



TESIS - EE185401

**ANALISIS SENTIMEN KEBIJAKAN PEMERINTAH
DALAM PEMBANGUNAN INFRASTRUKTUR PADA
TWITTER MENGGUNAKAN MULTILAYER
PERCEPTRON CLASSIFIER**

EKO YUDHI PRASTOWO
NRP 07111750067001

DOSEN PEMBIMBING
Dr. Ir. Endroyono, DEA
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA - PETIK
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2019



TESIS - EE185401

**ANALISIS SENTIMEN KEBIJAKAN PEMERINTAH
DALAM PEMBANGUNAN INFRASTRUKTUR PADA
TWITTER MENGGUNAKAN MULTILAYER
PERCEPTRON CLASSIFIER**

EKO YUDHI PRASTOWO
NRP 07111750067001

DOSEN PEMBIMBING
Dr. Ir. Endroyono, DEA
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA - PETIK
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2019

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)

di

Institut Teknologi Sepuluh Nopember

oleh:

Eko Yudhi Prastowo
NRP 07111750067001

Tanggal Ujian : 26 Juni 2019
Periode Wisuda : September 2019

Disetujui oleh :



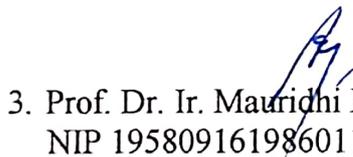
1. Dr. Ir. Endroyono, DEA
NIP 196504041991021001

(Pembimbing I)



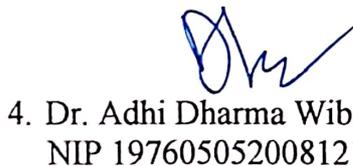
2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.
NIP 196806011995121009

(Pembimbing II)



3. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
NIP 195809161986011001

(Penguji)



4. Dr. Adhi Dharma Wibawa, S.T., M.T.
NIP 197605052008121003

(Penguji)



5. Reza Fuad Rachmadi, S.T., M.T., Ph.D.
NIP 198504032012121001

(Penguji)



Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul **“ANALISIS SENTIMEN KEBIJAKAN PEMERINTAH DALAM PEMBANGUNAN INFRASTRUKTUR PADA TWITTER MENGGUNAKAN MULTILAYER PERCEPTRON CLASSIFIER”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 18 Juli 2019

Eko Yudhi Prastowo
NRP 07111750067001

Halaman ini sengaja dikosongkan

ANALISIS SENTIMEN KEBIJAKAN PEMERINTAH DALAM PEMBANGUNAN INFRASTRUKTUR PADA TWITTER MENGUNAKAN MULTILAYER PERCEPTRON CLASSIFIER

Nama mahasiswa : Eko Yudhi Prastowo
NRP : 07111750067001
Pembimbing : 1. Dr. Ir. Endroyono, DEA
2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.

ABSTRAK

Pembangunan infrastruktur merupakan prioritas kebijakan pemerintah untuk peningkatan pertumbuhan dan pemerataan pembangunan dalam rangka meningkatkan kesejahteraan masyarakat dan pembangunan daerah. Dalam siklus kebijakan publik, evaluasi berfungsi sebagai pertanggungjawaban publik, terlebih masyarakat makin kritis menilai kinerja pemerintah karena masyarakat yang merasakan dampak kebijakan. Pada beberapa penelitian tentang kebijakan publik, pengukuran dampak kebijakan dilakukan dengan metode kuantitatif dengan sumber data yang diperoleh dari wawancara atau kuesioner dengan responden adalah masyarakat. Pengumpulan data melalui wawancara atau kuesioner dengan responden jumlah yang besar membutuhkan waktu yang lama dan biaya yang cukup besar. Pada penelitian ini, data yang digunakan adalah data Twitter yang berisi opini dan komentar terhadap kebijakan pemerintah. Opini dan komentar masyarakat di Twitter diolah menggunakan analisis sentimen untuk menilai apakah masyarakat merasakan dampak positif atau dampak negatif dari kebijakan pemerintah pada pembangunan infrastruktur. Analisis sentimen dilakukan dengan pendekatan *machine learning* menggunakan *Multilayer Perceptron*. Transformasi data teks Twitter menjadi vektor menggunakan *Word2vec* sebagai masukan *Multilayer Perceptron*. *Multilayer Perceptron* menunjukkan nilai akurasi tertinggi sebesar 0,7625 saat menggunakan algoritme pelatihan *Adam* dengan *learning rate* 0,001. Dari data Twitter yang diamati mulai Januari 2016 sampai dengan April 2018, klasifikasi dan analisis sentimen *Multilayer Perceptron* menunjukkan sentimen negatif pada tengah tahun pertama 2016 pada saat kebijakan infrastruktur mulai dicanangkan. Sentimen positif masyarakat mulai muncul mulai tengah tahun kedua 2016 sampai dengan 2018 saat pembangunan infrastruktur menunjukkan hasil dan masyarakat mulai menggunakan infrastruktur utamanya jalan tol dan jalan nasional.

Kata kunci: analisis sentimen, infrastruktur, kebijakan pemerintah, multilayer perceptron, twitter

Halaman ini sengaja dikosongkan

SENTIMENT ANALYSIS OF GOVERNMENT POLICY IN INFRASTRUCTURE DEVELOPMENT ON TWITTER USING MULTILAYER PERCEPTRON CLASSIFIER

By : Eko Yudhi Prastowo
Student Identity Number : 07111750067001
Supervisor(s) : 1. Dr. Ir. Endroyono, DEA
2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.

ABSTRACT

Infrastructure development is a priority for government policies to increase growth and equitable development in order to improve people welfare and regional development. In the public policy cycle, evaluation functions as a public responsibility, especially as people increasingly critically assess government performance because people feel impact of policy. In several studies on public policy, measurement of policy impact was carried out with quantitative methods with data sources obtained from interviews or questionnaires with respondents being people. Data collection through interviews or questionnaires with large numbers of respondents requires a long time and considerable costs. In this study, data used is Twitter data that contains opinions and comments on government policies. Public opinion and comments on Twitter are processed using sentiment analysis to assess whether people feel the positive impact or negative impact of government policies on infrastructure development. Sentiment analysis carried out with a machine learning approach using Multilayer Perceptron. Transforming Twitter text data into vectors using Word2vec as a Multilayer Perceptron input. Multilayer Perceptron shows the highest accuracy value of 0.7625 when using Adam's training algorithm with a learning rate of 0.001. From the Twitter data observed from January 2016 to April 2018, classification and analysis of Multilayer Perceptron sentiments showed negative sentiment in the middle of the first year of 2016 at the time the infrastructure policy began to be launched. Positive public sentiment began to emerge from the middle of the second year of 2016 to 2018 when infrastructure development showed results and the community began to use infrastructure mainly toll roads and national roads.

Key words: sentiment analysis, government policy, infrastructure development, twitter, multilayer perceptron

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillah, tesis dengan judul “ANALISIS SENTIMEN KEBIJAKAN PEMERINTAH DALAM PEMBANGUNAN INFRASTRUKTUR PADA TWITTER MENGGUNAKAN MULTILAYER PERCEPTRON CLASSIFIER” akhirnya dapat diselesaikan sesuai dengan waktu yang ditargetkan. Tesis ini disusun sebagai syarat kelulusan program pascasarjana serta memenuhi salah satu syarat untuk memperoleh gelar Magister Teknik di Institut Teknologi Sepuluh Nopember.

Dalam proses penulisan buku tesis ini tidak terlepas dari dukungan banyak pihak. Untuk itu, penulis mengucapkan banyak terima kasih kepada :

1. Ibuku, Ibu Suryati atas segala kesabaran dan doa yang tak pernah putus.
2. Istriku Arum dan anakku Prasta atas segala dukungan dan pengertian selama menjalani kuliah dua tahun ini.
3. Bapak Dr. Ir. Endroyono, DEA dan Bapak Dr. Eko Mulyanto Yuniarno, S.T., M.T. selaku dosen pembimbing yang telah memberi bimbingan dan arahan dalam penyusunan buku tesis ini,
4. Direktorat Jenderal Pajak Kementerian Keuangan yang telah memberikan kesempatan dan dukungan pada studi S2 dalam negeri sebagai pegawai tugas belajar.
5. Kementerian Komunikasi dan Informatika Republik Indonesia yang telah memberikan beasiswa pascasarjana bidang informatika Pengelola Teknologi Informasi dan Komunikasi Pemerintahan (PETIK) di Institut Teknologi Sepuluh Nopember Surabaya.
6. Bapak Dr. Tri Arief Sarjono, S.T., M.T. selaku Dekan Fakultas Teknologi Elektro Institut Teknologi Sepuluh Nopember Surabaya.
7. Bapak Dr. Ir. Wirawan, DEA selaku Ketua Program Studi Pascasarjana Departemen Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya.
8. Ibu Dr. Trihastuti Agustinah, S.T., M.T. selaku Sekretaris Program Studi Pascasarjana S2 Departemen Teknik Elektro

9. Bapak Dr. Adhi Dharma Wibawa, S.T., M.T. selaku dosen wali dan juga Koordinator Bidang Keahlian Telematika yang membawahi program PETIK.
10. Seluruh Dosen Pengajar S2 Departemen Teknik Elektro Bidang Keahlian Telematika yang bersedia mentransfer ilmu selama masa perkuliahan.
11. Seluruh Staf Akademik atas bantuan demi kelancaran pengurusan administrasi dari awal perkuliahan sampai dengan selesainya tesis ini.
12. Teman-teman PETIK 2017 yang telah banyak berdiskusi dan saling memberikan solusi atas permasalahan selama penyusunan buku tesis.
13. Anggota tetap ERENAM, Mas Gan dan Mbah Mul.
14. Dan semua pihak yang tidak bisa disebutkan satu persatu.

Semoga tesis ini dapat memberikan kontribusi keilmuan dan menjadi sumber amal jariyah bagi kita semua.

Surabaya, 12 Juni 2019

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Kontribusi	3
BAB 2 KAJIAN PUSTAKA.....	5
2.1 Kajian Penelitian Terkait	5
2.2 Teori Dasar.....	7
2.2.1 Kebijakan Pemerintah.....	7
2.2.2 Twitter.....	10
2.2.3 Infrastruktur.....	10
2.2.4 Text Mining.....	14
2.2.5 Analisis Sentimen.....	16
2.2.6 Word Embedding	17
2.2.7 SentiStrength.....	20
2.2.8 Multilayer Perceptron.....	20
2.2.9 Metode Holdout	25
2.2.10 Uji Kinerja Klasifikasi	26
BAB 3 METODE PENELITIAN.....	29

3.1	Alur Penelitian.....	29
3.2	Spesifikasi Data.....	30
3.3	Pengumpulan Data	30
3.4	Penyiapan Dataset Berlabel.....	32
3.5	Pembagian Dataset	35
3.6	Preprocessing.....	35
3.7	Ekstraksi Fitur	38
3.7.1	Model Word2vec	38
3.7.2	Transformasi Tweet ke Vektor	39
3.8	Pelatihan dan Pengujian Multilayer Perceptron	39
3.8.1	Pelatihan Multilayer Perceptron	40
3.8.2	Pengujian Multilayer Perceptron.....	46
3.9	Analisis Hasil	46
3.10	Analisis Sentimen Seluruh Data.....	47
BAB 4 HASIL DAN PEMBAHASAN		49
4.1	Penentuan Parameter	49
4.1.1	Jumlah Layer pada Hidden Layer.....	49
4.1.2	Jumlah Neuron pada Hidden Layer	50
4.1.3	Fungsi Aktivasi pada Hidden Layer	51
4.1.4	Fungsi Loss.....	52
4.2	Multilayer Perceptron Algoritme Pelatihan Gradient Descent Learning Rate 0,01.....	52
4.3	Multilayer Perceptron Algoritme Pelatihan Gradient Descent Learning Rate 0,001.....	55
4.4	Multilayer Perceptron Algoritme Pelatihan Adam Learning Rate 0,01 ..	58
4.5	Multilayer Perceptron Algoritme Pelatihan Adam Learning Rate 0,001	61
4.6	Perbandingan Keseluruhan Hasil	64
4.7	Analisis Sentimen Seluruh Data.....	67
BAB 5 KESIMPULAN		69
DAFTAR PUSTAKA.....		71
BIOGRAFI PENULIS		73

DAFTAR GAMBAR

Gambar 2.1 Model CBOW [21].....	19
Gambar 2.2 Model Skip-Gram [21].....	19
Gambar 2.3 Ilustrasi <i>Multilayer Perceptron</i> [24]	21
Gambar 2.4 Fungsi <i>Softmax</i> [25]	23
Gambar 2.5 <i>Confusion Matrix</i> [26].....	27
Gambar 3.1 Alur Penelitian.....	29
Gambar 3.2 Visualisasi Vektor dalam Bentuk 2 Dimensi	39
Gambar 3.3 Arsitektur <i>Multilayer Perceptron</i>	40
Gambar 4.3 Grafik <i>Cross-Entropy Loss</i> Pelatihan Algoritme <i>Gradient Descent</i> <i>Learning Rate</i> 0,01.....	53
Gambar 4.4 Grafik <i>Cross-Entropy Loss</i> Pelatihan Algoritme <i>Gradient Descent</i> <i>Learning Rate</i> 0,001.....	56
Gambar 4.5 Grafik <i>Cross-Entropy Loss</i> Pelatihan Algoritme <i>Adam Learning</i> <i>Rate</i> 0,01	59
Gambar 4.6 Grafik <i>Cross-Entropy Loss</i> Pelatihan Algoritme <i>Adam Learning</i> <i>Rate</i> 0,001	62
Gambar 4.7 Grafik Hasil Analisis Sentimen Terhadap Keseluruhan Data.....	67

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.1 Contoh <i>Tweet</i>	31
Tabel 3.2 Rincian Jumlah <i>Tweet</i> Unduhan	32
Tabel 3.3 Distribusi Dataset Berlabel	33
Tabel 3.4 Dataset <i>Tweet</i> dengan Label Sentimen	34
Tabel 3.5 Penghapusan sintaks HTML	35
Tabel 3.6 Penghapusan URL.....	36
Tabel 3.7 Penghapusan <i>hashtags</i> dan <i>mentions</i>	36
Tabel 3.8 Penghapusan <i>Unicode</i> selain ASCII	36
Tabel 3.9 Penghapusan Karakter Tanda Baca dan Khusus.....	37
Tabel 3.10 Konversi <i>Slangword</i>	37
Tabel 3.11 <i>Case Folding</i>	38
Tabel 4.1 Distribusi Pembagian Dataset	49
Tabel 4.2 Hasil Validasi Uji Coba Jumlah <i>Layer</i> pada <i>Hidden Layer</i>	50
Tabel 4.3 Hasil Validasi Uji Coba Jumlah Neuron pada <i>Hidden Layer</i>	51
Tabel 4.4 Hasil Validasi Uji Coba Fungsi Aktivasi pada <i>Hidden Layer</i>	51
Tabel 4.5 Hasil Validasi Uji Coba Fungsi <i>Loss</i>	52
Tabel 4.6 <i>Confusion Matrix</i> Pengujian dengan Dataset Pelatihan Algoritme <i>Gradient Descent Learning Rate</i> 0,01	54
Tabel 4.7 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pelatihan Algoritme <i>Gradient Descent Learning Rate</i> 0,01	54
Tabel 4.8 <i>Confusion Matrix</i> Pengujian dengan Dataset Pengujian Algoritme <i>Gradient Descent Learning Rate</i> 0,01	55
Tabel 4.9 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pengujian Algoritme <i>Gradient Descent Learning Rate</i> 0,01	55
Tabel 4.10 <i>Confusion Matrix</i> Pengujian dengan Dataset Pelatihan Algoritme <i>Gradient Descent Learning Rate</i> 0,001	57
Tabel 4.11 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pelatihan Algoritme <i>Gradient Descent Learning Rate</i> 0,001	57
Tabel 4.12 <i>Confusion Matrix</i> Pengujian dengan Dataset Pengujian Algoritme <i>Gradient Descent Learning Rate</i> 0,001	58
Tabel 4.13 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pengujian Algoritme <i>Gradient Descent Learning Rate</i> 0,001	58
Tabel 4.14 <i>Confusion Matrix</i> Pengujian dengan Dataset Pelatihan Algoritme <i>Adam Learning Rate</i> 0,01	60
Tabel 4.15 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pelatihan Algoritme <i>Adam Learning Rate</i> 0,01	60
Tabel 4.16 <i>Confusion Matrix</i> Pengujian dengan Dataset Pengujian Algoritme <i>Adam Learning Rate</i> 0,01	61
Tabel 4.17 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pengujian Algoritme <i>Adam Learning Rate</i> 0,01	61

Tabel 4.18 <i>Confusion Matrix</i> Pengujian dengan Dataset Pelatihan Algoritme <i>Adam Learning Rate 0,001</i>	63
Tabel 4.19 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pelatihan Algoritme <i>Adam Learning Rate 0,001</i>	63
Tabel 4.20 <i>Confusion Matrix</i> Pengujian dengan Dataset Pengujian Algoritme <i>Adam Learning Rate 0,001</i>	64
Tabel 4.21 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pengujian Algoritme <i>Adam Learning Rate 0,001</i>	64
Tabel 4.22 Perbandingan Keseluruhan Hasil	65
Tabel 4.23 Hasil Prediksi <i>Tweet</i> Dataset Pengujian Menggunakan <i>Multilayer Perceptron</i> Algoritme <i>Adam Learning Rate 0,001</i>	66
Tabel 4.24 Hasil Analisis Sentimen Terhadap Keseluruhan Data.....	68

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pembangunan infrastruktur merupakan prioritas kebijakan pemerintah saat ini. Landasan hukum kebijakan pemerintah dalam pembangunan infrastruktur adalah Peraturan Presiden Nomor 74 Tahun 2014 tentang Percepatan Penyediaan Infrastruktur Prioritas dan Peraturan Presiden Nomor 3 Tahun 2016 tentang Percepatan Pelaksanaan Proyek Strategis Nasional sebagaimana telah beberapa kali diubah terakhir dengan Peraturan Presiden Nomor 56 Tahun 2018. Infrastruktur prioritas adalah infrastruktur yang berdampak signifikan terhadap perekonomian. Infrastruktur prioritas mencakup infrastruktur jalan, transportasi, pengairan, air minum, air limbah, sarana persampahan, telekomunikasi dan informatika, minyak dan gas bumi, dan ketenagalistrikan. Proyek strategis nasional memiliki sifat strategis untuk peningkatan pertumbuhan dan pemerataan pembangunan dalam rangka meningkatkan kesejahteraan masyarakat dan pembangunan daerah. Pembangunan infrastruktur prioritas masuk dalam proyek strategis nasional.

Salah satu pembangunan infrastruktur pada proyek strategis nasional yang mempunyai prioritas lebih adalah pembangunan infrastruktur jalan. Infrastruktur jalan meliputi infrastruktur jalan tol dan jalan umum/nasional. Jalan merupakan bagian dari prasarana transportasi yang mempunyai peran penting dalam bidang ekonomi, sosial budaya, dan dipergunakan untuk kemakmuran rakyat. Selain itu jalan berfungsi sebagai prasarana distribusi barang dan jasa sebagai urat nadi kehidupan masyarakat serta penghubung seluruh wilayah Indonesia. Berdasarkan data Kementerian Pekerjaan Umum dan Perumahan Rakyat sepanjang periode 2015 sampai dengan 2018, pemerintah telah berhasil membangun jalan umum 3.387 km dan 442 km jalan tol. Pembangunan infrastruktur jalan tol diharapkan dapat meningkatkan pelayanan kepada masyarakat. Selain dapat mempercepat waktu tempuh perjalanan, dengan makin panjang jalan tol diharapkan dapat menurunkan biaya logistik yang selama ini dianggap memberatkan konsumen.

Evaluasi kebijakan dalam proses kebijakan pemerintah/publik menempati posisi terakhir setelah implementasi kebijakan. Dari evaluasi akan diketahui keberhasilan atau kegagalan sebuah kebijakan, sehingga secara normatif akan diperoleh rekomendasi apakah kebijakan dapat dilanjutkan, perlu perbaikan sebelum dilanjutkan, atau bahkan harus dihentikan. Evaluasi juga menilai keterkaitan antara kebijakan dengan implementasi dalam bentuk dampak kebijakan, apakah dampak tersebut sesuai dengan yang diperkirakan atau tidak. Dari hasil evaluasi pula kita dapat menilai apakah sebuah kebijakan memberikan manfaat atau tidak bagi masyarakat yang dituju. Secara normatif fungsi evaluasi sangat dibutuhkan sebagai bentuk pertanggung jawaban publik, di mana saat ini masyarakat makin kritis menilai kinerja pemerintah. Salah satu bentuk kritisnya masyarakat sebagai pihak yang merasakan dampak dari kinerja kebijakan pemerintah adalah banyaknya penyampaian opini dan komentar terhadap kebijakan pemerintah menggunakan media sosial Twitter. Untuk memahami dan menanggapi opini dan komentar tersebut, diperlukan metode untuk mengolah data informasi, opini dan komentar, serta menganalisis opini dan komentar pengguna media sosial yang disebut dengan analisis sentimen [1]. Analisis sentimen atau penambangan opini adalah bidang studi yang menganalisis opini, sentimen, evaluasi, sikap, dan emosi seseorang dari bahasa tulisan. Analisis sentimen dilakukan untuk menilai opini dan komentar terhadap suatu objek apakah cenderung positif atau negatif [2].

