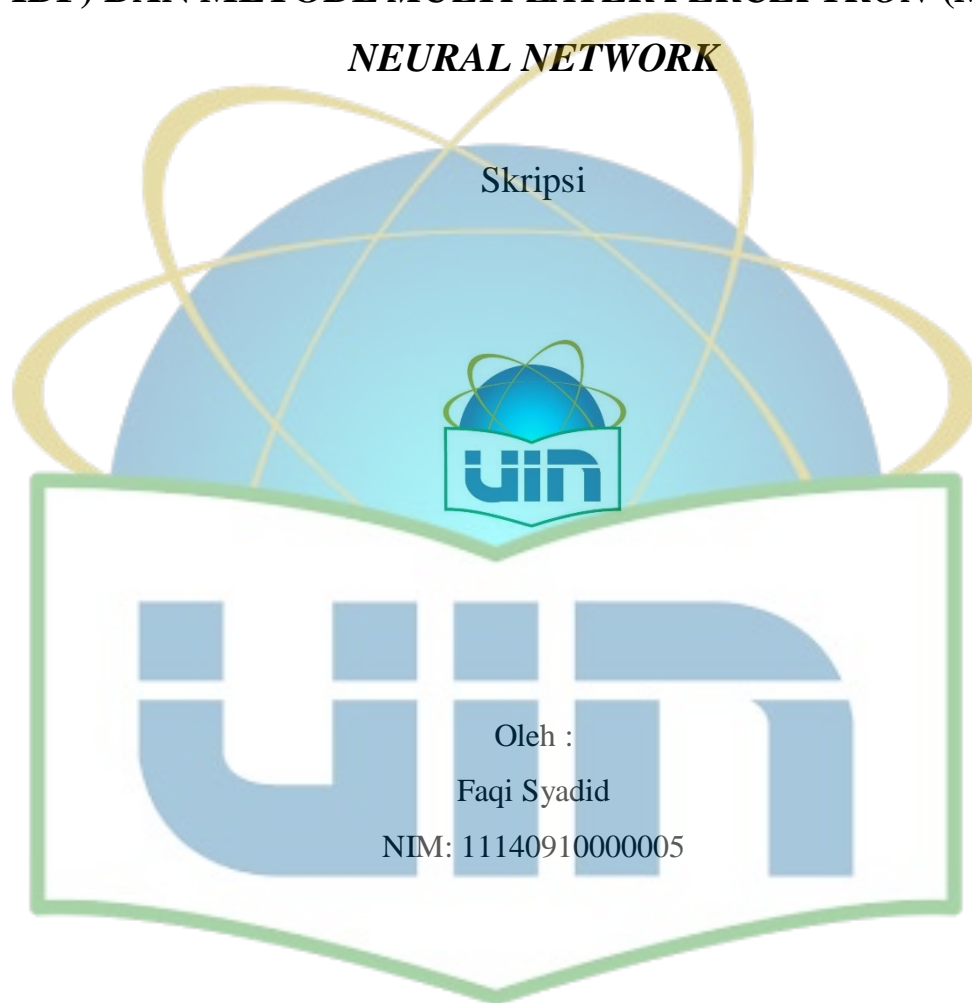


**ANALISIS SENTIMEN KOMENTAR NETIZEN
TERHADAP CALON PRESIDEN INDONESIA 2019 DARI
TWITTER MENGGUNAKAN ALGORITMA *TERM
FREQUENCY-INVERS DOCUMENT FREQUENCY* (TF-
IDF) DAN METODE *MULTI LAYER PERCEPTRON* (MLP)
*NEURAL NETWORK***



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH
JAKARTA
2019 M / 1440 H**

**ANALISIS SENTIMEN KOMENTAR NETIZEN
TERHADAP CALON PRESIDEN INDONESIA 2019 DARI
TWITTER MENGGUNAKAN ALGORITMA *TERM
FREQUENCY-INVERS DOCUMENT FREQUENCY* (TF-
IDF) DAN METODE *MULTI LAYER PERCEPTRON* (MLP)
*NEURAL NETWORK***

Skripsi

Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer (S.Kom)



Oleh :

Faqi Syadid

NIM: 11140910000005

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH
JAKARTA
2019 M / 1440 H**

HALAMAN PERSETUJUAN PEMBIMBING

**ANALISIS SENTIMEN KOMENTAR NETIZEN TERHADAP CALON
PRESIDEN INDONESIA 2019 DARI *TWITTER* MENGGUNAKAN
ALGORITMA *TERM FREQUENCY-INVERS DOCUMENT FREQUENCY*
(TF-IDF) DAN METODE *MULTI LAYER PERCEPTRON (MLP) NEURAL
NETWORK***

Skripsi

Diajukan sebagai salah satu syarat untuk memperoleh gelar

Sarjana Komputer (S.Kom)

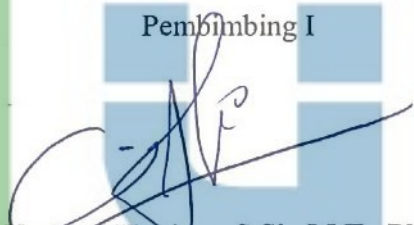
Oleh:

Faqi Syadid 11140910000005

Menyetujui,

Pembimbing I

Pembimbing II


Nashrul Hakiem, S.Si., M.T., Ph.D
NIP. 19710608 200501 1 005


Siti Umami Masrurroh, M.Sc
NIP. 19820823 201101 2 013

Mengetahui,

Ketua Program Studi Teknik Informatika



Arini, MT

NIP. 19760131 200901 2 001

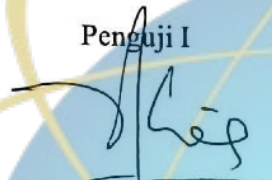
HALAMAN PENGESAHAN UJIAN

Skripsi berjudul **Analisis Sentimen Komentar Netizen terhadap Calon Presiden Indonesia 2019 dari Twitter menggunakan Algoritma *Term Frequency-Invers Document Frequency* (TF-IDF) dan Metode *Multi Layer Perceptron* (MLP) *Neural Network*** telah diujikan dalam sidang munaqasyah Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta pada 06 Maret 2019. Skripsi ini telah diterima sebagai salah satu syarat memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika.

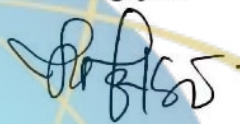
Jakarta, 06 Maret 2019

Tim Penguji

Penguji I

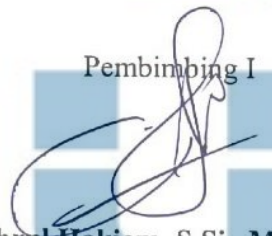

Khadijah Hullyah, M.Si
NIP. 19730402 200112 2 001

Penguji II


Anif Hanifa Setianingrum, M.Si
NIDN.07280 20340 389997

Tim Pembimbing

Pembimbing I


Nashrud Hakiem, S.Si., M.T., Ph.D
NIP. 19710608 200501 1 005

Pembimbing II



Siti Umami Masruroh, M.Sc
NIP. 19820823 201101 2 013

Mengetahui,

Dekan
Fakultas Sains dan Teknologi

Ketua
Program Studi Teknik Informatika




Prof. Dr. Lily Surraya Eka Putri,
M.Env.Stud
NIP. 19690404 200501 2 005


Arini, MT
NIP. 19760131 200901 2 001

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI

Sebagai civitas akademik UIN Syarif Hidayatullah Jakarta, saya yang bertanda tangan di bawah ini:

Nama : Faqi Syadid

NIM : 11140910000005

Program Studi : Teknik Informatika

Fakultas : Sains dan Teknologi

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Islam Negeri Syarif Hidayatullah Jakarta Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah yang berjudul:

**ANALISIS SENTIMEN KOMENTAR NETIZEN TERHADAP CALON
PRESIDEN INDONESIA 2019 DARI TWITTER MENGGUNAKAN
ALGORITMA *TERM FREQUENCY-INVERS DOCUMENT FREQUENCY*
(TF-IDF) DAN METODE *MULTI LAYER PERCEPTRON (MLP) NEURAL
NETWORK***

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non eksklusif ini Universitas Islam Negeri Syarif Hidayatullah Jakarta berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilih Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Jakarta

Pada tanggal: 6 Maret 2019

Yang menyatakan



(Faqi Syadid)

HALAMAN PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa:

1. Skripsi ini merupakan hasil karya asli saya yang diajukan untuk memenuhi salah satu persyaratan memperoleh gelar strata 1 di UIN Syarif Hidayatullah Jakarta
2. Semua sumber yang saya gunakan dalam penulisan ini telah saya cantumkan sesuai dengan ketentuan yang berlaku di UIN Syarif Hidayatullah Jakarta
3. Jika di kemudian hari terbukti bahwa karya ini bukan hasil karya asli saya atau merupakan hasil jiplakan dari karya orang lain, maka saya bersedia menerima sanksi yang berlaku di UIN Syarif Hidayatullah Jakarta.

Jakarta, 6 Maret 2019



Faqi Syadid

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena atas nikmat dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta. Dalam proses penyelesaian skripsi ini tidak luput dari berbagai macam bantuan, dukungan, saran, dan kritik yang telah penulis terima, oleh karena itu dalam kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua penulis, Bapak Rusmani Said & Ibu Eni Siti Royani yang selalu mendoakan dan selalu memberikan segala dukungan kepada penulis dalam menyelesaikan skripsi ini.
2. Ibu Prof. Dr. Lily Surraya Eka Putri, M.Env.Stud., selaku Dekan Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta
3. Ibu Arini, ST. MT., selaku Ketua Program Studi Teknik Informatika, serta Bapak Feri Fahrianto, M.Sc., selaku Sekretaris Program Studi Teknik Informatika.
4. Bapak Nashrul Hakiem, S.Si., M.T., Ph.D selaku Dosen Pembimbing I dan Ibu Siti Umami Masruroh, M.Sc., selaku Dosen Pembimbing II yang telah memberikan bimbingan, motivasi, dan arahan kepada penulis sehingga skripsi ini bisa selesai dengan baik.
5. Seluruh dosen Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta atas segala ilmu pengetahuan yang diberikan kepada penulis. Beserta seluruh staff akademik Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta.
6. Kepada Risma Ramjani yang selalu menemani dan memberikan memotivasi dalam penyusunan skripsi.
7. Kepada teman-teman kosan yang sudah membantu dan memberikan semangat.

8. Kepada seluruh teman-teman Teknik Informatika angkatan 2014 yang telah memberikan bantuan, motivasi dan do'a kepada penulis agar dapat menyelesaikan skripsi ini dengan baik.
9. Kepada seluruh teman-teman penerima beasiswa Karya Salemba Empat (KSE) yang telah memberikan bantuan, motivasi dan do'a kepada penulis agar dapat menyelesaikan skripsi ini dengan baik.
10. Seluruh pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dalam penyelesaian penulisan skripsi ini.

Akhir kata, penulis menyadari bahwa dalam penyajian skripsi ini masih jauh dari sempurna. Apabila ada kebenaran dari penulisan skripsi ini maka kebenaran tersebut datangnya dari Allah SWT, tetapi apabila ada kesalahan dalam penulisan ini maka kesalahan ini berasal dari penulis. Semoga skripsi ini dapat bermanfaat dan menambah wawasan serta pengetahuan bagi para pembaca.

Jakarta, 6 Maret 2019



Faqi Syadid

11140910000005

Nama : Faqi Syadid

Program Studi : Teknik Informatika

Judul : Analisis Sentimen Komentar Netizen terhadap Calon Presiden Indonesia 2019 dari *Twitter* menggunakan Algoritma *Term Frequency-Invers Document Frequency* (TF-IDF) dan Metode *Multi Layer Perceptron* (MLP) *Neural Network*

ABSTRAK

Penelitian ini bertujuan menerapkan algoritma *term frequency-invers document frequency* dan metode *multi layer perceptron* untuk mengklasifikasikan sentimen positif dan negatif pada tweet calon presiden indonesia 2019. Ada beberapa tahap untuk melakukan analisis sentimen, diantaranya adalah tahap pengumpulan data, *preprocesing* data, penghitungan bobot term menggunakan *term frequency-invers document frequency*, dan klasifikasi menggunakan *multi layer perceptron*. Menggunakan 3 skenario dalam percobaan klasifikasi, yaitu skenario 1 menggunakan jumlah data latih sebanyak 700 data, skenario 2 menggunakan jumlah data latih sebanyak 800 data, dan skenario 3 menggunakan jumlah data latih sebanyak 900 data. Hasil dari penelitian ini bahwa *term frequency-invers document frequency* dan metode *multi layer perceptron* dapat diimplementasikan untuk analisis sentimen dan didapatkan nilai akurasi tertinggi pada skenario 3 mencapai 88%.

Kata Kunci : Klasifikasi, *Multi layer Perceptron*, TF-IDF, Sentimen Analisis

Jumlah Pustaka : 3 Buku + 21 Jurnal + 5 Website

Jumlah Halaman : VI BAB + xiv Halaman + 74 Halaman + 12 Gambar + 19 Tabel

Name : Faqi Syadid
Study Program : Informatics Engineering
Title : Sentiment Analysis of the Netizen Comments on Indonesian President Candidates 2019 From Twitter Using Term Frequency-Invers Document Frequency Algorithm (TF-IDF) and Multilayer Perceptron (MLP) Neural Network Methods

ABSTRACT

This research aims to apply the Term Frequency-Inverse document Frequency algorithm and multilayer perceptron method to classify positive and negative sentiments in the Indonesian presidential candidates 2019. There are several stages to conduct sentiment analysis, including the stages of data collection, data preprocessing, calculation of term weights using Term Frequency-Inverse document Frequency and classification using multilayer perceptron. Using 3 scenarios in the classification experiment, that's scenario 1 uses the amount of training data as many as 700 data, scenario 2 uses the number of training data as many as 800 data, and scenario 3 uses the amount of training data as many as 900 data. The results of this research that the Term Frequency-Inverse Document Frequency and multilayer perceptron method can be implemented for sentiment analysis and obtained the highest accuracy value in scenario 3 reaching 88%.

Keywords : Classification, Multi layer Perceptron, TF-IDF, Sentiment Analysis
Bibliography : 3 Books + 21 Journals + 5 Websites
Number of Pages : VI BAB + xiv Pages + 74 Pages + 12 Pictures + 19 Tables

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN PEMBIMBING.....	ii
HALAMAN PENGESAHAN UJIAN	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI.....	iv
HALAMAN PERNYATAAN ORISINALITAS	v
KATA PENGANTAR	vi
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan Penelitian	3
1.3 Manfaat Penelitian	4
1.3.1 Bagi Penulis.....	4
1.3.2 Bagi Universitas	4
1.3.3 Bagi Masyarakat.....	4
1.4 Rumusan Masalah.....	4
1.5 Batasan Masalah	5
1.6 Metodologi Penelitian	5
1.6.1 Metode Pengumpulan Data	5
1.6.2 Analisis Permasalahan	5
1.6.3 Perancangan Desain Sistem	6
1.6.4 Implementasi	6
1.6.5 Uji Coba Sistem.....	6

1.6.6	Analisis Hasil Uji Coba Sistem	6
1.7	Sistematika Penelitian	6
BAB 2 LANDASAN TEORI		8
2.1	Sentimen Analisis	8
2.2	Twitter API.....	9
2.3	Natural Language Processing (NLP)	10
2.4	Text Mining	10
2.5	Preprocessing	12
2.5.1	Cleansing	12
2.5.2	Tokenisasi.....	12
2.5.3	Case Folding	12
2.5.4	Penghilangan Stopword	12
2.5.5	Stemming	13
2.6	Klasifikasi	13
2.7	Model Klasifikasi	14
2.8	Pengukuran Kinerja Klasifikasi.....	15
2.9	Teknik Klasifikasi Analisis Sentimen.....	16
2.9.1	Supervised Learning	16
2.9.2	Unsupervised Learning	17
2.10	Term Frequency – Inverse Document Frequency (TF-IDF).....	17
2.11	Artificial Neural Network.....	18
2.12	Multilayer Perceptron.....	21
2.13	Python	23
BAB 3 METODOLOGI PENELITIAN		25
3.1	Metode Pengumpulan Data	25
3.1.1	Studi Pustaka	25
3.1.2	Studi Literatur Sejenis	25
3.1.3	Streaming Data Twitter.....	28

3.2	Analisis Permasalahan	28
3.3	Perancangan Desain Sistem	29
3.4	Implementasi	29
3.5	Uji Coba Sistem.....	30
3.6	Analisis Hasil Uji Coba Sistem	30
3.7	Alur Penelitian.....	31
BAB 4 IMPLEMENTASI.....		32
4.1	Perancangan Desain Sistem	32
4.1.1	Pengumpulan Data.....	33
4.1.2	Data Latih dan Data Uji	35
4.1.3	<i>Preprocessing</i>	35
4.1.4	Pembobotan TF-IDF	43
4.1.5	Klasifikasi dengan Algoritma <i>Multilayer Perceptron</i>	47
4.2	Uji Coba Sistem.....	56
4.3	Analisis Hasil Uji Coba.....	58
BAB 5 Hasil dan pembahasan.....		59
5.1	Perangkat Penelitian	59
5.1.1	Perangkat Keras.....	59
5.1.2	Perangkat Lunak.....	59
5.2	Hasil Klasifikasi Sentimen Menggunakan Algoritma TF-IDF dan MLP.....	59
5.2.1	Hasil Tingkat Akurasi Skenario 1	66
5.2.2	Hasil Tingkat Akurasi Skenario 2	66
5.2.3	Hasil Tingkat Akurasi Skenario 3	67
BAB 6 KESIMPULAN DAN SARAN		71
6.1	Kesimpulan.....	71
6.2	Saran	71
DAFTAR PUSTAKA		72

DAFTAR GAMBAR

Gambar 2.1 Proses Pekerjaan Klasifikasi (Sumber : Prasetyo, 2012)	14
Gambar 2.2 Desain ANN Secara Umum (Sumber : Prasetyo, 2012)	19
Gambar 2.3 Arsitektur Multi Layer Perceptron	21
Gambar 3.1 Proses Pengumpulan Data.....	28
Gambar 3.2 Alur Penelitian.....	31
Gambar 4.1 Arsitektur Sistem Analisis Sentimen	32
Gambar 4.2 <i>Preprocessing</i>	35
Gambar 4.3 Rancangan Arsitektur Jaringan Multi Layer Perceptron	47
Gambar 5.1 Gambar Hasil perhitungan akurasi	68
Gambar 5.2 Hasil perhitungan precision.....	68
Gambar 5.3 Hasil perhitungan recall	69
Gambar 5.4 Hasil perhitungan f1-score	70



DAFTAR TABEL

Tabel 2.1 Matriks Konfusi Untuk Klasifikasi Dua Kelas	15
Tabel 3.1 Studi Literatur Sejenis.....	26
Tabel 4.1 Contoh Pelabelan Data	35
Tabel 4.2 Contoh Cleansing	36
Tabel 4.3 Contoh Tokenisasi.....	37
Tabel 4.4 Contoh Case Folding	38
Tabel 4.5 Contoh Stopword.....	40
Tabel 4.6 Contoh <i>Stemming</i>	42
Tabel 4.7 Tweet dan Label	43
Tabel 4.8 Menghitung Term Frequency	44
Tabel 4.9 Menghitung Inverse Document Frequency.....	45
Tabel 4.10 Menghitung TF-IDF	46
Tabel 4.11 Parameter Multilayer Perceptron	47
Tabel 4.12 Skenario Data Latih.....	56
Tabel 5.1 Hasil Klasifikasi Sentimen Menggunakan Algoritma TF-IDF dan MLP60	
Tabel 5.2 Rekapitulasi Hasil Klasifikasi dan Pengujian Skenario	64
Tabel 5.3 Hasil Pengujian Skenario 1	66
Tabel 5.4 Hasil Pengujian Skenario 2.....	66
Tabel 5.5 Hasil Pengujian Skenario 3	67

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Beberapa tahun terakhir pada era informasi seperti saat ini, pengguna internet di Indonesia telah tumbuh secara luar biasa cepatnya dan internet memegang peranan penting dalam segala aspek kehidupan manusia. Internet menjadi media yang banyak digunakan oleh hampir semua kalangan. Berdasarkan data dari Asosiasi Penyelenggara Jasa Internet Indonesia (www.apjii.or.id, 2017), Pertumbuhan pengguna internet tahun 2017 di Indonesia telah menembus angka 143,26 juta yang mengalami kenaikan dari tahun sebelumnya pada tahun 2016 yaitu jumlah pengguna internet mencapai 132,7 juta dan pengguna internet mulai dari umur 13 – 18 tahun sebanyak 16,68%, 19 – 34 tahun sebanyak 49,52%, 35 – 54 tahun sebanyak 29,55% dan umur lebih dari 54 tahun sebanyak 4,24%.

