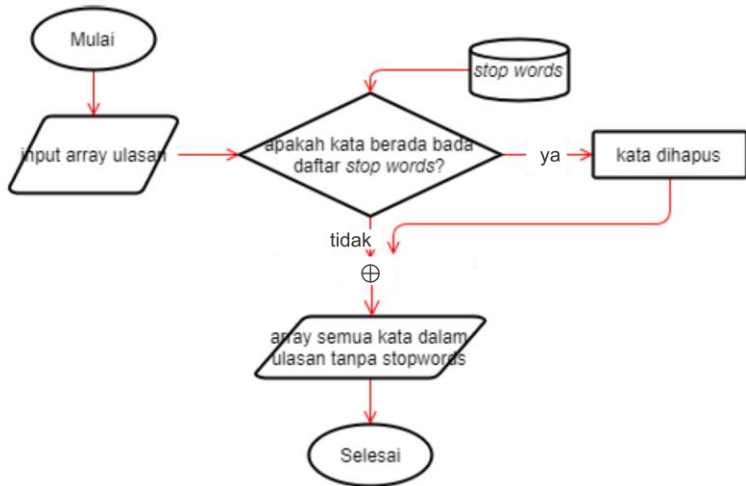
Tabel 4.3. Tabel array kata ulasan hasil tokenisasi

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

Tahap setelah tokenisasi adalah filterisasi. Pada tahap ini, yang akan kita lakukan adalah penyaringan kata-kata penting dari data ulasan, dengan kata lain membuang kata-kata yang tidak berpengaruh pada makna utama dari kalimat atau paragraph ulasan tersebut. Kata-kata yang dihapus antara lain seperti kata hubung, imbuhan dan lain-lain. Selain itu, dilakukan juga *case folding* atau pengubahan huruf capital menjadi huruf kecil agar sistem

dapat mengolah data lebih efisien dan efektif. Berikut adalah diagram alir proses filterisasi pada Gambar 4.4



Gambar 4.4. Diagram alir proses filterisasi

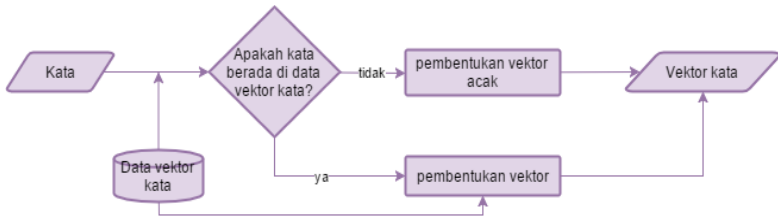
Setelah melalui proses filterisasi maka didapat hasil seperti pada Tabel 4.4.

Tabel 4.4. Array kata ulasan tanpa kata hubung atau *stopwrods*

No	Array Kata Ulasan
1	["book" "brilliant" "ending" "vile" "Wow" "finished" "last" "page" "obviously" "brilliant" ...]
2	["book" "brilliant" "ending" "vile" "Wow" "finished" "last" "page" "obviously" "brilliant" ...]
3	["Horrible" "book" "stunned" "how" "bad" "book" "two" "main" "characters"]

4.3 Ekstraksi Fitur

Ketika kita telah memiliki array kata yang bebas dari *stopwords* maka kita akan mengubahnya ke dalam bentuk vektor kata agar bisa dicerna oleh komputer. Vektor kata merupakan representasi kedekatan antar kata. Vektor yang akan digunakan adalah word2vec yang berupa vektor *Glove*. Kata yang tidak terdapat di vektor *Glove*, vektor katanya akan terbuat secara acak. Pada Gambar 4.5 akan ditunjukkan mekanisme pembuatan vektor kata pada ulasan.



Gambar 4.5. Mekanisme pembuatan vektor kata

Melalui ekstraksi fitur kita mendapatkan dari array kata ulasan tanpa *stopwords* menjadi matriks kata representasi dari kalimat ulasan. Akan terbentuk matriks representasi kata yang berukuran $n \times 50$ dimana n merupakan panjang kata pada ulasan dan 50 adalah panjang vektor kata. Hasil dari ekstraksi fitur akan ditunjukkan pada Tabel 4.5.

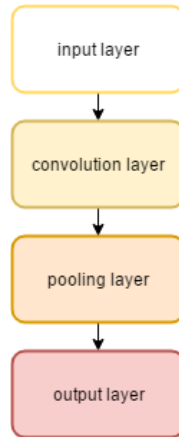
Tabel 4.5. Matriks hasil ekstraksi fitur

No	Array kata tulisan
1	$\begin{bmatrix} -0.0077 & 0.9346 & -0.7319 & \dots \\ -0.562 & 0.2454 & -0.7003 & \dots \\ -0.1733 & 0.2825 & -0.6046 & \dots \\ -0.2862 & -1.2141 & -1.0587 & \dots \\ -0.3636 & 0.2676 & 0.5763 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$
2	$\begin{bmatrix} -0.1497 & -0.4225 & 0.1674 & \dots \\ -0.1733 & 0.2825 & -0.7003 & \dots \\ -0.0918 & 0.0458 & -0.0146 & \dots \\ -0.1027 & 0.4431 & -0.629 & \dots \\ 0.3887 & 0.6805 & -0.1175 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$
3	$\begin{bmatrix} 0.3614 & -0.5514 & -0.7005 & \dots \\ -0.0077 & 0.9346 & -0.7319 & \dots \\ 0.0806 & -0.6435 & 0.1234 & \dots \\ 0.6894 & -0.1064 & 0.1708 & \dots \\ -0.1798 & -0.404 & -0.1653 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$

4.4 Arsitektur CNN-LSTM

Perbedaan antara algoritma *deep learning* dengan algoritma lain adalah kedalaman arsitekturnya. Arsitektur jaringan model *Convolutional Neural Network* (CNN) pada Tugas Akhir ini terdiri atas 4 layer utama, yaitu *input layer*, *convolutional layer*, *pooling layer*, dan *output layer*. Akan digunakan masing-masing satu *convolution layer*, *pooling layer* dan *output layer* dengan inputan berupa matriks ulasan dan vektor kelas berdimensi dua. Hal ini dikarenakan data yang diolah berupa teks yang mana tidak memiliki dimensi yang tinggi seperti data citra sehingga dengan arsitektur ini sudah cukup mampu mengklasifikasikan data teks ulasan dengan baik. Akan ditunjukkan arsitektur dari jaringan yang

akan dibentuk pada Tugas Akhir kali ini. Pada Tugas Akhir ini akan ditunjukkan arsitektur dari jaringan yang dibentuk pada Gambar 4.6.



