

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH**

SKRIPSI

**ANALISIS SENTIMEN DATA TWITTER TERHADAP BAKAL
CALON PRESIDEN REPUBLIK INDONESIA 2024 DENGAN
METODE *BACKPROPAGATION***



Diajukan oleh:

MUHAMMAD YUSUF

011180142

**Untuk Memenuhi Sebagian dari Syarat
Mencapai Gelar Sarjana Komputer**

PALEMBANG

2022

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH**

SKRIPSI

**ANALISIS SENTIMEN DATA TWITTER TERHADAP BAKAL
CALON PRESIDEN REPUBLIK INDONESIA 2024 DENGAN
METODE *BACKPROPAGATION***



Diajukan oleh:

MUHAMMAD YUSUF

011180142

**Untuk Memenuhi Sebagian dari Syarat
Mencapai Gelar Sarjana Komputer**

PALEMBANG

2022

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH**

HALAMAN PENGESAHAN PEMBIMBING SKRIPSI

NAMA : MUHAMMAD YUSUF
NOMOR POKOK : 011180142
PROGRAM STUDI : INFORMATIKA
JENJANG PENDIDIKAN : STRATA SATU
**JUDUL : ANALISIS SENTIMEN DATA
TWITTER
TERHADAP BAKAL CALON
PRESIDEN
REPUBLIK INDONESIA 2024 DENGAN
METODE
*BACKPROPAGATION***

Tanggal : 18 Agustus 2022
Pembimbing

Mengetahui,
Rektor

Rezania Agramanisti Azdy, S.Kom., M.Cs.
NIDN: 0215118601

Benedictus Effendi, S.T., M.T.
NIP: 09.PCT.13

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH**

HALAMAN PENGESAHAN PENGUJI SKRIPSI

NAMA : MUHAMMAD YUSUF
NOMOR POKOK : 011180142
PROGRAM STUDI : INFORMATIKA
JENJANG PENDIDIKAN : STRATA SATU
JUDUL : ANALISIS SENTIMEN DATA
TWITTER
TERHADAP BAKAL CALON
PRESIDEN
REPUBLIK INDONESIA 2024 DENGAN
METODE
BACKPROPAGATION

Tanggal : 18 Agustus 2022
Penguji 1

Tanggal : 15 Agustus 2022
Penguji 2

Hendra Effendi, S.Kom., M.Kom.
NIDN: 0217108001

Fadhila Tangguh Admojo, S.Kom., M.Cs.
NIDN: 0212088304

**Menyetujui,
Rektor**

Benedictus Effendi, S.T., M.T.

NIP: 09.PCT.13

MOTTO:

*“Coding is lifestyle.
This is the way. or, this is my way.”*

~Budi Rahardjo

*“If work so much fun,
Whyyy take a holiday.”*

~Yusuf

“Kalkulator yang di atas memang gak pernah salah ngitung!!!”

~Onno W. Purbo

“Ilmu tidak menciptakan IMAN.”

~Quraish Shihab

*“Dipersembahkan kepada Ibu, Ibu, Ibu, Ayah
dan Keluarga Tercinta.”*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan Laporan Skripsi ini dengan judul **“Analisis Sentimen Data Twitter Terhadap Bakal Calon Presiden Republik Indonesia 2024 Dengan Metode *Backpropagation*”**. Penulis ingin menghaturkan terima kasih kepada pihak-pihak yang telah memberikan dukungan selama penyusunan skripsi ini kepada.

1. Ibu Rezanisa Agramanisti Azdy, S.Kom., M.Cs., selaku pembimbing skripsi yang telah memberikan ilmu, meluangkan waktu, dan memberi masukan dalam skripsi ini hingga terselesaikan dengan baik.
2. Kedua Orangtua, Ibu Saniati dan Bapak Apendi yang senantiasa selalu mendoakan dan mendukung penulis dalam segala hal.
3. Seluruh jajaran Staff dan Dosen Informatika Institut Teknologi dan Bisnis Palcomtech.