Para pengguna internet rata – rata menghabiskan durasi waktu sehari selama 1 – 3 jam dengan jumlah rata – rata pengguna sebanyak 43,89%, dan per minggunya rata – rata pengguna internetnya sebanyak 65,98% (www.apjii.or.id, 2017). Layanan internet yang paling sering diakses adalah chatting dengan menggunakan (Whatsapp, Line, We Chat) dan lain-lain sebesar 89,35%. Dan sosial media 87,13% dengan menggunakan (Facebook, Instagram, *Twitter*) dan lain-lain (www.apjii.or.id, 2018). Situs microblogging seperti *Twitter* telah menjadi alat komunikasi yang sangat populer di kalangan pengguna internet di Indonesia. Pada tahun 2016 Indonesia mendapat peringkat ketiga negara dengan pengguna aktif *twitter* ketiga terbesar dunia (www.katadata.co.id, 2016) .

Kegunaan *twitter* selain sebagai media untuk berbagi informasi dengan mem-post berbagai macam *tweet*, *twitter* juga kerap sering digunakan untuk bersosialisasi antar pengguna dan mengungkapkan sentimen atau opini mereka terhadap suatu topik atau isu-isu yang sedang hangat diperbincangkan, tidak hanya opini yang positif tapi juga yang

negatif. Kebiasaan tersebut terkadang tidak dianggap begitu penting untuk disikapi karena terlalu banyak *tweet*. Namun kebiasaan tersebut begitu penting apabila dipelajari lebih lanjut karena kebiasaan tersebut bisa dimanfaatkan juga sebagai sebuah analisis sentimen atau opini terhadap isu–isu yang sedang hangat dan dapat di jadikan sebuah informasi untuk mengetahui sentimen atau opini masyarakat Indonesia pada permasalahan tertentu.

Menjelang pemilihan Presiden Indonesia yang akan dilangsungkan pada tahun 2019, Komisi Pemilihan Umum (KPU) sudah menetapkan jadwal rangkaian Pemilihan Presiden Indonesia tahun 2019 dan sudah menetapkan daftar pencalonan pemilihan presiden 2019 yaitu dengan pasangan calon nomor urut satu adalah Ir. H. Joko Widodo bersama Prof. Dr. (HC). KH. Ma'ruf Amin dan pasangan nomor urut dua adalah H. Prabowo Subianto bersama Sandiaga Salahuddin Uno, MBA (www.infopemilu.kpu.go.id, 2018). Sejak sudah ditetapkan nama calon presiden indonesia 2019, nama-nama calonnya mulai banyak diperbincang, baik di dunia nyata maupun dunia maya melalui media sosial yang dimana semua orang bebas berpendapat atau beropini yang positif dan juga negatif tentang calon Presiden Indonesia 2019.

Hal ini menjadi menarik dan penting bagi pihak – pihak tertentu yang ingin mengetahui baik buruknya sentimen atau opini yang diberikan masyarakat indonesia terhadap para calon presiden indonesia 2019 menjelang pemilihan tersebut. Atas hal tersebut penelitian ini mencoba memanfaatkan hasil *tweet* pada *twitter* tersebut karena mudah diakses, jumlah pengikut tidak terbatas, data yang relatif banyak, *twitter* yang berasal dari berbagai sumber (terdistribusi), waktu penyajian yang *realtime*, dan jumlah karakter pesan singkat hanya 140 karakter sehingga mampu menyampaikan maksud dan tujuan dengan singkat, padat dan jelas.

Dari permasalahan tersebut, dibutuhkan sistem pengklasifikasian opini berupa teks sentimen pada *Twitter* dengan menganalisa *tweet* berbahasa indonesia yang mengandung sentimen atau opini tentang para

calon presiden indonesia 2019. Untuk melakukan hal itu, bisa menggunakan salah satu fungsi dari *text mining*, dalam hal ini adalah klasifikasi sentimen. Ada beberapa penelitian yang sudah pernah dilakukan berkaitan dengan klasifikasi dan yang sering digunakan untuk metode klasifikasi adalah *Naive Bayes Classifier* dan *Support Vector Machine* (Hadna, Santosa, & Winarno, 2016). Penelitian yang dilakukan oleh (Pang, Lee, Rd, & Jose, 2002) telah membandingkan algoritma *Naive Bayes Classifier*, *maximum entropy*, dan *Support Vector Machine* didapatkan hasil yang terbaik adalah *Support Vector Machine* (SVM). *Support Vector Machine* (SVM) memiliki kelebihan bisa diterapkan untuk data yang berdimensi tinggi, tetapi memiliki kekurangan yaitu sulit digunakan untuk data dengan jumlah yang besar (Nugroho, Witarto, & Handoko, 2003) .

Penelitian selanjut dilakukan oleh (Moraes, Valiati, & Neto, 2013) yang melakukan perbandingan *Support Vector Machine* (SVM) dengan *Artificial Neural Network* (ANN) dengan hasil metode klasifikasi terbaik adalah *Artificial Neural Network* (ANN). ANN mempunyai kelebihan dalam hal kemampuan untuk generalisasi, yang bergantung pada seberapa baik ANN meminimalkan resiko empiris dan dapat diterapkan pada yang jumlahnya besar (Chandani, 2015). Penulis pada penelitian ini akan melakukan analisis sentimen pada pengguna *twitter* terhadap kedua pasangan calon presiden pada Pilpres Indonesia 2019 menggunakan kombinasi metode *term frequency - invers document frequency* (TF-IDF) dan metode *multilayer perceptron neural network*. Dengan input berupa data *tweet* dalam bahasa indonesia dan output berupa dua jenis sentimen yaitu positif dan negatif.

1.2 Tujuan Penelitian

Untuk mengimplementasikan algoritma *Term Frequency - Inverse Document Frequency* (TF-IDF) dan metode *Multi Layer Perceptron* (MLP) *Neural Network* terhadap analisis sentimen *twitter*.

1.3 Manfaat Penelitian

1.3.1 Bagi Penulis

Mengaplikasikan ilmu-ilmu akademis yang didapat selama perkuliahan ke dalam aplikasi *Natural Language Processing* (NLP) untuk identifikasi sentimen *twitter* terkait calon presiden indonesia 2019 dengan menggunakan algoritma *Term Frequency - Inverse Document Frequency* (TF-IDF) dan *Multi Layer Perceptron* (MLP) *Neural Network*.

1.3.2 Bagi Universitas

1. Memberikan gambaran terhadap penerapan ilmu pengetahuan yang telah diterima selama kuliah.
2. Menjadi sumbangan literatur karya ilmiah dalam disiplin ilmu teknologi khususnya bidang jaringan komputer.
3. Mengukur tingkat kemampuan mahasiswa dalam menerapkan ilmu akademis maupun non-akademis di lingkungan masyarakat

1.3.3 Bagi Masyarakat

1. Menambah wawasan pembaca mengenai kombinasi algoritma *Term Frequency - Inverse Document Frequency* (TF-IDF) dan *Multi Layer Perceptron* (MLP) *Neural Network* dalam melakukan analisis orientasi sentimen terhadap data *twitter*
2. Membantu pembaca dalam menerapkan kombinasi algoritma *Term Frequency - Inverse Document Frequency* (TF-IDF) dan *Multi Layer Perceptron* (MLP) *Neural Network* dalam aplikasi berbasis Python.

1.4 Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas, maka dirumuskan masalah penelitian ini, yaitu : “Bagaimana mengimplementasikan algoritma *Term Frequency - Inverse Document Frequency* (TF-IDF) dan metode *Multi*

Layer Perceptron (MLP) Neural Network terhadap analisis sentimen twitter?''.

1.5 Batasan Masalah

Agar pembahasan lebih terarah berdasarkan perumusan masalah dan sesuai dengan batasan kemampuan penulis, maka batasan masalah dalam penelitian ini dibagi menjadi tiga bagian, yaitu:

1. Menggunakan metode *Term Frequency - Inverse Document Frequency* (TF-IDF) dan *Multi Layer Perceptron (MLP) Neural Network* untuk pengklasifikasian dalam penelitian ini.
2. Data yang digunakan untuk penelitian ini adalah data *tweet* dalam bahasa indonesia
3. Data yang digunakan adalah *tweet* dari pengguna *twitter* yang mengandung salah satu *keyword* berikut : #jokowi, #prabowo, @jikowi dan @prabowo.
4. *Tweet* yang digunakan hanya *tweet* yang berupa teks, tidak mengandung gambar.
5. Pengklasifikasian *tweet* berdasarkan dua kategorif yaitu positif dan negatif.
6. Sistem dibangun menggunakan bahasa pemograman Python.

1.6 Metodologi Penelitian

1.6.1 Metode Pengumpulan Data

Peneliti menggunakan metode pengumpulan data dilakukan dengan studi pusta dan studi literatur, yaitu dengan mencari buku-buku yang berkaitan serta jurnal sebagai referensi dan melakukan observasi dengan *streaming* data dari *twitter*.

1.6.2 Analisis Permasalahan

Pada tahap ini dilakukan analisis terhadap studi literatur untuk mendapatkan pemahaman mengenai metode yang akan digunakan, yaitu *Term Frequency - Inverse Document Frequency* (TF-IDF) dan *Multi*

Layer Perceptron (MLP) Neural Network untuk menyelesaikan masalah dalam analisis sentimen.

1.6.3 Perancangan Desain Sistem

Pada tahap ini dilakukan perancangan berdasarkan spesifikasi yang telah ditentukan sebelumnya. Perancangan terfokus pada software, pengumpulan data, arsitektur, dan algoritma yang digunakan. Proses perancangan dilakukan berdasarkan hasil analisis studi literatur sejenis yang telah didapatkan.

1.6.4 Implementasi

Pada tahap ini dilakukan implementasi ke dalam bentuk kode sesuai dengan analisis dan perancangan yang telah dilakukan sebelumnya.

1.6.5 Uji Coba Sistem

Pada tahap ini dilakukan uji coba sistem yang telah dibuat sesuai kriteria dan kebutuhan yang diinginkan dan memastikan apakah program berjalan sesuai dengan yang diinginkan. Uji coba sistem dilakukan berdasarkan skenario pengujian yang telah dibuat oleh penulis

1.6.6 Analisis Hasil Uji Coba Sistem

Pada tahap ini dilakukan analisis terhadap hasil kinerja sistem analisis sentimen yang akan diukur tingkat akurasi.

1.7 Sistematika Penelitian

Sistematika penulisan laporan Tugas Akhir ini terdiri dari pokok-pokok permasalahan yang dibahas pada masing-masing yang akan diuraikan menjadi beberapa bagian.

BAB 1 PENDAHULUAN

Bab ini membahas tentang gambaran umum isi tugas akhir yang meliputi latar belakang masalah, rumusan masalah, batasan masalah,

tujuan, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini menguraikan tentang pengertian dan teori-teori yang dibutuhkan sebagai landasan atau dasar dalam penelitian yang akan dilakukan.

BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan mengenai metode yang digunakan dalam penelitian yaitu metode pengumpulan data, metode pengembangan sistem dan lain-lainnya.

BAB 4 IMPLEMENTASI

Bab ini berisi tentang simulasi sistem dari perancangan sampai pengujian sistem sesuai dengan metode yang digunakan pada sistem.

BAB 5 HASIL DAN PEMBAHASAN

Bab ini membahas tentang output yang dihasilkan berdasarkan analisis perancangan dan implementasi yang dilakukan pada sistem

BAB 6 KESIMPULAN DAN SARAN

Bagian ini berisi tentang uraian mengenai kesimpulan berdasarkan penelitian yang telah dilakukan dan tentang saran-saran bagi pembaca untuk pengembangan penelitian yang telah dilakukan agar menjadi lebih baik.

BAB 2

LANDASAN TEORI

2.1 Sentimen Analisis

Sentimen analisis atau bisa disebut juga *opinion mining*, adalah bidang studi yang menganalisis opini, sentimen, evaluasi, penilaian, sikap, dan emosi orang-orang terhadap entitas seperti produk, layanan, organisasi, individu, masalah, peristiwa, topik, dan atributnya. *Opinion mining* dapat digunakan dalam berbagai kemungkinan domain/entitas dari produk dan jasa, peristiwa sosial dan politik serta kegiatan tertentu lainnya. *Opinion* atau pendapat adalah pusat dari semua aktifitas manusia karena merupakan pemberi pengaruh utama perilaku kita. Analisis sentimen dan *Opinion mining* terutama berfokus pada opini yang mengekspresikan atau menyiratkan sentimen positif atau negatif. Pada awal tahun 2000, analisis sentimen sudah mulai berkembang menjadi salah satu penelitian aktif dalam *natural language processing* (NLP). Dalam beberapa tahun terakhir, analisis sentimen pada kegiatan industri juga terus berkembang.

Banyaknya penelitian yang mengenai analisis sentimen disebabkan karena analisis sentimen memberikan hasil manfaat yang cukup besar. Salah satu dari kegunaannya adalah dapat membantu untuk mengetahui pendapat dari masyarakat atau orang lain terhadap suatu produk dengan menggunakannya sebagai alat bantu untuk mengetahui respon pendapat terhadap produk tersebut. Tidak hanya itu analisis sentimen juga dapat digunakan untuk mengamati tanggapan masyarakat terhadap berbagai isu seperti, sehingga bisa digunakan untuk mengetahui respon pada politik (Liu, 2012).

(Liu, 2012) mencontohkan permasalahan kalimat sentimen pada kasus pemilihan berikut: "(1) Saya membeli kamera Canon G12 enam bulan lalu. (2) Saya hanya menyukai itu. (3) Kualitas gambarnya menakjubkan. (4) Ketahanan pemakaian baterai juga menakjubkan. (5) Namun, istri saya berpikir itu terlalu berat baginya".

Dari tulisan tersebut dapat dijelaskan sebagai berikut (Liu, 2012) : 1. Tulisan ini memiliki pendapat tentang suatu entitas yaitu Kamera Canon G12. Pada kalimat (2) dan kalimat (3) kecendrungan pendapat positif terhadap kamera Canon G12 masing masing tentang kamera yang ia sukai dan kesetaraan gambarnya. 2. Sementara itu pada kalimat (4) menyatakan opini positif tentang daya ketahanan baterai. 3. Sedangkan kalimat (5) mengekspresikan pendapat negatif tentang berat kamera. Dari pendapat tersebut dapat dilihat terdapat dua komponen yaitu (G, s) yang mana G merupakan target atau entitas dan s adalah positif, negatif, atau netral. Misalnya, target pendapat dalam kalimat (2) adalah Canon G12, dan target pendapat dalam kalimat (3) adalah kualitas gambar dari Canon G12. Target juga disebut topik dalam literatur. Definisi opini menurut (Liu, 2012) adalah (G, s, h, t), dimana g adalah opini (atau sentimen) sasaran, s adalah sentimen tentang Target, h adalah pemegang pendapat dan t adalah waktu ketika pendapat itu diungkapkan

2.2 Twitter API

Twitter diciptakan oleh Jack Dorsey di tahun 2006 dan pertama meluncur di dunia maya saat Juli 2006 dengan alamat <http://www.twitter.com> yang masih digunakan hingga saat ini. *Twitter* memiliki *Application Programming Interface* sedemikian hingga *developer* dapat mengembangkan aplikasi sesuai dengan kebutuhannya masing-masing. Dokumentasi mengenai *twitter* API dapat dilihat pada <http://dev.twitter.com>. Ada beberapa jenis *twitter* API seperti :

1. *Twitter REST API*

Terdiri dari *Twitter REST* dan *Twitter Search*. *Twitter REST* memberikan *core data* dan *core twitter objects*. *Twitter search* berfungsi untuk melakukan pencarian mengenai suatu *instance* objek *Twitter* maupun mencari trend.

2. *Twitter Streaming API*

API ini biasa digunakan untuk penggalian data karena melalui API ini informasi bisa didapatkan secara realtime dengan volume yang sangat tinggi (Monarizqa, Nugroho, & Hantono, 2014).

2.3 *Natural Languge Processing (NLP)*

Natural Language Processing (NLP) merupakan salah satu cabang ilmu AI yang berfokus pada pengolahan bahasa natural. Bahasa natural adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh untuk diproses dan dipahami terlebih dahulu supaya maksud dari user bisa dipahami dengan baik oleh komputer. Ada berbagai terapan aplikasi dari NLP. Diantaranya adalah *Chatbot* (aplikasi yang membuat *user* bisa seolah-olah melakukan komunikasi dengan *computer*), *Stemming* atau *Lemmatization* (pemotongan kata dalam bahasa tertentu menjadi bentuk dasar pengenalan fungsi setiap kata dalam kalimat), *Summarization* (ringkasan dari bacaan), *Translation Tools* (menterjemahkan bahasa) dan aplikasi-aplikasi lain yang memungkinkan komputer mampu memahami instruksi bahasa yang diinputkan oleh *user* (<http://socs.binus.ac.id>, 2013).

Natural Language Processing (NLP) adalah salah satu bidang ilmu komputer yang merupakan cabang dari kecerdasan buatan, dan bahasa (linguistik) yang berkaitan dengan interaksi antara komputer dan bahasa alami manusia, seperti bahasa Indonesia atau bahasa Inggris. Tujuan utama dari studi NLP adalah membuat mesin yang mampu mengerti dan memahami makna bahasa manusia lalu memberikan respon yang sesuai (Alamanda, Suhery, & Brianorman, 2016).

2.4 *Text Mining*

Menurut jurnal (Herwijayanti, Ratnawati, & Muflikhah, 2018) yang mengutip dari buku (Ronen Feldman, 2007) menjelaskan *Text mining* merupakan proses analisis dalam data yang berupa teks dimana sumber data didapatkan dari dokumen. Konsep *text mining* biasanya digunakan dalam

klasifikasi dokumen tekstual dimana dokumen-dokumen tersebut akan diklasifikasikan sesuai dengan topik dokumen tersebut. Dengan bantuan *text mining* suatu artikel dapat diketahui jenis kategorinya melalui kata-kata yang terdapat pada artikel tersebut.

Kata-kata yang dapat mewakili isi dari artikel tersebut dianalisa dan dicocokkan pada basis data kata kunci yang telah ditentukan sebelumnya. Sehingga dengan adanya *text mining* dapat membantu melakukan pengelompokan suatu dokumen dengan waktu yang singkat. Tahapan dalam melakukan analisa pada *text mining* yaitu melakukan pengumpulan data kemudian melakukan ekstraksi terhadap fitur yang akan digunakan. Sedangkan menurut (Indriati & Ridok, 2016) *Text mining* adalah proses penemuan pola yang sebelumnya tidak terlihat pada dokumen atau sumber tertentu menjadi pola yang diinginkan untuk tujuan tertentu .

Text mining dapat didefinisikan sebagai suatu proses menggali informasi dimana seorang user berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen- komponen dalam *data mining* yang salah satunya adalah peringkatan dokumen. Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. Jadi, sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Adapun tugas khusus dari *text mining* antara lain yaitu pengkategorisasian teks (*text categorization*) dan pengelompokan teks (*text clustering*). Permasalahan yang dihadapi pada *text mining*.

sama dengan permasalahan yang terdapat pada data mining, yaitu jumlah data yang besar, dimensi yang tinggi, data dan struktur yang terus berubah, dan data *noise*. Perbedaan di antara keduanya adalah pada data yang digunakan. Pada data mining, data yang digunakan adalah *structured data*, sedangkan pada *text mining*, data yang digunakan *text mining* pada umumnya adalah *unstructured data*, atau minimal *semistructured*. Hal ini menyebabkan adanya tantangan tambahan pada *text mining* yaitu struktur teks yang kompleks

dan tidak lengkap, arti yang tidak jelas dan tidak standar, dan bahasa yang berbeda ditambah translasi yang tidak akurat (Sasmita & Falani, 2018).