Gambar 4.6. Diagram alir jaringan CNN

Lapisan pertama adalah *layer input* yang terdapat matriks dari gabungan vektor representasi kata dalam satu ulasan. Dimensi vektor kata yang digunakan pada Tugas Akhir ini adalah 50 sehingga input pada *input layer* adalah matriks berukuran $k \times 50$ dengan k adalah jumlah kata dalam satu ulasan.

Selanjutnya matriks pada *input layer* memasuki *convolution layer*. Pada *convolution layer* ini akan dibentuk vektor *feature map* sebanyak *filter* yang digunakan dalam proses konvolusi. Tiap *filter* akan digunakan untuk semua *window* yang mungkin sehingga masing-masing *feature map* memiliki beragam *window*.

Tahap selanjutnya adalah *pooling layer*, pada *layer* ini akan diambil nilai *feature map* terbaik dari setiap *filter* yang digunakan

hingga didapat nilai *feature* yang paling penting dari ulasan. Hasil dari *pooling layer* adalah vektor yang memiliki panjang sama dengan jumlah filter yang digunakan. Vektor tersebut akan digunakan pada *output layer* dengan jaringan yang terhubung penuh (*fully connected*). *Long Short Term Memory* (LSTM) digunakan untuk mendapatkan skor tiap kelas sehingga dapat diklasifikasikan dalam kategori positif atau negatif.

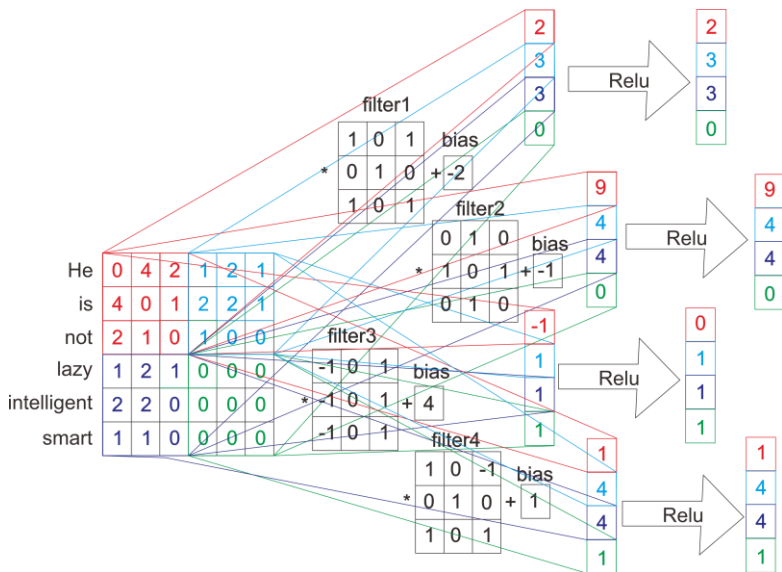
4.5 Implementasi CNN-LSTM

Arsitektur algoritma CNN yang telah dibangun sebelumnya adalah acuan konstruksi model jaringan. Model merupakan persamaan antara pola fitur dengan pola fitur lainnya. Berikut ini adalah deskripsi model pada setiap arsitektur jaringan:

a. *Convolution layer*

Pada *layer* pertama dalam algoritma CNN dilakukan proses konvolusi untuk mendapatkan *feature map* dari data input awal. *Convolutional layer* terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi. Sebagai contoh, penulis menggunakan korpus “*He is not lazy. He is intelligent. He is smart*” yang sama dengan contoh yang ada pada Gambar 2.1. Dapat dilihat pada Gambar 4.7 bahwa data input berukuran 6x6 dan memiliki 4 filter. Keempat filter ini akan digeser ke seluruh bagian dari data input lalu dilakukan operasi dot produk antara input dan

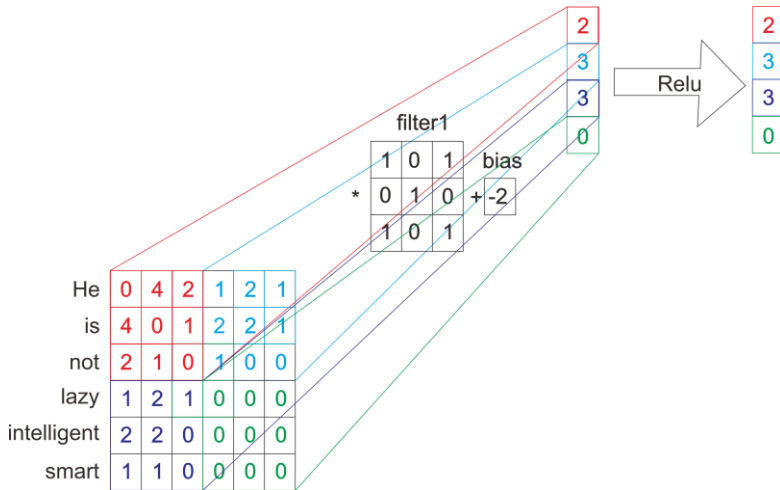
nilai filter pada setiap pergeseran sehingga hasil dari operasi tersebut akan dioperasikan menggunakan *activation function*. Output dari *activation function* tersebut adalah *feature map*. Tidak terdapat aturan khusus dalam pemilihan nilai filter dan bias. Nilai yang digunakan di atas adalah nilai yang umum digunakan pada penelitian sebelumnya.