Penulis menyadari bahwa penyusunan laporan skripsi ini masih jauh dari sempurna, untuk itu penulis tidak menutup kritik dan saran yang membangun. Semoga laporan skripsi ini berguna bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

Palembang, 20 Juli 2022

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN PEMBIMBING SKRIPSI.....	ii
HALAMAN PENGESAHAN PENGUJI SKRIPSI	iii
HALAMAN MOTTO DAN PERSEMBAHAN	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xi
ABSTRAK.....	xiii

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Ruang Lingkup	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
1.5.1 Manfaat Bagi Penulis	5
1.5.2 Manfaat Bagi Masyarakat Umum	6
1.5.3 Manfaat Bagi Akademik	6
1.6 Sistematika Penulisan	6

BAB II GAMBARAN UMUM PERANGKAT LUNAK YANG DIKEMBANGKAN

2.1 Fenomena Perangkat Lunak yang Dikembangkan	9
--	---

BAB III TINAJUAN PUSTAKA

3.1 Teori Pendukung	12
3.1.1 Data Mining	12

3.1.2 <i>Text Mining</i>	13
3.1.3 Analisis Sentimen	15
3.1.4 Twitter	15
3.1.5 <i>Term Frequency-Inverse Document Frequency</i>	16
3.1.6 Klasifikasi	17
3.1.7 <i>Backpropagation</i>	18
3.1.8 <i>Confusion Matrix</i>	23
3.1.9 <i>Text Preprocessing</i>	25
3.1.10 <i>Python</i>	27
3.1.11 <i>Natural Language Processing (NLTK)</i>	28
3.1.12 <i>Pandas</i>	28
3.1.13 <i>Preference value</i>	28
3.1.14 <i>Scikit Learn</i>	29
3.1.15 <i>Matplotlib</i>	30
3.1.16 Sastrawi	30
3.1.17 Fungsi Aktivasi <i>ReLU</i>	30
3.2 Hasil Penelitian Terdahulu	31

BAB IV METODE PENELITIAN

4.1 Lokasi dan Waktu Penelitian	35
4.1.1 Lokasi	35
4.1.2 Waktu Penelitian	35
4.2 Alur Penelitian	36
4.2.1 Pengumpulan Data	37
4.2.2 Pelabelan Data	38
4.2.3 <i>Text Preprocessing</i>	38
4.2.4 Pembobotan <i>Term Frequency–Inverse Document Frequency</i>	39
4.2.5 Permodelan <i>Backpropagation</i>	40
4.2.6 Evaluasi Model	41

BAB V HASIL DAN PEMBAHASAN

5.1	Gambaran Umum Aplikasi	42
5.2	Pengumpulan Data	43
5.3	Pelabelan Data	45
5.4	<i>Text Preprocessing</i>	46
5.5	Ekstraksi Fitur TF-IDF	57
5.6	Permodelan <i>Backpropagation</i>	60
5.6.1	<i>Encode Label</i>	69
5.6.2	Konfigurasi Model Pada Modul <i>Sklearn</i>	69
5.6.3	Pembagian Data <i>Training</i> dan Data <i>Testing</i>	70
5.7	Evaluasi Model	71
5.8	<i>Preference Value</i>	75