2.5 Preprocessing

Preprocessing adalah mempersiapkan dokumen teks yang tidak terstruktur menjadi data terstruktur yang siap digunakan untuk proses selanjutnya (Ipmawati, Kusrini, & Luthfi, 2017). Tahapan *text processing* yang dilakukan, diantaranya adalah:

2.5.1 Cleansing

Cleansing, yaitu proses membersihkan *tweets* dari kata yang tidak diperlukan untuk mengurangi noise. Kata yang dihilangkan adalah karakter HTML, katakunci, ikon emosi, *hashtag* (#), *username* (@username), url (<http://situs.com>), dan email (<nama@situs.com>) (Aditya, 2015).

2.5.2 Tokenisasi

Tokenisasi adalah tugas memisahkan deretan kata di dalam kalimat, paragraf atau halaman menjadi token atau potongan kata tunggal atau *termmed word*. Pada saat bersamaan, tokenisasi juga membuang beberapa karakter tertentu yang dianggap sebagai tanda baca (Susilowati, Sabariah, & Gozali, 2015).

2.5.3 Case Folding

Case-folding adalah proses penyamaan *case* dalam sebuah dokumen. Ini dilakukan untuk mempermudah pencarian (Susilowati et al., 2015).

2.5.4 Penghilangan Stopword

Stopword didefinisikan sebagai *term* yang tidak berhubungan (*irrelevant*) dengan subyek utama dari *database* meskipun kata tersebut sering kali hadir di dalam dokumen. Berikut ini adalah contoh *stopwords* dalam bahasa Indonesia: yang, juga, dari, dia, kami, kamu, aku, saya, ini, itu, atau, dan, tersebut, pada, dengan, adalah, yaitu, ke,

tak, tidak, di, pada, jika, maka, ada, pun, lain, saja, hanya, namun, seperti, kemudian, dll (Susilowati et al., 2015).

2.5.5 Stemming

Kata-kata yang muncul di dalam dokumen sering mempunyai banyak varian morfologik. Karena itu, setiap kata yang bukan *stop-words* direduksi ke bentuk *stemmed word (term)* yang cocok. Kata tersebut distem untuk mendapatkan bentuk akarnya dengan menghilangkan awalan atau akhiran. Dengan cara ini, diperoleh kelompok kata yang mempunyai makna serupa tetapi berbeda wujud sintaktis satu dengan lainnya. Kelompok tersebut dapat direpresentasikan oleh satu kata tertentu. Sebagai contoh, kata menyebutkan, tersebut, disebut dapat dikatakan serupa atau satu kelompok dan dapat diwakili oleh satu kata umum sebut (Susilowati et al., 2015).

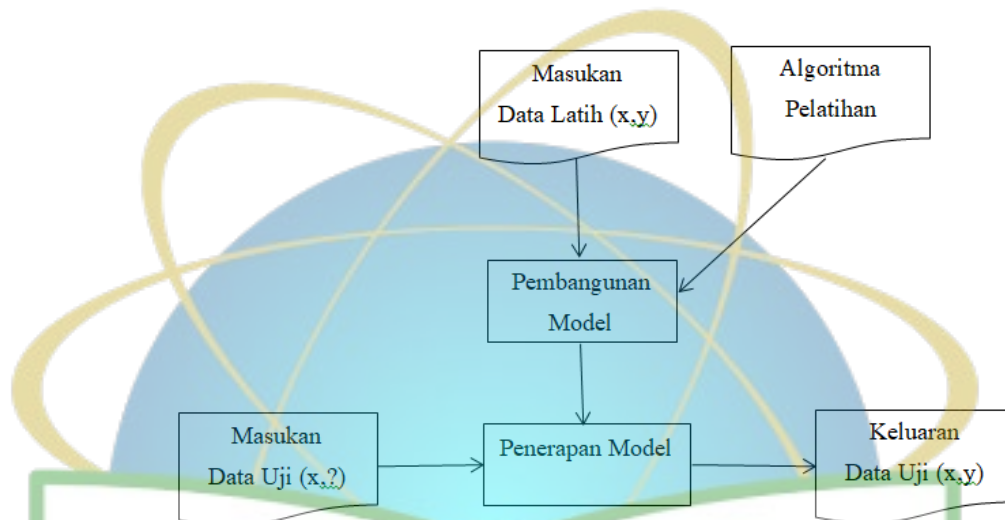
2.6 Klasifikasi

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu (Prasetyo, 2012) :

1. Pembangunan model sebagai prototipe untuk disimpan sebagai memori.
2. Penggunaan model tersebut untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya.

2.7 Model Klasifikasi

Model dalam klasifikasi mempunyai arti yang sama dengan kotak hitam, dimana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut, dan dapat memberikan jawaban sebagai keluaran dari hasil pemikirannya. Kerangka kerja (*framework*) klasifikasi ditunjukkan pada gambar berikut :



Gambar 2.1 Proses Pekerjaan Klasifikasi (Sumber : Prasetyo, 2012)

Ada banyak algoritma klasifikasi yang sudah dikembangkan oleh para peneliti, seperti *K-Nearest Neighbor*, *Artificial Neural Network*, *Support Vector Machine*, dan sebagainya. Berdasarkan cara pelatihan, algoritma-algoritma klasifikasi dapat dibagi menjadi dua macam, yaitu *eager learner* dan *lazy learner*. Algoritma-algoritma yang termasuk dalam kategori *eager learner* didesain untuk melakukan pembacaan/pelatihan/pembelajaran pada data latih agar dapat memetakan dengan benar setiap vektor masukan ke label kelas keluarannya sehingga di akhir proses pelatihan, model sudah dapat memetakan semua vektor data uji ke label kelas keluarannya dengan benar. Selanjutnya, setelah proses pelatihan tersebut selesai, model (biasanya berupa bobot atau sejumlah nilai kuantitas tertentu) disimpan sebagai memori, sedangkan semua data latihnya dibuang. Proses prediksi dilakukan dengan model yang tersimpan, tidak melibatkan data latih sama sekali. Cara ini mengakibatkan proses prediksi berjalan dengan cepat, tetapi harus dibayar dengan proses pelatihan yang lama. Algoritma-algoritma klasifikasi yang

masuk kategori ini, di antaranya, adalah *Artificial Neural Network* (ANN), *Support Vector Machine* (SVM), *Decision Tree*, *Bayesian*, dan sebagainya (Prasetyo, 2012).

2.8 Pengukuran Kinerja Klasifikasi

Sebuah sistem yang melakukan klasifikasi diharapkan dapat melakukan klasifikasi semua set data dengan benar, tetapi tidak dapat dimungkiri bahwa kinerja suatu sistem tidak bisa 100% benar sehingga sebuah sistem klasifikasi juga harus diukur kinerjanya. Umumnya, pengukuran kinerja klasifikasi dilakukan dengan matriks konfusi (*confusion matrix*).

Matriks konfusi merupakan tabel pencatat hasil kerja klasifikasi. Tabel 2.1 merupakan contoh matriks konfusi yang melakukan klasifikasi masalah biner (dua kelas), hanya ada dua kelas, yaitu kelas 0 dan 1. Setiap sel dalam matriks menyatakan jumlah record/data dari kelas i yang hasil prediksinya masuk ke kelas j . Misalnya, sel f_{11} adalah jumlah data dalam kelas 1 yang secara benar dipetakan ke kelas 1, dan f_{10} adalah data dalam kelas 1 yang dipetakan secara salah ke kelas 0.

Tabel 2.1 Matriks Konfusi Untuk Klasifikasi Dua Kelas

f_{ij}		Kelas hasil prediksi (j)	
		Kelas = 1	Kelas = 0
Kelas Asli (i)	Kelas = 1	f_{11}	f_{10}
	Kelas = 0	f_{01}	f_{00}

Berdasarkan isi matriks konfusi, kita dapat mengetahui jumlah data dari masing-masing kelas yang diprediksi secara benar, yaitu $(f_{11} + f_{00})$, dan data yang diklasifikasikan secara salah, yaitu $(f_{10} + f_{01})$. Kuantitas matriks konfusi dapat diringkas menjadi dua nilai, yaitu akurasi dan laju eror. Dengan mengetahui jumlah data yang diklasifikasikan secara benar, kita dapat mengetahui akurasi hasil prediksi, dan dengan mengetahui jumlah data yang diklasifikasikan secara salah, kita dapat mengetahui laju *error* dari

prediksi yang dilakukan. Dua kuantitas ini digunakan sebagai metrik kinerja klasifikasi. Untuk menghitung akurasi digunakan formula :

$$\text{Akurasi} = \frac{\text{Jumlah data yang diprediksi secara benar}}{\text{Jumlah prediksi yang dilakukan}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Untuk menghitung laju eror (kesalahan prediksi) digunakan formula

$$\text{Laju error} = \frac{\text{Jumlah data yang diprediksi secara salah}}{\text{Jumlah prediksi yang dilakukan}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Semua algoritma klasifikasi berusaha membentuk model yang mempunyai akurasi tinggi (laju *error* yang rendah). Umumnya, model yang dibangun dapat memprediksi dengan benar pada semua data yang menjadi data latihnya, tetapi ketika model berhadapan dengan data uji, barulah kinerja model dari sebuah algoritma klasifikasi ditentukan (Prasetyo, 2012).

2.9 Teknik Klasifikasi Analisis Sentimen

Ada dua teknik utama yang digunakan untuk menentukan orientasi sentimen (Liu, 2012) yaitu, teknik *supervised learning* dan *unsupervised learning*.

2.9.1 Supervised Learning

Supervised Learning adalah teknik yang paling banyak digunakan. Teknik ini sama dengan “*programming by example*”. Teknik ini melibatkan fase pelatihan di mana data pelatihan historis yang karakter-karakternya dipetakan ke hasil-hasil yang telah diketahui diolah dalam algoritma data mining. Proses ini melatih algoritma untuk mengenali variabel-variabel dan nilai-nilai kunci yang nantinya akan digunakan sebagai dasar dalam membuat perkiraan-perkiraan ketika diberikan data baru.

Berikut ini adalah beberapa nama algoritma tipe supervised learning (Faisal, 2017), yaitu :

1. *K-Nearest Neighbors*
2. *Naïve Bayes*
3. *Support Vector Machine*
4. *Decision Trees*

5. *Linear Regression*

6. *Neural Network*

2.9.2 *Unsupervised Learning*

Teknik pembelajaran ini tidak melibatkan fase pelatihan seperti yang terdapat pada *supervised learning*. Teknik ini bergantung pada penggunaan algoritma yang mendeteksi semua pola, seperti *associations* dan *sequences* yang muncul dari kriteria penting yang spesifik dalam data masukan. Pendekatan ini mengarah pada pembuatan banyak aturan (*rules*) yang mengkarakterisasikan penemuan *associations*, *clusters*, dan *segments*. Aturan-aturan ini kemudian dianalisis untuk menemukan hal-hal yang penting.

Berikut ini adalah beberapa nama algoritma tipe *supervised learning* (Faisal, 2017), yaitu :

1. *Association rule*
2. *K-Mean clustering*

2.10 *Term Frequency – Inverse Document Frequency (TF-IDF)*

Metode TF-IDF merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan inverse frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen (Nurjannah, Hamdani, & Astuti, 2013).

Pada algoritma TF-IDF digunakan rumus untuk menghitung bobot (W) masing-masing dokumen terhadap kata kunci dengan rumus yaitu (Melita, Amrizal, Suseno, & Dirjam, 2018):

$$W_{dt} = TF_{dt} * IDF_{ft}$$

Dimana:

W_{dt} = bobot dokumen ke-d terhadap kata ke-t

TF_{dt} = banyaknya kata yang dicari pada sebuah dokumen

IDF_{ft} = Inversed Document Frequency ($\log (\frac{N}{df})$)

N = total dokumen

df = banyak dokumen yang mengandung kata yang dicari.

2.11 Artificial Neural Network

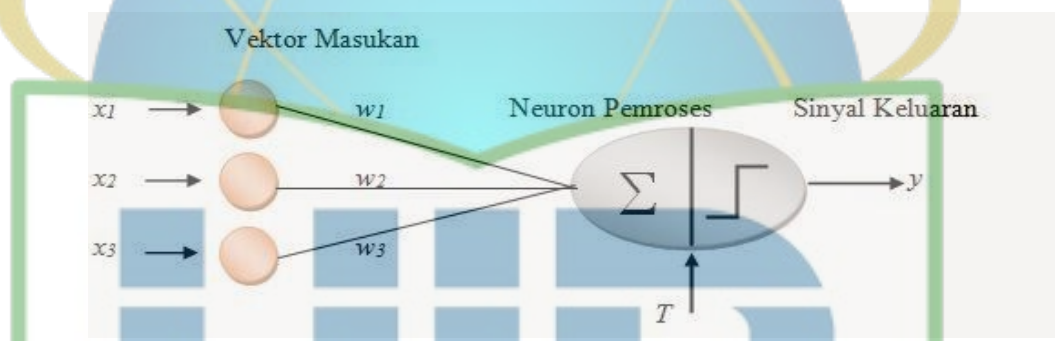
Artificial Neural Network (ANN) merupakan suatu konsep rekayasa pengetahuan dalam bidang kecerdasan buatan yang didesain dengan mengadopsi sistem saraf manusia, di mana pemrosesan utama sistem saraf manusia ada di otak. Bagian terkecil dari otak manusia adalah sel saraf yang merupakan unit dasar pemroses informasi. Unit ini sering disebut sebagai *neuron*. Ada sekitar 10 miliar *neuron* dalam otak manusia dan sekitar 60 triliun koneksi (disebut *sinapsis*) antar *neuron* dalam otak manusia (Shepherd dan Koch, 1990). Dengan menggunakan *neuron-neuron* tersebut secara simultan, otak manusia dapat memproses informasi secara paralel dan cepat, bahkan lebih cepat dari komputer tercepat saat ini.

Sebuah *neuron* biologis terdiri dari elemen-elemen sebagai berikut: badan sel (disebut *soma*), sejumlah serat yang menyalurkan informasi ke *neuron* (disebut *dendrit*), dan sebuah serat tunggal yang keluar dari *neuron* (disebut *akson*). Setiap sinyal luar yang diterima oleh dendrit akan melewati sinapsis untuk diteruskan ke *neuron* kemudian diproses di dalam soma, setelah selesai akan dikeluarkan melalui akson untuk diproses kembali oleh *neuron* yang lain ataupun keluar sebagai sinyal akhir hasil proses di otak.

Dengan menganalogikan sistem kerja otak manusia tersebut. ANN terdiri dari sebuah unit pemroses yang disebut *neuron* (akson kalau dalam

otak manusia) yang berisi *adder* dan *fungsi aktivasi*, sejumlah bobot (sinapsis dalam otak manusia), sejumlah vektor masukan (dendrit dalam otak manusia). Fungsi aktivasi yang berguna untuk mengatur keluaran yang diberikan oleh *neuron*. Desain ANN secara umum ditunjukkan oleh Gambar 5.1. Pada gambar tersebut, vektor masukan terdiri dari sejumlah nilai (iitur) yang diberikan sebagai nilai masukan pada ANN, vektor masukan tersebut ada 3 nilai (x_1, x_2, x_3) sebagai fitur dalam vektor yang akan diproses dalam ANN, masing-masing nilai masukan melewati sebuah hubungan berbobot w , kemudian semua nilai digabungkan. Nilai gabungan tersebut kemudian diproses oleh fungsi aktivasi untuk menghasilkan sinyal y sebagai keluaran. Fungsi aktivasi menggunakan sebuah nilai ambang batas untuk membatasi nilai keluaran agar selalu dalam batas nilai yang ditetapkan.

ANN menggunakan fungsi aktivasi yang dipakai untuk membatasi



Gambar 2.2 Desain ANN Secara Umum (Sumber : Prasetyo, 2012)

keluaran dari *neuron* agar sesuai dengan batasan sinyal/nilai keluarannya. Secara umum, ada empat macam fungsi aktivasi yang dipakai diberbagai jenis ANN, yaitu (Prasetyo, 2012) :

Fungsi Aktivasi :

1. Fungsi aktivasi *linear*

Fungsi aktivasi ini biasanya digunakan untuk keluaran ANN yang nilai keluarannya diskret. Jika v adalah nilai gabungan dari semua *vector* oleh penambah, sinyal keluaran y didapatkan dengan memberikan nilai v apa adanya untuk menjadi nilai keluaran. Nilai y diformulasikan dengan $y = \text{sign}(v) = v$ (Prasetyo, 2012).

Keterangan:

v = hasil perkalian input dan bobot

2. Fungsi aktivasi *step*

Jika v adalah nilai gabungan dari semua vector oleh penambah. Keluaran y didapatkan dengan melakukan pengambang (*thresholding*) pada nilai v berdasarkan nilai T yang diberikan. Nilai y diformulasikan

$$y = \text{sign}(v) = \begin{cases} 1, & \text{jika } v \geq T \\ -1, & \text{jika } v < T \end{cases}$$

Bentuk diatas disebut juga *threshold bipolar*, ada juga yang berbentuk *step/threshold biner*. Formulasinya seperti berikut (Prasetyo, 2012):

$$y = \text{sign}(v) = \begin{cases} 1, & \text{jika } v \geq T \\ 0, & \text{jika } v < T \end{cases}$$

3. Fungsi aktivasi *sigmoid biner*

ANN dengan nilai keluaran kontinu biasanya menggunakan fungsi aktivasi *sigmoid*. Nilai sinyal keluaran, y dihitung fungsi *sigmoid* kurva *sigmoid* dengan interval nilai keluaran mulai dari 0 sampai 1.

$$\text{Nilai } y \text{ diformulasikan dengan } Y = \text{sign}(v) = \frac{1}{1+e^{-av}}$$

Parameter a adalah parameter meniringan (*slope*) pada kurva *sigmoid* yang dihasilkan. Semakin besar nilai a , semakin tegak kurva yang diberikan, dan semakin kecil nilainya, semakin landai kurva diberikan. Umumnya nilai a yang digunakan adalah 1 sehingga formula yang umum digunakan menjadi (Prasetyo, 2012).

$$Y = \text{sign}(v) = \frac{1}{1 + e^{-v}}$$

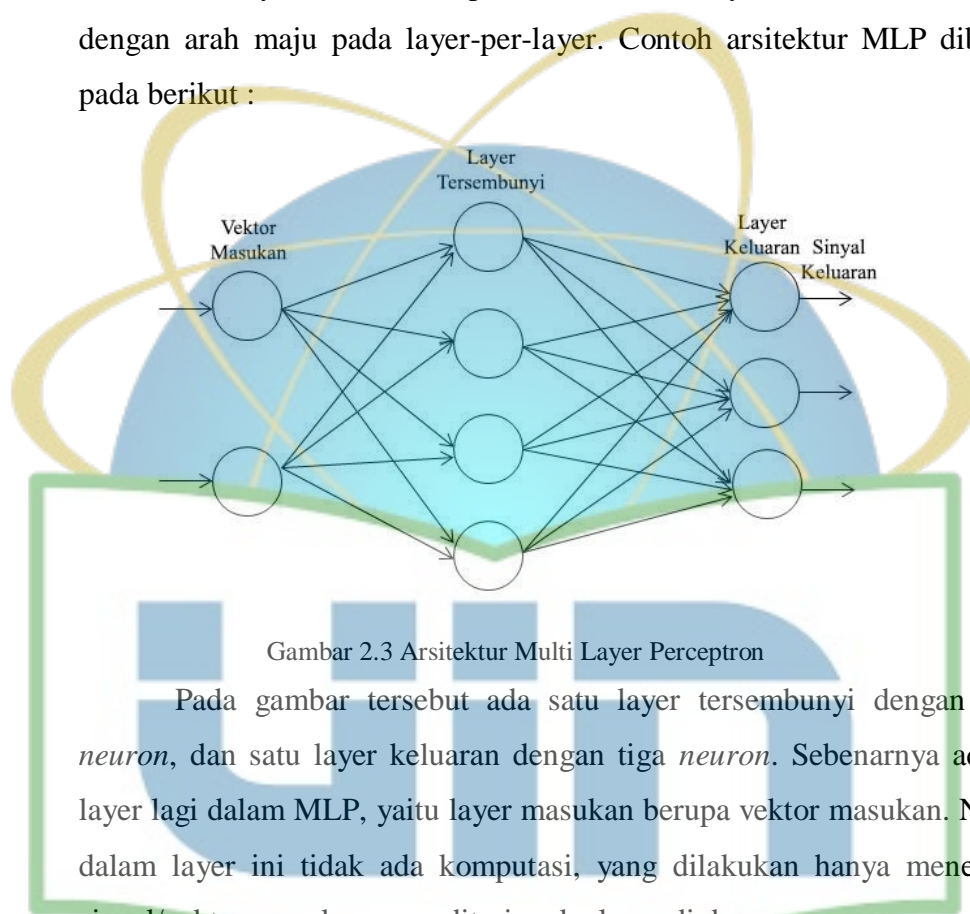
4. Fungsi aktivasi *sigmoid bipolar*

Fungsi aktivasi ini sebenarnya sama dengan *sigmoid biner*. Hanya saja batas nilai keluaran yang diberikan adalah -1 sampai 1. Fungsi aktivasi ini sangat baik digunakan untuk proses klasifikasi karena lebih tangguh dalam menangani data – data yang banyak didominasi oleh nilai nol. Nilai y diformulasikan dengan (Prasetyo, 2012).

$$Y = \text{sign}(v) = \frac{2}{1 + e^{-av}} - 1$$

2.12 Multilayer Perceptron

Multilayer Perceptron (MLP) merupakan ANN turunan dari Perceptron, berupa ANN umpan balik (*feedforward*) dengan satu atau lebih layer tersembunyi (*hidden layer*). Biasanya, jaringan terdiri atas satu layer masukan, setidaknya satu layer *neuron* komputasi di tengah (tersembunyi), dan sebuah layer *neuron* komputasi keluaran. Sinyal masukan ditambatkan dengan arah maju pada layer-per-layer. Contoh arsitektur MLP diberikan pada berikut :



Gambar 2.3 Arsitektur Multi Layer Perceptron

Pada gambar tersebut ada satu layer tersembunyi dengan empat *neuron*, dan satu layer keluaran dengan tiga *neuron*. Sebenarnya ada satu layer lagi dalam MLP, yaitu layer masukan berupa vektor masukan. Namun, dalam layer ini tidak ada komputasi, yang dilakukan hanya meneruskan sinyal/vektor masukan yang diterima ke layer di depannya.