Gambar 4.7. Arsitektur jaringan CNN

Pergeseran filter yang dibahas pada paragraf sebelumnya dikenal dengan sebutan *stride*. Jika nilai *stride* 1 maka filter akan bergeser sebanyak 1 kolom ke samping lalu 1 baris ke

bawah. Pada contoh di atas nilai *stride* adalah 3. *Activation function* yang digunakan untuk mendapatkan *feature map* di atas adalah *Relu function (Rectified Linear Unit)*, menggunakan persamaan (2.9). Berikut *breakdown* dari proses Gambar 4.7 yang akan ditunjukkan pada Gambar 4.8 sampai Gambar 4.11.



Gambar 4.8. Operasi konvolusi pertama

Perhitungan yang terjadi pada Gambar 4.8 adalah sebagai berikut:

$$\begin{aligned}
 X * W + b &= \begin{bmatrix} 0 & 4 & 2 \\ 4 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} + [-2] \\
 &= (0 * 1 + 4 * 0 + 2 * 1 + 4 * 0 + 0 * 1 + 1 * 0 + 2 * 1 + 1 * 0 \\
 &\quad + 0 * 1) - 2 \\
 X * W + b &= 2
 \end{aligned}$$

$$c_1 = \max(0, 2)$$

$$c_1 = 2$$

$$X * W + b = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} + [-2]$$

$$= (1 * 1 + 2 * 0 + 1 * 1 + 2 * 0 + 2 * 1 + 0 * 0 + 1 * 1 + 1 * 0 + 0 * 1) - 2$$

$$X * W + b = 3$$

$$c_2 = \max(0, 3)$$

$$c_2 = 3$$

$$X * W + b = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} + [-2]$$

$$= (1 * 1 + 2 * 0 + 1 * 1 + 2 * 0 + 2 * 1 + 1 * 0 + 1 * 1 + 0 * 0 + 0 * 1) - 2$$

$$X * W + b = 3$$

$$c_3 = \max(0, 3)$$

$$c_3 = 3$$

$$X * W + b = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} + [-2]$$

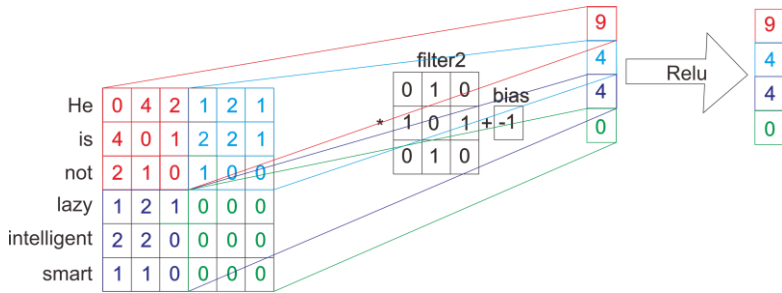
$$= (0 * 1 + 0 * 0 + 0 * 1 + 0 * 0 + 0 * 1 + 0 * 0 + 0 * 1 + 0 * 0 + 0 * 1) - 2$$

$$X * W + b = 0$$

$$c_4 = \max(0, 0)$$

$$c_4 = 0$$

$$c_4 = [2 \quad 3 \quad 3 \quad 0]$$



Gambar 4.9. Operasi konvolusi kedua

Perhitungan yang terjadi pada Gambar 4.9 adalah sebagai berikut:

$$X * W + b = \begin{bmatrix} 0 & 4 & 2 \\ 4 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} + [-1]$$

$$= (0 * 0 + 4 * 1 + 2 * 0 + 4 * 1 + 0 * 0 + 1 * 1 + 2 * 0 + 1 * 1 + 0 * 0) - 1$$

$$X * W + b = 9$$

$$c_1 = \max(0, 9)$$

$$c_1 = 9$$

$$\begin{aligned} X * W + b &= \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} + [-1] \\ &= (1 * 0 + 2 * 1 + 1 * 0 + 2 * 1 + 2 * 0 + 0 * 1 + 1 * 0 + 1 * 1 \\ &\quad + 0 * 0) - 1 \end{aligned}$$

$$X * W + b = 4$$

$$c_2 = \max(0, 4)$$

$$c_2 = 4$$

$$\begin{aligned} X * W + b &= \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} + [-1] \\ &= (1 * 0 + 2 * 1 + 1 * 0 + 2 * 1 + 2 * 0 + 1 * 1 + 1 * 0 + 0 * 1 \\ &\quad + 0 * 0) - 1 \end{aligned}$$

$$X * W + b = 4$$

$$c_3 = \max(0, 4)$$

$$c_3 = 4$$

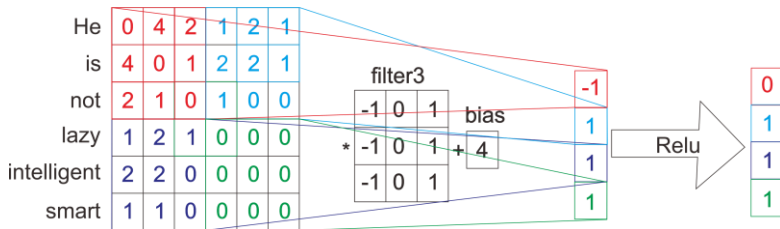
$$\begin{aligned} X * W + b &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} + [-1] \\ &= (0 * 0 + 0 * 1 + 0 * 0 + 0 * 1 + 0 * 0 + 0 * 1 + 0 * 0 + 0 * 1 \\ &\quad + 0 * 0) - 1 \end{aligned}$$