Dalam analisis sentimen terdapat dua metode yaitu *unsupervised* dan *supervised*. Metode *unsupervised* dilakukan menggunakan basis aturan dan kamus kata. Sedangkan metode *supervised* dilakukan menggunakan *machine learning* diantaranya *Naive Bayes*, *Maximum Entropy* dan *Support Vector Machine*. *Machine learning* adalah metode yang paling banyak digunakan. Namun *machine learning* memiliki masalah dalam mengekstrak fitur yang kompleks dan menemukan jenis fitur yang lebih baik. Fitur yang sering digunakan adalah frekuensi kata. Fitur semantik antarkata jarang dipertimbangkan dalam klasifikasi sentimen. Fitur semantik dapat mengungkap hubungan semantik yang dalam dan implisit antar kata yang dapat lebih bermanfaat dalam klasifikasi sentimen. Representasi vektor dari pembelajaran kata dengan menggunakan model *Word2vec* dapat menemukan makna semantik antar kata.

Terdapat teknik pada *machine learning* yang memanfaatkan banyak lapisan pengolahan informasi nonlinear untuk melakukan pengenalan pola, dan klasifikasi yang disebut *deep learning*. Salah satu bentuk algoritme *deep learning* adalah *Multilayer Perceptron*.

1.2 Rumusan Masalah

Kebijakan pemerintah seharusnya memberikan efek dan dampak pada masyarakat. Pengukuran efek dan dampak sebuah kebijakan dilakukan dengan analisis kuantitatif dengan sumber data yang diperoleh dari wawancara atau kuesioner dengan responden adalah masyarakat. Pengumpulan data melalui wawancara atau kuesioner dengan responden jumlah yang besar membutuhkan waktu yang lama dan biaya yang cukup besar. Oleh karena itu, sebagai alternatif solusi dapat menggunakan media sosial Twitter. Media sosial Twitter dapat mewakili populasi masyarakat. Opini dan komentar masyarakat terhadap kebijakan pemerintah di media sosial Twitter diolah menggunakan analisis sentimen. Analisis sentimen dilakukan untuk menilai apakah masyarakat merasakan dampak positif atau dampak negatif dari kebijakan pemerintah pada pembangunan infrastruktur.

1.3 Tujuan

Penelitian ini bertujuan untuk melakukan analisis sentimen terhadap kebijakan pemerintah dalam pembangunan infrastruktur. Analisis sentimen dilakukan menggunakan pendekatan *machine learning* menggunakan *Multilayer Perceptron* dengan sumber data yang diperoleh dari Twitter.

1.4 Batasan Masalah

Pada penelitian ini dilakukan pembatasan yaitu kebijakan pemerintah dalam pembangunan infrastruktur hanya terkait tentang jalan tol, pembangunan jalan dan tol laut.

1.5 Kontribusi

Penelitian ini dapat memberikan kontribusi antara lain :

1. Mengklasifikasikan sentimen di Twitter dalam jumlah yang besar secara otomatis.
2. Pengetahuan bagi masyarakat terhadap kebijakan pemerintah dalam pembangunan infrastruktur.
3. Memperkaya penelitian berkaitan dengan analisis sentimen.

BAB 2

KAJIAN PUSTAKA

Analisis sentimen menggunakan *text mining* sebagai cara untuk yang dapat mengetahui dari isi dari suatu dokumen sehingga dapat dilakukan analisis opini atau sentimen penulis yang terkandung di dalam dokumen tersebut [3]. Di dalam *text mining*, terdapat beberapa proses diantaranya *preprocessing*, ekstraksi fitur dan proses pembelajaran mesin (*machine learning*). *Preprocessing* merupakan proses awal untuk melakukan pengolahan teks agar teks lebih terstruktur dari segi penulisan. Ekstraksi fitur adalah proses mentransformasi sebuah teks menjadi sebuah bentuk yang dapat dikenali oleh algoritme pembelajaran mesin. Algoritme pembelajaran mesin akan memproses masukan dalam bentuk numerik atau vektor dan mengeluarkan keluaran berupa nilai dalam bentuk numerik juga. Algoritme pembelajaran mesin akan berusaha agar nilai hasil keluaran mendekati nilai asli (konvergen) menggunakan pengulangan dan tingkat pembelajaran tertentu.

2.1 Kajian Penelitian Terkait

Penelitian pada Twitter berbahasa Indonesia dan situs resmi pemerintah yang bertujuan untuk memperoleh sentimen dari keluhan publik [4]. Pemilihan fitur dengan *unigram* serta translasi ke bahasa Inggris. Untuk metode klasifikasi menggunakan sumber daya leksikal dari leksikon sentimen berbahasa Indonesia dan *SentiWordNet*.

Selanjutnya penelitian dan analisis terhadap media sosial Twitter yang bertujuan untuk mengevaluasi penggunaan data Badan Pusat Statistik terutama ekonomi oleh pengguna media sosial [5]. Metode klasifikasi yang digunakan adalah *Decision Tree* dengan membagi menjadi empat kategori. Ekstraksi dan pemilihan fitur dengan *Term Frequency Inverse Document Frequency* (TFIDF) dan *N-Gram* (*unigram*, *bigram* dan *trigram*).

Penelitian terhadap Twitter berbahasa Indonesia yang bertujuan untuk analisis sentimen untuk implementasi program kerja pemerintah kota Bandung serta referensi pengambilan keputusan [6]. Penelitian tersebut menggunakan *Part of*

Speech Tagger dan *Stemming* untuk seleksi dan ekstraksi fiturnya dilanjutkan proses pembobotan dengan *Feature Term Frequency*, *Feature Term Presence* dan TFIDF. Klasifikasi menggunakan metode *Support Vector Machine* (SVM).

Penelitian tentang analisis sentimen *tweet* menggunakan *Backpropagation Neural Network* [7]. Penelitian ini menggunakan tiga kelas sentimen yaitu positif, netral dan negatif. *Term Frequency Inverse Document Frequency* (TF*IDF) digunakan untuk melakukan transformasi teks ke dalam bentuk skor. Peneliti melakukan pemeringkatan pada skor TF*IDF dan mengambil beberapa *term* terbaik yang mewakili setiap kelas positif, netral dan negatif. Arsitektur *Backpropagation Neural Network* terdiri dari 3 *layer*. Peneliti menggunakan data latihan sebanyak 630 *tweet* dan data uji sebanyak 314 *tweet*.

Penelitian tentang analisis sentimen pada Twitter mahasiswa menggunakan metode *backpropagation* [8]. Tujuan penelitian ini adalah untuk mengetahui kecenderungan emosi mahasiswa untuk digunakan sebagai acuan menentukan perlakuan yang sesuai terhadap mahasiswa pada saat proses belajar. Peneliti menggunakan kelas sentimen positif, negatif dan netral. Peneliti menggunakan data latihan sebanyak 1.500 *tweet* dari 30 mahasiswa responden. Pemberian label sentimen pada data latihan dilakukan secara manual.

Penelitian tentang analisis sentimen Twitter menggunakan metode *deep learning* atau *deep neural network* [1]. Arsitektur *neural network* yang digunakan adalah 5 *layer*. Peneliti menggunakan dua kelas sentimen yaitu positif dan negatif. Data yang digunakan dalam penelitian ini *tweet* dalam bahasa Inggris dan Korea dengan komposisi 1.000 *tweet* untuk masing-masing positif dan negatif untuk data latihan dan data uji.

Penelitian tentang analisis sentimen tentang pemilihan kepala daerah di Indonesia menggunakan algoritme *term frequency* dan *neural network* [9]. Kelas sentimen yang digunakan adalah positif, netral dan negatif. Peneliti menggunakan dataset sebanyak 130 *tweet*.

Penelitian terkait menjadi studi literatur untuk melakukan analisis sentimen terhadap kebijakan pemerintah dengan data Twitter yang bertujuan untuk mendapatkan sentimen publik terhadap kebijakan yang telah dikeluarkan oleh pemerintah.

2.2 Teori Dasar

2.2.1 Kebijakan Pemerintah

Kebijakan menurut Kamus Besar Bahasa Indonesia dapat diartikan sebagai rangkaian konsep yang menjadi pedoman dan dasar pelaksanaan suatu pekerjaan, kepemimpinan dan juga cara bertindak [10]. Menurut Lasswell, kebijakan adalah suatu program pencapaian tujuan, nilai-nilai dan juga praktik-praktik yang terarah (*a projected program of goals values and practices*) sedangkan menurut Anderson kebijakan adalah serangkaian tindakan yang memiliki tujuan tertentu yang mesti diikuti serta dilakukan oleh para pelakunya untuk menyelesaikan suatu masalah (*a purposive course of problem or matter of concern*).

Istilah kebijakan dapat diterapkan pada pemerintahan yang biasa disebut kebijakan pemerintah atau kebijakan publik. Kebijakan publik ialah suatu aturan yang dibuat oleh pemerintah yang juga merupakan bagian dari keputusan politis untuk mengatasi berbagai persoalan dan isu yang ada dan berkembang di masyarakat. Kebijakan publik juga merupakan keputusan yang dibuat pemerintah untuk melakukan atau tidak melakukan tindakan yang orientasinya pada tujuan tertentu untuk memecahkan masalah-masalah publik atau demi kepentingan publik. Ada dua karakteristik kebijakan publik. Kebijakan publik merupakan sesuatu yang mudah untuk dipahami, karena maknanya adalah hal-hal yang dikerjakan untuk mencapai tujuan nasional. Kebijakan publik merupakan sesuatu yang mudah diukur, karena ukurannya jelas yakni sejauh mana kemajuan pencapaian cita-cita sudah ditempuh. Kebijakan berbeda dengan peraturan dan hukum. Hukum dapat memaksakan atau melarang suatu perilaku, sedangkan kebijakan menjadi pedoman tindakan untuk memperoleh hasil yang diinginkan. Ciri-ciri khusus yang ada pada kebijakan publik, sumbernya berasal dari proses kebijakan tersebut dirumuskan. Ciri-ciri kebijakan publik diantaranya adalah:

1. Kebijakan publik adalah suatu perbuatan yang mengarah kepada tujuan dibandingkan sebagai pelaku atau perbuatan yang serba acak dan kebetulan.
2. Kebijakan hakikatnya terdiri atas perbuatan-perbuatan yang saling terhubung dan mempunyai pola yang mengarah ke tujuan tertentu yang

dijalankan oleh pejabat-pejabat pemerintah dan bukan merupakan keputusan yang berdiri sendiri.

3. Kebijakan menyangkut dengan apa yang sebenarnya dilakukan pemerintah dalam bidang tertentu.
4. Kebijakan publik bisa berbentuk positif, bisa berbentuk negatif, berpengaruh terhadap keputusan-keputusan pejabat pemerintah untuk melakukan maupun tidak melakukan tindakan apapun dalam masalah-masalah yang mana membuat campur tangan pemerintah dibutuhkan.

Pemerintah sebagai pembuat kebijakan publik dimana dalam penyusunannya melewati berbagai tahapan. Tahapan dalam penyusunan kebijakan publik antara lain:

1. Penyusunan Agenda

Para pejabat publik sebagai perumus kebijakan memposisikan masalah pada agenda publik. Masalah sebelum masuk dalam kebijakan publik akan berkompetisi terlebih dahulu, dan pada akhirnya beberapa masalah masuk pada agenda kebijakan para perumus kebijakan. Di tahap ini mungkin suatu masalah tidak disentuh sama sekali, sementara masalah yang lain ditetapkan menjadi fokus yang dibahas, atau terdapat pula masalah karena alasan-alasan tertentu ditunda dalam waktu yang lama.

2. Formulasi Kebijakan

Masalah yang sudah masuk dalam agenda kebijakan selanjutnya dibahas oleh para pembuat kebijakan. Masalah tersebut diartikan untuk selanjutnya dicari pemecahan masalah yang terbaik. Dalam memecahkan masalah dicari dari asal dan berbagai alternatif atau pilihan kebijakan yang ada.

3. Adopsi Kebijakan

Sekian banyak alternatif kebijakan yang disajikan para perumus kebijakan, nanti di akhir salah satu dari alternatif kebijakan tersebut diadopsi dengan dukungan dari mayoritas legislatif, konsensus antara pimpinan lembaga hukum dan peradilan.

4. Implementasi Kebijakan

Sebuah program kebijakan hanya akan menjadi catatan-catatan elit apabila program tersebut tidak di implementasikan. Yaitu dijalankan oleh badan-badan administrasi ataupun agen-agen pemerintah pada tingkat bawah. Kebijakan yang sudah diambil dijalankan oleh unit-unit administrasi yang memobilisasikan sumber-sumber daya finansial dan manusia. Di tahap implementasi ini beberapa kepentingan akan saling bersaing.

5. Evaluasi Kebijakan

Dalam tahapan ini kebijakan yang sudah dijalankan akan dinilai atau dilakukan evaluasi, untuk melihat sejauh mana kebijakan yang dibuat untuk meraih efek dampak yang dikehendaki, yakni memecahkan masalah yang dihadapi masyarakat. Oleh sebab itu ditentukan ukuran-ukuran atau kriteria-kriteria yang menjadi dasar dalam penilaian apakah kebijakan publik yang sudah dilaksanakan telah mencapai dampak atau tujuan yang dikehendaki atau belum.

Evaluasi kebijakan publik yakni aktivitas yang berhubungan dengan estimasi atau penilaian kebijakan yang meliputi substansi, implementasi dan dampak [11]. Evaluasi merupakan sebuah aktivitas fungsional, yang berarti evaluasi kebijakan bukan saja dilaksanakan pada tahap akhir saja tetapi juga terhadap semua proses kebijakan. Efek kebijakan yaitu berbagai hal yang dilakukan oleh pemerintah. Seperti, pembangunan dan rehabilitasi jalan raya, pembayaran tunjangan kesejahteraan atau tunjangan suatu profesi, ditangkannya pelaku tindakan kriminal, diselenggarakannya sekolah umum.

Dampak kebijakan publik merupakan semua efek yang timbul oleh suatu kebijakan dalam kondisi kehidupan nyata [12]. Suatu kebijakan pasti akan menimbulkan dampak, baik itu yang positif atau negatif. Dampak positif yang dimaksud adalah dampak yang diinginkan akan terjadi akibat kebijakan dan memberikan manfaat yang berguna untuk lingkungan kebijakan. Sedangkan dampak negatif yang dimaksud adalah dampak yang tidak memberikan suatu manfaat untuk lingkungan kebijakan dan tidak dikehendaki terjadi.

2.2.2 Twitter

Twitter adalah salah satu layanan media sosial yang populer di internet, membiarkan percakapan berisi “kicauan” individual diantara kumpulan banyak orang. Twitter diciptakan oleh Jack Dorsey pada pertengahan tahun 2006 dengan link URL <http://www.twitter.com>. Twitter mengakui kekuatan dari data historisnya untuk memungkinkan penelitian tentang garis waktu atau lini masa perubahan dalam karakteristik yang disimpulkan dari data mereka, seperti, “mood (suasana hati)”. Aplikasi *microblog* Twitter memiliki aplikasi pemrograman antarmuka (*Application Programming Interface*) agar developer dapat mengembangkan aplikasi tersebut sesuai dengan keinginan dan kebutuhan masing-masing. Adapun dokumentasi perihal Twitter API dapat dilihat di <http://dev.twitter.com>.

2.2.3 Infrastruktur

Infrastruktur dapat dipahami sebagai kebutuhan dasar fisik dari pengorganisasian sistem struktur yang dibutuhkan untuk jaminan ekonomi sektor publik dan sektor privat. Infrastruktur sebagai layanan dan fasilitas yang dibutuhkan agar perekonomian berfungsi dengan baik. Infrastruktur ini berbentuk teknis atau fisik sehingga dapat mendukung jaringan struktur seperti fasilitas berupa jalan, kereta api, waduk, tanggul, air bersih, kanal, pengolahan limbah, perlistrikan, telekomunikasi, bandara, pelabuhan.

World Bank mendefinisikan infrastruktur dalam konteks ekonomi sebagai sebuah terminologi yang memayungi banyak aktivitas terkait “*social overhead capital*” [13]. Lebih jauh “*social overhead capital*” ini dipandang sebagai fondasi bagi peningkatan standar kehidupan, penggunaan lahan nasional dengan lebih baik dan pertumbuhan ekonomi yang *sustainable*. Secara konteks sesuai pernyataan bahwa infrastruktur memiliki peranan yang luas serta dianggap sebagai pendorong dalam pertumbuhan ekonomi.

Karakteristik infrastruktur adalah eksternalitas, baik positif maupun negatif dan adanya monopoli alamiah (*natural monopoly*) yang disebabkan oleh tingginya biaya tetap serta tingkat kepentingannya dalam perekonomian. Selain itu, infrastruktur juga bersifat nonekklusif (tidak ada orang yang dapat dikesampingkan), *nonrivalry* (konsumsi seorang individu tidak mengurangi

konsumsi individu yang lainnya) serta umumnya biaya marginal adalah nol. Infrastruktur juga umumnya tidak diperjualbelikan (*nontradable*).

Infrastruktur merupakan elemen struktural ekonomi yang memfasilitasi arus barang dan jasa antara pembeli dan penjual. Sedangkan pengertian lain yang lebih luas bahwa infrastruktur juga merupakan pelayanan utama dari suatu negara yang membantu kegiatan ekonomi dan kegiatan masyarakat sehingga dapat berlangsung melalui penyediaan transportasi dan fasilitas pendukung lainnya.

Infrastruktur merupakan pondasi atau rancangan kerja yang mendasari pelayanan pokok, fasilitas dan institusi dimana bergantung pada pertumbuhan dan pembangunan dari suatu area, komunitas dan sistem. Infrastruktur meliputi variasi yang luas dari jasa, institusi dan fasilitas yang mencakup sistem transportasi dan sarana umum untuk membiayai sistem, hukum dan penegakan hukum pendidikan dan penelitian.

Dalam hubungan infrastruktur dengan pembangunan ekonomi, beberapa ekonom juga memberikan pendapatnya mengenai infrastruktur. Infrastruktur sebagai sesuatu yang sangat dibutuhkan, tanpa infrastruktur kegiatan produksi pada berbagai sektor kegiatan ekonomi dan industri tidak dapat berfungsi dengan baik.

Keberadaan infrastruktur bagi suatu daerah dapat berdampak terhadap aktivitas ekonomi. Adapun beberapa manfaat infrastruktur, meliputi :

- fasilitasi kegiatan ekonomi masyarakat
- mendukung kelancaran aktivitas ekonomi masyarakat
- mendorong distribusi aliran produksi barang dan jasa
- mengembangkan wilayah dan output ekonomi
- mendukung kegiatan ekonomi, sosial – budaya, kesatuan dan persatuan.
- sebagai katalisator di antara proses produksi, pasar dan konsumen akhir.
- sebagai modal sosial masyarakat.
- memfasilitasi cakrawala masyarakat agar lebih terbuka.
- mempertemukan budaya antar masyarakat.
- mengikat dan menghubungkan antar daerah.

- melalui dukungan Ketahanan Pangan, dapat memberikan rasa aman akan tercukupinya kebutuhan dasar.

Dari berbagai manfaat infrastruktur ini, dapat diketahui bahwa selain perannya dalam sektor ekonomi, infrastruktur pekerjaan umum memiliki peran vital dalam mendukung sektor lain yang juga penting dalam kesejahteraan masyarakat. Dampak pembangunan infrastruktur terhadap aktivitas ekonomi daerah memang jelas adanya. Fakta empiris menunjukkan bahwa perkembangan kapasitas infrastruktur suatu wilayah berjalan seiring dengan adanya perkembangan output ekonomi. Ini berarti secara langsung, infrastruktur dapat berdampak pada peningkatan kegiatan ekonomi secara positif.

Dalam pengadaan infrastruktur, boleh dikatakan bahwa kebijakan publiklah yang memainkan peran besar. Peranan kebijakan publik ini terjadi lantaran tidak adanya mekanisme harga yang jelas pada pengadaan infrastruktur. Adapun peningkatan pengadaan infrastruktur terhadap pendapatan tidak dapat diinterpretasikan sebagai elastisitas pendapatan dari permintaan (*income elasticity of demand*) karena biaya infrastruktur yang dapat berbeda di masing-masing negara.

Data dari The World Bank sendiri menunjukkan bahwa biaya pembangunan jalan di negara dengan pendapatan menengah, hanya kurang lebih 2/3 dari negara kaya dan negara miskin. Kondisi ini menunjukkan bahwa sesungguhnya hubungan GDP per kapita dengan infrastruktur tidaklah sederhana.

Kaitan antara dua hal ini merupakan hasil interaksi kompleks yang lebih dari sekedar penawaran dan permintaan. Selain itu, pengeluaran dana bagi pembangunan infrastruktur juga merupakan sebuah strategi untuk mempromosikan pembangunan ekonomi. Negara – negara berkembang misalnya rata-rata melakukan investasi untuk pembangunan infrastruktur baru sebesar US\$ 200 milyar per tahun. Nilai ini \pm 4 persen dari output nasional serta 1/5 dari total investasi. Namun, dampak investasi yang diharapkan sangat besar ini justru menunjukkan performa infrastruktur yang sering mengecewakan.

Salah satu penyebab dari buruknya performa infrastruktur dan ketidakefektivan infrastruktur ini terletak pada kesalahan dalam pengalokasian dana. Hal ini seperti yang biasanya dilakukan adalah dengan terus melakukan pembangunan infrastruktur baru, sementara tidak ada perawatan terhadap

infarstruktur yang telah ada. Padahal, tingkat perawatan yang minim ini adalah hal penting.