BAB VI PENUTUP

6.1	Kesimpulan	80
6.2	Saran.	80

DAFTAR PUSTAKA	xiv
-----------------------------	------------

HALAMAN LAMPIRAN	xivi
-------------------------------	-------------

DAFTAR GAMBAR

Gambar 2. 1	Gambaran Umum Sistem	11
Gambar 3. 1	Proses Dasar <i>Text Mining</i>	14
Gambar 3. 2	Arsitektur <i>Backpropagation Neural Network</i>	19
Gambar 4. 1	Alur Penelitian	37
Gambar 4. 2	Alur Pembobotan TF-IDF	40
Gambar 4. 3	Alur Proses Pelatihan <i>Backpropagation</i>	41
Gambar 5. 1	Inisialisasi Twitter API	44
Gambar 5. 2	<i>Crawling</i> Data Cuitan Twitter	44
Gambar 5. 3	<i>Source Code Cleaning</i>	48
Gambar 5. 4	<i>Source Code Case Folding</i>	49
Gambar 5. 5	<i>Source Code Tokenizing</i>	51
Gambar 5. 6	<i>Source Code</i> Normalisasi	52
Gambar 5. 7	Proses <i>Filtering Stopwords</i>	53
Gambar 5. 8	<i>Source Code Filtering Stopwords</i>	55
Gambar 5. 9	<i>Source Code Stemming</i>	56
Gambar 5. 10	<i>Source Code TF-IDF</i>	59
Gambar 5. 11	Arsitektur <i>Neural Network</i>	68
Gambar 5. 12	<i>Label Encoding</i>	69
Gambar 5. 13	Model BPNN <i>Sklearn</i>	71
Gambar 5. 14	<i>Confusion Matrix</i> Ganjar Pranowo	72
Gambar 5. 15	<i>Accuracy Confusion Matrix</i> Ganjar Pranowo	73
Gambar 5. 16	<i>Confusion Matrix</i> Model Prabowo Subianto	73
Gambar 5. 17	<i>Accuracy</i> Model Prabowo Subianto	73
Gambar 5. 18	<i>Confusion Matrix</i> Model Anies	74
Gambar 5. 19	<i>Accuracy</i> Model Anies	74
Gambar 5. 20	Persentase Sentimen Prabowo Subianto	77
Gambar 5. 21	Persentase Sentimen Ganjar Pranowo	77
Gambar 5. 22	Persentase Sentimen Anies Baswedan	78
Gambar 5. 23	Perbandingan <i>Preference Value</i>	79

DAFTAR TABEL

Tabel 3. 1 <i>Confusion Matrix</i>	24
Tabel 3. 2 Penelitian Terdahulu	31
Tabel 4. 1 Jadwal Penelitian	35
Tabel 5. 1 Contoh Hasil Pelabelan Secara Manual	45
Tabel 5. 2 Hasil Cleaning	47
Tabel 5. 3 Hasil Case Folding	49
Tabel 5. 4 Hasil Tokenizing	50
Tabel 5. 5 Cuitan Hasil Normalisasi	51
Tabel 5. 6 Hasil Filtering Stopwords	54
Tabel 5. 7 Cuitan Hasil Stemming	55
Tabel 5. 8 Pembobotan TF-IDF	57
Tabel 5. 9 Hasil TF-IDF untuk Klasifikasi Backpropagation	60
Tabel 5. 10 Inisialisasi Bobot dan Bias Awal	61
Tabel 5. 11 Inisialisasi Bobot dan Bias Awal Output	61
Tabel 5. 12 Perubahan Bobot (v_{ij})	65
Tabel 5. 13 Inisialisasi Bobot Input Iterasi Kedua	66
Tabel 5. 14 Inisialisasi Bobot Output Iterasi Kedua	67
Tabel 5. 15 Konfigurasi Model Pada Sklearn	70
Tabel 5. 16 Pembagian Data Training dan Data Testing	70
Tabel 5. 17 Prediksi Sentimen Dataset Prabowo Subianto	75
Tabel 5. 18 Prediksi Sentimen Dataset Ganjar Pranowo	76
Tabel 5. 19 Prediksi Sentimen Anies Baswedan	76

DAFTAR LAMPIRAN

1. Lampiran 1. *Form* Topik dan Judul (Fotokopi)
2. Lampiran 2. *Form* Konsultasi (Fotokopi)
3. Lampiran 3. Surat Pernyataan (Fotokopi)
4. Lampiran 4. *Form* Revisi Ujian Pra Sidang (Fotokopi)
5. Lampiran 5. *Form* Revisi Ujian Kompre (Asli)
6. Lampiran 6. *Listing Code*

ABSTRACT

MUHAMMAD YUSUF. *Twitter Data Sentiment Analysis About Indonesia's 2024 Presidential Candidate Using Backpropagation Method.*