$$X * W + b = 0$$

$$c_4 = \max(0, 0)$$

$$c_4 = 0$$

$$c_4 = [9 \quad 4 \quad 4 \quad 0]$$



Gambar 4.10. Operasi konvolusi ketiga

$$X * W + b = \begin{bmatrix} 0 & 4 & 2 \\ 4 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} + [4]$$

$$= (0 * (-1) + 4 * (-1) + 2 * (-1) + 4 * 0 + 0 * 0 + 1 * 0 + 2 * 1 + 1 * 1 + 0 * 1) + 4$$

$$X * W + b = -1$$

$$c_1 = \max(0, -1)$$

$$c_1 = 0$$

$$X * W + b = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} + [4]$$

$$= (1 * (-1) + 2 * (-1) + 1 * (-1) + 2 * 0 + 2 * 0 + 0 * 0 + 1 * 1 + 1 * 1 + 0 * 1) + 4$$

$$X * W + b = 1$$

$$c_2 = \max(0, 1)$$

$$c_2 = 1$$

$$X * W + b = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} + [4]$$

$$= (1 * (-1) + 2 * (-1) + 1 * (-1) + 2 * 0 + 2 * 0 + 1 * 0 + 1 * 1 + 0 * 1 + 0 * 1) + 4$$

$$X * W + b = 1$$

$$c_3 = \max(0, 1)$$

$$c_3 = 1$$

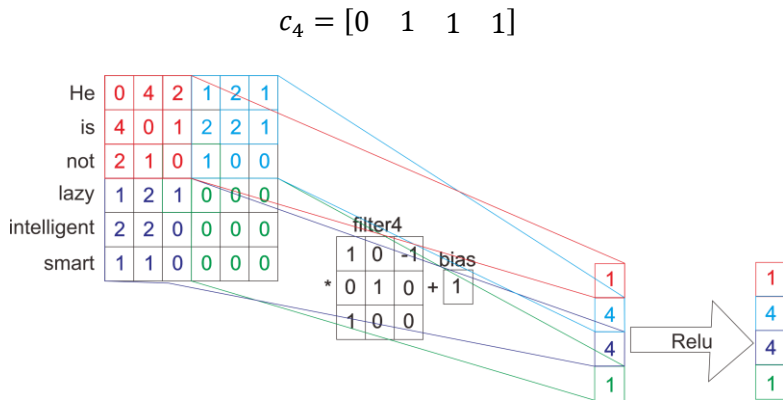
$$X * W + b = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} + [4]$$

$$= (0 * (-1) + 0 * (-1) + 0 * (-1) + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 1 + 0 * 1 + 0 * 1) + 4$$

$$X * W + b = 1$$

$$c_4 = \max(0, 1)$$

$$c_4 = 1$$



Gambar 4.11. Operasi konvolusi keempat

$$X * W + b = \begin{bmatrix} 0 & 4 & 2 \\ 4 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + [1]$$

$$= (0 * 1 + 4 * 0 + 2 * 1 + 4 * 0 + 0 * 1 + 1 * 0 + 2 * (-1) + 1 * 0 + 0 * 0) + 1$$

$$X * W + b = 1$$

$$c_1 = \max(0, 1)$$

$$c_1 = 1$$

$$X * W + b = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + [1]$$

$$= (1 * 1 + 2 * 0 + 1 * 1 + 2 * 0 + 2 * 1 + 0 * 0 + 1 * (-1) + 1 * 0 + 0 * 0) + 1$$

$$X * W + b = 4$$

$$c_2 = \max(0, 4)$$

$$c_2 = 4$$

$$X * W + b = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + [1]$$

$$= (1 * 1 + 2 * 0 + 1 * 1 + 2 * 0 + 2 * 1 + 1 * 0 + 1 * (-1) + 0 * 0 + 0 * 0) + 1$$

$$X * W + b = 4$$

$$c_3 = \max(0, 4)$$

$$c_3 = 4$$

$$X * W + b = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + [1]$$

$$= (0 * 1 + 0 * 0 + 0 * 1 + 0 * 0 + 0 * 1 + 0 * 0 + 0 * (-1) + 0 * 0 + 0 * 0) + 1$$

$$X * W + b = 1$$

$$c_4 = \max(0, 1)$$

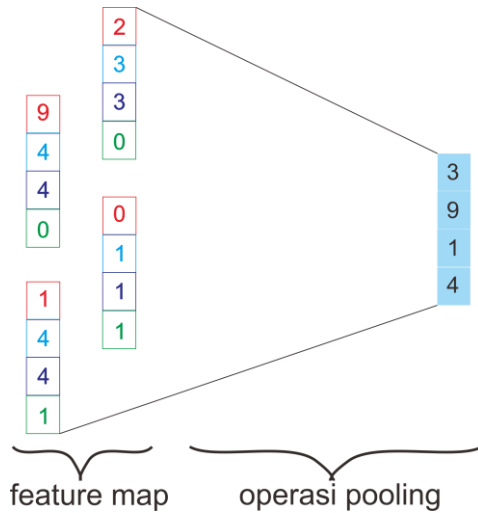
$$c_4 = 1$$

$$c_4 = [1 \quad 4 \quad 4 \quad 1]$$

Didapat *feature map* $c = [c_1, c_2, c_3, c_4]$

b. Pooling layer

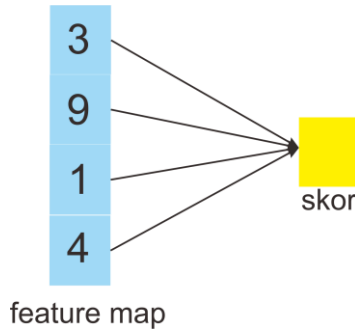
Pada *pooling layer* akan diambil *feature* yang paling penting dari vektor ulasan yang diinputkan. Berikut gambaran proses dari *pooling layer* pada Gambar 4.12:



Gambar 4.12. Ilustrasi jaringan pada *pooling layer*

c. Output layer

Output dari *pooling layer* dimasukkan ke tahap *output layer* yang dapat diilustrasikan seperti pada Gambar 4.10 berikut:



Gambar 4.13. Ilustrasi jaringan pada *output layer*

Skor yang didapatkan adalah penentu klasifikasi dari ulasan tersebut. *Feature map* yang didapat dari *pooling layer* akan dioperasikan dot produk dengan vektor W baru dan ditambah dengan nilai bias baru yang didapat menggunakan persamaan (2.24).