Sebagai contoh, tingkat efektivitas tenaga listrik yang dibangun di negara-negara berkembang hanya 60 persen dari kapasitas terpasangnya. Padahal, optimalnya dapat mencapai 80 persen. Hal ini terjadi akibat perawatan yang buruk. Perawatan yang buruk inilah yang mengurangi jasa pelayanan serta meningkatkan biaya bagi penggunaannya.

Ketika infrastruktur rendah, baik dari segi kuantitas maupun kualitas, hal ini dapat menyebabkan adanya perlambatan pertumbuhan ekonomi dan tenaga kerja. Pada akhirnya, kondisi ini akan berimbas pada kesejahteraan masyarakat secara masif. Contoh nyata, perusahaan-perusahaan yang merasa biaya produksinya melambung akibat buruknya infrastruktur akan keluar dari bisnis atau membatalkan ekspansinya. Alhasil, kesempatan kerja akan berkurang. Akses terhadap produk berkualitas juga akan berkurang. Begitu pula tetesan-tetesan ekonomi juga akan macet.

Karena itulah, keberadaan infrastruktur sangat berperan dalam proses produksi. Infrastruktur juga merupakan pra-kondisi yang sangat dibutuhkan untuk dapat menarik akumulasi modal sektor swasta. Ketiadaan infrastruktur yang baik dapat mengurangi produktivitas dan menambah biaya produksi. Dalam hal ini infrastruktur jalan adalah infrastruktur yang berperan paling strategis terutama pada tahap awal proses pembangunan suatu negara atau daerah. Sebab, ketersediaan jalan tidak hanya berperan dalam mendorong aktivitas ekonomi semata, melainkan juga mendorong ketersediaan dari berbagai jenis infrastruktur lain seperti pembangunan jaringan infrastruktur listrik, jaringan telepon, irigasi, pipa air bersih, rel kereta api, pelabuhan, bandar udara, dan infrastruktur lainnya.

Bisa dikatakan, hampir tidak mungkin untuk bisa menyediakan aneka infrastruktur ini tanpa didahului dengan pembangunan jaringan jalan. Sebab, jaringan jalan inilah yang mempermudah distribusi peralatan untuk mendukung pembangunan berbagai infrastruktur. Sekalipun pengaruh pembangunan infrastruktur dikatakan berperan penting dalam pertumbuhan ekonomi, akan tetapi secara nyata perannya dalam mengurangi ketimpangan ekonomi masih menjadi

tanda tanya. Hal ini karena pembangunan infrastruktur yang dikatakan tidak selalu memberikan dampak positif terhadap peningkatan kualitas pembangunan sosial.

2.2.4 Text Mining

Text mining atau penambangan teks adalah proses ekstraksi pola didalamnya informasi dan pengetahuan dari sejumlah sumber data yang tidak terstruktur. Penambangan teks mempunyai cara dan tujuan yang sama dengan penambangan data, akan tetapi memiliki masukan yang berbeda. Masukan untuk penambangan teks adalah data berupa teks yang tidak terstruktur, seperti dokumen pengolah kata, PDF, kutipan teks, dll, sedangkan masukan untuk penambangan data adalah data yang terstruktur [14].

Penambangan teks melalui dua tahap proses yang diawali dengan penerapan struktur terhadap sumber data teks dan dilanjutkan dengan ekstraksi informasi dan pengetahuan dari data teks dengan menggunakan teknik dan alat yang sama dengan penambangan data. Penambangan teks dapat menghasilkan banyak fitur yang dapat merepresentasikan suatu dokumen. Terdapat empat fitur yang sering digunakan dalam penambangan teks antara lain:

a. *Character*

Character atau karakter merupakan fitur paling sederhana yang membentuk sebuah teks. Fitur ini terdiri dari huruf A-Z, a-z, 0-9, tanda baca dan tanda lainnya. Fitur ini jarang digunakan dalam penelitian tentang penambangan teks karena keterbatasan fitur ini.

b. *Word*

Word atau kata merupakan kombinasi atau kumpulan karakter yang membentuk sebuah makna tertentu. Kata sentimen merupakan kombinasi dari karakter s, e, n, t, i, dan m. Fitur ini lebih banyak digunakan pada penelitian karena jumlah pembeda atau keunikan yang lebih banyak.

c. *Term*

Term juga sering diartikan seperti *word*, tetapi *term* bisa lebih unik daripada *word*. *Term* dapat terdiri dari dua kata atau lebih yang membentuk sebuah makna yang baru. Sebagai contoh analisis

sentimen merupakan sebuah *term* yang dibentuk dari dua buah kata yaitu analisis dan sentimen.

d. *Concept*

Dalam kasus tertentu sering menemukan fitur (*word* dan *term*) memiliki kesamaan makna sehingga perlu untuk melakukan generalisasi terhadap fitur tersebut. Generalisasi terhadap fitur-fitur yang memiliki kesamaan makna yaitu menggunakan *concept*. Sebagai contoh orang, anak, bocah, bapak dan ibu dapat diganti dengan fitur yang lebih umum yaitu manusia. Hal ini bermanfaat karena fitur yang diperoleh menjadi lebih sedikit.

Area implementasi penambangan teks antara lain :

1. Ektaraksi informasi (*informatian extraction*), identifikasi frasa kunci dan keterkaitan di dalam teks dengan melihat sesuatu melalui pencocokan pola.
2. Pelacakan topik (*topic tracking*), penentuan dokumen lain yang menarik seorang pengguna berdasarkan profil dan dokumen yang dilihat pengguna tersebut.
3. Perangkuman (*summarization*), pembuatan rangkuman dokumen untuk mengefisiensikan proses membaca.
4. Kategorisasi (*categorization*), penentuan tema utama suatu teks berdasarkan tema tersebut ke dalam kategori yang telah ditentukan.
5. Penggugusan (*clustering*), pengelompokan dokumen yang serupa tanpa penentuan kategori sebelumnya.
6. Penautan konsep (*concept linking*), penautan dokumen terkait dengan identifikasi konsep yang dimiliki bersama sehingga membantu pengguna untuk menemukan informasi yang mungkin tidak akan ditemukan dengan hanya menggunakan metode pencarian tradisional.
7. Penjawab pertanyaan (*question answering*), pemberi jawaban terbaik terhadap suatu pertanyaan dengan pencocokan pola berdasarkan pengetahuan.

8. Analisis sentimen (*sentiment analysis*), untuk melihat kecenderungan baik positif maupun negatif.

Proses komputerisasi membutuhkan data yang terstruktur dengan baik. Pada *text mining*, data dan informasi yang digali masih mempunyai struktur yang sembarang. Untuk itu diperlukan proses pengubahan dari struktur yang sembarang menjadi data yang terstruktur dengan baik serta pemrosesan menjadi bentuk numerik.

2.2.5 Analisis Sentimen

Analisis sentimen adalah sebuah teknik atau cara yang digunakan untuk mengidentifikasi bagaimana sebuah sentimen diekspresikan menggunakan teks dan bagaimana sentimen tersebut bisa dikategorikan sebagai sentimen positif maupun sentimen negatif. Sentimen sendiri dapat diartikan sebagai pendapat atau pandangan yang didasarkan pada perasaan yang berlebihan terhadap sesuatu biasanya bertentangan dengan pertimbangan pikiran. Pendapat yang hampir senada dimana analisis sentimen digunakan untuk memahami komentar yang diciptakan oleh pengguna internet dan menjelaskan bagaimana sebuah produk maupun *brand* diterima oleh mereka [15]. Analisis sentimen adalah proses yang digunakan untuk menentukan opini, emosi dan sikap yang dicerminkan melalui teks, dan biasanya diklasifikasikan menjadi opini negatif dan positif [16].

Dalam analisis sentimen mencakup beberapa tugas yang berbeda seperti *emotion analysis*, *opinion extraction*, *subjectivity analysis*, *affected analysis*, dan *review mining* [2]. Tujuan utama analisis sentimen adalah bagaimana sebuah mesin dapat mengenali tingkat emosional positif atau negatif dari sampel teks. Adanya polaritas sentimen tersebut merupakan ciri khusus dari teks.

Analisis sentimen dapat dipahami pada beberapa level yaitu level dokumen, paragraf, kalimat atau klausa [17]. Berdasarkan level sumber datanya, analisis sentimen terbagi menjadi dua :

1. *Coarse-grained Sentiment Analysis*

Proses analisis sentimen yang dilakukan pada level dokumen. Klasifikasi sentimen dilakukan pada sebuah dokumen secara keseluruhan untuk menilai sentimen positif dan negatif

2. *Fine-grained Sentiment Analysis*

Proses analisis sentimen dilakukan pada level kalimat. Klasifikasi sentimen dilakukan pada setiap kalimat pada suatu dokumen, sehingga kemungkinan terjadinya perbedaan antara sentimen level kalimat dan dokumen.

2.2.6 Word Embedding

Word embedding merupakan sebuah cara untuk melakukan transformasi sebuah kata menjadi vektor numerik dan merupakan cara paling umum mentransformasikan sebuah kata menjadi vektor. Pada mulanya kata berbentuk teks (*string*). Kata yang berbentuk *string* tidak menjelaskan hubungan antara kata yang satu dengan kata yang lain. *Word embedding* memetakan dengan kata atau frasa dari kosakata ke vektor numerik dalam ruang berdimensi rendah. Vektor tersebut dibandingkan dengan ukuran kosakata dan kemiripan antara vektor yang berkorelasi dengan kata berdasarkan kesamaan semantik.

Model ruang vektor telah digunakan dalam semantik distribusi sejak 1990-an. Sejak itu, kita telah melihat pengembangan sejumlah model yang digunakan untuk memperkirakan representasi kata yang berkelanjutan, *Latent Dirichlet Allocation* (LDA) dan *Latent Semantic Analysis* (LSA) sebagai dua contoh tersebut. Istilah *embeddings* kata awalnya diciptakan pada tahun 2003 yang melatih mereka dalam model bahasa saraf bersama dengan parameter model [18]. Pada tahun 2008 *embeddings* kata pra-terlatih menunjukkan kekuatannya dalam sebuah makalah. Sebuah arsitektur terpadu untuk pemrosesan bahasa alami, dimana membangun *embedding* kata sebagai alat yang sangat efektif apabila digunakan dalam tugas sehari-hari. Sementara arsitektur jaringan saraf banyak digunakan untuk pendekatan *embeddings*. Mikolov pada tahun 2013 membawa kata *embedding* ke permukaan melalui penciptaan *Word2vec*, *toolkit* yang memungkinkan pelatihan dan penggunaan *embeddings* pra-pelatihan [19]. Setahun kemudian, Pennington memperkenalkan *GloVe*, serangkaian pra-pelatihan yang kompetitif [20].

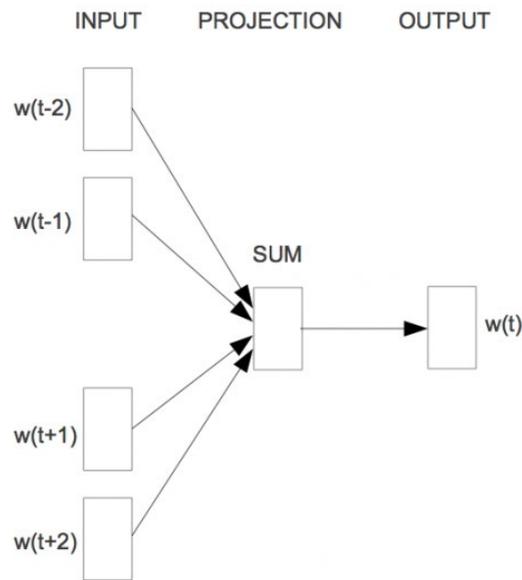
Perkawinan kata dianggap sebagai salah satu dari sejumlah kecil aplikasi pembelajaran tanpa pengawasan yang berhasil saat ini. Fakta bahwa mereka tidak memerlukan anotasi yang mahal mungkin adalah manfaat utama mereka. Sebaliknya, mereka dapat diturunkan dari korpora yang sudah tidak tercatat dari yang telah tersedia..

Salah satu kelebihan *word embedding* adalah dapat diterapkan ke berbagai macam algoritme *machine learning*. Beberapa metode *word embedding* diantaranya *CountVectorizing*, *Term Frequency Inverse Document Frequency* (TDIDF), *Word2vec* dan *GloVe* (*Global Vectors for Word Representation*).

2.2.6.1 *Word2vec*

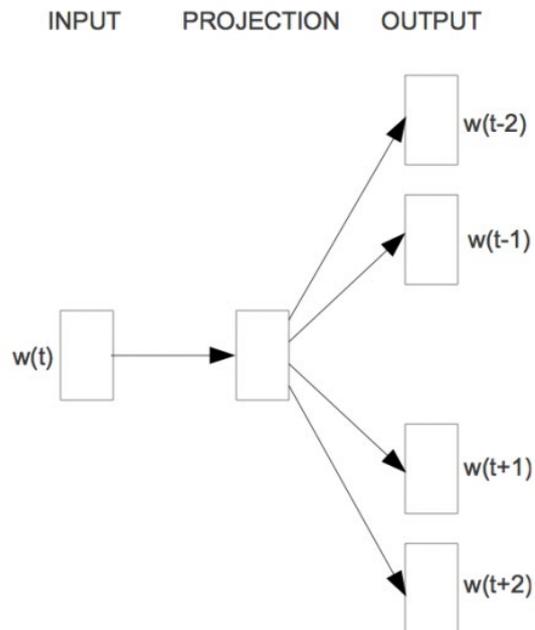
Word2vec adalah teknik *word embedding* yang dikembangkan oleh Mikolov [19]. *Word2vec* merepresentasikan setiap kata di dalam teks sebagai vektor dengan n-dimensi. *Word2vec* menghitung kesamaan konteks dan semantik dari setiap kata yang diinput. Hasil dari penghitungan kesamaan konteks dan semantik dapat merepresentasikan relasi suatu kata dengan kata lainnya. Hasil dari relasi-relasi tersebut menjadi referensi dalam merepresentasikan suatu kata menjadi vektor. *Word2vec* mempunyai dua model, yaitu :

1. *Continous Bag of Word* (CBOW), model *Word2vec* yang dapat memprediksi kata target dari konteks disekitarnya. CBOW mempelajari distribusi probabilitas dari konteks dengan *windows* yang telah ditentukan. Model CBOW cenderung lebih mudah *smooth* terhadap informasi distribusional karena semua kata-kata konteks langsung diproses menjadi satu vektor sebelum akhirnya digunakan untuk memprediksi vektor kata target. Untuk korpus yang ukurannya kecil, model CBOW lebih tepat untuk digunakan.



Gambar 2.1 Model CBOW [21]

2. *Skip-Gram* merupakan kebalikan dari CBOW. *Skip-Gram* menggunakan kata untuk memprediksi konteks disekitarnya. *Skip-Gram* akan mempelajari distribusi probabilitas dari kata-kata di dalam konteks dengan *windows* yang telah ditentukan. Untuk korpus yang ukurannya besar, model *Skip-Gram* lebih tepat untuk digunakan.



Gambar 2.2 Model Skip-Gram [21]

2.2.7 SentiStrength

SentiStrength adalah sebuah algoritme yang menggunakan pendekatan berbasis kamus atau leksikon yang dikembangkan oleh Thelwal [22]. Algoritme ini digunakan untuk mengidentifikasi sentimen dan kekuatan sentimen. *SentiStrength* digunakan untuk membantu memahami peran emosi dalam komunikasi informal. Algoritme ini juga dapat mengidentifikasi ucapan afektif yang tidak pantas atau tidak wajar yang berpotensi dikaitkan dengan perilaku yang mengancam diri sendiri maupun orang lain. *SentiStrength* mengekstraksi kekuatan sentimen dari bahasa Inggris informal.

SentiStrength menggunakan metode baru untuk mengeksploitasi gaya tata bahasa *de-facto* dan gaya penulisan di dunia maya. *SentiStrength* menggunakan aturan khusus dan informasi linguistik tambahan untuk mendeteksi kekuatan sentimen dalam teks pendek dalam bahasa Inggris. *SentiStrength* menggunakan sistem skala ganda (positif – negatif) karena menurut penelitian psikologis, pada saat yang sama manusia dapat merasakan emosi positif sekaligus emosi negatif sampai batas tertentu .

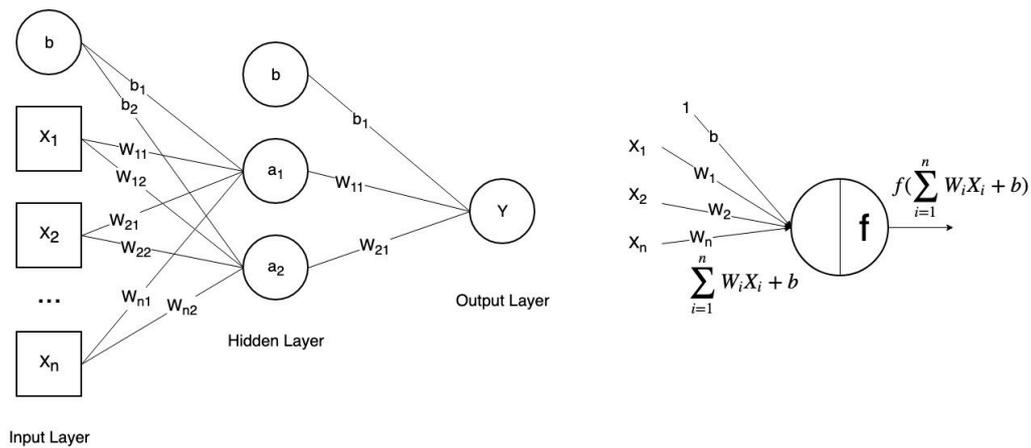
2.2.8 Multilayer Perceptron

Multilayer Perceptron adalah metode klasifikasi jaringan syaraf tiruan *feed-forward* yang terdiri dari sejumlah neuron yang terhubung oleh bobot-bobot penghubung dan menggunakan algoritme pelatihan *Backpropagation* [23]. Jaringan syaraf tiruan merupakan sistem pemrosesan yang mempunyai karakteristik kinerja yang sama dengan jaringan syaraf biologis. Para peneliti mengembangkan jaringan syaraf tiruan sebagai model matematika dari syaraf biologi manusia dengan asumsi sebagai berikut :

1. Pengolahan informasi terjadi pada elemen yang disebut neuron.
2. Perpindahan sinyal dari neuron satu ke neuron lainnya melalui penghubung.
3. Sebuah penghubung mempunyai bobot yang akan mengubah nilai sinyal yang ditransmisikan.
4. Pada penentuan output, setiap neuron mempunyai fungsi aktivasi yang dikenakan pada semua inputnya.

Multilayer Perceptron terdiri dari banyak elemen proses yang disebut neuron. Setiap neuron akan terhubung dengan neuron lainnya dengan bobot tertentu sebagai representasi dari informasi. Nilai bobot akan terus berubah seiring dengan proses pelatihan yang dilakukan sehingga *Multilayer Perceptron* memiliki akurasi yang baik untuk memecahkan masalah.

Sebuah arsitektur *Multilayer Perceptron* dengan tiga lapisan yaitu lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*) dan lapisan keluaran (*output layer*). Dalam penerapannya *hidden layer* dapat dibuat lebih dari satu. Pada Gambar 2.3 terlihat neuron antar lapisan saling terhubung dengan bobot tertentu.



Gambar 2.3 Ilustrasi *Multilayer Perceptron* [24]

Operasi dasar dalam *Multilayer Perceptron* meliputi penghitungan nilai input berbobot dan penerapan fungsi aktivasi. Setiap neuron mempunyai fungsi aktivasi nonlinear dari input yang diterima neuron tersebut. Misal terdapat neuron Y menerima sinyal input dari sejumlah n neuron X (X_1, X_2, \dots, X_n) dengan bobot penghubung adalah W (W_1, W_2, \dots, W_n). Maka penghitungan nilai output seperti ((2.1)

$$Y = f\left(\sum_{i=1}^n W_i X_i\right) \tag{2.1}$$

Keterangan

Y adalah nilai output

f adalah fungsi aktivasi

W_i adalah nilai bobot penghubung

X_i adalah nilai input

Kinerja *Multilayer Perceptron* ditentukan oleh hal-hal sebagai berikut :

1. Arsitektur jaringan yang menentukan pola antar neuron.
2. Algoritme pelatihan, yaitu algoritme yang digunakan untuk menentukan dan mengubah bobot penghubung.
3. Fungsi aktivasi.

2.2.8.1 Bias

Bias adalah salah satu input yang bersifat khusus dengan nilai selalu 1. Penggunaan bias dapat mempercepat pelatihan karena bias mempunyai fungsi sebagai faktor koreksi terhadap kecukupan variabel-variabel input yang ditetapkan. Ketika menggunakan bias (b) maka ((2.1) berubah menjadi ((2.2).

$$Y = f\left(b + \sum_{i=1}^n W_i X_i\right) \quad (2.2)$$

2.2.8.2 Fungsi Aktivasi

Perilaku *Multilayer Perceptron* ditentukan fungsi aktivasi dari bobot dan nilai input. Fungsi aktivasi adalah suatu fungsi berupa persamaan matematis yang memungkinkan mengenali pola data atau memberikan hasil penghitungan dari data input yang belum pernah dipelajarinya. Hasil penjumlahan semua input yang mempunyai bobot akan diproses oleh fungsi aktivasi setiap neuron dan dibandingkan hasil penjumlahan dengan nilai ambang (*threshold*) tertentu. Jika melebihi *threshold* maka aktivasi neuron akan dibatalkan. Sebaliknya jika masih dibawah *threshold*, neuron akan diaktifkan. Setelah aktif neuron akan mengirim nilai output melalui bobot outputnya ke semua neuron yang berhubungan. Proses ini akan terus berulang pada input-input selanjutnya.