Setiap layer dalam MLP mempunyai fungsi khusus. Layer masukan berfungsi menerima sinyal/ vektor masukan dari luar dan mendistribusikannya ke semua *neuron* dalam layer tersembunyi. Layer keluaran menerima sinyal keluaran (atau dengan kata lain, stimulus pola) dari layer tersembunyi dan memunculkan sinyal/nilai/kelas keluaran dari keseluruhan jaringan.

Neuron dalam layer tersembunyi mendeteksi fitur-fitur tersembunyi. Bobot dari *neuron* dalam layer tersembunyi merepresentasikan fitur

tersembunyi dalam vektor masukan. Fitur-fitur tersembunyi ini kemudian digunakan oleh layer keluaran dalam penentuan pola/kelas keluaran. Dengan satu layer tersembunyi, kita dapat merepresentasikan sembarang fungsi kontinu dari sinyal masukan, dan dengan dua layer tersembunyi, fungsi diskontinu pun dapat direpresentasikan.

Layer tersembunyi “menyembunyikan” keluaran yang diinginkan. *Neuron* dalam layer tersembunyi tidak dapat diamati melalui perilaku masukan/keluaran jaringan secara keseluruhan. Juga tidak ada cara yang jelas untuk mengetahui apa keluaran yang diinginkan oleh layer tersembunyi. Dengan kata lain, keluaran yang diinginkan oleh layer tersembunyi ditentukan oleh layer itu sendiri.

ANN komersial biasanya terdiri atas tiga atau bahkan empat layer, termasuk satu atau dua layer tersembunyi. Setiap layer bisa berisi 10 sampai 1.000 *neuron*. ANN eksperimental bisa mempunyai lima atau bahkan enam layer (tiga atau empat layer tersembunyi) di mana setiap layer menggunakan jutaan *neuron*, tetapi dalam kebanyakan aplikasi menggunakan dua layer (satu layer tersembunyi) karena setiap penambahan satu layer akan meningkatkan beban komputasi secara eksponensial (Prasetyo, 2012).

Berikut ini adalah algoritma Multilayer Perceptron (Muliantara & Widiartha, 2001) :

1. Inisialisasi semua bobot dengan bilangan acak kecil.
2. Jika kondisi penghentian belum dipenuhi, lakukan langkah 2-8.
3. Untuk setiap pasang data pelatihan, lakukan langkah 3-8
4. Tiap unit masukan menerima sinyal dan meneruskan ke unit tersembunyi di atasnya.
5. Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$).

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$z_i = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}}$$

6. Hitung semua keluaran jaringan di unit keluaran y_k ($k = 1, 2, \dots, m$).

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-0,0347}} = 0,5$$

7. Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1, 2, \dots, m$).

$$\delta_k = (t_k - y_k)f'(y_{net_k}) = (t_k - y_k)y_k(1 - y_k), \quad t_k = target$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer dibawahnya. Hitung perubahan bobot w_{kj} dengan laju pemahaman α .

$$\Delta w_{kj} = \alpha \delta_k z_j, \quad k = 1, 2, \dots, m; j = 0, 1 \dots p$$

8. Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1$).

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj}$$

Faktor δ unit tersembunyi.

$$\delta_j = \delta_{net_j} f'(Z_{net_j}) = \delta_{net_j} z_j (1 - z_j)$$

Hitung suku perubahan bobot v_{ji}

$$\Delta v_{ji} = \alpha \delta_j x_i, \quad j = 1, 2, \dots, p; i = 1, 2, \dots, n$$

9. Hitung semua perubahan bobot. Perubahan bobot garis yang menuju ke unit keluaran, yaitu:

$$w_{kj}(baru) = w_{kj}(lama) + \Delta w_{kj}, \quad (k = 1, 2, \dots, m; j = 0, 1 \dots, p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi, yaitu:

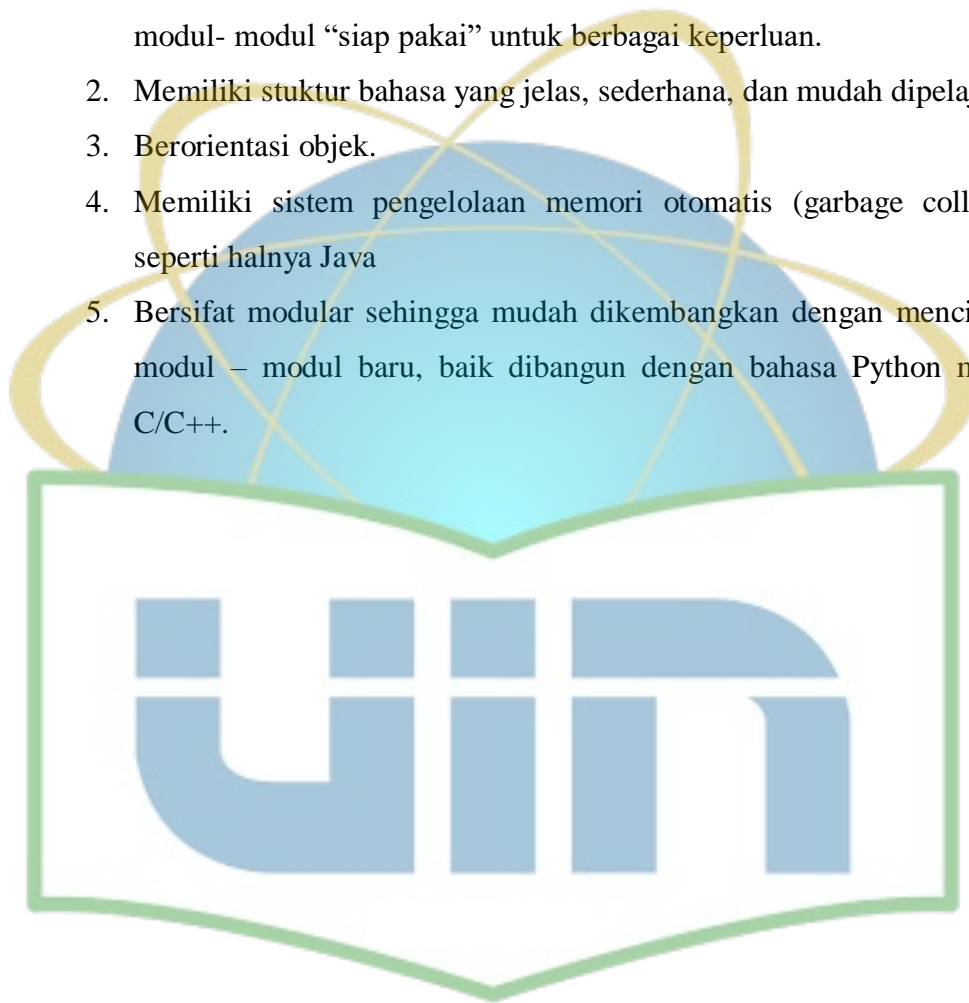
$$V_{ji}(baru) = v_{ji}(lama) + \Delta v_{ji}, \quad (j = 1, 2, \dots, p; i = 0, 1, \dots, n)$$

2.13 Python

Python adalah bahasa pemrograman interpretatif yang dianggap mudah dipelajari serta berfokus pada keterbacaan kode. Dengan kata lain, Python dikalim sebagai bahasa pemrograman yang memiliki kode-kode pemrograman yang sangat jelas, lengkap, dan mudah untuk dipahami. Python

secara umum berbentuk pemograman berorientasi objek, pemogaman imperatif, dan pemograman fungsional. Python dapat digunakan untuk berbagai keperluan pembangunan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Python memiliki beberapa fitur dan kelebihan adalah (Jubilee Enterprise, 2017) :

1. Memiliki koleksi kepustakaan yang banyak, itu artinya, telah tersedia modul- modul “siap pakai” untuk berbagai keperluan.
2. Memiliki stuktur bahasa yang jelas, sederhana, dan mudah dipelajari.
3. Berorientasi objek.
4. Memiliki sistem pengelolaan memori otomatis (garbage collection) seperti halnya Java
5. Bersifat modular sehingga mudah dikembangkan dengan menciptakan modul – modul baru, baik dibangun dengan bahasa Python maupun C/C++.



BAB 3

METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

Pada penelitian ini penulis mengumpulkan data dan informasi yang dapat menunjang proses dalam penelitian dimana proses pengumpulan data nya sebagai berikut.

3.1.1 Studi Pustaka

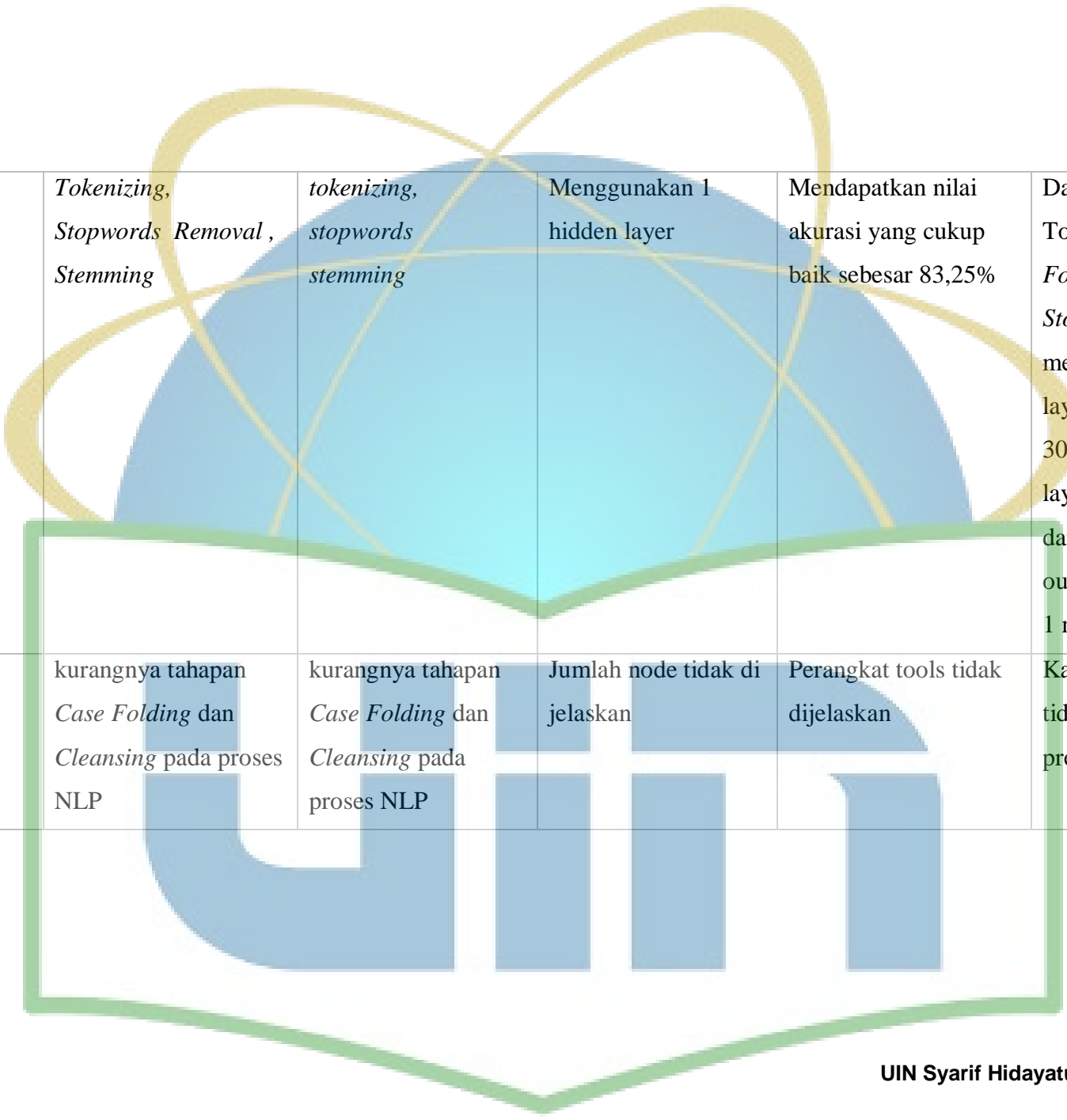
Pada tahap pengumpulan data dengan studi pustaka ini penulis melakukan pengumpulan teori-teori yang berkaitan dengan penulisan skripsi sebagai bahan untuk melengkapi penelitian ini. Sumber teori berasal dari buku referensi, hasil penelitian (jurnal dan skripsi) dan artikel-artikel terkait. Selain itu peneliti juga mengunjungi beberapa situs-situs yang terkait aplikasi *natural language processing*, *text mining*, *preprocessing*, *python*, algoritma. *Term Frequency*, *Inverse Document Frequency* (TF-IDF) dan *Multilayer Perceptron* (MLP) *Neural Network*

3.1.2 Studi Literatur Sejenis

Merupakan sumber data sekunder dalam penelitian. Studi literatur dilakukan dengan dengan cara mempelajari jurnal-jurnal hasil penelitian sebelumnya yang telah dibuat, penulis menggunakan 5 jurnal penelitian terdahulu yang sejenis untuk dijadikan acuan supaya mengetahui kelebihan dan kekurangan dari masing-masing penelitian terdahulu dan penulis melakukan perbandingan pada penelitian terdahulu tersebut yang dapat dilihat pada tabel berikut.

Tabel 3.1 Studi Literatur Sejenis

Hal yang dibandingkan	(Herwijayanti <i>et al.</i> , 2018)	(Saputro, Aristian, & Tyas, 2018)	(Al-Batah, Mrayyen, & Alzaqebah, 2018)	(Jotheeswaran & Koteeswaran, 2015)	Penelitian penulis yang sekarang
Topik sentimen yang dianalisis	Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan <i>Cosine Similarity</i>	Klasifikasi Lagu Daerah Indonesia Berdasarkan Lirik Menggunakan Metode TF-IDF Dan Naïve Bayes	Arabic Sentiment Classification using MLP Network Hybrid with Naive Bayes Algorithm	Decision Tree Based Feature Selection and Multilayer Perceptron for Sentiment Analysis	Analisis Sentimen Calon Presiden Indonesia 2019 Dari <i>Twitter</i> Menggunakan Algoritma <i>Term Frequency-Invers Document Frequency (Tf-Idf)</i> Dan Metode Multilayer Perceptron Neural
Algoritma yang digunakan	TF-IDF dan <i>Cosine Similarity</i>	TF-IDF dan <i>Naïve Bayes</i>	Multilayer Perceptron dan Naïve Bayes	Decision Tree dan Multilayer Perceptron	Multilayer Perceptron dan TF-IDF

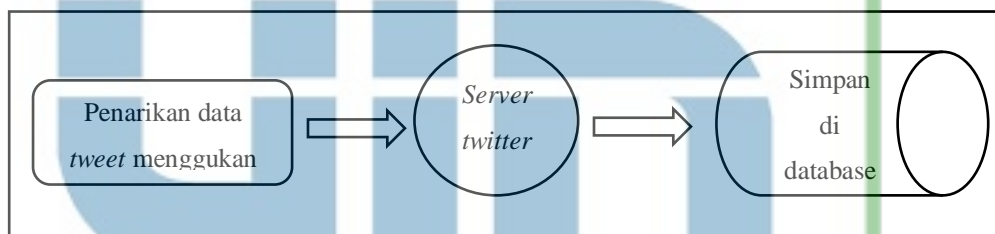


Kelebihan	<i>Tokenizing, Stopwords Removal, Stemming</i>	<i>tokenizing, stopwords stemming</i>	Menggunakan 1 hidden layer	Mendapatkan nilai akurasi yang cukup baik sebesar 83,25%	Data <i>Cleansing</i> , Tokenisasi, <i>case Folding</i> , Penghilangan <i>Stopword</i> , <i>Stemming</i> , menggunakan 1 input layer yang terdiri dari 300 node, 2 hidden layer teridiri dari f node dan 117 node, dan 1 output layer terdiri dari 1 node
Kekurangan	kurangnya tahapan <i>Case Folding</i> dan <i>Cleansing</i> pada proses NLP	kurangnya tahapan <i>Case Folding</i> dan <i>Cleansing</i> pada proses NLP	Jumlah node tidak di jelaskan	Perangkat tools tidak dijelaskan	Kata singkatan yang tidak dapat di preprosesing

3.1.3 Streaming Data Twitter

Dalam penelitian ini data yang di gunakan adalah data *tweet* dari hasil penarikan pada *server twitter* selama 2 hari, penarikan data di dapat dengan memanfaatkan fasilitas *Application Programming Interface* (API) yang telah disediakan oleh *twitter*. API digunakan untuk mengambil data *tweet* dari *server twitter* kemudian data ini dikumpulkan kedalam sebuah file dengan format json.

Pada saat melakukan pengumpulan data *tweets*, peneliti menggunakan kata kunci hashtag dengan nama calon presiden indonesia 2019 yaitu #Jokowi dan #Prabowo. Kemudian proses pengambilan data *tweet* diambil dari pengguna *twitter* yang memiliki *tweet* yang sesuai dengan kata kunci menggunakan API *twitter*. Kemudian *server twitter* akan memberikan data *tweet* yang sesuai kata kuci, selanjutnya data *tweet* didapat dari *server twitter* dapat di simpan kedalam file yang berformat json. Berikut adalah gambar proses pengumpulan data :



Gambar 3.1 Proses Pengumpulan Data

3.2 Analisis Permasalahan

Setelah penulis mendapatkan data dari studi pustaka dan penelitian sejenis sebelumnya pada tabel 3.1. maka Maka penulis mendapatkan sebuah masalah yaitu mengimplemntasikan analisis sentimen calon presiden indonesia 2019 menggunakan algoritma *term frequency - invers document frequency* (TF-IDF) dan metode *multilayer perceptron* (MLP) *neural network* untuk menentukan analisis sentimen pada *twitter* sehingga dapat diketahui hasil sentimen tersebut berisi positif atau negatif dan didapatkan nilai akurasi.