Setelah melalui *layer-layer* pada jaringan CNN-LSTM, dilakukan perhitungan *error* dengan persamaan (2.18) dan dilakukan tahap umpan mundur sesuai persamaan (2.23), (2.24), (2.25) dan (2.26) lalu dilakukan pembaruan parameter dan bobot menggunakan persamaan (2.22).

4.6 Library

Untuk membentuk model *deep learning* yang akan dibangun pada Tugas Akhir ini, digunakan *library* yang ada pada Python Anaconda. Terdapat dua *library* utama pada Tugas Akhir ini yaitu

Keras dan Gensim. *Library* tersebut membantu mempermudah dalam membangun model *deep learning*.

Keras adalah API (*Application Programming Interface*) jaringan syaraf tiruan tingkat tinggi yang ditulis dengan Python dan mampu berjalan diatas Tensorflow, CNTK atau Theano, dengan kata lain Keras merupakan *framework* pada Python yang diperuntukkan untuk membuat aplikasi atau *prototype deep neural network*.

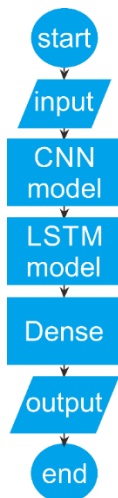
Library Gensim merupakan *framework* pada Python yang digunakan untuk pemodelan topik, pengindeksan dokumen dan pengambilan kesamaan dengan korpus yang besar. Pada Tugas Akhir ini akan dilakukan pengubahan data teks ulasan menjadi vektor representasi kata ulasan dengan menggunakan model Word2vec yang terdapat dalam *library* Gensim.

4.7 Algoritma

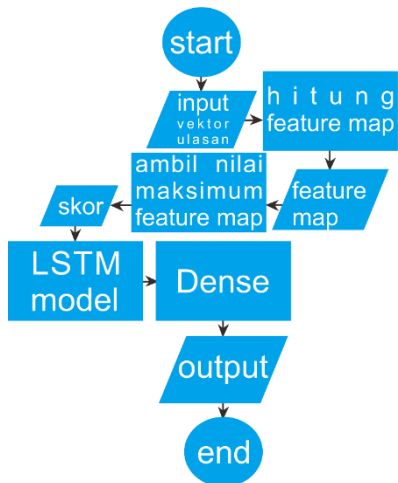
Flowchart dari perpaduan algoritma CNN dan LSTM yang digunakan pada penelitian ini ditunjukkan pada Gambar 4.14.

Seperti yang telah dijelaskan pada Gambar 4.6, 4.8, 4.9 adalah penjabaran dari *block* CNN model. Maka *flowchart* menjadi seperti yang diilustrasikan pada Gambar 4.15.

Dense adalah kata lain dari *fully connected layer* yang merupakan istilah yang menggambarkan kondisi ketika setiap *node* pada *neural network* terhubung ke setiap *node* sebelum dan sesudahnya seperti yang telah ditunjukkan pada Gambar 2.4.



Gambar 4.14. Diagram alir algoritma CNN-LSTM secara umum



Gambar 4.15. Diagram alir algoritma CNN-LSTM

4.8 Implementasi sistem

Pada implementasi sistem, akan dibentuk tiga modul antara lain, modul data, modul Word2vec dan modul utama.

1. Modul data

Tujuan dari modul data adalah mengambil data, membersihkan data, tokenisasi data dan pembentukan data sebelum dilakukan ekstraksi fitur menggunakan Word2vec. Pada awalnya data ulasan buku dipisah menjadi tiga kumpulan data, yaitu data positif, data negatif dan data berskor 3. Hal itu dilakukan untuk mempermudah sistem dalam mengolah data. Penyimpanan data dalam direktori bernama “data_positif.pos” untuk dataset ulasan positif dan “data_negatif.pos” untuk dataset ulasan negatif. Berikut adalah cara untuk mengambil data ulasan dari direktori:

```
# 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]
```

Hasil dari script diatas merupakan kumpulan kalimat ulasan yang akan diproses.

Implementasi selanjutnya adalah tokenisasi dan filterisasi. Tujuan dari keduanya adalah untuk meningkatkan efisiensi algoritma dalam proses perulangannya. Bentuk implementasinya

```
# Load data from directory

positive_examples =
list(open("../data/data_positif.pos",encoding="utf
8",errors='ignore').readlines())

positive_examples = [s.strip() for s in
positive_examples]

negative_examples =
list(open("../data/data_negatif.pos",encoding="utf
8",errors='ignore').readlines())

negative_examples = [s.strip() for s in
negative_examples]
```

pada program adalah sebagai berikut:

Selain itu dibuat label untuk setiap data yang dapat dibentuk seperti ini:

```
# 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

Akan dilakukan perubahan data teks menjadi vektor kata pada tahap ini. Sumber basis data vektor kata yang digunakan merupakan hasil dari model Glove (*Global vektor*) dengan dimensi berukuran 50 setiap kata. Berikut adalah implementasi modul

```
# Initialization Word2vec model

    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)
```

Word2vec dalam program:

Jika ada kata yang diinginkan tidak ditemukan pada data Word2vec maka kata akan dibentuk secara acak. Berikut program implementasinya:

```
# 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 ini, terdapat proses pembentukan model dan tahap pembelajaran dengan dilakukan validasi untuk setiap iterasinya. Iterasi yang digunakan sebanyak 10 kali. Berikut adalah implementasi pembentukan model jaringan:

```
#Building model

model = Sequential()

model.add(Embedding(max_features, embedding_size,
input_length=maxlen))

model.add(Dropout(0.25))

model.add(Conv1D(filters,
                  kernel_size,
                  padding='valid',
                  activation='relu',
                  strides=1))

model.add(MaxPooling1D(pool_size=pool_size))

model.add(LSTM(lstm_output_size))

model.add(Dense(1))

model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Proses selanjutnya adalah proses pembelajaran dengan menggunakan data yang telah dibentuk. Berikut implementasinya dalam program:

```
model.fit(x_latih, y_train,  
          batch_size=batch_size,  
          epochs=epochs,  
          validation_data=(x_uji, y_test))  
score, acc = model.evaluate(x_uji, y_test,  
                             batch_size=batch_size)  
print('Test score:', score)  
print('Test accuracy:', acc)
```

BAB V

UJI COBA DAN EVALUASI SISTEM

Bab ini akan menjelaskan tahap-tahap ujicoba berdasarkan implementasi sistem yang dibuat. Hasil uji coba akan dianalisa dan divalidasi sehingga dapat dilakukan evaluasi sistem.

5.1 Data Uji Coba

Pasca proses pembelajaran maka dilakukan proses pengujian. Data ulasan buku *Gone Girl* berjumlah 41.974 ulasan. Data tersebut dibagi menjadi tiga jenis yaitu data positif, data negatif dan netral. Data positif adalah yang memiliki skor ulasan 4 atau 5 sedangkan data negatif memiliki skor ulasan 2 atau 1. Data netral memiliki skor ulasan 3 dan tidak digunakan pada penelitian ini. Berikut akan ditunjukkan rincian data pada Tabel 5.1:

Tabel 5.1 Rincian jumlah data

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

Demi meringankan proses komputasi maka data yang digunakan sebanyak 10.662 data ulasan dengan rincian 5331 data ulasan positif dan 5331 data ulasan negatif. Hal ini dilakukan untuk pemerataan penyebaran data positif dan negatif. Total data ulasan yang digunakan akan dibagi menjadi dua bagian yaitu data uji dan data latih dengan proporsi pembagian 90% untuk data latih dan

10% untuk data uji. Berikut akan ditunjukkan rincian pembagian data uji dan latih pada Tabel 5.2.

Tabel 5.2. Rincian data uji dan latih

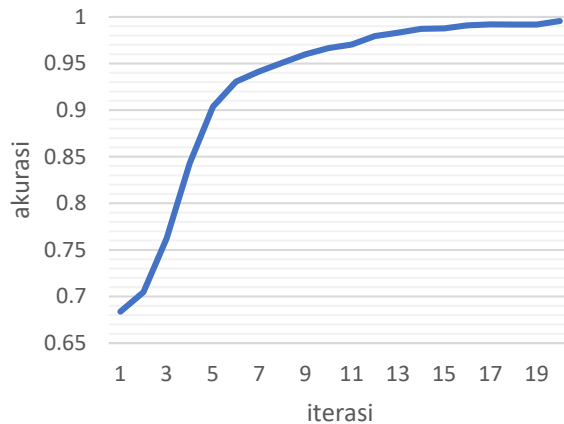
Jenis Data	Jumlah Data
Data latih	9595
Data uji	1067
Total data	10662

5.2 Hasil Uji Coba Sistem

Proses pembelajaran menghasilkan model atau *classifier* yang telah menyesuaikan data yang diinputkan. Model yang dihasilkan dari proses pembelajaran perlu dilakukan pengujian terhadap akurasinya agar dapat diketahui apakah model yang dihasilkan adalah model yang tepat.

Pengujian pada penelitian ini dilakukan dengan cara membagi data menjadi 10 bagian. Pada iterasi pertama, model akan diuji dengan bagian pertama dari 10 bagian pada data. Pada iterasi kedua, data uji adalah bagian kedua dari 10 bagian data. Bagian ketiga pada data adalah data uji untuk iterasi ketiga, begitu seterusnya. Akurasi data uji dari setiap iterasi akan dirata-rata untuk mengetahui performansi model. Perhatikan Gambar 5.1 yang menjelaskan ilustrasi pengujian model.

Data latih dan data uji merupakan data yang tidak saling berketergantungan sehingga dapat diketahui kemampuan model saat mengolah data asing. Sebelum diuji menggunakan data uji,

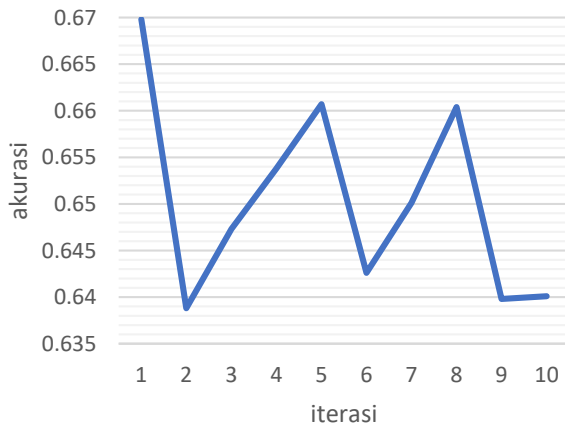


Gambar 5.2. Grafik pergerakan akurasi proses pembelajaran

Terjadi perubahan yang tidak terlalu tinggi pada hasil dari proses pengujian. Saat memproses data uji, model menghasilkan akurasi akhir sebesar 64.01%, dapat dilihat pada grafik akurasi pengujian bahwa akurasi untuk data uji mencapai akurasi tertinggi sebesar 66.98% dan akurasi terendah 63.88% dan berakhir 64.01%. Grafik pergerakan akurasi model terhadap data uji cukup stabil. Perhatikan Gambar 5.3 yang menunjukkan grafik pergerakan akurasi model pada proses pengujian.

Perbedaan grafik akurasi antara data latih dan data uji tersebut dikarenakan adanya pengawasan terhadap ketepatan klasifikasi pada setiap pengujian menggunakan data latih maka akurasi data latih akan selalu meningkat atau tetap pada setiap iterasinya. Berbeda dengan data uji, pengujian model menggunakan data uji tidak diawasi.

Perbandingan antara pergerakan grafik hasil pembelajaran dan pengujian terlihat sangat jauh berbeda. Hal ini menunjukkan model mengalami *overfitting*. *Overfitting* adalah kejadian dimana model terlalu baik dalam mengklasifikasi data latih namun buruk dalam klasifikasi data uji. Hal ini umum terjadi pada klasifikasi menggunakan jaringan syaraf tiruan.