Terdapat beberapa fungsi aktivasi nonlinear yang biasa digunakan. Diantaranya *Logistic function (Sigmoid)* dan *Rectified Linear Unit function (ReLU)*.

2.2.8.3 Softmax Layer

Softmax layer diimplementasikan pada *layer* terakhir untuk mendapatkan output. Tidak berbeda dengan neuron-neuron sebelumnya, neuron *softmax* menerima input kemudian melakukan penghitungan bobot dan penambahan bias. Tetapi neuron pada *softmax layer* tidak menerapkan fungsi aktivasi melainkan menerapkan fungsi *softmax*. Fungsi *softmax* mengubah nilai tertimbang menjadi nilai probabilitas dan apabila nilai probabilitas dijumlahkan hasilnya adalah 1 seperti yang dilihat pada Gambar 2.4.

$$\begin{bmatrix} 1.2 \\ 0.9 \\ 0.4 \end{bmatrix} \rightarrow \text{softmax} \rightarrow \begin{bmatrix} 0.46 \\ 0.34 \\ 0.20 \end{bmatrix}$$

Gambar 2.4 Fungsi *Softmax*[25]

Secara matematis fungsi *softmax* ditunjukkan pada ((2.3)).

$$S(L_n) = \frac{e^{L_n}}{\sum e^{L_n}} \quad (2.3)$$

Keterangan

L adalah vektor input pada *softmax layer*

$S(L_n)$ adalah nilai *softmax* dari vektor indeks ke- n

e^{L_n} adalah elemen vektor indeks ke- n

2.2.8.4 Fungsi Biaya

Fungsi biaya atau *cost function* atau *loss function* adalah sebuah fungsi mengukur seberapa baik sebuah prediksi dengan cara mengukur jarak hasil output prediksi (Y) dengan nilai target (Y'). Fungsi biaya menghasilkan nilai kesalahan atau eror antara output pelatihan dan target (*truth*), ini juga biasa disebut fungsi kehilangan (*loss function*). Pengukuran jarak biasanya menggunakan jarak Euclidean, tetapi untuk permasalahan klasifikasi satu jarak lebih efisien menggunakan *Cross-Entropy* ((2.4)).

$$C = - \sum_{i=1}^n Y_i' \log(Y_i) \quad (2.4)$$

Keterangan :

C adalah nilai *cost (loss)*

Y_i' adalah nilai output prediksi

Y_i adalah nilai target

Fungsi biaya dipengaruhi oleh nilai bobot dan nilai bias. Error dikatakan memiliki perbedaan yang sedikit dengan nilai target ketika nilai fungsi biaya sangat kecil. Sebaliknya ketika nilai fungsi biaya besar, maka itu menunjukkan hasil output prediksi masih terdapat banyak perbedaan dengan nilai target.

2.2.8.5 Algoritme Pelatihan

Algoritme pelatihan adalah algoritme yang digunakan untuk menemukan bobot yang tepat sehingga membuat fungsi biaya menghasilkan nilai sekecil mungkin. Algoritme pelatihan yang biasa digunakan adalah *gradient descent*. Teknik ini mengubah bobot dalam kenaikan kecil dengan bantuan perhitungan turunan atau gradien dari fungsi biaya, untuk melihat ke arah mana turun menuju titik paling minimum. Ini dilakukan dalam iterasi (*epoch*) dari semua data yang dimasukkan ke jaringan.

Beberapa algoritme *gradient descent* yang digunakan pada *Multilayer Perceptron* adalah *Gradient Descent*, *Stochastic Gradient Descent*, *Adam*, *Adamax*, *Adadelta*, dan *Adagrad*. Secara umum proses pembelajaran merupakan optimisasi dimana bobot dan bias harus disesuaikan sehingga fungsi biaya dapat diminimalkan. Pada banyak kasus bobot dan bias tidak dapat diselesaikan secara analitis, tetapi dapat didekati dengan iterasi atau optimisasi algoritme pelatihan.

Proses penghitungan gradien hanya pada satu sampel dataset pelatihan untuk memperbarui bobot dan bias disebut *Stochastic Gradient Descent*. Penggunaan beberapa sampel dataset pelatihan memberikan gradien yang lebih baik karena mewakili dari setiap sampel yang berbeda dan dapat mengurangi varians dari pembaruan parameter, yang dapat menyebabkan konvergensi yang

lebih stabil. Penggunaan beberapa sampel sekaligus disebut juga *mini-batch*. Ukuran *mini-batch* (*batch size*) adalah parameter yang dapat disesuaikan berkisar antara 50 dan 256.

2.2.8.6 Learning Rate

Learning rate merupakan salah satu parameter pelatihan untuk menghitung nilai koreksi bobot pada waktu proses pelatihan. Nilai *learning rate* ini berada pada range nol (0) sampai satu (1). Semakin besar nilai *learning rate*, maka proses pelatihan akan berjalan semakin cepat. Namun apabila nilai *learning rate* relatif terlalu besar, pada umumnya proses pelatihan dapat melampaui keadaan optimal yaitu pada saat dicapai nilai eror yang paling minimal. Dengan kata lain, *learning rate* mempengaruhi ketelitian jaringan *Multilayer Perceptron*. Semakin besar *learning rate*, maka ketelitian jaringan akan semakin berkurang, tetapi berlaku sebaliknya, apabila *learning rate*-nya semakin kecil, maka ketelitian jaringan akan semakin besar atau bertambah dengan konsekuensi proses pelatihan akan memakan waktu yang semakin lama.

2.2.9 Metode Holdout

Metode *cross validation* adalah metode statistik yang digunakan untuk mengevaluasi dan membandingkan algoritme yang dipelajari dengan cara membagi data menjadi dua bagian, satu digunakan untuk mempelajari atau melatih model dan satunya digunakan untuk memvalidasi model. Bagian untuk mempelajari algoritma dan melatih model ialah dataset pelatihan sedangkan bagian untuk memvalidasi model dan algoritma yang telah dipelajari disebut dataset pengujian. Dalam tipikal *cross validation*, dataset pelatihan dan dataset pengujian harus saling silang berturut-turut hingga setiap titik data memiliki kesempatan untuk tervalidasi.

Tujuan dari *cross validation* ada dua. Pertama, mengukur kinerja model pembelajaran dari data yang tersedia menggunakan suatu algoritme. Singkatnya, untuk mengukur generalisasi sebuah algoritmae Kedua, untuk membandingkan kinerja dua atau lebih kierja yang berbeda dan mengetahui algoritme terbaik untuk data yang ada, atau alternatif lain untuk membandingkan kinerja dua atau lebih varian model parameter. Salah satu bentuk *cross validation* adalah metode *holdout*.

Metode ini juga merupakan metode pengujian kinerja *classifier* hasil pelatihan. Ide dari metode ini adalah membagi data menjadi dua yang pada umumnya dengan perbandingan 3:1, 3 untuk data latih, dan 1 untuk data uji . Data latih digunakan untuk melatih jaringan *classifier*. Jaringan yang terbentuk akan diuji keandalannya dalam menangani data yang sama sekali belum digunakan dalam proses pelatihan jaringan (*blind data*) tetapi memiliki keanggotaan label yang sama dengan data latih. Metode pengukuran bisa menggunakan apa saja asalkan metode yang digunakan dalam proses pelatihan sama dengan yang digunakan untuk proses pengujian.

2.2.10 Uji Kinerja Klasifikasi

Membentuk model klasifikasi merupakan bagian penting di dalam proses analisis sentimen, tetapi lebih penting lagi adalah mengetahui seberapa baik model-model tersebut dikembangkan. Kinerja model klasifikasi biasanya berdasarkan seberapa baik model tersebut memprediksi hasil untuk data yang baru. Kinerja model klasifikasi ini diukur terhadap sebuah pengujian atau dataset pengujian yang berisi data yang tidak digunakan untuk melatih *classifier* atau model klasifikasi. Dataset pengujian memiliki struktur yang sama dengan dataset yang digunakan untuk melatih model klasifikasi.

Pembelajaran mesin (*machine learning*) membagi klasifikasi ke dalam bentuk biner, multikelas, multilabel, dan hierarki. Pembelajaran mesin terawasi (*supervised machine learning*) memungkinkan akses ke label data selama tahap pelatihan dan pengujian. Saat memasukkan dataset, label kategorik harus ditempatkan ke kelas yang telah ditentukan. Untuk kasus biner, inputnya harus diklasifikasikan menjadi dua kelas yang tidak tumpang tindih. Kategori yang ditentukan bisa objektif (tidak tergantung pada evaluasi manual), atau subjektif (bergantung pada evaluasi manual, misalnya ulasan positif atau negatif). Kelas bisa didefinisikan dengan baik (contoh: label pungut suara), ambigu (contoh: ulasan label pendapat), atau keduanya (contoh: teks medis).

Pembelajaran mesin meminjam langkah-langkah dari bermacam-macam disiplin yang secara tradisional bergantung pada bukti empiris. Dalam beberapa hal, klasifikasi teks meminjam dari *Information Extraction* yang mengawali

penggunaan pembelajaran mesin dalam pemrosesan dan pemahaman teks otomatis, misalnya, analisis otomatis dan pembuatan teks sinonim. Metrik *Information Extraction* dan *Information Retrieval* dalam evaluasi algoritma pembelajaran adalah contoh peminjaman tersebut. Metrik yang dimaksud adalah metrik evaluasi umum yang digunakan dalam klasifikasi teks. Beberapa perhitungan yang menentukan kinerja dari prediksi model dalam klasifikasi teks adalah akurasi, presisi (*precision*) dan sensitivitas (*recall*). Ketepatan klasifikasi dapat dievaluasi dengan menghitung jumlah sampel kelas yang diakui dengan benar (*True Positive*), jumlah sampel yang diakui dengan benar namun tidak digolongkan dalam kelas (*True Negative*), sampel yang salah penetapan kelas (*False Positive*) atau yang tidak dikenali sebagai sampel kelas (*False Negative*). Keempat perhitungan ini merupakan *confusion matrix* yang ditunjukkan pada Gambar 2.5.

		Kelas Prediksi	
		Kelas A	Bukan Kelas A
Kelas Target	Kelas A	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
	Bukan Kelas A	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

Gambar 2.5 *Confusion Matrix*[26]

Akurasi merupakan perbandingan antara data yang diprediksi benar sesuai kelas target dengan keseluruhan data oleh model klasifikasi. Akurasi dapat diformulasikan pada ((2.5)

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.5)$$

Precision merupakan perbandingan data yang teridentifikasi dengan benar sebagai positif dari seluruh data yang diprediksi sebagai positif oleh model klasifikasi. *Precision* dapat diformulasikan pada ((2.6).

$$Precision = \frac{TP}{TP + FP} \quad (2.6)$$

Recall adalah jumlah data kelas positif yang diprediksi dengan benar oleh model klasifikasi. *Recall* dapat diformulasikan pada ((2.7)

$$Recall = \frac{TP}{TP + FN} \quad (2.7)$$

BAB 3

METODE PENELITIAN

3.1 Alur Penelitian

Secara garis besar alur yang dilakukan pada penelitian ini adalah pengumpulan data Twitter, pelabelan untuk menyiapkan dataset berlabel, pembagian dataset menjadi dataset pelatihan dan dataset pengujian, *preprocessing* teks *tweet* dalam dataset, ekstraksi fitur untuk mengubah dataset berupa teks *tweet* menjadi vektor, pelatihan dan pengujian *Multilayer Perceptron*, analisis hasil pelatihan dan pengujian serta analisis sentimen terhadap keseluruhan data.



Gambar 3.1 Alur Penelitian

3.2 Spesifikasi Data

Data input adalah data yang digunakan sebagai masukan pada penelitian ini. Data input akan diproses dan digunakan lebih lanjut untuk mendapatkan output yang dikehendaki. Data input yang digunakan adalah data teks (*tweet*) yang di-*posting* di Twitter. Data output adalah data hasil keluaran dari proses pada penelitian ini. Data output meliputi data *tweet* hasil proses pengumpulan data, data *tweet* bersih hasil *preprocessing*, fitur vektor hasil proses ekstraksi fitur, model klasifikasi hasil proses pelatihan atau pembelajaran algoritme klasifikasi menggunakan data pelatihan, hasil dari proses prediksi data pengujian terhadap model klasifikasi dan nilai hasil pengukuran antara hasil prediksi dan data pengujian.

3.3 Pengumpulan Data

Data yang digunakan pada penelitian ini adalah data teks yang berasal dari *tweet* di Twitter. Pengambilan data *tweet* dilakukan menggunakan metode *scraping* halaman pencarian lanjutan (*advanced search*) di *website* Twitter. Pengambilan data tidak dilakukan menggunakan *Application Programming Interface* (API) yang disediakan oleh Twitter karena keterbatasan API Twitter yang hanya dapat melakukan pengambilan data *tweet* tujuh hari ke belakang dari tanggal pada saat pengambilan data.

Dalam proses *scraping*, secara otomatis akan mengambil data *tweet* yang mengandung kata dan dalam jangka waktu yang telah ditentukan. Teknik *scraping* yaitu membuat sebuah program komputer seperti peramban atau *browser* yang melakukan pencarian melalui halaman pencarian lanjutan di Twitter. Di dalam peramban terdapat pemuat gulir (*scroll*). Sebuah pemuat gulir akan menggulir ke bawah untuk mendapatkan data yang lebih banyak. Pada saat itu data *tweet* secara otomatis diambil dan disimpan dalam sebuah berkas teks. Hal ini dilakukan berulang-ulang sampai dengan pemuat gulir selesai memuat semua hasil pencarian dari *website* Twitter.

Data *tweet* yang diambil adalah *tweet* dari tanggal 1 Januari 2016 sampai dengan 30 April 2018. Pengambilan data dilakukan pada tanggal 18 Mei 2018

sampai dengan 25 Mei 2018. Contoh *tweet* dapat dilihat pada Tabel 3.1. Rincian *tweet* unduhan sebagaimana Tabel 3.2.

Tabel 3.1 Contoh *Tweet*

<i>Tweet</i>
jalan pintas GPP utk sementara s/d lebaran, utk Rakyat banyak, selanjutnya perlu dibangun 'jalan tol' utk stabilitas harga pangan
mending bangun jalan tol dari sukabumi sampai banjar.. wilayah selatan udah mati gaya. https://twitter.com/fadjroeL/status/741398999114993665 ...
TOL Laut untuk ternak saja merugi .. semoga yang ini gak rugi dan STOP.
Ini baru bs disebut pelabuhan internasional. Super cargo sdh bs bersandar manis di dermaga. Tol laut pak Presiden sukses https://twitter.com/detikcom/status/854284229458370560 ...
JKW hutang dan bangun jalan jembatan bukan buat rakyat indo, ente salah, semua itu untuk warga china yg sebentar lg mirgrasi gede2an ke indo, n ente nnt cma kebagian jd kacung2 mrk, bagi jkw, negara ini hrs dibangun dgn hutang agar china bisa ikut nikmati, lalu komunis berkuasa
Nah poinnya di sini mas. Semua naik kecuali pendapatan krn dibuat bangun jalan semoga ke depan membaik

Tabel 3.2 Rincian Jumlah *Tweet* Unduhan

Kata Kunci	Jumlah <i>Tweet</i> Unduhan
jalan tol	315.857
bangun jalan	13.590
tol laut	27.166
Total	356.613

3.4 Penyiapan Dataset Berlabel

Klasifikasi merupakan metode pembelajaran mesin yang terawasi yang membutuhkan dataset yang mempunyai label. Label berperan sebagai supervisor atau guru yang mengawasi proses pembelajaran dalam mencapai tingkat akurasi atau tingkat eror tertentu. Pada penelitian ini, dataset yang akan digunakan mempunyai atribut berupa teks *tweet* dan kelas atau label sentimen yang terdiri positif dan negatif.

Dataset berlabel diperoleh dengan cara melabeli setiap *tweet* secara manual berdasarkan sentimen positif atau sentimen negatif dan menggunakan algoritme *SentiStrength*. Pelabelan *tweet* secara manual dilakukan pada 1.682 *tweet*. Pelabelan dengan cara *generate* label positif dan label negatif menggunakan algoritme *SentiStrength* untuk mendapatkan sebanyak 5.132 *tweet*. *SentiStrength* merupakan kamus leksikon sentimen seperti *SentiWordNet* [27] dengan penggunaan lebih mudah karena tidak menggunakan *synset* tetapi hanya menghitung skor dari kata yang ada di dalam kamus leksikon. Karena *SentiStrength* berdasarkan bahasa Inggris, sedangkan dataset dalam penelitian ini tentang bahasa Indonesia, jadi perlu proses penerjemahan. Data *tweet* diterjemahkan ke dalam bahasa Inggris menggunakan program komputer yang mengadopsi *Google Translate*. Setelah data *tweet* diterjemahkan dalam bahasa Inggris proses selanjutnya adalah penentuan skor menggunakan algoritme *SentiStrength*. Algoritme *SentiStrength* memberikan skor positif (*positive strength*) dan skor negatif (*negative strength*) dari setiap data *tweet* yang diproses. Penentuan label menggunakan aturan sebagai berikut :

- Jika penjumlahan skor positif dan skor negatif lebih besar dari nol, maka label data *tweet* tersebut adalah positif.
- Jika penjumlahan skor positif dan skor negatif lebih kecil dari nol, maka label data *tweet* tersebut adalah negatif.
- Jika penjumlahan skor positif dan skor negatif sama dengan nol, maka data *tweet* tersebut tidak dipakai.

Proses validasi dari pelabelan menggunakan algoritme *SentiStrength* dengan menggunakan penganotasi secara manual. Distribusi dataset *tweet* dengan label sentimen seperti ditunjukkan pada Tabel 3.3.

Tabel 3.3 Distribusi Dataset Berlabel

Topik	Jumlah Dataset Berlabel
jalan tol	2.736
pembangunan jalan	2.080
tol laut	1.998
Total	6.814

Setiap dataset pada Tabel 3.3 memiliki label sentimen yang digunakan sebagai dataset pelatihan dan dataset pengujian *Multilayer Perceptron*. Tabel 3.4 menunjukkan contoh dataset *tweet* beserta label sentimen hasil dari proses pelabelan.

Tabel 3.4 Dataset *Tweet* dengan Label Sentimen

<i>Tweet</i>	Label Sentimen
<p>jalan pintas GPP utk sementara s/d lebaran, utk Rakyat banyak, selanjutnya perlu dibangun 'jalan tol' utk stabilitas harga pangan</p>	Positif
<p>mending bangun jalan tol dari sukabumi sampai banjar.. wilayah selatan udah mati gaya.</p> <p>https://twitter.com/fadjroeL/status/7413989991 14993665 ...</p>	Positif
<p>TOL Laut untuk ternak saja merugi .. semoga yang ini gak rugi dan STOP.</p>	Negatif
<p>Ini baru bs disebut pelabuhan internasional. Super cargo sdh bs bersandar manis di dermaga. Tol laut pak Presiden sukses</p> <p>https://twitter.com/detikcom/status/854284229 458370560 ...</p>	Positif
<p>JKW hutang dan bangun jalan jembatan bukan buat rakyat indo, ente salah, semua itu untuk warga china yg sebentar lg mirgrasi gede2an ke indo, n ente nnt cma kebagian jd kacung2 mrk, bagi jkw, negara ini hrs dibangun dgn hutang agar china bisa ikut nikmati, lalu komunis berkuasa</p>	Negatif
<p>Nah poinnya di sini mas. Semua naik kecuali pendapatan krn dibuat bangun jalan semoga ke depan membaik</p>	Positif

Melalui proses ini akan didapatkan pasangan data *tweet* dan label yang disimpan dalam sebuah berkas teks.

3.5 Pembagian Dataset

Untuk kebutuhan pengujian model klasifikasi menggunakan *cross validation* dengan metode *holdout*, penulis mengambil 682 *tweet* dari dataset yang telah dilabeli secara manual untuk digunakan sebagai dataset pengujian. Sisanya digunakan sebagai dataset pelatihan untuk proses pelatihan atau pembelajaran *Multilayer Perceptron*. Dataset pengujian tidak diikuti selama proses pelatihan *Multilayer Perceptron*.

3.6 Preprocessing

Preprocessing dilakukan terhadap atribut teks *tweet* pada dataset pelatihan dan dataset pengujian. Tujuan *preprocessing* adalah untuk membersihkan atribut teks *tweet* sebelum proses ekstraksi fitur dilakukan. Tahapan yang dilakukan dalam *preprocessing* antara lain,

1. Penghapusan karakter atau sintaks *HyperText Markup Language* (HTML). Sintaks HTML yang muncul di dalam *tweet* antara lain ``, ``, `
`, ``, ``, `<h3>`, `</h3>`, dan lain-lain. Tabel 3.5 menunjukkan penghapusan sintaks `<b<` dan `` pada sebuah *tweet*.

Tabel 3.5 Penghapusan sintaks HTML

Masukan	Keluaran
<code></code> Jalan tol <code></code> digunakan juga untuk jalur distribusi barang . Apapun membangun jalan adalah suatu kebaikan!	Jalan tol digunakan juga untuk jalur distribusi barang . Apapun membangun jalan adalah suatu kebaikan!