Metode yang diajukan penulis untuk menentukan analisis sentimen calon presiden indonesia 2019 menggunakan algoritma *term frequency - invers document frequency* (TF-IDF) dan metode *multilayer perceptron* (MLP) *neural network* pada *twitter* terdiri dari beberapa proses. Proses-proses yang akan dilakukan adalah sebagai berikut: pengumpulan *tweet* dari publik pengguna *twitter*, selanjutnya memasuki tahap pre-processing, hasil dari pre-processing dilakukan perhitungan bobot pada setiap *term* menggunakan TF-IDF, selanjutnya dilakukan proses klasifikasi dengan algoritma *multilayer perceptron neural network* pada tahap ini dilakukan untuk mengklasifikasikan menjadi 2 kategori sentimen yaitu positif dan negatif. Setelah didapat hasil sentimen selanjutnya akan dihitung tingkat akurasi.

3.3 Perancangan Desain Sistem

Pada tahap ini, penulis akan melakukan perancangan sistem dalam menerapkan masing-masing metode pada sistem analisis sentimen yang akan digunakan. hal-hal yang dilakukan adalah :

1. Pengumpulan data *tweet* dari *twitter*.
2. Membagi data *tweet* menjadi data latih dan data uji.
3. Preprocessing.
4. Pembobotan *Term Frequency - Inverse Document Frequency* (TF-IDF).
5. Membuat arsitektur *Multilayer Perceptron* (MLP) *Neural Network*.

3.4 Implementasi

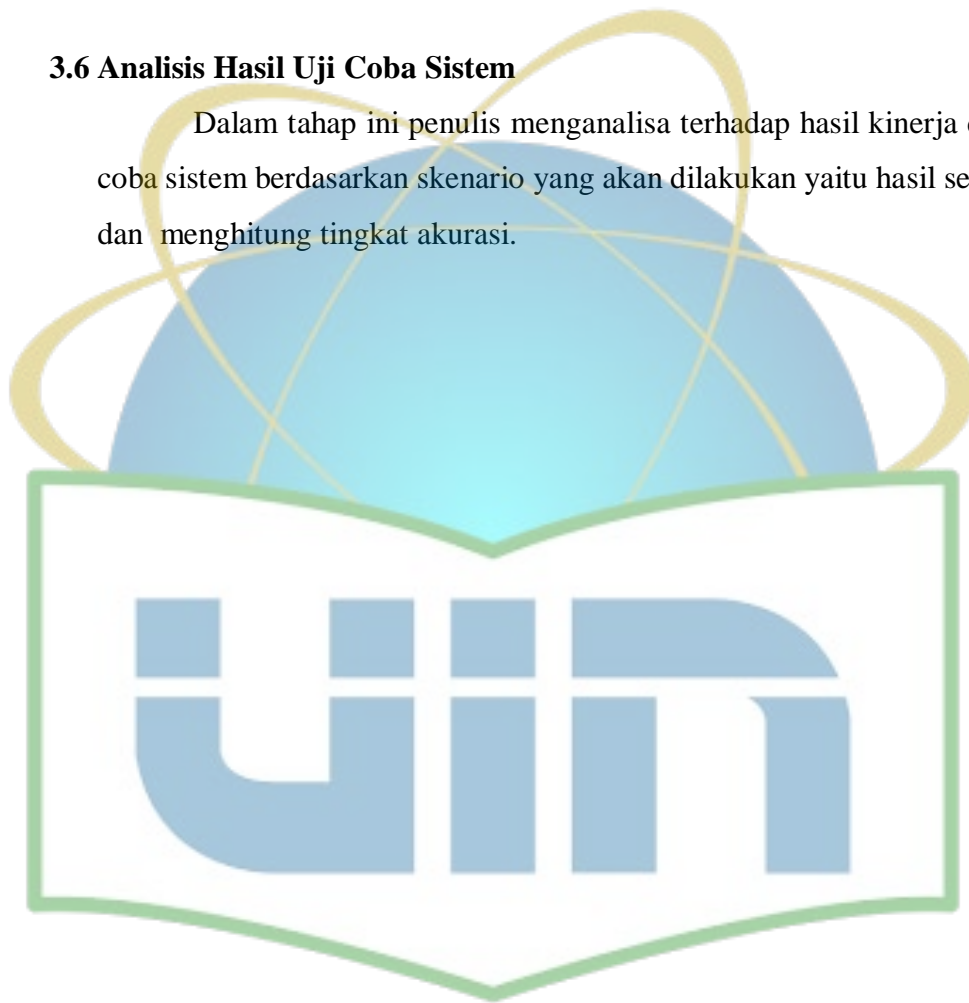
Pada tahap ini dilakukan implementasi sistem sesuai ke dalam bentuk kode sesuai dengan analisis dan perancangan yang telah dilakukan sebelumnya. Tahap ini dilakukan agar terwujudnya penelitian yang penulis lakukan terhadap sistem analisis sistemen calon presiden indonesia 2019. Bahasa pemrograman yang digunakan adalah dengan menggunakan bahasa python.

3.5 Uji Coba Sistem

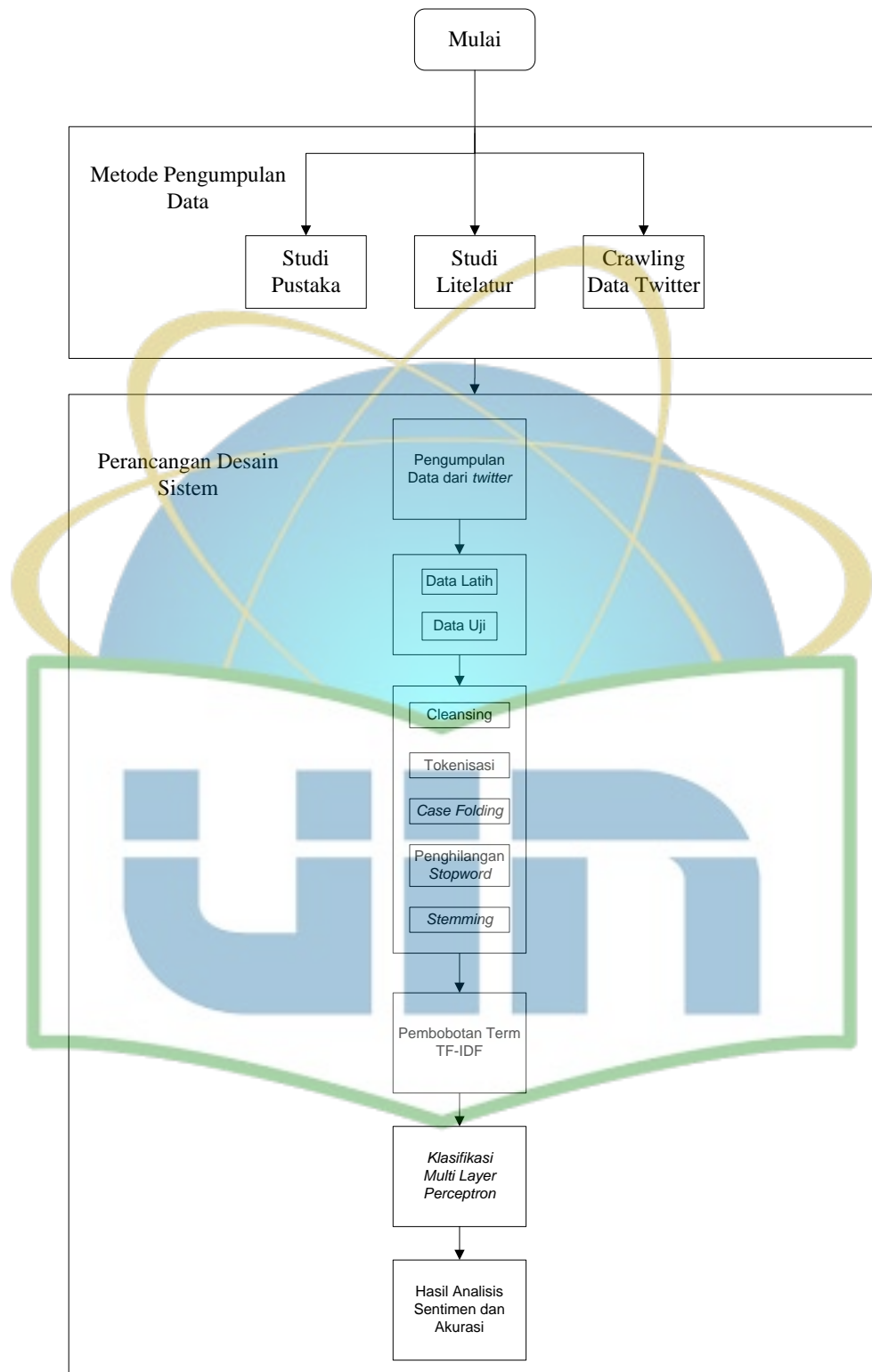
Pada tahap ini dilakukan uji coba terhadap sistem dilakukan sesuai dengan skenario pengujian. Skenario pengujian sistem dilakukan yaitu dengan memasukkan jumlah data berbeda pada setiap skenario. Banyaknya skenario adalah 3 skenario. Dan akan diukur hasil sentimen dan menghitung tingkat akurasi setiap skenario yang akan dilakukan.

3.6 Analisis Hasil Uji Coba Sistem

Dalam tahap ini penulis menganalisa terhadap hasil kinerja dari uji coba sistem berdasarkan skenario yang akan dilakukan yaitu hasil sentimen dan menghitung tingkat akurasi.



3.7 Alur Penelitian



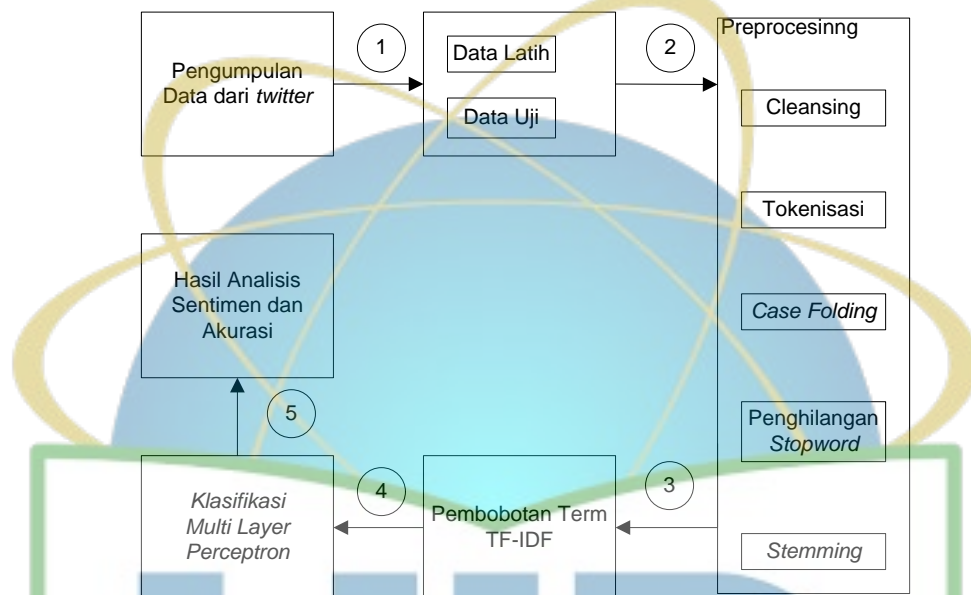
Gambar 3.2 Alur Penelitian

BAB 4

IMPLEMENTASI

4.1 Perancangan Desain Sistem

Arsitektur analisis sentimen calon presiden indonesia 2019 yang akan digunakan pada penelitian adalah sebagai berikut :



Gambar 4.1 Arsitektur Sistem Analisis Sentimen

Pada arsitektur diatas membahas tentang langkah-langkah yang dilakukan untuk analisis sentimen calon presiden indonesia 2019. Adapun langkah-langkahnya adalah sebagai berikut: pengumpulan data komentar publik dari *twitter*, data komentar memasuki preprocessing, hasil preprocessing dilakukan perhitungan bobot pada setiap *term* menggunakan tf-df, selanjutnya dilakukan proses klasifikasi dengan algortima Multilayer perceptron pada tahap ini dilakukan untuk mengklasifikasikan sentimen. langkah-langkah tersebut dilakukan maka akan diperoleh hasil sentimen . sentimen dibagi jadi 2 kategori yaitu positif dan negatif. Selanjutny akan dihitung tingkat nilai akurasi.

4.1.1 Pengumpulan Data

Pengumpulan data yang digunakan dalam penelitian ini merupakan kumpulan data *tweet* yang di peroleh dengan menggunakan stream *twitter* API yang disediakan oleh *twitter* dengan menggunakan bahasa pemograman python. Dalam melakukan stream *twitter* API dibutuhkan *keys* dan *access token* sebagai autentifikasi dengan cara mendaftarkan program yang akan dibuat pada aplikasi developer *twitter* pada situs <https://developer.twitter.com/>. Dalam source ini juga dituliskan kata kunci “#Jokowi”, #Prabowo, @Prabowo dan @Jokowi. Seluruh *tweet* yang telah diunduh disimpan kedalam dokumen .csv untuk dilakun proses selanjutnya. Berikut adalah source code stream *twitter* API untuk mendapatkan *tweet* dengan kata kunci yang tadi disebutkan.

Langkah pertama dalam tahap ini adalah memasukan *keys* dan *access token* untuk dapat terhubung dengan API *Twitter*. Adapun isi dari script tersebut sebagai berikut :

```
access_token = "401870074-
hDmVvARsnI6gWnJuC9ykvq4y653hv4zvgMAVIbPP"
access_secret =
"HaGprBweldnBhluS5B5oR4lidSzeuMXWRx0UIXVXzabJI"
consumer_key = "SfmswzxcYr43vuHbVmg9Fkleh"
consumer_secret =
"IMUwcE5jqy0ir4tL1VECqjrbgIiaNxnauo1m19H4wb5rEe5kui"
```

Source Code 4.1 Filter Data

Setelah mendapatkan hak akses API *Twitter*, untuk mendapatkan data *tweet* dibutuhkan kata kunci y dalam penginputan *keyword* (kata kunci) sebagai memfilter data yang sesuai. Berikut adalah script yang berisi kata kunci yakni #Jokowi, #Prabowo, @jokowi dan @prabowo.

```

import tweepy
import csv #Import csv

access_token = "401870074-
hDmVvARsnI6gWnJuC9ykvq4y653hv4zvgMAVIbPP"
access_secret =
"HaGprBweldnBhluS5B5oR4lidSzeuMXWRx0UIXVXzabJI"
consumer_key = "SfmswzxcYr43vuHbVmg9Fkleh"
consumer_secret =
"IMUwcE5jqy0ir4tL1VECqjrbgIiaNxnaUo1m19H4wb5rEe5kui
"

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)
api = tweepy.API(auth, wait_on_rate_limit=True)

csvFile = open('#prabowo.csv', 'a')
csvWriter = csv.writer(csvFile)

for tweet in tweepy.Cursor(api.search,
                             q = "#prabowo",
                             #since = "2018-10-19",
                             #until = "2018-10-20",
                             lang = "").items():

    csvWriter.writerow([tweet.created_at,
tweet.text.encode('utf-8')])
    print (tweet.created_at, tweet.text)
csvFile.close()

```

Source Code 4.2 Stream API Twitter

4.1.2 Data Latih dan Data Uji

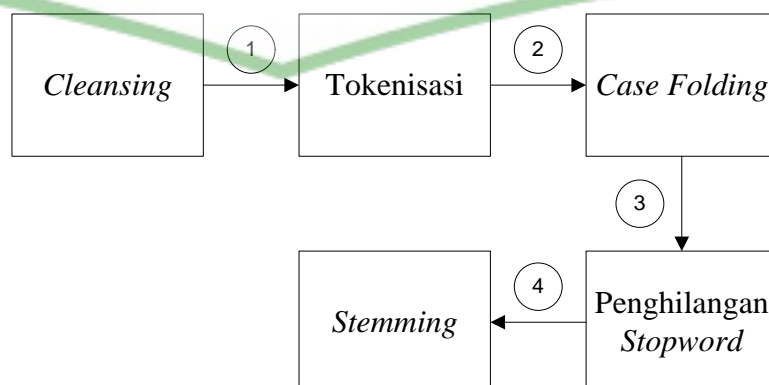
Data yang digunakan dalam penelitian ini terdiri dari dua jenis, yaitu data latih dan data uji. Data latih yang digunakan diambil dari kumpulan data *tweet* yang telah dilebeli sesuai sentimennya secara manual, label dibagi kedalam sentimen yaitu positif dan negatif. Kemudian data uji yang digunakan adalah kumpulan data *tweet* yang belum memiliki label. Berikut adalah contoh dalam pelabelan secara manual :

Tabel 4.1 Contoh Pelabelan Data

<i>Tweet</i>	Label
@jokowi @prabowo \n2 sahabat sejati jokowi-prabowo ada kalanya saling mendukung ada kalanya juga saling bersaing\n#JokowiAmin	Positif
@prabowo Semoga bapak terpilih memimpin negeri kami.. balasan terbaik bagi rakyat adalah kepemimpinan berbasis kejujuran...	Positif

4.1.3 Preprocessing

Preprocessing adalah mempersiapkan dokumen teks yang tidak terstruktur menjadi data terstruktur yang siap digunakan untuk proses selanjutnya. *Preprocessing* terdiri dari lima proses yaitu *cleansing*, tokenisasi, case folding, penghilangan *stopword*, dan *stemming*.