Gambar 5.3. Grafik pergerakan akurasi proses pengujian

Ada banyak faktor yang menyebabkan *overfitting* yaitu, jumlah data kurang besar, jumlah variabel input yang terlalu sedikit atau kualitas data yang kurang baik (*outlier* data, *noise*, konsistensi, korelasi, *input* dan *output* atau sebaran data). *Overfitting* pada penelitian ini disebabkan oleh kualitas data yang kurang baik dan jumlah data yang kurang besar. Ditemukan beberapa data ulasan positif yang berlabelkan negatif dan

sebaliknya. Tabel 5.3 akan menunjukkan contoh data yang berlabel salah.

Tabel 5.3. Contoh data ulasan yang salah label

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 twist and turns in this one. Definitely worth the read

5.3 Evaluasi Model

Model yang telah dibentuk perlu untuk dievaluasi agar dapat dilihat performansinya dalam melakukan tugas yang diperintahkan. Terdapat beberapa cara evaluasi yang digunakan pada model klasifikasi. Pada penelitian evaluasi yang akan dilakukan adalah akurasi, presisi, *recall* dan F1.

Sebelumnya, dilakukan tahap evaluasi sistem, hasil dari proses klasifikasi dikategorikan menjadi empat jenis, yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), *false negative* (FN). TP atau *true positive* adalah jumlah data yang positif yang diklasifikasikan ke kelas positif. TN atau *true negative* adalah jumlah data negatif yang diklasifikasikan ke kelas negatif. FP atau *false positive* adalah jumlah data positif yang diklasifikasikan

ke kelas negatif. FN atau *false negative* adalah jumlah data negatif yang diklasifikasikan ke kelas positif.

Akurasi adalah salah satu evaluasi untuk *task* klasifikasi untuk mengetahui seberapa sering data diklasifikasikan secara benar. Akurasi mengukur ketepatan dan kemiripan hasil pada waktu yang sama dengan membandingkannya terhadap nilai absolut. Akurasi menggambarkan kedekatan hasil klasifikasi dengan target klasifikasi. Perhitungan akurasi secara umum dirumuskan sebagai berikut:

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

Presisi adalah ukuran ketepatan antara informasi yang diminta oleh *user* dan jawaban yang diberikan sistem. Perbedaan antara akurasi dan presisi adalah, akurasi menunjukkan kedekatan hasil pengukuran dengan nilai sesungguhnya, presisi menunjukkan seberapa dekat perbedaan nilai pada saat dilakukan pengulangan pengukuran. Presisi juga dikenal dengan istilah reproduktifitas. Perhitungan presisi secara umum dirumuskan sebagai berikut:

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

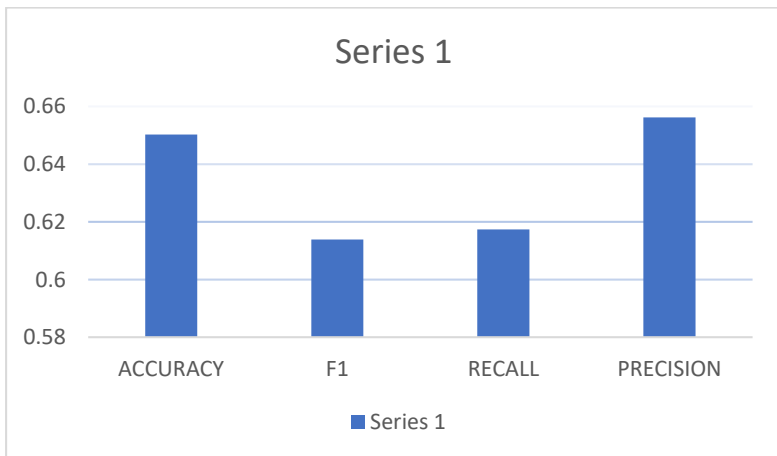
Recall merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. *Recall* dapat dicari dengan menggunakan persamaan berikut:

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

F1 dapat diinterpretasikan sebagai rata-rata dari presisi dan *recall*. F1 dapat diperoleh dengan menggunakan persamaan berikut:

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

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 5.4.



Gambar 5.4. Diagram hasil evaluasi model

Evaluasi sistem pada Tugas Akhir ini menghasilkan nilai akurasi sebesar 65.03% lalu untuk nilai presisi sebesar 65.62%, nilai *recall* sebesar 61.74% dan nilai F1 sekitar 61.39%. Sistem memiliki nilai presisi yang cukup tinggi seperti yang tertera pada Gambar 5.4.

Hasil evaluasi diatas menunjukkan bahwa model memiliki performansi yang cukup baik dalam klasifikasi data ulasan buku dengan tingkat akurasi, presisi, recall dan F1 diatas 61%.

5.4 Perbandingan Hasil Penelitian Terdahulu

Penelitian mengenai *opinion mining* atau *sentiment analysis* terdahulu menghasilkan akurasi yang baik. Penulis berusaha membandingkan metode yang digunakan dengan metode yang telah digunakan sebelumnya yang menggunakan jenis data yang berbeda pula.

Pada penelitian Tang [5] digunakan perpaduan antara metode *Convolutional Neural Network* dengan *Support Vector Machine* untuk mengklasifikasikan data citra. Hasilnya didapat *error* yang lebih kecil dibandingkan yang tidak menggunakan *Support Vector Machine* dengan hasil sebesar 11.9% yang mana nilai *error* penelitian sebelumnya sebesar 14%.

Pada penelitian Kim [6] digunakan model *Convolutional Neural Network* untuk klasifikasi opini berbentuk kalimat. Kim melakukan beberapa cara untuk mengevaluasi modelnya, hasilnya pun berbeda-beda berdasarkan data yang digunakan. Hasil dari data SST-1 hanya didapat hasil 48% sedangkan untuk yang lain mencapai lebih dari 80%.