Twitter is estimated to have 217 million active users by the end of 2021. This causes a flood of text data that Twitter produces through tweets or comments from its active users. The generated tweet data can be any topic. However, this research specifically discusses the 2024 Indonesian presidential candidates. The Indonesian presidential election will be held in 2024. However, several survey institutions have issued survey results regarding the electability of the presidential candidates. Some of the candidates with the highest electability are Prabowo Subianto, Ganjar Pranowo, Anies Baswedan. Many Twitter users express their opinions about the 2024 presidential candidate for Indonesia, whether they have positive, neutral, or negative sentiments. This study aims to conduct a sentiment analysis regarding the 2024 presidential candidates with the previously mentioned names. The research was conducted by crawling the tweet data related to the name of the candidate, after that the tweet data would be preprocessed, TF-IDF, and modeled using the Backpropagation algorithm. The results obtained in this study are the accuracy of each model built based on the Ganjar Pranowo, Anies Baswedan, and Prabowo Subianto dataset with Backpropagation getting an average of 81%. While the results of the evaluation based on the confusion matrix obtained an average precision of 80.3%, recall 81%, and f-measure 80%. Based on the results of the Backpropagation classification for each dataset, the preference value for Prabowo Subianto was 31%, Anies Baswedan 29%, Ganjar Pranowo 23%. So it can be seen that Prabowo Subianto has the best electability based on a Twitter tweet in July 2022.

Keywords : *Data Mining, Text Mining, Sentiment analysis, Twitter*

ABSTRAK

MUHAMMAD YUSUF. Analisis Sentimen Data Twitter Terhadap Bakal Calon Presiden Republik Indonesia 2024 Dengan Metode *Backpropagation*.

Twitter diperkirakan memiliki 217 juta pengguna aktif pada akhir tahun 2021. Hal ini menyebabkan banjirnya data teks yang diproduksi Twitter melalui cuitan ataupun komentar dari pengguna aktifnya. Data cuitan yang dihasilkan dapat berupa topik apa saja. Namun, pada penelitian ini spesifik membahas bakal calon Presiden Indonesia 2024. Pemilu Presiden Indonesia akan dilaksanakan pada tahun 2024. Beberapa lembaga survei telah mengeluarkan hasil survei mengenai elektabilitas dari para bakal calon Presiden. Nama bakal calon yang memiliki elektabilitas teratas adalah Prabowo Subianto, Ganjar Pranowo, dan Anies Baswedan. Pengguna Twitter banyak mengekspresikan pendapatnya mengenai bakal calon Presiden Republik Indonesia 2024 tersebut baik itu yang memiliki sentimen positif, netral, ataupun negatif. Penelitian ini bertujuan untuk melakukan analisis sentimen mengenai bakal calon Presiden Republik Indonesia 2024 dengan nama yang telah disebutkan sebelumnya. Penelitian diawali dengan melakukan *crawling* data cuitan Twitter yang berhubungan dengan nama bakal calon tersebut, setelah itu data cuitan akan dilakukan *preprocessing*, TF-IDF, dan membuat model dengan algoritme *Backpropagation*. Hasil yang didapatkan pada penelitian ini adalah *accuracy* masing-masing model yang dibangun berdasarkan dataset Ganjar Pranowo, Anies Baswedan, dan Prabowo Subianto dengan *Backpropagation* mendapatkan rata-rata 81%. Sedangkan hasil evaluasi yang dilakukan menggunakan *confusion matrix* didapatkan rata-rata *precision* 80,3%, *recall* 81%, dan *f-measure* 80%. Berdasarkan hasil klasifikasi *Backpropagation* untuk setiap dataset didapatkan *preference value* untuk Prabowo Subianto 31%, Anies Baswedan 29%, Ganjar Pranowo 23%. Sehingga dapat diketahui Prabowo Subianto memiliki elektabilitas paling baik berdasarkan cuitan Twitter bulan Juli 2022.

Kata Kunci : Data Mining, Text Mining, Analisis Sentimen, Twitter

BAB I

PENDAHULUAN

1.1 Latar Belakang

Beberapa tahun terakhir terdapat peningkatan jumlah pengguna jejaring sosial, salah satu jejaring sosial yang memiliki pengguna aktif paling banyak adalah Twitter. Dikutip dari (www.omnicoreagency.com) salah satu media yang menyajikan demografi jejaring sosial, Twitter diperkirakan memiliki 217 juta pengguna aktif pada akhir tahun 2021. Hal ini menyebabkan banjirnya data teks yang diproduksi Twitter melalui cuitan ataupun komentar dari pengguna aktifnya. Pengguna juga sangat dibebaskan untuk mengekspresikan pikiran, perasaan, dan juga opini mereka melalui cuitan.