2. Penghapusan *Uniform Resource Locator* (URL). URL di dalam *tweet* biasanya digunakan untuk menautkan ke situs berita online, tautan gambar, media sosial lainnya seperti Instagram. Tabel 3.6 menunjukkan penghapusan sebuah url `t.co/QnhdhW` pada *tweet*.

Tabel 3.6 Penghapusan URL

Masukan	Keluaran
pembangunan jalan tol di kalimantan good job pak t.co/QnhdhW	pembangunan jalan tol di kalimantan good job pak

3. Penghapusan tanda *hashtags* (#) dan *mentions* (@) karena umumnya tidak mengandung informasi penting. Tabel 3.7 menunjukkan penghapusan tanda *mentions* @FPKSDPRRI dan tanda *hashtags* #Kom5 Sigit S, #Jatim1:.

Tabel 3.7 Penghapusan *hashtags* dan *mentions*

Masukan	Keluaran
#Kom5 Sigit S @FPKSDPRRI #Jatim1: sy ingin mengucapkan selamat hebat kpd anak" yg berhasil membangun jalan tol.	sy ingin mengucapkan selamat hebat kpd anak" yg berhasil membangun jalan tol.

4. Penghapusan karakter *Unicode* selain ASCII. Karakter *Unicode* hampir mirip dengan karakter pembentuk *emoticon*. Dalam pemrosesan teks biasa terlalu sulit untuk memproses karakter *Unicode*, untuk itu dilakukan penghapusan karakter ini. Tabel 3.8 menunjukkan penghapusan *Unicode* © (*copyright*) dan *emoticon* ♥♥♥♥.

Tabel 3.8 Penghapusan *Unicode* selain ASCII

Masukan	Keluaran
© Gw harap pembangunan jalan tol Bekasi-Cawang-Kampung Melayu ini beneran jadi ♥♥♥♥	Gw harap pembangunan jalan tol Bekasi-Cawang-Kampung Melayu ini beneran jadi

5. Penghapusan karakter angka, *enter*, *tab*, *double space*. Karakter-karakter ini tidak mengandung informasi yang penting. Tabel 3.9

menunjukkan penghapusan angka 69 dan 328, tanda titik dan tanda *newline*.

Tabel 3.9 Penghapusan Karakter Tanda Baca dan Khusus

Masukan	Keluaran
Saat ini sebanyak 39 ruas jalan Tol telah dibangun dgn skema kerjasama pemerintah dan badan usaha dgn nilai Rp. 328 trilyun.\nKepastian penyelesaian jalan tol sangat penting karena sebelumnya sempat terhenti karena progres pembebasan lahan yg berlarut	Saat ini sebanyak ruas jalan Tol telah dibangun dgn skema kerjasama pemerintah dan badan usaha dgn nilai Rp trilyun Kepastian penyelesaian jalan tol sangat penting karena sebelumnya sempat terhenti karena progres pembebasan lahan yg berlarut

- Konversi *slangword* menjadi menjadi bentuk baku dan memperbaiki kesalahan penulisan. Tabel 3.10 menunjukkan penggantian kata yang ditulis tidak baku atau *slangword*, seperti sy diganti dengan saya, kpd diganti dengan kepada, yg diganti dengan yang.

Tabel 3.10 Konversi *Slangword*

Masukan	Keluaran
sy ingin mengucapkan selamat hebat kpd anak yg brhasil membangun Jalan Tol	saya ingin mengucapkan selamat hebat kepada anak yang berhasil membangun Jalan Tol

- Konversi semua karakter menjadi huruf kecil (*case folding*). Tabel 3.11 menunjukkan bahwa karakter pada *tweet* dirubah menjadi huruf kecil semua. Hal ini dilakukan untuk menghindari perbedaan huruf besar dan huruf kecil yang nantinya akan memperbesar dimensi.

Tabel 3.11 *Case Folding*

Masukan	Keluaran
saya ingin mengucapkan selamat hebat kepada anak yang berhasil membangun Jalan Tol	saya ingin mengucapkan selamat hebat kepada anak yang berhasil membangun jalan tol

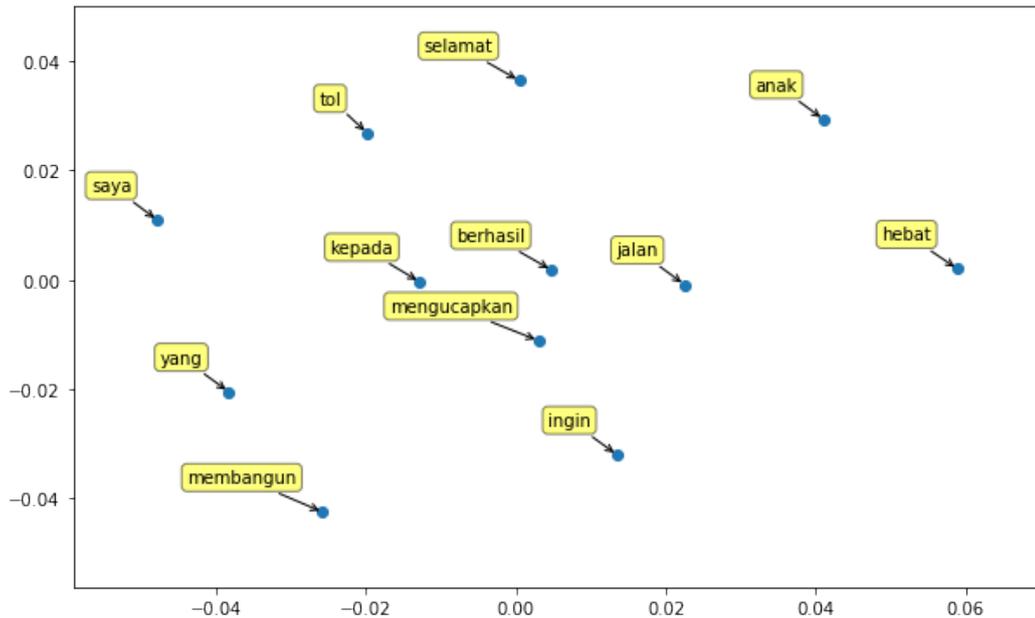
8. Penghapusan duplikasi *tweet* agar *tweet* yang diproses merupakan *tweet* yang unik.

3.7 Ekstraksi Fitur

3.7.1 Model *Word2vec*

Untuk membuat model *Word2vec* bahasa Indonesia diperlukan teks bahasa Indonesia dalam jumlah yang banyak. Cara paling mudah untuk mendapatkan teks tersebut dengan menggunakan korpus yang telah disediakan oleh Wikipedia bahasa Indonesia. Korpus Wikipedia disediakan secara terbuka dan biasa dipakai untuk kegiatan apapun sepanjang tidak melanggar lisensinya. Korpus tersebut dapat diunduh pada laman <https://dumps.wikimedia.org/idwiki/>. Hasil unduhan berupa berkas *idwiki-latest-pages-articles.xml.bz2*.

Pertama yang dilakukan adalah mengekstrak berkas *idwiki-latest-pages-articles.xml.bz2*, selanjutnya memproses hasil ekstrak dengan *preprocessing* kemudian hasilnya akan disimpan sebuah berkas teks. Proses selanjutnya adalah melatih *Word2vec* menggunakan jaringan syaraf tiruan. Dalam proses pelatihan, *Word2vec* menggunakan *Skip-Gram* karena korpus Wikipedia memiliki ukuran yang sangat besar. *Skip-Gram* menggunakan *range* antara kata-kata konteks dengan posisi *current word* sebanyak 5. Dimensi vektor yang dihasilkan *Word2vec* sebanyak 800. Setelah proses pelatihan *Word2vec* selesai, model disimpan dan digunakan lagi untuk melakukan transformasi dataset ke vektor. Gambar 3.2 menunjukkan plot vektor dari beberapa kata untuk melihat kedekatan dari beberapa kata setelah dilakukan transformasi ke vektor.



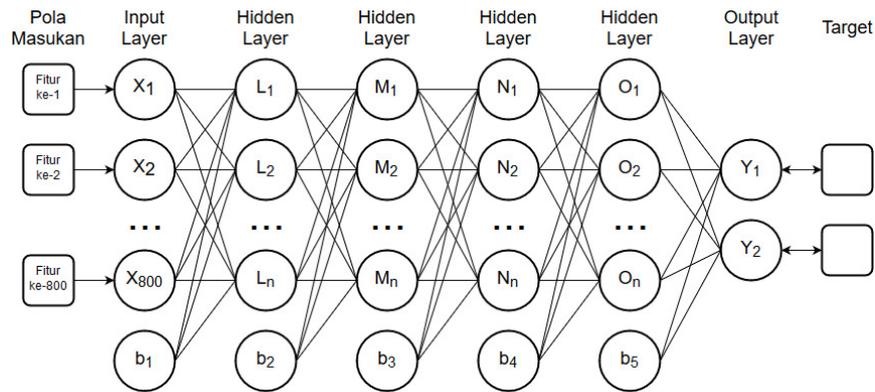
Gambar 3.2 Visualisasi Vektor dalam Bentuk 2 Dimensi

3.7.2 Transformasi Tweet ke Vektor

Setelah model *Word2vec* terbentuk, dataset pelatihan dan dataset pengujian berupa *tweet* hasil *preprocessing* ditransformasikan ke vektor menggunakan *Word2vec*. Setiap *tweet* yang mempunyai panjang berbeda-beda ditransformasi ke dalam vektor dengan dimensi 800 sehingga menjadi vektor numerik yang mempunyai panjang atau dimensi yang sama. Hasil dari transformasi adalah kumpulan vektor numerik dengan dimensi 800 dari setiap *tweet* pada dataset pelatihan dan dataset pengujian akan digunakan sebagai masukan atau *input* untuk pelatihan maupun pengujian model klasifikasi *Multilayer Perceptron*.

3.8 Pelatihan dan Pengujian Multilayer Perceptron

Pada tahap ini dilakukan proses pelatihan dan pengujian *Multilayer Perceptron*. Pada proses pelatihan digunakan untuk penentuan parameter dan membuat model klasifikasi *Multilayer Perceptron*. Proses pengujian digunakan untuk menguji kinerja dari model klasifikasi *Multilayer Perceptron* yang dibuat pada proses pelatihan. Secara visual arsitektur *Multilayer Perceptron* ditunjukkan pada Gambar 3.3.



Gambar 3.3 Arsitektur *Multilayer Perceptron*

3.8.1 Pelatihan *Multilayer Perceptron*

Pada tahap pelatihan *Multilayer Perceptron* yang pertama kali dilakukan adalah pelatihan untuk menentukan parameter. Setelah parameter ditentukan selanjutnya dilakukan pelatihan untuk membuat model klasifikasi *Multilayer Perceptron*.

3.8.1.1 Penentuan Parameter

Proses pelatihan digunakan untuk menentukan parameter antara lain :

- a. Jumlah *layer* pada *hidden layer*. Penentuan jumlah *layer* pada *hidden layer* dilakukan dengan menggunakan uji coba. Untuk uji coba pertama kali, jumlah *layer* ditentukan sebanyak 2 *layer* pada *hidden layer* [28]. Uji coba selanjutnya, *layer* pada *hidden layer* jumlahnya ditambah menjadi 4 *layer* dan 6 *layer*. Uji coba pelatihan dilakukan dengan menggunakan dataset pelatihan sedangkan untuk validasi menggunakan dataset pengujian. Hasil uji coba yang memiliki akurasi validasi tertinggi yang diambil sebagai parameter jumlah *layer* pada *hidden layer* dan digunakan untuk penentuan parameter selanjutnya yaitu penentuan jumlah neuron pada masing-masing *hidden layer*.
- b. Jumlah neuron pada masing-masing *hidden layer*. Setelah parameter jumlah *layer* ditentukan, selanjutnya penentuan jumlah neuron pada masing-masing *hidden layer* dilakukan dengan menggunakan uji coba. Sebagai acuan, jumlah neuron pada *layer* pertama *hidden layer* paling banyak menggunakan $\frac{2}{3}$ dari neuron *input layer* dan jumlah neuron pada *layer* selanjutnya menggunakan $\frac{1}{2}$ dari jumlah neuron pada *layer* sebelumnya [29]. Uji coba pelatihan dilakukan

dengan menggunakan dataset pelatihan sedangkan untuk validasi menggunakan dataset pengujian. Hasil uji coba yang memiliki akurasi validasi tertinggi yang diambil sebagai parameter jumlah neuron pada masing-masing *hidden layer* dan digunakan untuk penentuan parameter selanjutnya yaitu penentuan fungsi aktivasi pada masing-masing *hidden layer*.

- c. Fungsi aktivasi pada masing-masing *hidden layer*. Setelah parameter jumlah *layer* pada *hidden layer* dan jumlah neuron pada masing-masing *hidden layer* ditentukan, selanjutnya fungsi aktivasi ditentukan melalui ujicoba fungsi aktivasi *Sigmoid* dan *Rectified Linear Unit*. Uji coba pelatihan dilakukan dengan menggunakan dataset pelatihan sedangkan untuk validasi menggunakan dataset pengujian. Hasil uji coba yang memiliki akurasi validasi tertinggi yang diambil sebagai parameter fungsi aktivasi pada masing-masing *hidden layer*.
- d. Fungsi *loss/cost*. Penentuan fungsi *loss* dilakukan menggunakan uji coba fungsi *Mean Square Error* dan *Cross-Entropy*. Ujicoba pelatihan dilakukan dengan menggunakan dataset pelatihan sedangkan untuk validasi menggunakan dataset pengujian. Hasil ujicoba yang memiliki akurasi validasi tertinggi yang diambil sebagai parameter fungsi *loss*.
- e. Algoritme pelatihan dan *learning rate*.

Parameter lainnya yang ditentukan tanpa melalui proses pelatihan diantaranya :

- a. Jumlah neuron pada *input layer* sesuai dengan dimensi vektor dari ekstraksi fitur *Word2vec*.
- b. Jumlah neuron pada *output layer* sesuai dengan kelas atau label.
- c. Iterasi proses pelatihan ditentukan sebanyak 10.000
- d. *Batch size* ditentukan sebesar 100.

3.8.1.2 Pembuatan Model Klasifikasi

Pada proses pembuatan model klasifikasi *Multilayer Perceptron* terdapat beberapa parameter yang harus didefinisikan antara lain,

- a. *Input Layer*

Input layer mewakili secara langsung vektor fitur numerik dari input. Jumlah neuronnya menyesuaikan dari jumlah fitur yang dihasilkan pada proses ekstraksi fitur. Banyaknya jumlah neuron menyesuaikan dimensi vektor pada *Word2vec*. Dataset yang ditransformasikan ke vektor menggunakan *Word2vec* dimasukkan ke *input layer*.

Proses yang dilakukan pertama kali pada *input layer* berupa inisialisasi bobot dan bias. Inisialisasi bobot yang menghubungkan *input layer* dengan lapisan pertama *hidden layer* menggunakan metode pengacakan yang dibatasi dengan *mean* sebesar 0 dan standar deviasi sebesar 0,1. Sedangkan bobot yang menghubungkan bias dengan lapisan pertama *hidden layer* diinisialisasi menggunakan nilai 0. Proses ini inisialisasi membangkitkan nilai acak sebanyak jumlah neuron *input layer* dikalikan dengan jumlah neuron lapisan pertama *hidden layer* dan bobot bias menghubungkan bias dengan lapisan pertama *hidden layer* sebanyak jumlah neuron pada lapisan pertama *hidden layer*. Dimana jumlah neuron lapisan Pada *input layer* yang terhubung dengan lapisan pertama *hidden layer* terdapat penghitungan perkalian antara nilai input dari nilai fitur dengan nilai bobot serta penambahan dengan nilai bias dan hasilnya akan diproses di setiap neuron pada lapisan pertama *hidden layer*. Bobot dan bias akan diperbarui secara terus menerus menggunakan optimisasi algoritme pelatihan dengan meminimalkan nilai *loss*. Pembaruan dilakukan sebanyak iterasi yang ditentukan.

b. *Hidden Layer*

Hidden layer adalah layer tersembunyi yang terletak antara *input layer* dan *output layer*. Terdiri dari neuron-neuron saling terhubung dengan neuron pada *layer* sebelumnya dan setelahnya. Tidak ada ketentuan terkait jumlah neuron pada setiap *hidden layer*. Pada setiap lapisan akan menerapkan fungsi aktivasi untuk menghitung nilai keluaran dari setiap neuronnya. Nilai ini akan menjadi nilai masukan bagi lapisan selanjutnya.

Penentuan parameter jumlah neuron pada *hidden layer* dilakukan dengan cara uji coba *trial and error*. Uji coba dengan melakukan pelatihan pada *Multilayer Perceptron* dengan jumlah neuron pada *hidden layer* secara acak untuk mencari nilai *loss* terkecil.

Penentuan parameter fungsi aktivasi pada *hidden layer* juga dilakukan dengan cara uji coba *trial and error*. Uji coba dengan melakukan pelatihan pada *Multilayer Perceptron* dengan membandingkan fungsi aktivasi *Sigmoid* dan fungsi aktivasi *Rectified Linear Unit* pada setiap *hidden layer* untuk mencari nilai *loss* terkecil.

Proses inisialisasi dilakukan pada setiap lapisan pada *hidden layer*. Inisialisasi bobot dan bias yang menghubungkan antara lapisan pertama dan lapisan kedua *hidden layer* menggunakan metode pengacakan yang dibatasi dengan *mean* sebesar 0 dan standar deviasi sebesar 0,1. Proses inisialisasi ini membangkitkan nilai acak sebanyak jumlah neuron lapisan pertama *hidden layer* dikalikan dengan jumlah neuron lapisan kedua *hidden layer*) dan bobot bias menghubungkan bias dengan lapisan kedua *hidden layer* sebanyak jumlah neuron lapisan kedua *hidden layer*. Hasil perkalian antara nilai *input layer* dan bobot serta penambahan bias diproses menggunakan fungsi aktivasi pada lapisan pertama *hidden layer*. Nilai keluaran fungsi aktivasi akan dikalikan dengan nilai bobot yang telah diinisialisasi serta ditambahkan dengan bias. Hasilnya akan diproses pada lapisan kedua *hidden layer*. Bobot dan bias akan diperbarui secara terus menerus menggunakan optimisasi algoritme pelatihan dengan meminimalkan nilai *loss*. Pembaruan dilakukan sebanyak iterasi yang ditentukan.

Proses inisialisasi juga dilakukan terhadap bobot dan bias yang menghubungkan layer kedua *hidden layer* dan lapisan ketiga *hidden layer*. Inisialisasi bobot menggunakan metode pengacakan dengan dibatasi *mean* sebesar 0 dan standar deviasi sebesar 0,1. Inisialisasi bias menggunakan nilai 0. Proses inisialisasi bobot yang menghubungkan lapisan kedua dan ketiga *hidden layer* membangkitkan nilai acak sebanyak jumlah neuron lapisan kedua *hidden layer* dikalikan dengan jumlah neuron lapisan ketiga *hidden layer* dan bobot yang menghubungkan bias dengan lapisan ketiga *hidden layer* sebanyak jumlah neuron lapisan ketiga *hidden layer*. Pada lapisan kedua *hidden layer* terdapat proses aktivasi menggunakan fungsi aktivasi yang mana nilai masukannya adalah hasil perkalian nilai pada lapisan pertama *hidden layer* dikalikan dengan bobot penghubung serta penambahan bias. Nilai keluaran dari fungsi aktivasi pada lapisan kedua *hidden layer* juga akan dikalikan dengan bobot dan ditambahkan dengan bias

yang telah diinisialisasi. Hasilnya akan di proses menggunakan fungsi aktivasi pada lapisan ketiga *hidden layer*. Bobot dan bias antara lapisan kedua dan ketiga *hidden layer* akan diperbarui secara terus menerus menggunakan optimisasi algoritme pelatihan dengan meminimalkan nilai *loss*. Pembaruan dilakukan sebanyak iterasi yang ditentukan.

Bobot yang menghubungkan lapisan ketiga *hidden layer* dan lapisan keempat *hidden layer* juga dilakukan inisialisasi menggunakan metode pengacakan yang dibatasi dengan *mean* sebesar 0 dan standar deviasi 0,1. Sedangkan bias diinisialisasi dengan nilai 0. Proses inisialisasi bobot yang menghubungkan lapisan ketiga dan keempat *hidden layer* membangkitkan nilai acak sebanyak jumlah neuron lapisan ketiga *hidden layer* dikalikan dengan jumlah neuron lapisan keempat *hidden layer* dan bobot bias yang menghubungkan bias dengan lapisan keempat *hidden layer* sebanyak jumlah neuron lapisan keempat *hidden layer*. Pada lapisan ketiga *hidden layer* terdapat proses aktivasi menggunakan fungsi aktivasi yang mana nilai masukannya adalah hasil perkalian nilai pada lapisan kedua *hidden layer* dikalikan dengan bobot penghubung serta penambahan bias. Nilai keluaran dari fungsi aktivasi pada lapisan ketiga *hidden layer* juga akan dikalikan dengan bobot dan ditambahkan dengan bias yang telah diinisialisasi. Hasilnya akan di proses menggunakan fungsi aktivasi pada lapisan keempat *hidden layer*. Bobot dan bias antara lapisan ketiga dan keempat *hidden layer* akan diperbarui secara terus menerus menggunakan optimisasi algoritme pelatihan dengan meminimalkan nilai *loss*. Pembaruan dilakukan sebanyak iterasi yang ditentukan.