Gambar 4.2 Preprocessing

a. *Cleansing*

Cleansing dilakukan untuk menghilangkan delimiter koma (,), titik(.), seluruh tanda baca, angka dalam *tweet* dan beberapa komponen khas yang biasa ada di *tweet*, yakni username (@username), URL, karakter HTML, dan hashtag(#) karena tidak memiliki pengaruh apapun dalam proses analisis sentimen, maka komponen – komponen tersebut akan dihilangkan dengan tujuan untuk mengurangi *noise*. Adapun contoh *cleansing* dari data *tweet* yang sudah didapat sebagai berikut :

Tabel 4.2 Contoh Cleansing

No.	Teks <i>Tweet</i>	Hasil <i>Cleansing</i>
1	@jokowi @prabowo \n2 sahabat sejati jokowi-prabowo ada kalanya saling mendukung ada kalanya juga saling bersaing\n#JokowiAmin	sahabat sejati jokowi prabowo ada kalanya saling mendukung ada kalanya juga saling bersaing
2	@prabowo Semoga bapak terpilih memimpin negeri kami.. balasan terbaik bagi rakyat adalah kepemimpinan berbasis kejujuran...	Semoga bapak terpilih memimpin negeri kami balasan terbaik bagi rakyat adalah kepemimpinan berbasis kejujuran

Berikut adalah script untuk melakukan *cleansing* :

```

import re,string

def preprocess(tweet):
    def hapus_tanda(tweet):
        tanda_baca = set(string.punctuation)
        tweet = ''.join(ch for ch in tweet if ch not in
            tanda_baca)
        return tweet
    def hapus_katadouble(s):#look for 2 or more
        repetitions of character and replace with the
        character itself

```

```

pattern = re.compile(r"(\.)\{1,}", re.DOTALL)
return pattern.sub(r"\1\1", s)
tweet=re.sub(r'http\S+', '',tweet)
tweet=re.sub('@[^\s]+', '',tweet)
tweet = re.sub(r'#([^\s]+)', r'\1', tweet)
tweet=hapus_tanda(tweet)
tweet=re.sub(r'\w*\d\w*', '',tweet).strip()
tweet=hapus_katadouble(tweet)

```

Source Code 4.3 *Cleansing*b. *Tokenisasi*

Tokenisasi adalah sebuah proses yang dilakukan untuk memisahkan deretan kata di dalam kalimat, paragraf atau halaman menjadi token atau potongan kata tunggal atau *termmed word*. Pada saat bersamaan, tokenisasi juga membuang beberapa karakter tertentu yang dianggap sebagai tanda baca. Adapun contoh tokenisasi dari data *tweet* yang sudah didapat sebagai berikut :

Tabel 4.3 Contoh Tokenisasi

No.	Teks <i>Tweet</i>	Hasil Tokenisasi
1	sahabat sejati jokowi prabowo ada kalanya saling mendukung ada kalanya juga saling bersaing	‘sahabat’, ‘sejati’, ‘jokowi’, ‘prabowo’, ‘ada’, ‘kalanya’, ‘saling’, ‘mendukung’, ‘ada’, ‘kalanya’, ‘juga’, ‘saling’, ‘bersaing’
2	Semoga bapak terpilih memimpin negeri kami balasan terbaik bagi rakyat adalah kepemimpinan berbasis kejujuran	‘Semoga’, ‘bapak’, ‘terpilih’, ‘memimpin’, ‘negeri’, ‘kami’, ‘balasan’, ‘terbaik’, ‘bagi’, ‘rakyat’, ‘adalah’, ‘kepemimpinan’, ‘berbasis’, ‘kejujuran’

Berikut adalah script untuk melakukan tokenisasi:

```
import nltk

class Preproses:
    def tokenize(self, tweet):
        token = nltk.word_tokenize(tweet)
        return token
```

Source Code 4.4 Tokenisasi

c. Case Folding

Case folding dilakukan untuk mengubah seluruh ukuran huruf pada kata menjadi suatu bentuk ukuran huruf yang sama. Karena tidak semua *tweet* konsisten dalam penggunaan ukuran huruf. *Case folding* dilakukan dengan mengubah kata menjadi *lower case* atau huruf kecil. Adapun contoh *case folding* dari data *tweet* yang sudah didapat sebagai berikut :

Tabel 4.4 Contoh Case Folding

No.	Teks <i>Tweet</i>	Hasil <i>Case folding</i>
1	sahabat sejati jokowi prabowo ada kalanya saling mendukung ada kalanya juga saling bersaing	sahabat sejati jokowi prabowo ada kalanya saling mendukung ada kalanya juga saling bersaing
2	Semoga bapak terpilih memimpin negeri kami balasan terbaik bagi rakyat adalah kepemimpinan berbasis kejujuran	semoga bapak terpilih memimpin negeri kami balasan terbaik bagi rakyat adalah kepemimpinan berbasis kejujuran

Berikut adalah script untuk melakukan *Case folding*:

```
def preprocess(tweet):
    tweet.lower()
```

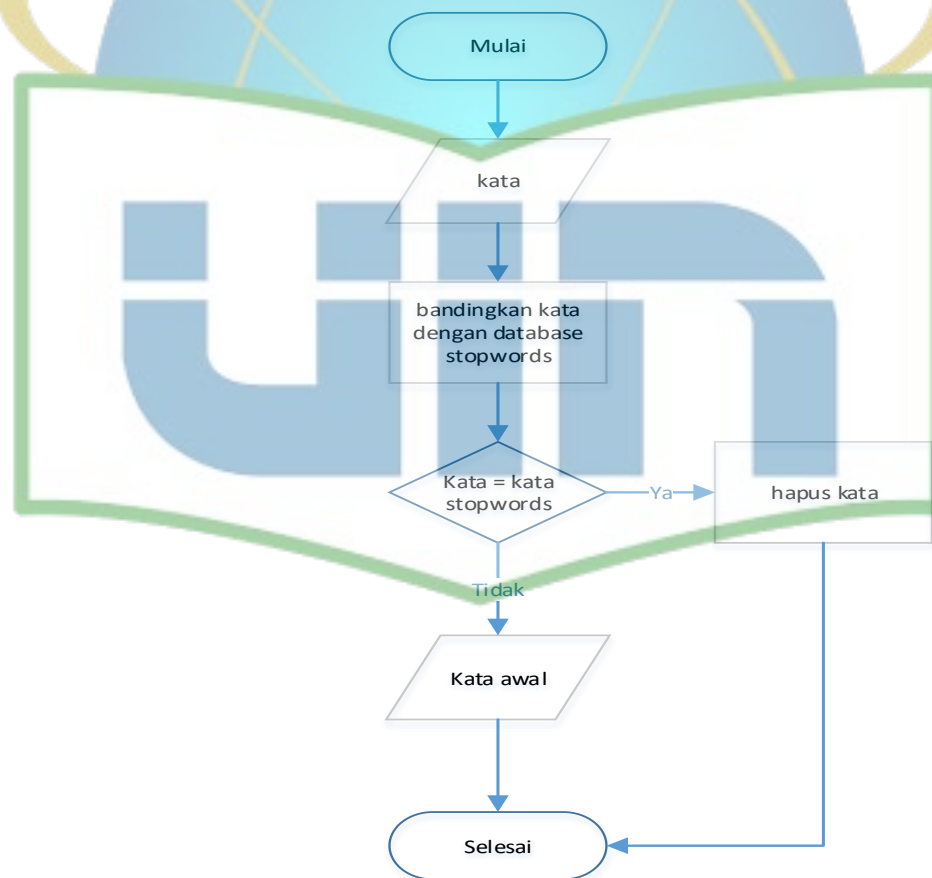
Source Code 4.5 Case Folding

d. Penghilangan *Stopword*

Tahapan *Stopword*, adalah proses menghapus kata kata yang dianggap tidak penting dan tidak berpengaruh terhadap proses kategorisasi. Contohnya adalah kata penghubung seperti 'dan', 'atau', 'di', 'ke', dan seterusnya. Adapun langkah-langkah dalam melakukan *stopword* adalah sebagai berikut :

1. Hasil dari tahapan *tokenisasi* berupa kata per kata akan dijadikan masukan.
2. Setiap kata akan dibandingkan dengan kata-kata yang ada pada *database stopwords*.
3. Apabila kata yang dibandingkan terdapat sama, maka kata tersebut akan dihapus. Namun jika tidak sama maka kata tersebut tidak dihapus.

Berikut adalah gambar *flowchart* pada Proses *stopwords* dibawah ini :



Adapun contoh Penghilangan *stopword* dari data *tweet* yang sudah didapat sebagai berikut :

Tabel 4.5 Contoh Stopword

No.	Teks <i>Tweet</i>	Hasil <i>Stopword</i>
1	sahabat sejati jokowi prabowo ada kalanya saling mendukung ada kalanya juga saling bersaing	sahabat sejati jokowi prabowo kalanya saling mendukung kalanya saling bersaing
2	semoga bapak terpilih memimpin negeri kami balasan terbaik bagi rakyat adalah kepemimpinan berbasis kejujuran	semoga bapak terpilih memimpin negeri balasan terbaik bagi rakyat kepemimpinan berbasis kejujuran

Berikut adalah script untuk melakukan Penghilangan *Stopword*:

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory
import StopWordRemoverFactory

def preprocess(tweet):
    factory1 = StopWordRemoverFactory()
    stopword = factory1.create_stop_word_remover()
    text = stopword.remove(text)
```

Source Code 4.6 Penghilangan *Stopword*

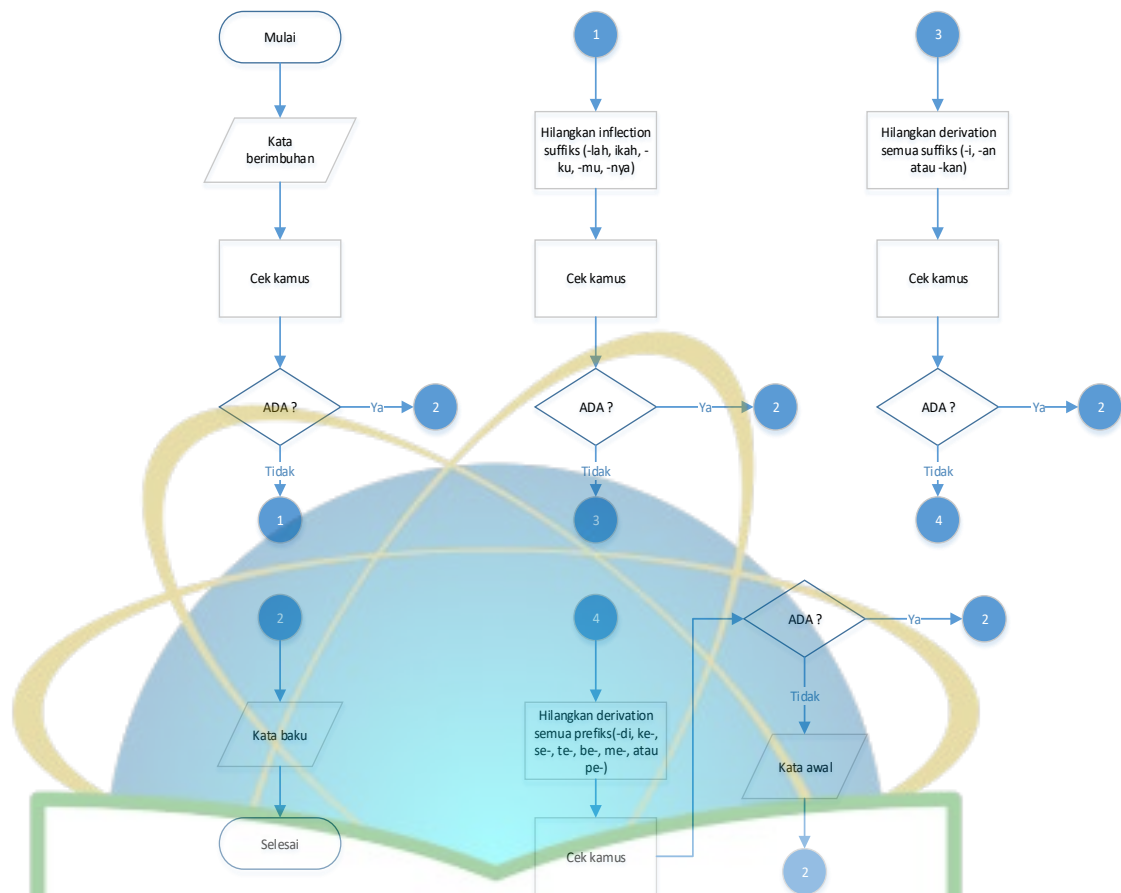
e. *Stemming*

Tahapan *Stemming*, adalah proses untuk mengubah kata yang berimbuhan menjadi stem (kata dasar) dari kata hasil *stopword*. Tahapan *stemming* dilakukan menggunakan algoritma nazief dan adriani dijelaskan berikut ini :

1. Langkah pertama adalah memeriksa apakah kata tersebut merupakan akar kata (root) terdapat dalam daftar akar kata (root). Jika kata tersebut merupakan akar kata, maka proses dihentikan pada tahap pertama ini.

2. Menghilangkan Inflection Suffixes (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”). Jika kata berupa particles (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possesive Pronouns (“-ku”, “-mu”, atau “-nya”).
3. Menghilangkan derivational suffix (imbuhan turunan). Hilangkan imbuhan -i, -kan, -an.
4. Menghilangkan derivational prefix (awalan turunan). Hilangkan awalan be-, di-, ke-, me-, pe-, se- dan te-.
5. Bila dari langkah 4 di atas belum ketemu juga. Maka lakukan analisis apakah kata tersebut masuk dalam tabel diambiguitas kolom terakhir atau tidak.
6. Bila dari langkah 4 di atas belum ketemu juga. Maka lakukan analisis apakah kata tersebut masuk dalam tabel diambiguitas kolom terakhir atau tidak.
7. Bila semua proses di atas gagal, maka algoritma mengembalikan kata aslinya.





Adapun contoh *stemming* dari data *tweet* yang sudah didapat sebagai berikut :

Tabel 4.6 Contoh *Stemming*

No.	Teks <i>Tweet</i>	Hasil <i>Stemming</i>
1	sahabat sejati jokowi prabowo kalanya saling mendukung kalanya saling bersaing	sahabat sejati jokowi prabowo kala saling dukung kala saling bersaing jokowi amin
2	semoga bapak terpilih memimpin negeri balasan terbaik bagi rakyat kepemimpinan berbasis kejujuran	moga bapak pilih pimpin negeri balas baik rakyat pimpin basis jujur

Berikut adalah script untuk melakukan stemming :

```
from Sastrawi.Stemmer.StemmerFactory import
StemmerFactory
def preprocess(tweet):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    tweet= stemmer.stem(tweet)
```

Source Code 4.7 Stemming

4.1.4 Pembobotan TF-IDF

Pada pembobotan *term* dilakukan proses pemberian nilai atau bobot terhadap setiap *term* yang ada pada setiap *tweet* yang telah melewati tahap proses *pre-processing*. Dalam melakukan pemberian bobot terhadap *term* ini menggunakan metode TF-IDF. Pembobotan ini bertujuan untuk memberikan nilai kepada suatu *term* yang dimana nilai dari *term* tersebut akan dijadikan sebagai input pada proses klasifikasi.

1. Tweet yang sudah melalui tahap preprocessing dan memiliki label menjadi masukan untuk perhitungan TF-IDF

Tabel 4.7 Tweet dan Label

NO	Tweet	Label
1	dukung fanatik sebar fitnah tuju politik	negatif
2	sebar isu fitnah hoax tuju politik	negatif

2. Menghitung *Term-Frequency* (TF)

Tabel 4.8 Menghitung Term Frequency

Kosa Kata	Tf		Df
	D1	D2	
Amin	1	0	1
Baik	0	1	1
balas	0	1	1
bapak	0	1	1
basis	0	1	1
bersaing	1	0	1
dukung	1	0	1
jokowi	2	0	2
jujur	0	1	1
kala	2	0	2
moga	0	1	1
negeri	0	1	1
pilih	0	1	1
pimpin	0	1	1
prabowo	1	0	1
rakyat	0	1	1
sahabat	1	0	1
saling	1	0	1
Sejati	1	0	1

3. Menghitung *Invers Document Frequency* (IDF)

Tabel 4.9 Menghitung Inverse Document Frequency

Kosa Kata	Tf		Df	Idf
	D1	D2		
Amin	1	0	1	0.277
Baik	0	1	1	0.277
balas	0	1	1	0.277
bapak	0	1	1	0.277
basis	0	1	1	0.277
bersaing	1	0	1	0.235
dukung	1	0	1	0.235
jokowi	2	0	2	0.471
jujur	0	1	1	0.277
kala	2	0	2	0.471
moga	0	1	1	0.277
negeri	0	1	1	0.277
pilih	0	1	1	0.554
pimpin	0	1	1	0.554
prabowo	1	0	1	0.235
rakyat	0	1	1	0.277
sahabat	1	0	1	0.471
saling	1	0	1	0.471
Sejati	1	0	1	0.235

4. Menghitung TF-IDF

Tabel 4.10 Menghitung TF-IDF

Kosa Kata	Tf		Df	Idf	TF-IDF	
	D1	D2			D1	D2
Amin	1	0	1	0.277	0.277	0
Baik	0	1	1	0.277	0	0.277
balas	0	1	1	0.277	0	0.277
bapak	0	1	1	0.277	0	0.277
basis	0	1	1	0.277	0	0.277
bersaing	1	0	1	0.235	0.235	0
dukung	1	0	1	0.235	0.235	0
jokowi	2	0	2	0.471	0.471	0
jujur	0	1	1	0.277	0	0.277
kala	2	0	2	0.471	0.471	0
moga	0	1	1	0.277	0	0.277
negeri	0	1	1	0.277	0	0.277
pilih	0	1	1	0.554	0	0.554
pimpin	0	1	1	0.554	0	0.554
prabowo	1	0	1	0.235	0.235	0
rakyat	0	1	1	0.277	0	0.277
sahabat	1	0	1	0.471	0.471	0
saling	1	0	1	0.471	0.471	0
Sejati	1	0	1	0.235	0.235	0

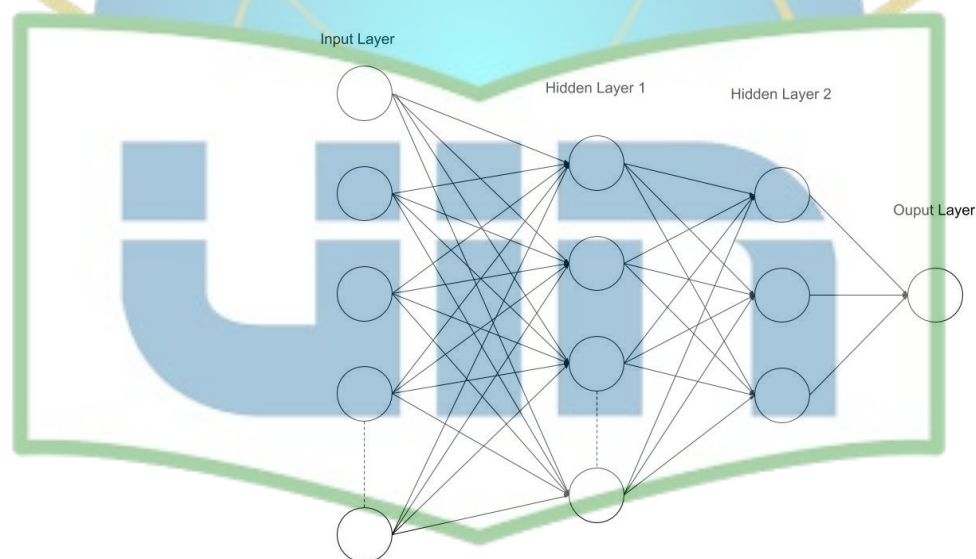
Berikut adalah script untuk melakukan TF-IDF :

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(analyzer='word',
max_features=300,max_df=0.5,use_idf=True, norm='l2')
itung_tfidf = vectorizer.fit_transform(X_train)
```

Source Code 4.8 TF-IDF

4.1.5 Klasifikasi dengan Algoritma *Multilayer Perceptron*

Klasifikasi yang dilakukan dengan menggunakan algoritma *multi layer perceptron* (MLP) dengan menggunakan bobot dari setiap *term* yang dipilih dari *tweet- tweet* yang sudah didapat. Nilai yang digunakan adalah nilai TF-IDF dari setiap kata yang memiliki bobot tertinggi. Berikut adalah arsitektur dan parameter *multilayer perceptron* yang digunakan dalam melakukan klasifikasi :



Gambar 4.3 Rancangan Arsitektur Jaringan Multi Layer Perceptron

Tabel 4.11 Parameter Multilayer Perceptron

No	Parameter	Keterangan
1	Input layer	300 node
2	Hidden layer 1	117 node

3	Hidden layer 2	117 node
4	Output layer	1 node
5	Fungsi aktivasi	Sigmoid biner
6	Learning rate	0.05

4.1.5.1 Menghitung *Multilayer Perceptron*

Berikut perhitungan manual dari metode *multilayer perceptron* (MLP). Jika diketahui nilai tf-idf dan labelnya sebagai berikut:

Table 4.1 Nilai TF-IDF dan Label

Kosa kata	Nilai TF-IDF
Amin	0.277
Baik	0.277
balas	0.277
bapak	0.277
basis	0.277
bersaing	0.235
dukung	0.235
jokowi	0.471
jujur	0.277
kala	0.471
moga	0.277
negeri	0.277
pilih	0.554
pimpin	0.554
prabowo	0.235
rakyat	0.277
sahabat	0.471
saling	0.471
Sejati	0.235

1. Inisialisasi semua bobot dengan bilangan acak kecil. Inisialisasi ini dilakukan dengan cara memberikan nilai random pada masing-masing sisi yang menghubungkan antar node.
2. Kemudian menghitung nilai dari semua node input yang dikalikan dengan sisi yang menghubungkan antara node untuk mendapatkan nilai yang akan dimasukkan kedalam fungsi aktivasi sigmoid, proses ini disebut sebagai feedforward.
3. Menghitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$).