Pengguna aktif Twitter cenderung beropini terhadap topik permasalahan tertentu, salah satu yang sedang hangat diperbincangkan adalah mengenai bakal calon Presiden Republik Indonesia 2024. Beberapa lembaga survei Indonesia telah mengeluarkan hasil mengenai bakal calon Presiden Republik Indonesia 2024, seperti hasil yang dikeluarkan oleh *SMRC Research*, elektabilitas Prabowo Subianto berada di tingkat paling atas dengan persentase sebesar 19,7% diikuti dengan Ganjar Pranowo dengan persentase 19,2% lalu Anies Baswedan 13,4% dan seterusnya. Lembaga Survei Nasional (LSN) juga telah merilis hasil survei mengenai hal tersebut, hasil yang didapatkan adalah Prabowo Subianto dengan presentase 21,9% Anies Baswedan 19,2% Ganjar Pranowo 18,8% Sandiaga

Uno 8,7% dan Ridwan Kamil 7,5%. Berdasarkan hasil dari survei terdapat banyak cuitan pengguna Twitter yang beropini mengenai bakal calon Presiden tersebut, ada yang menyertakan dukungannya ataupun sebaliknya terhadap bakal calon Presiden Republik Indonesia 2024. Dengan opini sentimen yang dihasilkan Twitter tersebut, dapat diketahui bagaimana masyarakat menyikapi figur bakal calon Presiden Indonesia 2024. Sebagai contoh, untuk figur Ganjar Pranowo rata-rata opini masyarakat terhadap beliau itu positif, hal tersebut menunjukkan bahwa Ganjar Pranowo mendapatkan dukungan dan juga elektabilitas yang baik dari pengguna Twitter untuk menjadi Calon Presiden Republik Indonesia 2024.

Banyaknya data dan juga informasi yang dihasilkan oleh cuitan Twitter terhadap suatu topik permasalahan tertentu tidak akan memiliki makna apabila tidak dilakukan proses penemuan pengetahuan dari informasi (*Knowledge Discovery*). *Text mining* adalah bagian dari bidang ilmu data *mining* yang terkhusus untuk mengelola dataset yang tidak terstruktur. *Text mining* juga merupakan suatu metodologi atau alat yang digunakan untuk memproses sejumlah teks agar memperoleh informasi yang berkualitas dengan menggunakan teknik statistik (Purbo, 2019).

Analisis sentimen adalah salah satu dari beberapa contoh penerapan *text mining*. Metode yang dapat digunakan untuk melakukan analisis sentimen dengan pendekatan *supervised learning* berbasis klasifikasi beberapa di antaranya adalah *Naïve Bayes*, *Support Vector Machine*, dan juga *Neural Network*.

Penelitian yang dilakukan oleh (Assuja & Saniati, 2016) yang membahas analisis sentimen Twitter menggunakan metode *Backpropagation Neural Network*, didapatkan hasil yang baik dengan akurasi yang didapat adalah 78.34% dan presisi 84.21%. Penelitian tersebut membagi polaritas menjadi tiga kelas yakni positif, negatif, dan netral. Sedangkan penelitian yang dilakukan (Ricky et al., 2020) dengan permasalahan yang dibahas adalah BPJS kesehatan, akurasi yang didapat adalah 87.14% dan presisi sebesar 86.97%.