Pada lapisan keempat *hidden layer* terdapat proses aktivasi menggunakan fungsi aktivasi yang mana nilai masukannya adalah hasil perkalian nilai pada lapisan ketiga *hidden layer* dikalikan dengan bobot penghubung serta penambahan bias. Nilai keluaran dari fungsi aktivasi pada lapisan keempat *hidden layer* juga akan dikalikan dengan bobot dan ditambahkan dengan bias yang telah diinisialisasi antara lapisan keempat *hidden layer* dengan *output layer*.

c. *Output Layer*

Output layer terdiri dari dua neuron. Dua neuron ini mewakili dari dua label yakni label positif dan label negatif. Dua neuron juga merupakan representasi vektor 1×2 dari label yang dihasilkan oleh *One Hot Encoding*. Label negatif

menghasilkan vektor $[1 \ 0]$ dan label positif menghasilkan vektor $[0 \ 1]$. *Layer* ini terhubung secara penuh dengan *layer* sebelumnya. *Output layer* merupakan *softmax layer* sehingga jika nilai aktivasinya pada masing-masing neuron ditotal bernilai sama dengan 1.

Proses inisialisasi dilakukan terhadap bobot yang menghubungkan antara lapisan keempat *hidden layer* dan output layer menggunakan metode pengacakan yang dibatasi dengan *mean* sebesar 0 dan standar deviasi sebesar 0,1. Sedangkan bias diinisialisasi dengan nilai 0. Proses inisialisasi bobot yang menghubungkan lapisan keempat *hidden layer* dan *output layer* membangkitkan nilai acak sebanyak jumlah neuron lapisan keempat *hidden layer* dikalikan dengan jumlah neuron *output layer* dan bobot bias yang menghubungkan bias dengan *output layer* sebanyak jumlah neuron *output layer*. Pada *output layer* terdapat perkalian antara nilai lapisan keempat *hidden layer* dengan bobot dan ditambahkan bias yang telah diinisialisasi. Hasilnya akan diproses menggunakan fungsi *softmax*. Bobot dan bias antara lapisan keempat *hidden layer* dan *output layer* akan diperbarui secara terus menerus menggunakan optimisasi algoritme pelatihan dengan meminimalkan nilai *loss*. Pembaruan dilakukan sebanyak iterasi yang ditentukan

d. Fungsi Biaya

Fungsi biaya atau *loss* digunakan untuk menghitung jarak antara hasil penghitungan pada *output layer* dengan target. Fungsi *loss* yang digunakan adalah *Mean Square Error* dan *Cross-Entropy*. Pada setiap iterasi, *loss* selalu dihitung dan dievaluasi untuk mencari *loss* terkecil. Algoritme pelatihan untuk melakukan optimisasi dengan meminimalkan nilai *loss* yang nantinya digunakan untuk memperbarui semua bobot dan bias pada *Multilayer Perceptron* dengan bantuan *learning rate*.

e. Algoritme Pelatihan dan *Learning Rate*

Algoritme pelatihan yang digunakan untuk penelitian ini adalah dengan *Gradient Descent* [30] dan *Adam* [31]. Seluruh dataset pelatihan akan langsung diterima oleh *Multilayer Perceptron* untuk dihitung nilai erornya dan dilakukan pembaruan bobot dan bias. Proses *gradient descent* [30], dimulai dari menghitung biaya lalu turunannya hingga memperbarui bobot, dilakukan berulang-ulang

sebanyak iterasi yang ditentukan sebelumnya yang diharapkan cukup untuk memberikan nilai *loss* paling kecil.

Algoritme *Adam* [31] bekerja dengan baik di pelbagai arsitektur *deep learning*. Algoritma ini merupakan kombinasi *gradient descent* dengan momentum dan *adaptive learning*. Proses algoritme *Adam* dimulai dengan menghitung rata-rata eksponensial dari gradien dan eksponensial kuadrat dari gradien. Kemudian diperbarui parameternya dengan menggabungkan informasi tersebut.

Dengan menggunakan *learning rate* yang tetap dengan nilai yang cukup kecil yang dipadukan dengan ukuran *batch* yang kecil, *loss* maupun *gradient* dapat konvergen di titik nol [32]. Proses pelatihan dilakukan untuk membuat suatu model klasifikasi yang dapat digunakan untuk memprediksi atau mengklasifikasi data baru yang belum pernah dilatih sesuai target yang diharapkan. *Multilayer Perceptron* menggunakan dataset pelatihan selama proses pelatihan. Pelatihan *Multilayer Perceptron* dilakukan menggunakan empat algoritme pelatihan yaitu :

1. *Gradient Descent* dengan *learning rate* 0,01
2. *Gradient Descent* dengan *learning rate* 0,001
3. *Adam* dengan *learning rate* 0,01
4. *Adam* dengan *learning rate* 0,001

3.8.2 Pengujian Multilayer Perceptron

Model klasifikasi hasil pelatihan menggunakan empat algoritme pelatihan diuji untuk mengukur kinerja. Pengujian dilakukan menggunakan dataset pelatihan untuk mengetahui kinerja model terhadap dataset yang digunakan untuk melatih model tersebut. Pengujian juga dilakukan menggunakan dataset pengujian yang sejak awal dipisahkan pada proses pembagian dataset dan dataset pengujian tidak pernah digunakan pada proses pelatihan untuk mengetahui kinerja model terhadap data baru. Hasil pengujian selanjutnya dilakukan analisis.

3.9 Analisis Hasil

Hasil pengujian dianalisis untuk mengukur kinerja *Multilayer Perceptron* dalam merespon setiap pola masukan yang diberikan terhadap target yang diharapkan. Analisis hasil pengujian antara lain :

1. Menganalisis *Multilayer Perceptron* dalam meminimalkan nilai kesalahan antara keluaran dengan target yang diharapkan.
2. Menganalisis kinerja *Multilayer Perceptron* dalam mengklasifikasikan dataset pelatihan dan dataset pengujian dalam *confusion matrix* dengan cara menghitung,
 - a. Seberapa banyak target positif yang terprediksi positif dari tiap kelas atau label dataset;
 - b. Seberapa banyak target positif yang terprediksi negatif dari tiap kelas atau label dataset;
 - c. Seberapa banyak target negatif yang terprediksi positif dari tiap kelas atau label dataset;
 - d. Seberapa banyak target negatif yang terprediksi negatif dari tiap kelas atau label dataset.
3. Menganalisis kinerja akurasi, *precision* dan *recall Multilayer Perceptron* berdasarkan hasil klasifikasi dataset pelatihan dan dataset pengujian.

3.10 Analisis Sentimen Seluruh Data

Setelah model *Multilayer Perceptron* menghasilkan kinerja tertinggi yang didapatkan dari hasil analisis pelatihan dan pengujian, langkah selanjutnya adalah melakukan analisis sentimen terhadap keseluruhan data yang diperoleh pada tahap pengumpulan data menggunakan *Multilayer Perceptron*. Hasil dari analisis sentimen akan ditampilkan dalam bentuk visual grafik.

Halaman ini sengaja dikosongkan

BAB 4

HASIL DAN PEMBAHASAN

Tahap ini merupakan pembahasan hasil dari metode penelitian dan pengujian untuk mengetahui kinerja model klasifikasi. Pengujian model klasifikasi digunakan untuk mengukur kinerja akurasi, *precision* dan *recall*. Berdasarkan hasil eksperimen dan pengujian maka dapat diketahui akurasi, *precision* dan *recall* dari setiap model klasifikasi.

Pada proses pelatihan dan pengujian, dataset yang digunakan sebanyak 6.814 *tweet* yang dibagi dalam dataset pelatihan sebanyak 6.132 *tweet* dan dataset pengujian sebanyak 682 *tweet* dengan distribusi yang ditunjukkan pada Tabel 4.1

Tabel 4.1 Distribusi Pembagian Dataset

Dataset	Jumlah	Kelas / Label	Jumlah Data
Dataset pelatihan	6.132	Positif	3.057
		Negatif	3.075
Dataset pengujian	682	Positif	350
		Negatif	332

4.1 Penentuan Parameter

4.1.1 Jumlah Layer pada Hidden Layer

Penentuan parameter jumlah *layer* pada *hidden layer* dilakukan dengan cara uji coba *trial and error*. Uji coba dilakukan melalui pelatihan pada *Multilayer Perceptron* dengan jumlah *layer* pada *hidden layer* antara 2, 4 dan 6 *layer*. Pelatihan *Multilayer Perceptron* menggunakan dataset pelatihan sedangkan untuk validasi menggunakan dataset pengujian pada Tabel 4.1. Jumlah neuron pada *input layer* 800, jumlah neuron *output layer* 2, fungsi aktivasi *Sigmoid*, fungsi *loss* adalah *cross-entropy* dan algoritme pelatihan *Gradient Descent learning rate* 0,01. Uji coba yang pertama menggunakan jumlah *layer* pada *hidden layer* sebanyak 2, nilai akurasi terhadap data validasi yang diperoleh sebesar 0,4838. Pada uji coba yang kedua dengan jumlah *layer* pada *hidden layer* sebanyak 4, nilai akurasi terhadap data validasi yang diperoleh sebesar 0,5131. Pada uji coba yang ketiga dengan jumlah

layer pada *hidden layer* sebanyak 6, nilai akurasi terhadap data validasi yang diperoleh sebesar 0,5023. Hasil akurasi terhadap data validasi dari ketiga uji coba ditunjukkan pada Tabel 4.2. Dari ketiga uji coba, nilai akurasi terhadap data validasi tertinggi diperoleh dari uji coba dengan jumlah *layer* pada *hidden layer* sebanyak 4 *layer*. Jumlah *layer* 4 dipilih sebagai parameter jumlah *layer* pada *hidden layer*.

Tabel 4.2 Hasil Validasi Uji Coba Jumlah *Layer* pada *Hidden Layer*

Jumlah Layer pada Hidden Layer	Akurasi
<i>2 layer</i>	0,4838
<i>4 layer</i>	0,5131
<i>6 layer</i>	0,5023

4.1.2 Jumlah Neuron pada Hidden Layer

Penentuan parameter jumlah neuron pada *hidden layer* dilakukan dengan cara uji coba *trial and error*. Jumlah lapisan pada *hidden layer* sebanyak 4 yang diperoleh dari penentuan parameter jumlah *layer* pada *hidden layer*. Sebagai acuan, jumlah neuron pada lapisan pertama *hidden layer* paling banyak $\frac{2}{3}$ dari jumlah neuron pada *input layer* yaitu sebanyak 600. Uji coba dilakukan melalui pelatihan pada *Multilayer Perceptron* dengan jumlah neuron pada *hidden layer* secara acak antara 200 sampai dengan 600. Jumlah neuron pada *input layer* 800, jumlah neuron *output layer* 2, fungsi aktivasi *Sigmoid*, fungsi *loss* adalah *cross-entropy*, iterasi pelatihan sebanyak 10.000 dan algoritme pelatihan *Gradient Descent learning rate* 0,01. Pelatihan *Multilayer Perceptron* menggunakan dataset pelatihan sedangkan untuk validasi menggunakan dataset pengujian pada Tabel 4.1. Uji coba yang pertama dengan jumlah neuron 200, 100, 50, 25, diperoleh nilai akurasi terhadap data validasi sebesar 0,5507. Uji coba yang kedua dengan jumlah neuron 300, 150, 75, 37 diperoleh nilai akurasi terhadap data validasi sebesar 0,5346. Uji coba yang ketiga dengan jumlah neuron 400, 200, 100, 50 diperoleh nilai akurasi terhadap data validasi sebesar 0,5492. Uji coba yang keempat dengan jumlah neuron 500, 150, 125, 62 diperoleh nilai akurasi terhadap data validasi sebesar 0,5258. Uji coba yang kelima dengan jumlah neuron 600, 300, 150, 75 diperoleh nilai akurasi terhadap data validasi sebesar 0,5463. Hasil akurasi terhadap data validasi ketiga uji coba ditunjukkan pada Tabel 4.3. Dari kelima uji coba, nilai akurasi terhadap data

validasi tertinggi sebesar 0,5507 diperoleh oleh jumlah neuron 200, 100, 50, 25. Jumlah neuron 200, 100, 50, 25 ini dipilih sebagai parameter jumlah neuron pada *hidden layer*.

Tabel 4.3 Hasil Validasi Uji Coba Jumlah Neuron pada *Hidden Layer*

Jumlah Neuron pada Hidden Layer	Akurasi
200,100,50,25	0,5507
300,150,75,37	0,5346
400,200,100,50	0,5492
500,250,125,62	0,5258
600,300,150,75	0,5463

4.1.3 Fungsi Aktivasi pada Hidden Layer

Penentuan parameter fungsi aktivasi pada *hidden layer* juga dilakukan dengan cara uji coba *trial and error*. Uji coba dilakukan melalui pelatihan pada *Multilayer Perceptron* dengan jumlah neuron pada *hidden layer* masing-masing sebesar 200, 100, 50, 25 yang diperoleh dari penentuan parameter jumlah neuron pada *hidden layer*. Jumlah neuron pada *input layer* 800, jumlah neuron *output layer* 2, fungsi *loss* adalah *cross-entropy*, iterasi pelatihan sebanyak 10.000 dan algoritme pelatihan *Gradient Descent learning rate* 0,01. Pelatihan *Multilayer Perceptron* menggunakan dataset pelatihan sedangkan untuk validasi menggunakan dataset pengujian pada Tabel 4.1. Uji coba fungsi aktivasi yang pertama menggunakan *Sigmoid*. Pada uji coba fungsi aktivasi *Sigmoid*, nilai akurasi terhadap data validasi yang diperoleh sebesar 0,5507. Uji coba fungsi aktivasi yang kedua menggunakan *Rectified Linear Unit*. Pada uji coba fungsi aktivasi *Rectified Linear Unit*, nilai akurasi terhadap data validasi yang diperoleh sebesar 0,5982. Hasil akurasi terhadap data validasi dari kedua uji coba ditunjukkan pada Tabel 4.4. Dari kedua uji coba fungsi aktivasi, nilai akurasi terhadap data validasi tertinggi sebesar 0,5982 yang diperoleh oleh fungsi aktivasi *Rectified Linear Unit*. *Rectified Linear Unit* dipilih sebagai parameter fungsi aktivasi pada *hidden layer*.

Tabel 4.4 Hasil Validasi Uji Coba Fungsi Aktivasi pada *Hidden Layer*

Fungsi Aktivasi pada Hidden Layer	Akurasi
<i>Sigmoid</i>	0,5507
<i>Rectified Linear Unit</i>	0,5982

4.1.4 Fungsi Loss

Penentuan parameter fungsi *loss* pada *Multilayer Perceptron* dilakukan dengan cara uji coba *trial and error*. Uji coba dilakukan melalui pelatihan pada *Multilayer Perceptron* dengan jumlah neuron pada *hidden layer* masing-masing sebesar 200, 100, 50, 25 yang diperoleh dari penentuan parameter jumlah neuron pada *hidden layer*. Jumlah neuron pada *input layer* 800, jumlah *neuron output layer* 2, fungsi aktivasi menggunakan *Rectified Linear Unit* yang diperoleh dari penentuan parameter fungsi aktivasi pada *hidden layer*, iterasi pelatihan sebanyak 10.000 dan algoritme pelatihan *Gradient Descent learning rate* 0,01. Pelatihan *Multilayer Perceptron* menggunakan dataset pelatihan sedangkan untuk validasi menggunakan dataset pengujian pada Tabel 4.1. Uji coba fungsi *loss* yang pertama menggunakan *Mean Square Error*. Pada uji coba fungsi *loss Mean Square Error*, nilai akurasi terhadap data validasi yang diperoleh sebesar 0,5703. Uji coba fungsi *loss* yang kedua menggunakan *Root Mean Square Error*. Pada uji coba fungsi *loss Root Mean Square Error*, nilai akurasi terhadap data validasi yang diperoleh sebesar 0,5733. Uji coba fungsi *loss* yang ketiga menggunakan *Cross-Entropy*. Pada uji coba fungsi *loss Cross-Entropy*, nilai akurasi terhadap data validasi yang diperoleh sebesar 0,6129. Hasil akurasi terhadap data validasi dari ketiga uji coba ditunjukkan pada Tabel 4.5. Dari ketiga uji coba fungsi *loss*, nilai akurasi terhadap data validasi tertinggi sebesar 0,6129 yang diperoleh oleh fungsi *loss Cross-Entropy*. *Cross-Entropy* dipilih sebagai parameter fungsi *loss* pada *Multilayer Perceptron*.

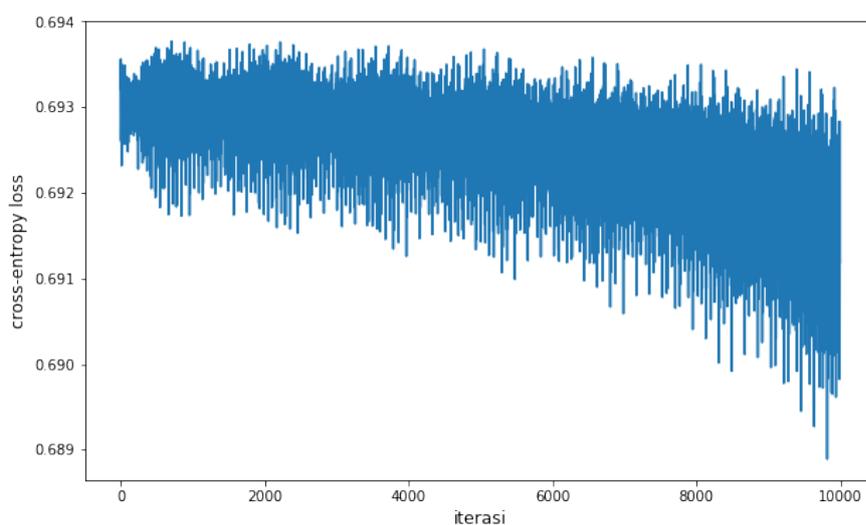
Tabel 4.5 Hasil Validasi Uji Coba Fungsi Loss

Fungsi Loss	Akurasi
<i>Mean Square Error</i>	0,5703
<i>Root Mean Square Error</i>	0,5733
<i>Cross-Entropy</i>	0,6129

4.2 Multilayer Perceptron Algoritme Pelatihan Gradient Descent Learning Rate 0,01

Pelatihan dilakukan dengan menggunakan algoritme pelatihan *Gradient Descent* dengan *learning rate* 0,01, *batch size* 100 dan iterasi 10.000. Proses

pelatihan *Multilayer Perceptron* menggunakan dataset pelatihan sebagaimana yang ditunjukkan pada Tabel 4.1. *Cross-Entropy Loss* pelatihan *Multilayer Perceptron* algoritme *Gradient Descent* *learning rate* 0,01 ditampilkan pada Gambar 4.1. Pada iterasi pertama proses pelatihan, *cross-entropy loss* yang didapatkan sebesar 0,6933029. Pada iterasi ke-2.000, *cross-entropy loss* yang didapatkan juga relatif sama sekitar 0,6933. Begitu juga pada iterasi ke-4.000 sampai dengan iterasi ke-10.000 cenderung mengalami penurunan. Penurunan *cross-entropy loss* rata-rata 0,003. Nilai *cross-entropy loss* yang didapatkan pada akhir proses pelatihan sebesar 0,69118667.



Gambar 4.1 Grafik *Cross-Entropy Loss* Pelatihan Algoritme *Gradient Descent* *Learning Rate* 0,01

Proses pelatihan menggunakan algoritme pelatihan *Gradient Descent* dengan *learning rate* 0,01 menghasilkan sebuah model klasifikasi. Dataset pelatihan dan dataset pengujian diklasifikasi menggunakan model yang dihasilkan pada proses pelatihan. Dengan begitu dapat dianalisis seberapa baik *Multilayer Perceptron* sebagai *classifier* mengenali beberapa data pada kelas atau label yang berbeda dengan menggunakan *confusion matrix*. *Confusion matrix* kelas atau label target klasifikasi terhadap prediksi dengan dataset pelatihan ditunjukkan pada Tabel 4.6. Dari dataset pelatihan dengan kelas target negatif sebanyak 3.075 berhasil diprediksi tepat pada kelas negatif sebanyak 2.307 dan kelas target positif sebanyak 3.057 berhasil diprediksi tepat pada kelas positif sebanyak 1.447.

Tabel 4.6 *Confusion Matrix* Pengujian dengan Dataset Pelatihan Algoritme *Gradient Descent Learning Rate 0,01*

		Prediksi		Total Target
		Neg	Pos	
Target	Neg	2307	768	3075
	Pos	1610	1447	3057
Total Prediksi		3917	2215	6132

Dari *confusion matrix* pada Tabel 4.6, dapat diukur kinerja akurasi, *precision* dan *recall* dari model klasifikasi. Pengukuran kinerja *Multilayer Perceptron* dengan algoritme *Gradient Descent Learning Rate 0,01* ditunjukkan pada Tabel 4.7. Akurasi, *precision* dan *recall* rata-rata yang didapatkan dari dataset pelatihan masing sebesar 0,6118, 0,6183 dan 0,6115.