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$\begin{aligned} z_{net_1} = & (0.277) (0.09) + (0.277) (0.01) + (0.277) (0.07) + (0.277) (0.03) + (0.277) (-0.02) + (0.235) (-0.06) \\ & + (0.235) (0.05) + (0.471) (-0.01) + (0.277) (-0.05) + (0.471) (0.08) + (0.277) (0.02) + (0.277) (0.04) \\ & + (0.554) (0.06) + (0.554) (-0.09) + (0.235) (0.04) + (0.277) (0.07) + (0.471) (0.01) + (0.471) (0.08) \\ & + (0.235) (0.07) + (-0.03) = 0.026 \end{aligned}$$

$$\begin{aligned} z_{net_2} = & (0.277) (0.01) + (0.277) (-0.07) + (0.277) (-0.07) + (0.277) (0.08) + (0.277) (-0.03) + (0.235) (-0.05) \\ & + (0.235) (-0.01) + (0.471) (0.05) + (0.277) (0.01) + (0.471) (0.01) + (0.277) (-0.03) + (0.277) (0.09) \\ & + (0.554) (-0.09) + (0.554) (0.04) + (0.235) (0.02) + (0.277) (-0.01) + (0.471) (0.06) + (0.471) (-0.07) \\ & + (0.235) (0.02) + (0.04) = 0.124 \end{aligned}$$

$$\begin{aligned} z_{net_3} = & (0.277) (-0.050) + (0.277) (-0.050) + (0.277) (-0.050) + (0.277) (-0.050) + (0.235) (-0.050) \\ & + (0.235) (-0.050) + (0.471) (-0.050) + (0.277) (-0.050) + (0.471) (-0.050) + (0.277) (-0.050) \\ & + (0.277) (-0.050) + (0.277) (-0.050) + (0.235) (-0.050) + (0.554) (-0.050) + (0.554) (-0.050) \\ & + (0.235) (-0.050) + (0.277) (-0.050) + (0.471) (-0.050) + (0.471) (-0.050) \end{aligned}$$

$$\begin{aligned}
 & (-0.050) + (0.235) (-0.050) + (0.020) + (0.277) (0.080) + \\
 & (0.277) (0.080) + (0.277) (0.080) + (0.277) (0.080) + \\
 & (0.277) (0.080) + (0.235) (0.080) + (0.235) (0.080) + \\
 & (0.471) (0.080) + (0.277) (0.080) + (0.471) (0.080) + \\
 & (0.277) (0.080) + (0.277) (0.080) + (0.554) (0.080) + \\
 & (0.554) (0.080) + (0.235) (0.080) + (0.277) (0.080) + \\
 & (0.471) (0.080) + (0.471) (0.080) + (0.235) \\
 & (0.080) + (0.020) = 0.213
 \end{aligned}$$

$$\begin{aligned}
 z_{net_4} = & (0.277) (-0.070) + (0.277) (-0.070) + (0.277) \\
 & (-0.070) + (0.277) (-0.070) + (0.277) (-0.070) + (0.235) \\
 & (-0.070) + (0.235) (-0.070) + (0.471) (-0.070) + (0.277) \\
 & (-0.070) + (0.471) (-0.070) + (0.277) (-0.070) + (0.277) \\
 & (-0.070) + (0.554) (-0.070) + (0.554) (-0.070) + (0.235) \\
 & (-0.070) + (0.277) (-0.070) + (0.471) (-0.070) + (0.471) \\
 & (-0.070) + (0.235) (-0.070) + (-0.030) + (0.277) \\
 & (0.080) + (0.277) (0.080) + (0.277) (0.080) + (0.277) \\
 & (0.080) + (0.277) (0.080) + (0.235) (0.080) + (0.235) \\
 & (0.080) + (0.471) (0.080) + (0.277) (0.080) + (0.471) \\
 & (0.080) + (0.277) (0.080) + (0.277) (0.080) + (0.554) \\
 & (0.080) + (0.554) (0.080) + (0.235) (0.080) + (0.277) \\
 & (0.080) + (0.471) (0.080) + (0.471) (0.080) + (0.235) \\
 & (0.080) + (-0.030) = 0.034
 \end{aligned}$$

Hitung sigmoid

$$z_i = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}}$$

$$f(z_{net_1}) = \frac{1}{1 + e^{0.026}} = 0.506$$

$$f(z_{net_2}) = \frac{1}{1 + e^{-0.124}} = 0.531$$

$$f(z_{net_3}) = \frac{1}{1 + e^{-0.213}} = 0.213$$

$$f(z_{net_4}) = \frac{1}{1 + e^{-0.034}} = 0.034$$

4. Hitung semua keluaran jaringan di unit keluaran y_k ($k = 1, 2, \dots, m$).

$$y_{net_j} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$y_{net_j} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$= (0,213)(-0,040) + (0,034)(0,070) + 0,010$$

$$= 0,068$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-0,068}} = 0,517$$

5. Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1, 2, \dots, m$).

$$\delta_k = (t_k - y_k)f'(y_{net_k})$$

$$= (t_k - y_k)y_k(1 - y_k), \quad t_k$$

$$= target$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer dibawahnya. Hitung perubahan bobot w_{kj} dengan laju pemahaman α .

$$\Delta w_{kj} = \alpha \delta_k Z_j, \quad k = 1, 2, \dots, m; j = 0, 1 \dots p$$

$$\delta_k = (t_k - y_k)f'(y_{net_k}) = (t_k - y_k)y_k(1 - y_k)$$

$$= (1 - 0,517)0,517(1 - 0,517)$$

$$= 0,121$$

6. Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1$).

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj}$$

$$\delta_1 = (-0,)(0,05) = -0,006$$

$$\delta_2 = (-0,125)(-0,04) = 0,005$$

$$\delta_3 = (-0,125)(0,04) + (-0,125)(0,05) = 0,0011$$

$$\delta_4 = (-0,125)(0,03) + (-0,125)(0,01) = -0,002$$

7. Selanjutnya menghitung nilai error di node output.
8. Setelah itu dilakukan langkah backpropagation yaitu mengupdate nilai bobot dari masing-masing sisi antar node yang terhubung sampai mendapatkan nilai bobot yang optimal.
9. Jika kondisi penghentian pada pencarian nilai bobot optimal belum dipenuhi, lakukan langkah 2-8.
10. Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.
11. Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$).

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$z_{net_1} = (0,05)(0,30) + (0,03)(0,30) + (-0,03) = -0,006$$

$$z_{net_2} = (-0,05)(0,30) + (-0,02)(0,30) + (0,04) = 0,019$$

$$z_{net_3} = (0,03)(0,30) + (0,04)(0,30) + (0,02) = 0,041$$

$$z_{net_4} = (-0,01)(0,30) + (0,05)(0,30) + (0,01) = 0,022$$

Hitung sigmoid

$$z_i = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}}$$

$$f(z_{net_1}) = \frac{1}{1 + e^{0,006}} = 0,49$$

$$f(z_{net_2}) = \frac{1}{1 + e^{-0,019}} = 0,49$$

$$f(z_{net_3}) = \frac{1}{1 + e^{-0,041}} = 0,49$$

$$f(z_{net_4}) = \frac{1}{1 + e^{-0,022}} = 0,49$$

12. Hitung semua keluaran jaringan di unit keluaran y_k ($k = 1, 2, \dots, m$).

$$y_{net_j} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$y_{net_j} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$= (0,49)(-0,04) + (0,49)(0,05) + 0,03 = 0,0349$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-0,0349}} = 0,5$$

13. Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1, 2, \dots, m$).

$$\delta_k = (t_k - y_k) f'(y_{net_k})$$

$$= (t_k - y_k) y_k (1 - y_k), \quad t_k = target$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer dibawahnya. Hitung perubahan bobot w_{kj} dengan laju pemahaman α .

$$\Delta w_{kj} = \alpha \delta_k Z_j, \quad k = 1, 2, \dots, m; j = 0, 1 \dots p$$

$$\begin{aligned} \delta_k &= (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \\ &= (0,5)(1 - 0,5)(0 - 0,5) \\ &= -0,125 \end{aligned}$$

14. Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1$).

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj}$$

$$\delta_1 = (-0,125)(0,05) = -0,006$$

$$\delta_2 = (-0,125)(-0,04) = 0,005$$

$$\delta_3 = (-0,125)(0,04) + (-0,125)(0,05) = 0,0011$$

$$\delta_4 = (-0,125)(0,03) + (-0,125)(0,01) = -0,002$$

Faktor δ unit tersembunyi.

$$\delta_j = \delta_{net_j} f'(Z_{net_j}) = \delta_{net_j} z_j (1 - z_j)$$

$$\delta_1 = (-0,006)(0,49)(1 - 0,49) = -0,001$$

$$\delta_2 = (0,005)(0,49)(1 - 0,49) = 0,001$$

$$\delta_3 = (0,011)(0,49)(1 - 0,49) = 0,002$$

$$\delta_4 = (-0,002)(0,49)(1 - 0,49) = -0,0004$$

Hitung suku perubahan bobot v_{ji}

$$\Delta v_{ji} = \alpha \delta_j x_i, \quad j = 1, 2, \dots, p; i = 1, 2, \dots, n$$

$$\Delta v_{50} = (0,05)(-0,125)(1) = -0,00625$$

$$\Delta v_{31} = (0,05)(-0,125)(0,49) = -0,00306$$

$$\Delta v_{32} = (0,05)(-0,125)(0,49) = -0,00306$$

$$\Delta v_{10} = (0,05)(0,001)(1) = 0,00005$$

$$\Delta v_{20} = (0,05)(-0,001)(1) = -0,00005$$

$$\Delta v_{11} = (0,05)(0,001)(0,30) = 0,000015$$

$$\Delta v_{12} = (0,05)(-0,001)(0,30) = -0,000015$$

$$\Delta v_{13} = (0,05)(0,001)(0,30) = 0,000015$$

$$\Delta v_{14} = (0,05)(-0,001)(0,30) = -0,000015$$

$$\Delta v_{30} = (0,05)(0,001)(1) = 0,00005$$

$$\Delta v_{40} = (0,05)(-0,001)(1) = -0,00005$$

$$\Delta v_{21} = (0,05)(0,001)(0,30) = 0,000015$$

$$\Delta v_{22} = (0,05)(-0,001)(0,30) = -0,000015$$

$$\Delta v_{23} = (0,05)(0,001)(0,30) = 0,000015$$

$$\Delta v_{24} = (0,05)(-0,001)(0,30) = -0,000015$$

15. Hitung semua perubahan bobot. Perubahan bobot garis yang menuju ke unit keluaran, yaitu:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj}, \quad (k = 1, 2, \dots, m; j = 0, 1, \dots, p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi, yaitu:

$$V_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji}, \quad (j = 1, 2, \dots, p; i = 0, 1, \dots, n)$$

$$v_{50} = (0,03) + (-0,006) = 0,024$$

$$v_{31} = (-0,04) + (-0,003) = -0,043$$

$$v_{32} = (0,05) + (-0,003) = 0,047$$

$$v_{10} = (-0,03) + (0,00005) = -0,02995$$

$$v_{20} = (0,04) + (-0,00005) = -0,039985$$

$$v_{11} = (0,05) + (0,000015) = 0,050015$$

$$v_{12} = (-0,05) + (-0,000015) = -0,050015$$

$$v_{13} = (0,03) + (0,000015) = 0,030015$$

$$v_{14} = (-0,02) + (-0,000015) = -0,020015$$

$$v_{30} = (0,02) + (0,00005) = 0,02005$$

$$v_{40} = (0,01) + (-0,00005) = 0,01005$$

$$v_{21} = (0,03) + (0,000015) = 0,030015$$

$$v_{22} = (-0,01) + (-0,000015) = -0,010015$$

$$v_{23} = (0,04) + (0,000015) = 0,040015$$

$$v_{24} = (0,05) + (-0,000015) = 0,049985$$

Berikut adalah script untuk melakukan *Multilayer Perceptron* :

```

from sklearn.neural_network import MLPClassifier

classifier= MLPClassifier(activation='tanh',
batch_size='auto',hidden_layer_sizes=(117,117),lea
rning_rate='constant',
max_iter=300,random_state=None, solver='sgd',
verbose=2)

classifier = classifier.fit(itung_tfidf, targets)
example_counts = vectorizer.transform(X_test)
prediksiin = classifier.predict(example_counts)

```

Source Code 4.9 Multi Layer Perceptron

4.2 Uji Coba Sistem

Uji coba terhadap sistem dilakukan sesuai dengan skenario pengujian yang sudah ditentukan sebelumnya sebanyak 3 skenario. Skenario pengujian sistem dilakukan yaitu dengan memasukan data latih dengan jumlah yang berbeda-beda pada setiap skenario dan menggunakan data uji yang sama pada setiap skenario dengan jumlah data uji sebanyak 100 data.

Tabel 4.12 Skenario Data Latih

Skenario	Jumlah Data Latih		Jumlah Data Uji
	Positif	Negatif	
1	350	350	100
2	400	400	100
3	450	450	100

Berikut adalah beberapa contoh *tweet* dari masing-masing skenario sebagai berikut:

Skenario	Tweet	
	Positif	Negatif
1	kinerja positif pemerintah jokowi dalam membangun infrastruktur yang ditujukan untuk kepentingan masyarakat Papua...	antek asing kriminalisasi ulama..saya yang mendzolimi atau ada yang mendzolimi ke saya ~ Jokowi
	Cuma di era Jokowi anggaran kedaulatan pangan naik	Berbagai isu fitnah...antek asing asing PKI gak laku!!! Malah bikin elektabilitas Jokowi semakin meroket
2	Percayalah mayoritas rakyat sudah pintar tahu mana yang kerja bener dan hasilnya benar	Salah satunya didemo karyawan kertas nusantara karena gak bayar gaji dan THR
	Kalo prabowojadi Presiden lunas deh tunggakan pajaknya.	Timses prabowo bisa gk sih bicara program...?? Jangan hanya cuma nyinyir
3	ternyata anda pandai juga menganalisa...mantap	Kasih pak prabowo kadernya banyak yg suka nyinyir dan nggak bermutu cuitannya
	tidak pernah berpikir berdiri diatas satu golongan saja karena Islam adalah rahmat bagi semesta alam Simak!	Prabowo adalah elite gagal dalam segala hal lemah pikir pemuja sistem barat pro asing pelaku kejahatan.

4.3 Analisis Hasil Uji Coba

Pada tahap ini penulis akan melakukan analisis terhadap hasil kinerja analisis sentimen dengan melihat nilai akurasi dari hasil uji coba.



BAB 5

HASIL DAN PEMBAHASAN

5.1 Perangkat Penelitian

5.1.1 Perangkat Keras

Adapun spesifikasi perangkat keras yang digunakan dalam pembuatan sistem ini adalah sebagai berikut:

1. Processor AMD A8-7410 APU with AMD Radeon R5 Graphics 2.20 Ghz
2. Memory (RAM) 4 GB
3. Kapasitas Hardisk 500 GB

5.1.2 Perangkat Lunak

Selain perangkat keras, sistem juga dibuat dalam lingkungan spesifikasi perangkat lunak sebagai berikut:

1. Operating System yang digunakan adalah Windows 10 Pro.
2. Software Integrated development environment (IDE) yang digunakan adalah Anaconda.

5.2 Hasil Klasifikasi Sentimen Menggunakan Algoritma TF-IDF dan MLP

Pada sub-bab ini hasil klasifikasi analisis sentimen dilakukan proses pengecekan terhadap hasil klasifikasi analisis sentimen masyarakat terhadap calon presiden 2019 di Indonesia dengan menggunakan algoritma *Term Frequency - Inverse Document Frequency* (TF-IDF) dan metode *multi layer perceptron* (MLP) *neural network* dan hasil tingkat akurasi dari algoritma tersebut. Pada penulisan skripsi ini, hasil klasifikasi sentimen penelitian ditampilkan pada sebuah output table dan grafik dari proses pengujian skenario pada sistem yang dibangun sendiri oleh penulis menggunakan bahasa pemrograman Python.

Berikut adalah tabel hasil output klasifikasi pada setiap skenario :

Tabel 5.1 Hasil Klasifikasi Sentimen Menggunakan Algoritma TF-IDF dan MLP

Data ke-n	Sentimen analisis menggunakan algoritma TF-IDF dan MLP			Kelas Sebenarnya
	Skenario 1	Skenario 2	Skenario 3	
1	Negatif	Negatif	Negatif	Negatif
2	Negatif	Negatif	Positif	Positif
3	Negatif	Negatif	Negatif	Negatif
4	Negatif	Negatif	Positif	Negatif
5	Negatif	Negatif	Negatif	Negatif
6	Positif	Positif	Positif	Positif
7	Positif	Positif	Positif	Positif
8	Positif	Positif	Positif	Positif
9	Positif	Positif	Positif	Positif
10	Positif	Positif	Positif	Positif
11	Positif	Positif	Positif	Positif
12	Negatif	Negatif	Negatif	Negatif
13	Positif	Positif	Positif	Positif
14	Negatif	Positif	Positif	Positif
15	Positif	Positif	Positif	Positif
16	Positif	Negatif	Positif	Positif
17	Positif	Positif	Positif	Positif
18	Positif	Positif	Positif	Positif
19	Positif	Positif	Positif	Positif
20	Positif	Positif	Positif	Positif
21	Positif	Positif	Positif	Positif
22	Positif	Positif	Positif	Positif
23	Positif	Negatif	Negatif	Negatif
24	Negatif	Positif	Positif	Negatif
25	Positif	Positif	Positif	Positif

26	Positif	Positif	Positif	Positif
27	Positif	Positif	Positif	Positif
28	Negatif	Negatif	Negatif	Negatif
29	Positif	Positif	Positif	Positif
30	Positif	Positif	Positif	Positif
31	Positif	Positif	Positif	Positif
32	Negatif	Negatif	Negatif	Negatif
33	Positif	Negatif	Positif	Positif
34	Positif	Positif	Negatif	Negatif
35	Negatif	Negatif	Negatif	Negatif
36	Positif	Negatif	Positif	Positif
37	Positif	Positif	Positif	Positif
38	Negatif	Negatif	Negatif	Positif
39	Negatif	Positif	Positif	Positif
40	Positif	Negatif	Positif	Negatif
41	Positif	Positif	Positif	Positif
42	Negatif	Negatif	Negatif	Negatif
43	Positif	Positif	Positif	Positif
44	Negatif	Negatif	Positif	Positif
45	Positif	Positif	Positif	Positif
46	Negatif	Positif	Positif	Positif
47	Negatif	Negatif	Negatif	Positif
48	Positif	Positif	Positif	Positif
49	Positif	Positif	Positif	Positif
50	Positif	Positif	Positif	Negatif
51	Negatif	Negatif	Negatif	Negatif
52	Positif	Positif	Negatif	Negatif
53	Negatif	Negatif	Negatif	Negatif
54	Negatif	Negatif	Negatif	Negatif

55	Negatif	Negatif	Negatif	Negatif
56	Negatif	Negatif	Negatif	Negatif
57	Negatif	Negatif	Negatif	Negatif
58	Negatif	Negatif	Negatif	Negatif
59	Negatif	Negatif	Negatif	Negatif
60	Negatif	Negatif	Negatif	Negatif
61	Positif	Positif	Positif	Positif
62	Positif	Positif	Positif	Positif
63	Negatif	Positif	Positif	Positif
64	Positif	Positif	Positif	Positif
65	Negatif	Positif	Positif	Positif
66	Positif	Positif	Positif	Positif
67	Positif	Positif	Negatif	Positif
68	Positif	Positif	Positif	Positif
69	Positif	Positif	Positif	Positif
70	Negatif	Positif	Positif	Positif
71	Negatif	Negatif	Negatif	Negatif
72	Negatif	Negatif	Negatif	Negatif
73	Negatif	Negatif	Negatif	Negatif
74	Positif	Positif	Positif	Positif
75	Negatif	Negatif	Negatif	Negatif
76	Negatif	Positif	Negatif	Negatif
77	Negatif	Negatif	Negatif	Negatif
78	Negatif	Negatif	Positif	Negatif
79	Positif	Positif	Positif	Positif
80	Negatif	Negatif	Negatif	Negatif
81	Negatif	Negatif	Negatif	Negatif
82	Negatif	Negatif	Negatif	Negatif
83	Negatif	Negatif	Negatif	Negatif

84	Positif	Positif	Positif	Negatif
85	Negatif	Negatif	Negatif	Negatif
86	Negatif	Negatif	Negatif	Negatif
87	Negatif	Negatif	Negatif	Negatif
88	Negatif	Negatif	Negatif	Negatif
89	Negatif	Negatif	Negatif	Negatif
90	Negatif	Negatif	Negatif	Negatif
91	Negatif	Negatif	Negatif	Negatif
92	Negatif	Negatif	Negatif	Negatif
93	Positif	Positif	Positif	Negatif
94	Negatif	Negatif	Negatif	Negatif
95	Negatif	Negatif	Negatif	Negatif
96	Negatif	Negatif	Negatif	Negatif
97	Negatif	Negatif	Negatif	Negatif
98	Positif	Positif	Negatif	Positif
99	Positif	Negatif	Negatif	Positif
100	Positif	Positif	Positif	Positif