Pada penelitian Jin Wang dan timnya digunakan perpaduan model CNN-LSTM untuk *dimensional sentiment analysis*. Peneliti melakukan tiga cara untuk mengevaluasi dan membandingkan

nilai error pada modelnya, yaitu *Root Mean Square Error* (RMS) yang bernilai error 1.341, *Mean Absolute Error* (MAE) yang bernilai error 0.987, *Pearson Correlation Coefficient* (r) yang bernilai error 0.778.

Terdapat juga penelitian yang telah dilakukan terhadap data ulasan buku Amazon.com oleh Taspinar. Peneliti melakukan analisis sentiment menggunakan SVM dengan *Bag-of-Word* (BOW) sebagai konstruksi vektor kata. Hasilnya diperoleh akurasi sebesar 60%.

Pada penelitian Fakhur Rozi, digunakan perpaduan model CNN dan L2-SVM untuk *sentiment analysis* menggunakan data ulasan buku yang sama dari Amazon.com. Peneliti menggunakan CNN untuk ekstraksi fitur dan L2-SVM untuk klasifikasi. Hasil dari penelitian tersebut memiliki akurasi 64.6%.

Mengacu pada hasil yang diperoleh pada penelitian sebelumnya, hasil yang didapat pada Tugas Akhir ini memiliki rentang yang tidak begitu jauh dari penelitian-penelitian yang sudah ada. Jika dilakukan perbandingan dengan penelitian Taspinar dan Fakhur Rozi, hasil dari Tugas Akhir ini lebih unggul.

Penggalian opini pada ulasan buku menggunakan algoritma CNN-LSTM menghasilkan performa yang cukup baik. Dibutuhkan data pembelajaran yang lebih banyak untuk dapat menghasilkan performa yang baik. Vektor representasi kata sangat membantu dalam penentuan sentimen dari ulasan.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil seluruh penelitian yang dilakukan dan saran dari peneliti untuk pembaca dan penelitian selanjutnya.

6.1 Kesimpulan

Konstruksi model CNN-LSTM mampu menentukan sentimen atau kecenderungan opini dari ulasan buku. Data terdiri dari 5331 ulasan positif dan 5331 ulasan negatif. Data dibagi menjadi sepuluh bagian secara acak, 10% sebagai data uji dan 90% sebagai data latih. Ulasan tersebut diproses menggunakan aplikasi *word embedding* yang bernama Word2vec untuk mendapatkan vektor representasi kata pada ulasan. Proses tersebut mengubah ulasan menjadi matriks berukuran (9596,1686) untuk data latih dan (1066, 1686) untuk data uji.

Setelah mengolah data latih atau melalui proses pembelajaran maka didapat model CNN-LSTM yang mampu mengklasifikasi ulasan berdasarkan data latih yang telah dipelajari. Model tersebut akan diuji menggunakan data uji dan data latih untuk mengetahui performansi model yang dihasilkan.

Nilai performansi yang didapat dari pengujian model ini menggunakan data uji yang memiliki akurasi sebesar 65.03%. Untuk klasifikasi ulasan buku, model ini memiliki performansi yang lebih baik dibandingkan model CNN - L2-SVM.

6.2 Saran

Saran dari penulis untuk penelitian selanjutnya adalah:

1. Menambahkan jumlah data hingga melebihi 25000 ulasan untuk meningkatkan performansi dan menghindari *overfitting*.
2. Menambahkan metode untuk menghindari *overfitting*.
3. Melakukan eksplorasi terhadap kedalaman jaringan dan jenis klasifikasi.
4. Menggunakan vektor representasi kata yang lebih baik.

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 University of California, Harvard University (7 ed.) (W. H. Freeman). p. Properties of RNA. ISBN 0-7167-3520-2. Diakses tanggal 2010-08-24.
- [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] Olah, Christopher. (2015), *Understanding LSTM Networks* <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [diakses 30 November 2016]

[16] Rozi, Fakhrur. (2017), Penggalian Opini pada Ulasan Buku Menggunakan Algoritma CNN – L2-SVM. Tugas Akhir Departemen Matematika ITS

[17] Chen, Long. He, Yuhan. Fan, Lei. (2016). Let the robot tell: Describe car image with natural language via LSTM

[18] Wang, Jin. Yu, Liang-chih. Lai, Robert. Zhang, Xuejie.
Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model

Biodata Penulis



Penulis memiliki nama lengkap Anshar Zamrudillah Arham, lahir di Pekanbaru 7 Januari 1996. Penulis berdomisili di Pekanbaru, menghabiskan masa kecil disana hingga lulus SD dan merantau sebatang kara ke kota Solo Jawa Tengah hingga SMA. Pendidikan formal yang telah ditempuh oleh penulis adalah SD Islam As-Shofa Pekanbaru (2001-2007), MTs Pondok Pesantren Modern Islam Assalaam Sukoharjo (2007-2010), SMA Pondok Pesantren Modern

Islam Assalaam Sukoharjo (2010-2013). Kemudian penulis melanjutkan studi di departemen Matematika ITS dengan rumpun mata kuliah Ilmu Komputer. Semasa menempuh masa perkuliahan penulis aktif berorganisasi di KM ITS di UKM sinematografi yang bernama Click ITS. Penulis dapat dihubungi melalui email: zamrud.arham@gmail.com