Tabel 4.7 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pelatihan Algoritme *Gradient Descent Learning Rate 0,01*

	Label		Avg
	Neg	Pos	
TP	2301	1447	
TN	1447	2301	
FP	1610	768	
FN	768	1610	
TPR = Sensitivity	0,7498	0,4733	
TNR = Specificity	0,4733	0,7498	
Recall	0,7498	0,4733	0,6115
Precision	0,5883	0,6533	0,6183
Akurasi	0,6118	0,6118	0,6118

Model klasifikasi *Multilayer Perceptron* dengan algoritme *Gradient Descent learning rate 0,01* juga dilakukan pengujian dengan dataset pengujian yang dari awal dipisahkan dan tidak diikutkan selama proses pelatihan. *Confusion matrix* kelas atau label target klasifikasi terhadap prediksi dengan dataset pengujian ditunjukkan pada Tabel 4.8. Dari total 682 dataset pengujian dengan kelas target negatif sebanyak 332 berhasil diprediksi tepat pada kelas negatif sebanyak 250 dan kelas target positif sebanyak 350 berhasil diprediksi tepat pada kelas positif sebanyak 145.

Tabel 4.8 *Confusion Matrix* Pengujian dengan Dataset Pengujian Algoritme *Gradient Descent Learning Rate 0,01*

		Prediksi		Total Target
		Neg	Pos	
Target	Neg	250	82	332
	Pos	205	145	350
Total Prediksi		455	227	682

Dari *confusion matrix* pada Tabel 4.8 dapat diukur kinerja akurasi, *precision* dan *recall* dari model klasifikasi yang diuji menggunakan data pengujian. Pengukuran kinerja hasil pengujian *Multilayer Perceptron* algoritme *Gradient Descent Learning Rate 0,01* ditunjukkan pada Tabel 4.9. Berdasarkan hasil pengujian *Multilayer Perceptron* algoritme *Gradient Descent Learning Rate 0,01* lebih dapat memprediksi kelas negatif dari pada kelas positif berdasarkan nilai *recall* kelas negatif sebesar 0,7530 sedangkan *recall* kelas positif hanya sebesar 0,4143. Akurasi rata-rata yang didapatkan sebesar 0,5792.

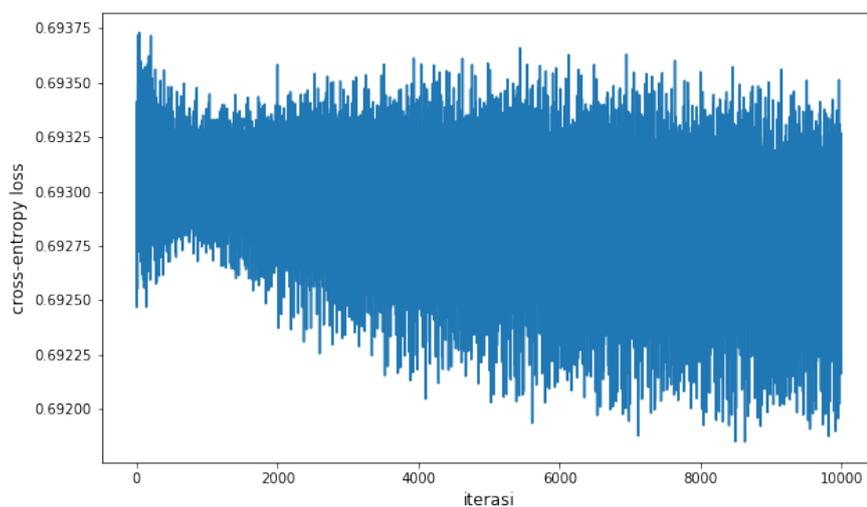
Tabel 4.9 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pengujian Algoritme *Gradient Descent Learning Rate 0,01*

	Label		Avg
	Neg	Pos	
TP	250	145	
TN	145	250	
FP	205	82	
FN	82	205	
TPR = Sensitivity	0,7530	0,4143	
TNR = Specificity	0,4143	0,7530	
Recall	0,7530	0,4143	0,5836
Precision	0,5495	0,6388	0,5941
Akurasi	0,5792	0,5792	0,5792

4.3 Multilayer Perceptron Algoritme Pelatihan Gradient Descent Learning Rate 0,001

Pelatihan dilakukan dengan menggunakan algoritme Gradient Descent dengan learning rate 0,001, batch size 100 dan iterasi 10.000. Proses pelatihan *Multilayer Perceptron* menggunakan dataset pelatihan sebagaimana yang ditunjukkan pada Tabel 4.1. *Cross-Entropy Loss* pelatihan *Multilayer Perceptron*

algoritme *Gradient Descent* *learning rate* 0,001 ditampilkan pada Gambar 4.2. Pada iterasi pertama proses pelatihan, *cross-entropy loss* yang didapatkan sebesar 0,6933375. Pada iterasi ke-2.000, *cross-entropy loss* yang didapatkan mengalami fluktuasi antara 0,69250 dan 0,69350. Pada iterasi ke-4.000, fluktuasi *cross-entropy loss* semakin melebar antara 0,69255 dan 0,69360. Pada iterasi ke-6.000, ke-8.000 dan ke-10.000, fluktuasi *cross-entropy loss* melebar antara 0,69160 dan 0,69350. Nilai *cross-entropy loss* yang didapatkan pada akhir proses pelatihan sebesar 0,6928074.



Gambar 4.2 Grafik *Cross-Entropy Loss* Pelatihan Algoritme *Gradient Descent* *Learning Rate* 0,001

Proses pelatihan menggunakan algoritme *Gradient Descent* dengan *learning rate* 0,001 menghasilkan sebuah model klasifikasi. Dataset pelatihan dan dataset pengujian diklasifikasi atau diprediksi menggunakan model yang dihasilkan pada proses pelatihan. Dengan begitu dapat dianalisis seberapa baik *Multilayer Perceptron* sebagai *classifier* mengenali beberapa data pada kelas atau label yang berbeda dengan menggunakan *confusion matrix*. *Confusion matrix* kelas atau label target klasifikasi terhadap prediksi dengan dataset pelatihan ditunjukkan pada Tabel 4.10. Dari 6.132 dataset pelatihan dengan kelas target negatif sebanyak 3.075 berhasil diprediksi tepat pada kelas negatif sebanyak 2.887 dan eror sebanyak 188. Kelas target positif sebanyak 3.057 berhasil diprediksi tepat pada kelas positif sebanyak 388 dan eror sebanyak 2.669.

Tabel 4.10 *Confusion Matrix* Pengujian dengan Dataset Pelatihan Algoritme *Gradient Descent Learning Rate 0,001*

		Prediksi		Total Target
		Neg	Pos	
Target	Neg	2887	188	3075
	Pos	2669	388	3057
Total Prediksi		5556	576	6132

Dari *confusion matrix* pada Tabel 4.10, dapat diukur kinerja akurasi, *precision* dan *recall*. Pengukuran kinerja hasil pelatihan *Multilayer Perceptron* dengan algoritme *Gradient Descent learning rate 0,001* ditunjukkan pada Tabel 4.11. Akurasi, *precision* dan *recall* rata-rata yang didapatkan dari dataset pelatihan masing sebesar 0,5341, 0,5966 dan 0,5329.

Tabel 4.11 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pelatihan Algoritme *Gradient Descent Learning Rate 0,001*

	Label		Avg
	Neg	Pos	
TP	2887	388	
TN	388	2887	
FP	2669	188	
FN	188	2669	
TPR = Sensitivity	0,9389	0,1269	
TNR = Specificity	0,1269	0,9389	
Recall	0,9389	0,1269	0,5329
Precision	0,5196	0,6736	0,5966
Akurasi	0,5341	0,5341	0,5341

Model klasifikasi *Multilayer Perceptron* dengan algoritme *Gradient Descent learning rate 0,001* juga dilakukan pengujian dengan dataset pengujian yang dari awal dipisahkan dan tidak diikuti selama proses pelatihan. *Confusion matrix* kelas atau label target klasifikasi terhadap prediksi dengan dataset pengujian ditunjukkan pada Tabel 4.12. Dari total 682 dataset pengujian dengan kelas target negatif sebanyak 332 berhasil diprediksi tepat pada kelas negatif sebanyak 315 dan eror sebanyak 17. Kelas target positif sebanyak 350 berhasil diprediksi tepat pada kelas positif sebanyak 45 dan eror sebanyak 305.

Tabel 4.12 *Confusion Matrix* Pengujian dengan Dataset Pengujian Algoritme *Gradient Descent Learning Rate 0,001*

		Prediksi		Total Target
		Neg	Pos	
Target	Neg	315	17	332
	Pos	305	45	350
Total Prediksi		620	62	682

Dari *confusion matrix* pada Tabel 4.12 dapat diukur kinerja akurasi, *precision* dan *recall* dari model klasifikasi yang diuji menggunakan data pengujian. Pengukuran kinerja hasil pengujian *Multilayer Perceptron* algoritme *Gradient Descent learning rate 0,001* ditunjukkan pada Tabel 4.13. Berdasarkan hasil pengujian, *Multilayer Perceptron* algoritme *Gradient Descent learning rate 0,001* sangat ekstrem dalam memprediksi kelas negatif dari pada kelas positif berdasarkan nilai *recall* kelas negatif sebesar 0,9488 sedangkan *recall* kelas positif hanya sebesar 0,1286. Akurasi rata-rata yang didapatkan sebesar 0,5792.

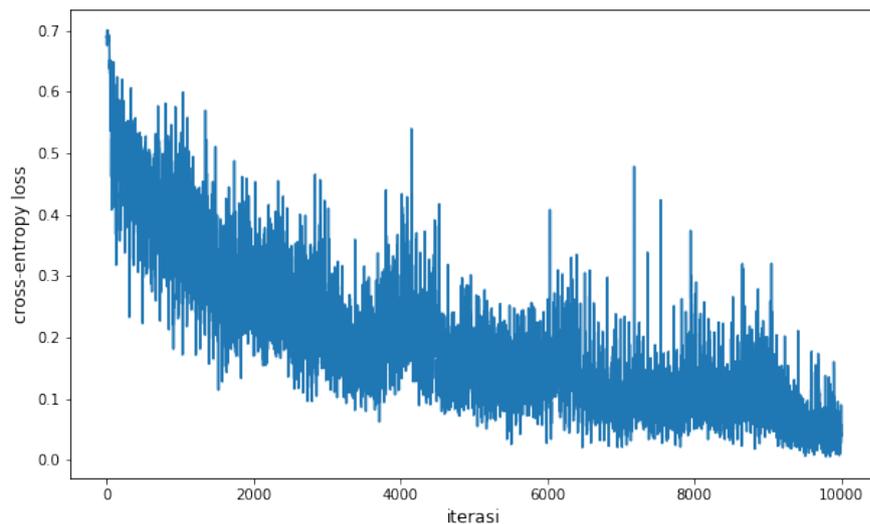
Tabel 4.13 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pengujian Algoritme *Gradient Descent Learning Rate 0,001*

	Label		Avg
	Neg	Pos	
TP	315	45	
TN	45	315	
FP	305	17	
FN	17	305	
TPR = Sensitivity	0,9488	0,1286	
TNR = Specificity	0,1286	0,9488	
Recall	0,9488	0,1286	0,5387
Precision	0,5080	0,7258	0,6169
Akurasi	0,5279	0,5279	0,5279

4.4 Multilayer Perceptron Algoritme Pelatihan Adam Learning Rate 0,01

Pelatihan dilakukan dengan menggunakan algoritme Adam dengan *learning rate 0,01*, *batch size 100* dan iterasi 10.000. Proses pelatihan *Multilayer Perceptron* menggunakan dataset pelatihan sebagaimana yang ditunjukkan pada Tabel 4.1. *Cross-Entropy Loss* pelatihan *Multilayer Perceptron* algoritme *Adam learning rate 0,01* ditampilkan pada Gambar 4.3. Pada iterasi pertama proses

pelatihan loss yang didapatkan sebesar 0.68921906. Pada iterasi ke-2.000, *cross-entropy loss* yang didapatkan turun menjadi 0.25070876. Pada iterasi ke-4.000, *cross-entropy loss* yang didapatkan sebesar 0,20259668. Pada iterasi ke-6.000, *cross-entropy loss* menjadi 0,17076403. Pada iterasi ke-800, *cross-entropy loss* turun menjadi 0,100791976. Nilai *cross-entropy loss* yang didapatkan pada akhir proses pelatihan sebesar 0.042749234. Algoritme Adam mampu meminimalkan nilai *cross-entropy loss* setiap iterasi selama proses pelatihan.



Gambar 4.3 Grafik *Cross-Entropy Loss* Pelatihan Algoritme *Adam Learning Rate* 0,01

Proses pelatihan menggunakan algoritme pelatihan *Adam* dengan *learning rate* 0,01 menghasilkan sebuah model klasifikasi. Dataset pelatihan dan dataset pengujian diklasifikasi menggunakan model yang dihasilkan pada proses pelatihan. Dengan begitu dapat dianalisis seberapa baik *Multilayer Perceptron* sebagai *classifier* mengenali beberapa data pada kelas atau label yang berbeda dengan menggunakan *confusion matrix*. *Confusion matrix* kelas atau label target klasifikasi terhadap prediksi dengan dataset pelatihan ditunjukkan pada Tabel 4.14. Dari 6.312 dataset pelatihan dengan kelas target negatif sebanyak 3.075 berhasil diprediksi tepat pada kelas negatif sebanyak 2.858 dengan eror sebanyak 217. Kelas target positif sebanyak 3.057 berhasil diprediksi tepat pada kelas positif sebanyak 2.986 dengan eror sebanyak 71.

Tabel 4.14 *Confusion Matrix* Pengujian dengan Dataset Pelatihan Algoritme *Adam Learning Rate 0,01*

	Prediksi		Total Target	
	Neg	Pos		
Target	Neg	2858	217	3075
	Pos	71	2986	3057
Total Prediksi		2929	3203	6132

Dari *confusion matrix* pada Tabel 4.14, dapat diukur kinerja akurasi, *precision* dan *recall*. Pengukuran kinerja hasil pelatihan *Multilayer Perceptron* algoritme *Adam learning rate 0,01* ditunjukkan pada Tabel 4.15. Akurasi, *precision* dan *recall* rata-rata yang didapatkan dari dataset pelatihan masing sebesar 0,9530, 0,9540 dan 0,9531.

Tabel 4.15 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pelatihan Algoritme *Adam Learning Rate 0,01*

	Label		Avg
	Neg	Pos	
TP	2858	2986	
TN	2986	2858	
FP	71	217	
FN	217	71	
TPR = Sensitivity	0,9294	0,9768	
TNR = Specificity	0,9768	0,9294	
Recall	0,9294	0,9768	0,9531
Precision	0,9758	0,9323	0,9540
Akurasi	0,9530	0,9530	0,9530

Model klasifikasi *Multilayer Perceptron* dengan algoritme *Adam learning rate 0,01* juga dilakukan pengujian dengan dataset pengujian yang dari awal dipisahkan dan tidak diikutkan selama proses pelatihan. *Confusion matrix* kelas atau label target klasifikasi terhadap prediksi dengan dataset pengujian ditunjukkan pada Tabel 4.16. Dari total 682 dataset pengujian dengan kelas target negatif sebanyak 332 berhasil diprediksi tepat pada kelas negatif sebanyak 222 dengan error sebanyak 110. Kelas target positif sebanyak 350 berhasil diprediksi tepat pada kelas positif sebanyak 287 dengan eror sebanyak 63.

Tabel 4.16 *Confusion Matrix* Pengujian dengan Dataset Pengujian Algoritme *Adam Learning Rate 0,01*

		Prediksi		Total Target
		Neg	Pos	
Target	Neg	222	110	332
	Pos	63	287	350
Total Prediksi		285	397	682

Dari *confusion matrix* pada Tabel 4.16 dapat diukur kinerja akurasi, *precision* dan *recall* dari model klasifikasi yang diuji menggunakan data pengujian. Pengukuran kinerja hasil pengujian *Multilayer Perceptron* algoritme *Adam learning rate 0,01* ditunjukkan pada Tabel 4.17. Berdasarkan hasil pengujian, *Multilayer Perceptron* algoritme *Adam learning rate 0,01* lebih presisi dalam memprediksi kelas positif dari pada kelas negatif berdasarkan nilai *precision* kelas positif sebesar 0,8200 sedangkan *precision* kelas negatif sebesar 0,6697. Akurasi rata-rata yang didapatkan sebesar 0,7463.

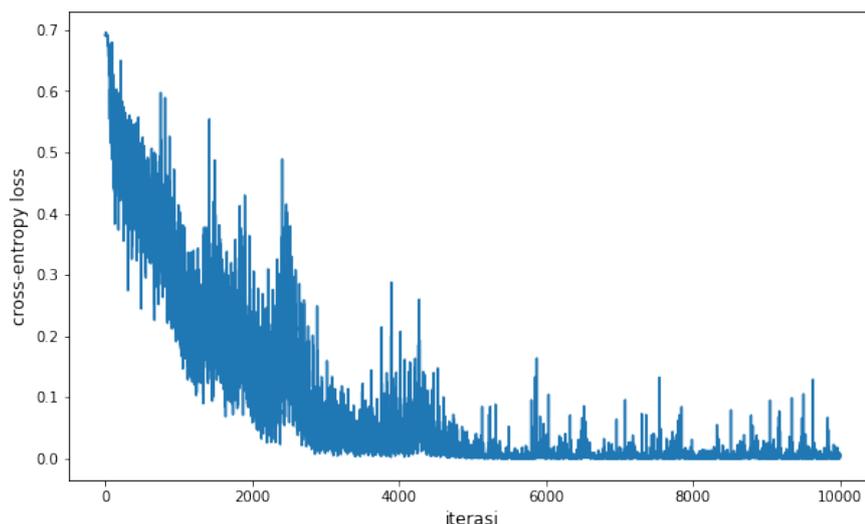
Tabel 4.17 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pengujian Algoritme *Adam Learning Rate 0,01*

	Label		Avg
	Neg	Pos	
TP	222	287	
TN	287	222	
FP	63	110	
FN	110	63	
TPR = Sensitivity	0,7789	0,7229	
TNR = Specificity	0,7229	0,7789	
Recall	0,7789	0,7229	0,7509
Precision	0,6697	0,8200	0,7448
Akurasi	0,7463	0,7463	0,7463

4.5 Multilayer Perceptron Algoritme Pelatihan Adam Learning Rate 0,001

Pelatihan dilakukan dengan menggunakan algoritme *Adam* dengan *learning rate 0,001*, *batch size 100* dan iterasi 10.000. Proses pelatihan *Multilayer Perceptron* menggunakan dataset pelatihan sebagaimana yang ditunjukkan pada Tabel 4.1. *Cross-Entropy Loss* pelatihan *Multilayer Perceptron* algoritme *Adam learning rate 0,001* ditampilkan pada Gambar 4.4. Pada iterasi pertama proses

pelatihan, *cross-entropy loss* yang didapatkan sebesar 0,69195086. Pada iterasi ke-2.000, *cross-entropy loss* yang didapatkan turun menjadi 0,07844079. Pada iterasi ke-4.000, *cross-entropy loss* yang didapatkan sebesar 0,032412693. Pada iterasi ke-6.000, *cross-entropy loss* menjadi 0,0043086647. Pada iterasi ke-800, *cross-entropy loss* turun menjadi 0,0027257553. Nilai *cross-entropy loss* yang didapatkan pada akhir proses pelatihan sebesar 0,005055798. Algoritme *Adam* mampu meminimalkan nilai *cross-entropy loss* setiap iterasi selama proses pelatihan.



Gambar 4.4 Grafik *Cross-Entropy Loss* Pelatihan Algoritme *Adam Learning Rate* 0,001

Proses pelatihan menggunakan algoritme pelatihan *Adam* dengan *learning rate* 0,001 menghasilkan sebuah model klasifikasi. Dataset pelatihan dan dataset pengujian diklasifikasi menggunakan model yang dihasilkan pada proses pelatihan. Dengan begitu dapat dianalisis seberapa baik *Multilayer Perceptron* sebagai *classifier* mengenali beberapa data pada kelas atau label yang berbeda dengan menggunakan *confusion matrix*. *Confusion matrix* kelas atau label target klasifikasi terhadap prediksi dengan dataset pelatihan ditunjukkan pada Tabel 4.18. Dari 6.312 dataset pelatihan dengan kelas target negatif sebanyak 3.075 berhasil diprediksi tepat pada kelas negatif sebanyak 3.003 dengan eror sebanyak 72. Kelas target positif sebanyak 3.057 berhasil diprediksi tepat pada kelas positif sebanyak 3.053 dengan eror sebanyak 4.

Tabel 4.18 *Confusion Matrix* Pengujian dengan Dataset Pelatihan Algoritme *Adam Learning Rate 0,001*

	Prediksi		Total Target	
	Neg	Pos		
Target	Neg	3003	72	3075
	Pos	4	3053	3057
Total Prediksi		3007	3125	6132

Dari *confusion matrix* pada Tabel 4.18, dapat diukur kinerja akurasi, *precision* dan *recall*. Pengukuran kinerja hasil pelatihan *Multilayer Perceptron* algoritme *Adam learning rate 0,001* ditunjukkan pada Tabel 4.19. Akurasi, *precision* dan *recall* rata-rata yang didapatkan dari dataset pelatihan masing sebesar 0,9876, 0,9878 dan 0,9876.