Tabel 5.2 Rekapitulasi Hasil Klasifikasi dan Pengujian Skenario

Skenario	Data latih			Data Uji	Hasil Tweet																					
	Positif	Negatif	Total Tweet																							
Skenario 1	350	350	700	100	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
					N	N	N	N	N	P	P	P	P	P	P	N	P	N	P	P	P	P	P	P	P	P
					21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40		
					P	P	P	N	P	P	P	N	P	P	P	N	P	P	N	P	P	N	N	P		
					41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60		
					P	N	P	N	P	N	N	P	P	P	N	P	N	N	N	N	N	N	N	N	N	N
					61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80		
					P	P	N	P	N	P	P	P	P	N	N	N	N	P	N	N	N	N	P	N		
					81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100		
					N	N	N	P	N	N	N	N	N	N	N	N	P	N	N	N	N	N	P	P	P	
Hasil Prediksi Keseluruhan pada Skenario 1				Positif									Negatif									Hasil Akurasi				
				a = 41									b = 7													
				d=10									c = 42											83%		
Skenario 2	400	400	800	100	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
					N	N	N	N	N	P	P	P	P	P	P	N	P	P	P	N	P	P	P	P	P	P
					21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40		
					P	P	N	P	P	P	P	N	P	P	P	N	N	P	N	N	P	N	P	N	N	
					41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60		
					P	N	P	N	P	P	N	P	P	P	N	P	N	N	N	N	N	N	N	N	N	N
					61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80		
					P	P	P	P	P	P	P	P	P	P	N	N	N	P	N	P	N	N	P	N	N	
					81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100		
					N	N	N	P	N	N	N	N	N	N	N	N	P	N	N	N	N	N	P	N	P	N
Hasil Prediksi Keseluruhan pada Skenario 2				Positif									Negatif									Hasil Akurasi				
				a = 43									b = 7													
																								85%		

					d=8								c = 42													
Skenario 3	450	450	900	100	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
					N	P	N	P	N	P	P	P	P	P	P	N	P	P	P	P	P	P	P	P	P	P
					21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40		
					P	P	N	P	P	P	P	N	P	P	P	N	P	N	N	P	P	N	P	P	P	
					41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60		
					P	N	P	P	P	P	N	P	P	P	N	N	N	N	N	N	N	N	N	N	N	N
					61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80		
					P	N	P	P	P	P	N	P	P	P	N	N	N	N	N	N	N	N	N	N	N	
					81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100		
					N	N	N	P	N	N	N	N	N	N	N	N	P	N	N	N	N	N	N	N	N	P
Hasil Prediksi Keseluruhan pada Skenario 3					Positif								Negatif								Hasil Akurasi					
					a = 46								b = 7								88%					
					d=5								c = 42													

5.2.1 Hasil Tingkat Akurasi Skenario 1

Tabel 5.3 Hasil Pengujian Skenario 1

Sentimen		Kelas hasil prediksi	
		Positif	Negatif
Kelas	Positif	a = 41	b = 7
sebenarnya	Negatif	d = 10	c = 42

Berdasarkan hasil pengujian dari tabel diatas dapat diambil nilai akurasi sebagai berikut :

$$Akurasi = \frac{41 + 42}{41 + 7 + 10 + 42} \times 100$$

$$Akurasi = \frac{83}{100} \times 100 = 83\%$$

5.2.2 Hasil Tingkat Akurasi Skenario 2

Tabel 5.4 Hasil Pengujian Skenario 2

Sentimen		Kelas hasil prediksi	
		Positif	Negatif
Kelas	Positif	a = 43	b = 7
sebenarnya	Negatif	d = 8	c = 42

Berdasarkan hasil pengujian dari tabel diatas dapat diambil nilai akurasi sebagai berikut :

$$Akurasi = \frac{43 + 42}{43 + 7 + 8 + 42} \times 100$$

$$Akurasi = \frac{85}{100} \times 100 = 85\%$$

5.2.3 Hasil Tingkat Akurasi Skenario 3

Tabel 5.5 Hasil Pengujian Skenario 3

Sentimen		Kelas hasil prediksi	
		Positif	Negatif
Kelas sebenarnya	Positif	a = 46	b = 7
	Negatif	d = 5	c = 42

Berdasarkan hasil pengujian dari tabel diatas dapat diambil nilai akurasi sebagai berikut :

$$Akurasi = \frac{46 + 42}{46 + 7 + 5 + 42} \times 100$$

$$Akurasi = \frac{88}{100} \times 100 = 88\%$$

Keterangan :

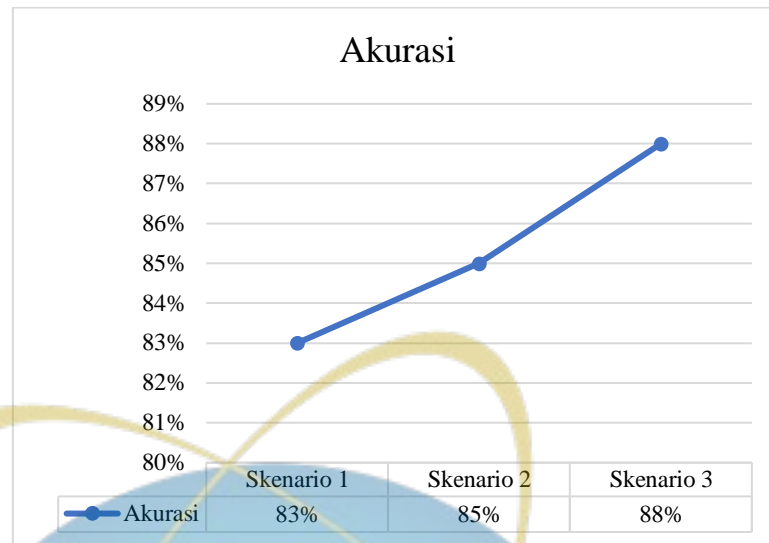
a = jumlah record positif yang diklasifikasikan sebagai positif

b = jumlah record positif yang diklasifikasikan sebagai negatif

c = jumlah record negatif yang diklasifikasikan sebagai negatif

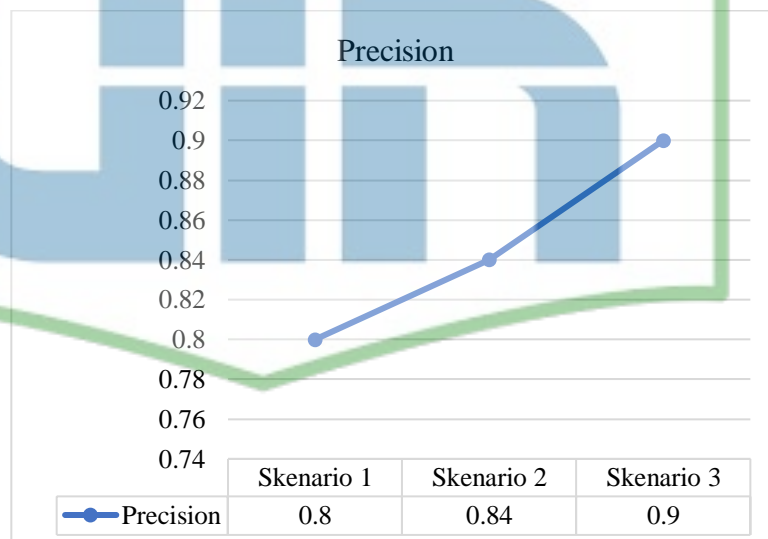
d = jumlah record negatif yang diklasifikasikan sebagai positif

Secara keseluruhan hasil eksperimen pada 3 skenario pada penelitian ini dirangkum berdasarkan gambar grafik dibawah ini.



Gambar 5.1 Gambar Hasil perhitungan akurasi

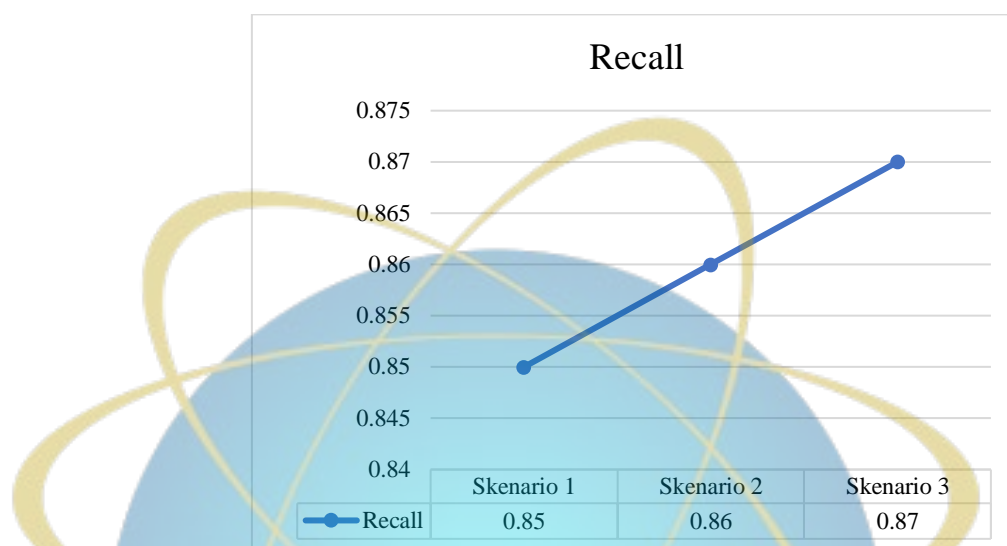
Hasil analisis sentimen pada gambar 5.1 menggunakan algoritma TF-IDF dan MLP tingkat akurasi dari ke 3 skenario memiliki nilai akurasi yang terus meningkat pada setiap skenario, nilai akurasi tertinggi berada di skenario 3 dengan nilai akurasi 88% dan nilai akurasi terendah berada di skenario 1 dengan nilai akurasi 83%.



Gambar 5.2 Hasil perhitungan precision

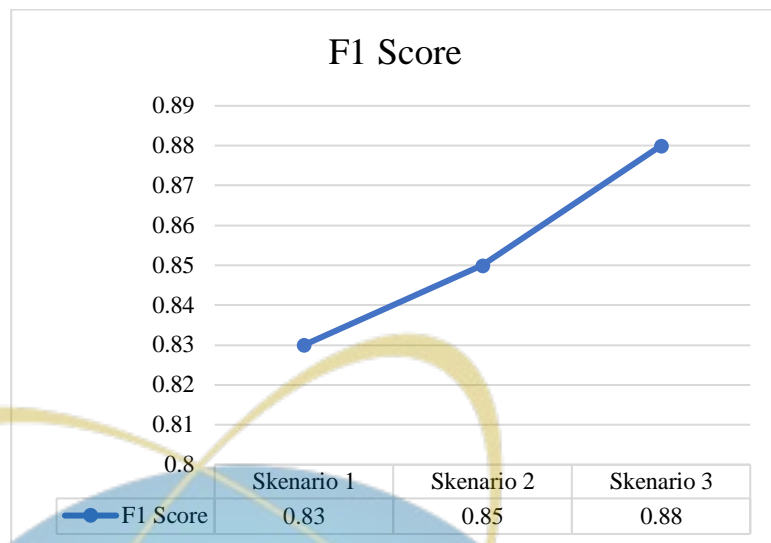
Hasil perhitungan precision pada gambar 5.2 menunjukkan nilai precision tertinggi ada pada skenario 3 yaitu sebesar 0.90 dan nilai precision terendah berada di skenario 1. Precision merupakan

persentase keakuratan hasil klasifikasi oleh sistem. Dari grafik tersebut terlihat nilai precision terus meningkat karena hasil klasifikasi dari sistem memiliki kemampuan mengklasifikasikan terbaik pada kalimat yang bernilai positif.



Gambar 5.3 Hasil perhitungan recall

Hasil perhitungan recall pada gambar 5.3 menunjukkan nilai tertinggi recall berada di skenario 3 yaitu sebesar 0.87 dan nilai recall terendah berada di skenario 1 yaitu 0.85. Semakin tinggi nilai recall yang dihasilkan maka sistem dapat memproporsikan kasus yang diidentifikasi dengan benar. Recall menunjukkan tingkat keberhasilan sistem dalam mengenali suatu kategori. Tingkat keberhasilan tersebut ditunjukkan melalui perbandingan dari jumlah data uji yang diklasifikasikan dengan benar dan jumlah data uji sebenarnya dari suatu kategori. Dari nilai rata-rata recall yang dihasilkan, dapat disimpulkan bahwa sistem dapat mengenali kategori data uji dengan cukup baik.



Gambar 5.4 Hasil perhitungan f1-score

Hasil perhitungan f1-score pada gambar 5.4 menjelaskan bahwa nilai f1-score pada ketiga skenario yang tertinggi berada di skenario 3 yaitu 0.88 dan nilai f1-score terendah berada di skenario 1 yaitu 0.83. Semakin tinggi nilai f1-score (mendekati nilai 1) yang dihasilkan maka sistem semakin bagus.

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan pembahasan yang sudah diuraikan pada bab sebelumnya, maka dapat ditarik kesimpulan bahwa algoritma *Term Frequency - Inverse Document Frequency* (TF-IDF) dan metode *Multilayer Perceptron* (MLP) *Neural Network* telah berhasil diimplementasikan terhadap analisis sentimen *twitter* dimana sistem dapat memberikan output berupa nilai sentimen positif dan negatif, dan mendapatkan nilai akurasi tertinggi sebesar 88% pada skenario 3.

6.2 Saran

Penulis menyadari bahwa aplikasi orientasi sentimen ini masih memiliki beberapa kekurangan dan keterbatasan. Oleh karena itu, ada beberapa hal yang perlu dipertimbangkan untuk mengembangkan aplikasi ini agar lebih baik, diharapkan dalam pengembangan sistem selanjutnya, sistem ini dapat diaplikasikan pada semua *platform* dan dapat dikombinasikan menggunakan algoritma klasifikasi lainnya seperti algoritma *K-Nearest Neighbor* (*K-NN*), *Support Vector Machine* (*SVM*), *CART* (*Classification and Regression Trees*).

DAFTAR PUSTAKA

- Aditya, B. R. (2015). Penggunaan Web Crawler untuk Menghimpun Tweets dengan Metode Pre-Processing Text Mining. *Infotel*, 7(2), 93–100.
- Al-Batah, M. S., Mrayyen, S., & Alzaqebah, M. (2018). Arabic Sentiment Classification using MLP Network Hybrid with Naive Bayes Algorithm. *Journal of Computer Science Original*.
<https://doi.org/10.3844/jcssp.2018.1104.1114>
- Alamanda, R., Suhery, C., & Brianorman, Y. (2016). Aplikasi Pendeteksi Plagiat Terhadap Karya Tulis Berbasis Web Menggunakan Natural Language Processing dan Algoritma Knuth-Morris-Pratt. *Jurnal Coding, Sistem Komputer Untan*, 04(1), 33–44.
- Chandani, V. (2015). Komparasi Algoritma Klasifikasi Machine Learning Dan Feature Selection pada Analisis Sentimen Review Film. *Journal of Intelligent Systems*, 1(1), 56–60. Retrieved from <http://journal.ilmukomputer.org>
- Faisal, M. R. (2017). *Seri Belajar Data Science: Klasifikasi dengan Bahasa Pemrograman R*.
- Hadna, N. M. S., Santosa, P. I., & Winarno, W. W. (2016). STUDI LITERATUR TENTANG PERBANDINGAN METODE UNTUK PROSES ANALISIS SENTIMEN DI TWITTER. *Seminar Nasional Teknologi Informasi Dan Komunikasi 2016 (SENTIKA 2016)*, 2016(Sentika), 18–19.
- Herwijayanti, B., Ratnawati, D. E., & Muflikhah, L. (2018). Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similarity. *Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(1), 306–312.
- <http://socs.binus.ac.id>. (2013). Natural Language Processing.
- Indriati, & Ridok, A. (2016). Sentiment Analysis For Review Mobile Applications Using Neighbor Method Weighted K-Nearest Neighbor (NWKNN). *Journal of Environmental Engineering & Sustainable Technology*, 03(01), 23–32.
- Ipmawati, J., Kusriani, & Luthfi, E. T. (2017). Komparasi Teknik Klasifikasi Teks Mining Pada Analisis Sentimen. *Indonesian Journal on Networking and*

- Security*, 6(1), 28–36.
- Jotheeswaran, J., & Koteeswaran, S. (2015). DECISION TREE BASED FEATURE SELECTION AND MULTILAYER PERCEPTRON FOR SENTIMENT ANALYSIS. *ARNP Journal of Engineering and Applied Sciences*, 10(14), 5883–5894.
- Liu, B. (2012). *Sentiment Analysis Analysis and Opinion Mining*. (G. Hirst & U. of Toronto, Eds.). Morgan & Claypool. <https://doi.org/10.2200/S00416ED1V01Y201204HLT016>
- Melita, R., Amrizal, V., Suseno, H. B., & Dirjam, T. (2018). Penerapan Metode Term Frequency Inverse Document Frequency (TF-IDF) dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Syarah Umdatil Ahkam). *JTI (Jurnal Teknik Informatika) UIN Syarif Hidayatullah*, 11(2), 149–164. <https://doi.org/http://dx.doi.org/10.15408/jti.v11i2.8623>
- Monarizqa, N., Nugroho, L. E., & Hantono, B. S. (2014). Penerapan Analisis Sentimen Pada Twitter Berbahasa Indonesia Sebagai Pemberi Rating. *Jurnal Penelitian Teknik Elektro Dan Teknologi Informasi*, 1(3), 151–155.
- Moraes, R., Valiati, J. F., & Neto, W. P. G. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems With Applications*, 40(2), 621–633. <https://doi.org/10.1016/j.eswa.2012.07.059>
- Muliantara, A., & Widiartha, I. M. (2001). Penerapan Multilayer Perceptron dalam Anotasi Image Secara Otomatis, 9–15.
- Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). Support Vector Machine. *Support Vector Machine (SVM) Pertama Kali Diperkenalkan Oleh Vapnik Pada Tahun 1992 Sebagai Rangkaian Harmonis Konsep-Konsep Unggulan Dalam Bidang Pattern Recognition. Sebagai Salah Satu Metode Pattern Recognition, Usia SVM Terbilang Masih Relatif Muda*. W. Retrieved from Kuliah Umum IlmuKomputer.Com
- Nurjannah, M., Hamdani, & Astuti, I. F. (2013). Penerapan Algoritma Term Frequency-Inverse Document Frequency (TF-IDF) untuk Text Mining. *Jurnal*

- Informatika Mulawarman (JIM)*, 8(3), 110–113.
<https://doi.org/10.1097/01.all.0000235903.10548.63>
- Pang, B., Lee, L., Rd, H., & Jose, S. (2002). Thumbs up ? Sentiment Classification using Machine Learning Techniques. *Association for Computational Linguistics. Thumbs*, (July), 79–86.
- Prasetyo, E. (2012). *Data Mining Konsep dan Aplikasi menggunakan MATLAB*. (N. WK, Ed.) (1st ed.). Yogyakarta: C.V ANDI OFFSET.
- Saputro, P. H., Aristian, M., & Tyas, D. L. (2018). Klasifikasi Lagu Daerah Indonesia Berdasarkan Lirik menggunakan Metode TF-IDF dan Naïve Bayes. *Jurnal Teknologi Informatika Dan Terapan*, 4(1), 45–50.
- Sasmita, R. A., & Falani, A. Z. (2018). Pemanfaatan Algoritma TF/IDF Pada Sistem Informasi Ecomplaint Handling. *JURNAL LINK*, 27(1), 27–33.
- Susilowati, E., Sabariah, M. K., & Gozali, A. A. (2015). Implementasi Metode Support Vector Machine untuk Melakukan Klasifikasi Kemacetan Lalu Lintas Pada Twitter. *E-Proceeding of Engineering*, 2(1), 1–7.
- www.apjii.or.id. (2017). Penetrasi & Perilaku Pengguna Internet Indonesia.
- Www.apjii.or.id. (2018). Penetrasi & Perilaku Pengguna Internet Indonesia.
- www.infopemilu.kpu.go.id. (2018). Daftar Pencalonan Pilpres 2019.
- www.katadata.co.id. (2016). Indonesia Pengguna Twitter Terbesar Ketiga di Dunia.