Tabel 4.19 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pelatihan Algoritme *Adam Learning Rate 0,001*

	Label		Avg
	Neg	Pos	
TP	3003	3053	
TN	3053	3003	
FP	4	72	
FN	72	4	
TPR = Sensitivity	0,9766	0,9987	
TNR = Specificity	0,9987	0,9766	
Recall	0,9766	0,9987	0,9876
Precision	0,9987	0,9770	0,9878
Akurasi	0,9876	0,9876	0,9876

Model klasifikasi *Multilayer Perceptron* dengan algoritme *Adam learning rate 0,001* juga dilakukan pengujian dengan dataset pengujian yang dari awal dipisahkan dan tidak diikuti selama proses pelatihan. *Confusion matrix* kelas atau label target klasifikasi terhadap prediksi dengan dataset pengujian ditunjukkan pada Tabel 4.20. Dari total 682 dataset pengujian dengan kelas target negatif sebanyak 332 berhasil diprediksi tepat pada kelas negatif sebanyak 222 dengan error sebanyak 110. Kelas target positif sebanyak 350 berhasil diprediksi tepat pada kelas positif sebanyak 298 dengan eror sebanyak 52.

Tabel 4.20 *Confusion Matrix* Pengujian dengan Dataset Pengujian Algoritme *Adam Learning Rate 0,001*

		Prediksi		Total Target
		Neg	Pos	
Target	Neg	222	110	332
	Pos	52	298	350
Total Prediksi		274	408	682

Dari *confusion matrix* pada Tabel 4.20 dapat diukur kinerja akurasi, *precision* dan *recall* dari model klasifikasi yang diuji menggunakan data pengujian. Pengukuran kinerja hasil pengujian *Multilayer Perceptron* algoritme *Adam learning rate 0,001* ditunjukkan pada Tabel 4.21. Berdasarkan hasil pengujian, *Multilayer Perceptron* algoritme *Adam learning rate 0,001* lebih presisi dalam memprediksi kelas negatif dari pada kelas positif berdasarkan nilai *precision* kelas negatif sebesar 0,8102 sedangkan *precision* kelas positif sebesar 0,7304. Akurasi rata-rata yang didapatkan sebesar 0,7625.

Tabel 4.21 Pengukuran Kinerja Hasil Pengujian dengan Dataset Pengujian Algoritme *Adam Learning Rate 0,001*

	Label		Avg
	Neg	Pos	
TP	222	298	
TN	298	222	
FP	52	110	
FN	110	52	
TPR = Sensitivity	0,6687	0,8514	
TNR = Specificity	0,8514	0,6687	
Recall	0,6687	0,8514	0,7600
Precision	0,8102	0,7304	0,7703
Akurasi	0,7625	0,7625	0,7625

4.6 Perbandingan Keseluruhan Hasil

Perbandingan dilakukan pada kinerja hasil pengujian menggunakan dataset pengujian terhadap empat model, yaitu *Multilayer Perceptron* algoritme *Gradient Descent learning rate 0,01*, *Multilayer Perceptron* algoritme *Gradient Descent learning rate 0,001*, *Multilayer Perceptron* algoritme *Adam learning rate*

0,01 dan *Multilayer Perceptron* algoritme *Adam* *learning rate* 0,001. Tabel 4.22 menunjukkan keseluruhan hasil dari empat model *Multilayer Perceptron*.

Tabel 4.22 Perbandingan Keseluruhan Hasil

	Multilayer Perceptron			
	<i>Gradient Descent</i>		<i>Adam</i>	
	<i>learning rate</i>	<i>learning rate</i>	<i>learning rate</i>	<i>learning rate</i>
	0,01	0,001	0,01	0,001
<i>Cross-Entropy Loss</i> pelatihan	0,69118667	0,6928074	0,042749234	0,005055798
Akurasi	0,5792	0,5279	0,7463	0,7625
<i>Precision</i>	0,5941	0,6169	0,7448	0,7703
<i>Recall</i>	0,5836	0,5387	0,7509	0,7600

Pada penelitian ini algoritme *Gradient Descent* belum mampu meminimalkan nilai *cross-entropy loss* mendekati nol. Termasuk perubahan atau penggantian *learning rate* dari 0,01 menjadi lebih kecil 0,001. Penggantian *learning rate* menjadi lebih kecil adalah upaya agar nilai *cross-entropy loss* bisa nol atau mendekati nilai nol. Algoritme *Gradient Descent* dengan *learning rate* 0,01 memberikan akurasi sebesar 0,5792 terhadap dataset pengujian, sedangkan *learning rate* 0,001 memberikan akurasi yang lebih rendah sebesar 0,5279.

Algoritme *Adam* lebih mampu untuk meminimalkan nilai *cross-entropy loss*, dimana jika dibandingkan dengan algoritme *Gradient Descent* hasilnya sangat jauh lebih kecil. Nilai *cross-entropy loss* pelatihan pada algoritme *Adam* dengan *learning rate* 0,01 sebesar 0,042749234 dan akurasi terhadap dataset pengujian sebesar 0,7463. Ketika *learning rate* diturunkan menjadi 0,001, nilai *cross-entropy loss* juga turun menjadi 0,005055798 dan akurasi terhadap dataset pengujian naik menjadi 0,7625. Berarti penurunan *learning rate* pada algoritme *Adam* menjadikan *Multilayer Perceptron* lebih akurat.

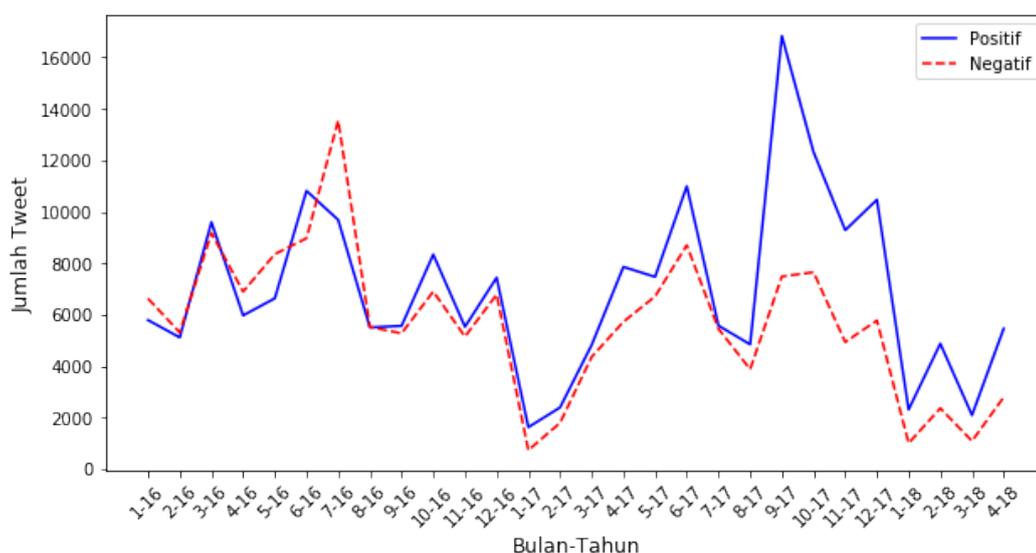
Tabel 4.23 menunjukkan beberapa hasil prediksi dari *tweet* dataset pengujian menggunakan model *Multilayer Perceptron* dengan algoritme *Adam learning rate* 0,001.

Tabel 4.23 Hasil Prediksi *Tweet* Dataset Pengujian Menggunakan *Multilayer Perceptron* Algoritme *Adam Learning Rate* 0,001

Tweet Dataset Pengujian	Label Sentimen	Prediksi
membangun ekonomi kelas atas jadi ingat sultan menolak jalan tol mematikan perekonomian dari pajak bersumber dari rakyat dipakai rakyat kok malah bayar	Negatif	Negatif
cebong bicara tol laut tapi tidak paham masalah	Negatif	Negatif
pembangunan wisata bikin tanah jadi mahal ada sengketa pula jalan tol medan binjai juga pembebasan lahan masih masalah kereta cepat jakarta juga pembebasan lahan sudah urus aja dulu sama pemerintah	Positif	Negatif
alfa tagar keadilan masalah transportasi beli bis bangun jalan tidak ada tentang	Negatif	Negatif
tidak semua jalan tol biaya kan oleh persero tapi biaya gabungan ada juga persero cuma mengerjakan pembangunan jalan tol serta ada juga yang oleh bank konsorsium jalan oleh swasta persero tidak ada hubungan dengan uang hasil pajak	Negatif	Negatif
apa yang memang jalan tol aja banyak rakyat miskin utang	Negatif	Positif
aku tak perlu orang lain selain aku bagi untuk aku aku boleh bangun jalan sendiri	Positif	Positif
kan sudah berhasil salah bikin tol laut pilih benar	Positif	Negatif
calon wakil gubernur barat berjanji akan membangun jalan tol selatan jabar mulai dari kabupaten bogor sampai jika terpilih jabar	Positif	Positif
membangun ekonomi kelas atas jadi ingat sultan menolak jalan tol mematikan perekonomian dari pajak bersumber dari rakyat dipakai rakyat kok malah bayar	Negatif	Negatif

4.7 Analisis Sentimen Seluruh Data

Data ini adalah data keseluruhan yang diperoleh pada tahap pengumpulan data, termasuk di dalamnya dataset yang digunakan untuk pelatihan dan dataset yang digunakan untuk pengujian model klasifikasi *Multilayer Perceptron*. Data ini kan diklasifikasikan menggunakan model *Multilayer Perceptron* dengan algoritme *Adam learning rate* 0,001 sebagai model yang memiliki kinerja paling tinggi. Hasilnya merupakan sentimen dari opini dan komentar pada Twitter terhadap kebijakan pemerintah dalam pembangunan infrastruktur. Gambar 4.5 menunjukkan hasil analisis sentimen terhadap keseluruhan data. Dari sebanyak 356.613 *tweet* berhasil diklasifikasi dan dilakukan analisis sentimen sebanyak 354.188 *tweet* sebagaimana ditunjukkan pada Tabel 4.24.



Gambar 4.5 Grafik Hasil Analisis Sentimen Terhadap Keseluruhan Data

Hasil analisis sentimen terhadap keseluruhan data menjelaskan bahwa pada periode semester satu tahun 2016 opini dan komentar pada Twitter tentang kebijakan pemerintah dalam pembangunan infrastruktur masih didominasi oleh sentimen negatif. Hal ini karena pembangunan infrastruktur masih dalam proses dan masyarakat belum merasakan dampak dari pembangunan infrastruktur. Peraturan Presiden Nomor 3 Tahun 2016 tentang Percepatan Pelaksanaan Proyek Strategis Nasional baru diundangkan pada tanggal 12 Januari 2016. Beberapa proyek strategis akan di-*groundbreaking* pada tahun 2016. Pada semester pertama

tahun 2016, beberapa proyek infrastruktur yang terbengkalai mulai dikebut seperti pembangunan tol trans jawa ruas Mojokerto-Kertosono dan beberapa proyek strategis pembangunan jalan nasional lainnya. Periode semester dua tahun 2016 sentimen positif mulai muncul karena bertepatan dengan lebaran tahun 2016. Beberapa proyek infrastruktur selesai dibangun dan beberapa ruas jalan tol terutama tol trans jawa mulai dioperasikan walaupun masih operasi secara fungsional. Pengoperasian jalan tol ini dimanfaatkan untuk mudik pada lebaran tahun 2016, hal ini yang membuat sentimen masyarakat menjadi positif. Masyarakat mulai merasakan dampak secara langsung dari pembangunan infrastruktur utamanya infrastruktur jalan tol. Pada tahun 2016, pemerintah meningkatkan kapasitas jalan nasional sepanjang 1.156 km, membangun jalan baru 567 kilometer dan jalan tol sepanjang 65 km (44 km dari pihak swasta dan 21 km dari pemerintah). Pembangunan jalan termasuk di daerah perbatasan di Pulau Kalimantan, Pulau Timor dan Pulau Papua.

Pada tahun 2017, sentimen positif juga mulai mendominasi opini dan komentar pada Twitter karena beberapa infrastruktur sudah selesai dikerjakan dan mulai diresmikan oleh pemerintah. Diantaranya merampungkan jalan tol sepanjang 392 km. Adapun jalan tol yang dapat dimanfaatkan oleh masyarakat pada tahun 2017 yaitu Akses Tanjung Priok sepanjang 11,4 km, Gempol-Pasuruan Seksi 1 sepanjang 15,7 km, Kertosono-Mojokerto Seksi 2 sepanjang 19,9 km, Semarang-Solo Seksi 3 sepanjang 17,6 km. Kemudian, Palembang-Indralaya seksi 1 sepanjang 7,4 km, Medan-Binjai Seksi 2 dan 3 sepanjang 10,45 km, Medan-Kualanamu-Tebing Tinggi seksi 2 hingga 6 sepanjang 42,1 km, dan Bekasi-Cawang-Kampung Melayu Seksi IB dan IC sepanjang 8,4 km. Pembangunan jalan baru mencapai 778 km. Untuk perbatasan Kalimantan Barat sudah mencapai 748 km.

Tabel 4.24 Hasil Analisis Sentimen Terhadap Keseluruhan Data

Label Sentimen	Jumlah	Persentase (%)
Positif	195.231	55,12
Negatif	158.957	44,88
Total	354.188	100

BAB 5

KESIMPULAN

Berdasarkan hasil penelitian yang diperoleh dapat disimpulkan bahwa :

1. Pada arsitektur *Multilayer Perceptron 6 layer*, jumlah neuron menggunakan konfigurasi 800, 200, 100, 50, 25 dan 2, fungsi aktivasi menggunakan *Rectified Linear Unit* pada *hidden layer* dan *softmax* pada *output layer*.
2. Penggunaan algoritme *Gradient Descent* pada *Multilayer Perceptron* belum bisa memberikan nilai *loss* paling kecil. Perubahan *learning rate* pada algoritme *Gradient Descent* dari 0,01 menjadi 0,001 membuat nilai *loss* meningkat dari 0,69118667 menjadi 0,6928074.
3. Penggunaan algoritme *Adam* pada *Multilayer Perceptron* dapat memperkecil nilai *loss* dari pada penggunaan algoritme *Gradient Descent*.
4. Penurunan *learning rate* pada penggunaan algoritme *Adam* dapat memperkecil nilai *loss* saat pelatihan *Multilayer Perceptron*.
5. Dari keempat ujicoba pelatihan yang dilakukan kinerja akurasi tertinggi sebesar 0,7625 didapatkan *Multilayer Perceptron* menggunakan algoritme *Adam* dengan *learning rate* 0,001.
6. Analisis sentimen masyarakat di Twitter terhadap kebijakan pemerintah dalam pembangunan infrastruktur menunjukkan sentimen negatif pada saat awal dicanangkannya proyek strategis nasional berdasarkan Peraturan Presiden Nomor 3 Tahun 2016 yang diundangkan tanggal 12 Januari 2016. Pada awal tahun 2016 beberapa proyek infrastruktur baru di-*groundbreaking* dan juga menyelesaikan proyek infrastruktur yang sebelumnya terbengkalai. Sentimen positif baru mulai dominan pada semester kedua tahun 2016 karena adanya agenda tahunan yaitu mudik lebaran yang memanfaatkan infrastruktur jalan tol baik yang beroperasi penuh maupun yang beroperasi secara fungsional. Masyarakat Pada akhir tahun 2016, pemerintah telah selesai membangun 567 km jalan baru, 65 km jalan tol dan peningkatan kapasitas 1.156 km jalan. Pada tahun 2017,

sentimen positif lebih banyak dari pada sentimen negatif karena proyek infrastruktur utamanya pembangunan infrastruktur jalan tol banyak yang telah selesai dan beroperasi penuh. Pemerintah telah menyelesaikan 392 km jalan tol dan 778 km jalan baru.

DAFTAR PUSTAKA

- [1] A. M. Ramadhani and H. S. Goo, "Twitter sentiment analysis using deep learning methods," in *2017 7th International Annual Engineering Seminar (InAES)*, Yogyakarta, Indonesia, 2017, pp. 1–4.
- [2] B. Liu, "Sentiment Analysis and Opinion Mining," p. 168, 2012.
- [3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Journal Foundations and Trends in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.
- [4] M. Lailiyah, S. Sumpeno, and I. K. E. Purnama, "Sentiment analysis of public complaints using lexical resources between Indonesian sentiment lexicon and Sentiwordnet," 2017, pp. 307–312.
- [5] P. Y. C. Kusuma, S. Sumpeno, and A. D. Wibawa, "Social media analysis of BPS data availability in economics using decision tree method," 2016, pp. 148–153.
- [6] Asniar and B. R. Aditya, "A Framework for Sentiment Analysis Implementation of Indonesian Language Tweet on Twitter," *J. Phys.: Conf. Ser.*, vol. 801, no. 1, p. 012065, 2017.
- [7] M. A. Assuja, "ANALISIS SENTIMEN TWEET MENGGUNAKAN BACKPROPAGATION NEURAL NETWORK," *ISSN*, vol. 10, no. 2, p. 6, 2016.
- [8] R. Habibi, D. B. Setyohadi, and E. Wati, "ANALISIS SENTIMEN PADA TWITTER MAHASISWA MENGGUNAKAN METODE BACKPROPAGATION," *Jurnal Informatika*, vol. 12, no. 1, Jun. 2016.
- [9] A. M. Siregar, "KLASIFIKASI ALGORITMA TF DAN NEURAL NETWORK DALAM SENTIMEN ANALISIS," *AIMS: Jurnal Accounting Information System*, vol. 1, no. 2, pp. 1–8, Sep. 2018.
- [10] Kementerian Pendidikan dan Kebudayaan, *Kamus Besar Bahasa Indonesia*. 2016.
- [11] James A. Anderson, *Public Policy Making: Basic Concept in Political*. New York: Praeger University Series, 1975.
- [12] T. R. Dye, *Understanding Public Policy (4th Edition)*. NJ: Prentice-Hall Inc, Ltd, 1981.
- [13] Y. Toyib and R. Nugroho, *Transformasi Public Private Partnership Indonesia*. Elex Media Computindo, 2018.
- [14] R. Feldman and J. Sanger, *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge; New York: Cambridge University Press, 2007.
- [15] I. P. Cvijikj and F. Michahelles, "Understanding Social Media Marketing: A Case Study on Topics, Categories and Sentiment on a Facebook Brand Page," *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, p. 8, 2011.
- [16] L. F. S. Coletta, N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka, "Combining Classification and Clustering for Tweet Sentiment Analysis," in *Brazilian Conference on Intelligent Systems*, 2014, pp. 210–215.

- [17] C. R. Fink, D. S. Chou, J. J. Kopecky, and A. J. Llorens, “Coarse- and Fine-Grained Sentiment Analysis of Social Media Text,” *JOHNS HOPKINS APL TECHNICAL DIGEST*, vol. 30, no. 1, p. 9, 2011.
- [18] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A Neural Probabilistic Language Model,” p. 19, 2003.
- [19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *arXiv:1301.3781 [cs]*, Jan. 2013.
- [20] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543.
- [21] “Word Embeddings - word2vec.” [Online]. Available: <https://i.imgur.com/vNYiUZi.jpg>. [Accessed: 06-Apr-2019].
- [22] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, “Sentiment strength detection in short informal text,” *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, Dec. 2010.
- [23] H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou, and M. Ettaouil, “Multilayer Perceptron: Architecture Optimization and Training,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. 1, p. 26, 2016.
- [24] “Multi-layer Perceptron.” [Online]. Available: https://www.saedsayad.com/images/Perceptron_bkp_1.png. [Accessed: 06-Apr-2019].
- [25] “Softmax in TensorFlow,” *Softmax in TensorFlow*. [Online]. Available: <https://www.oreilly.com/library/view/practical-convolutional-neural/9781788392303/2e8989a5-acd0-460a-bd66-44602f26a926.xhtml>. [Accessed: 12-Apr-2019].
- [26] “Understanding Confusion Matrix,” *Understanding Confusion Matrix*. [Online]. Available: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>. [Accessed: 12-Apr-2019].
- [27] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani, “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining,” in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10)*, 2010.
- [28] S. Asthana, F. Haneef, and R. K. Bhujade, “Handwritten Multiscript Numeral Recognition using Artificial Neural Networks,” vol. 1, no. 1, p. 5, 2011.
- [29] S. Karsoliya, “Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture,” *International Journal of Engineering Trends and Technology*, p. 4, 2012.
- [30] S. Amari, “Natural Gradient Works Efficiently in Learning,” *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.
- [31] Diederik P. Kingma and Jimmy Ba, “Adam: A Method for Stochastic Optimization,” in *Proceeding of 3rd International Conference for Learning Representations*, 2015.
- [32] M. S. Nacson, N. Srebro, and D. Soudry, “Stochastic Gradient Descent on Separable Data: Exact Convergence with a Fixed Learning Rate,” p. 9.

BIOGRAFI PENULIS

Eko Yudhi Prastowo, mahasiswa Magister Telematika (Pengelola Teknologi Informasi dan Komunikasi Pemerintahan), Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya. Penerima Beasiswa Pascasarjana (S2) Kementerian Komunikasi dan Informatika Republik Indonesia Tahun 2017. Lahir di Ngawi pada 21 Oktober 1986. Saat ini bekerja sebagai pegawai negeri sipil di Direktorat Jenderal Pajak Kementerian Keuangan Republik Indonesia. *E-mail* : ekoyudhiprastowo@gmail.com

Riwayat Pendidikan

1. Sekolah Dasar Negeri Jogorogo I, Ngawi
2. Sekolah Lanjutan Tingkat Pertama Negeri I Jogorogo, Ngawi
3. Sekolah Menengah Atas Negeri 2 Ngawi
4. Sekolah Tinggi Akuntansi Negara Jakarta
5. Universitas Terbuka

Halaman ini sengaja dikosongkan