Penelitian ini mengusulkan metode *Backpropagation Neural Network* untuk melakukan analisis sentimen terhadap bakal calon Presiden Republik Indonesia 2024 menggunakan data dari jejaring sosial Twitter. Data yang telah diolah akan dilakukan klasifikasi menjadi 3 kelas polaritas yakni positif, netral, dan negatif. Berdasarkan klasifikasi tersebut maka akan diketahui sentimen pengguna Twitter terhadap bakal calon Presiden Republik Indonesia 2024, sehingga dapat diketahui dukungan masyarakat terhadap bakal calon Presiden tersebut. Penelitian ini juga nantinya diharapkan dapat dimanfaatkan masyarakat sebagai referensi dalam memilih Presiden pada pemilu 2024 mendatang.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dipaparkan pada latar belakang sebelumnya, maka rumusan masalah pada penelitian ini yaitu:

1. Bagaimana sentimen masyarakat pengguna Twitter terhadap bakal calon Presiden Republik Indonesia tahun 2024?
2. Bagaimana metode *Backpropagation Neural Network* digunakan untuk melakukan klasifikasi cuitan pengguna Twitter mengenai bakal calon Presiden Republik Indonesia 2024?
3. Berapa akurasi yang diperoleh dengan menggunakan *Backpropagation Neural Network* dalam mengklasifikasikan cuitan mengenai bakal calon Presiden Republik Indonesia 2024?

1.3 Ruang Lingkup

Adapun ruang lingkup dalam melakukan penelitian ini yaitu:

1. Data cuitan pengguna Twitter diperoleh menggunakan Twitter API.
2. Data yang digunakan adalah cuitan pengguna Twitter berbahasa Indonesia yang dikumpulkan pada bulan Januari 2022 - Juni 2022.
3. Metode yang digunakan untuk proses klasifikasi yaitu *Backpropagation Neural Network*.
4. *Keyword* figur bakal calon Presiden Republik Indonesia 2024 diambil dari hasil survei SMRC *Research* dan Lembaga Survei Nasional (LSN) yang dibatasi hanya 3 nama yaitu, Prabowo Subianto, Ganjar Pranowo, dan Anies Baswedan.

5. Keluaran sistem akan memberikan informasi berupa persentase polaritas sentimen, serta nilai *preference value* dari setiap bakal calon.

1.4 Tujuan Penelitian

Adapun tujuan dilakukannya penelitian ini yaitu:

1. Mengetahui sentimen masyarakat pengguna Twitter terhadap bakal calon Presiden Republik Indonesia 2024.
2. Mengetahui elektabilitas bakal calon Presiden Republik Indonesia 2024 berdasarkan dukungan masyarakat pengguna Twitter.
3. Menerapkan metode *Backpropagation Neural Network* untuk melakukan klasifikasi cuitan pengguna Twitter mengenai bakal calon Presiden Republik Indonesia 2024.
4. Mengukur akurasi yang diperoleh dengan metode *Backpropagation Neural Network* dalam mengklasifikasikan cuitan pengguna Twitter mengenai bakal calon Presiden Republik Indonesia 2024.

1.5 Manfaat Penelitian

1.5.1 Manfaat Bagi Penulis

Manfaat yang didapatkan oleh penulis melalui penelitian ini adalah dapat mengaplikasikan ilmu yang didapatkan selama perkuliahan. Selain itu, penelitian ini juga memperdalam pengetahuan penulis mengenai algoritme *Backpropagation Neural Network* serta implementasinya di berbagai bidang, khususnya dalam proses klasifikasi sentimen.

1.5.2 Manfaat Bagi Masyarakat Umum

Mengetahui informasi opini masyarakat pengguna Twitter terhadap bakal calon Presiden Republik Indonesia 2024, serta mengetahui seberapa besar dukungan masyarakat terhadap calon tersebut.

1.5.3 Manfaat Bagi Akademik

Berkontribusi dalam perkembangan bidang ilmu *text mining* serta implementasinya dalam hal analisis sentimen. Menambah pengetahuan dalam pemanfaatan Twitter untuk dijadikan sebagai bahan penelitian untuk melakukan analisis sentimen terhadap topik permasalahan tertentu.

1.6 Sistematika Penulisan

Pada bagian ini, berisi struktur laporan skripsi yang akan dimulai dari bab pendahuluan sampai dengan bab penutup beserta deskripsi dari tiap bab secara ringkas. Bagian ini diperuntukkan bagi para pembaca agar lebih mudah memahami skripsi ini.

BAB I PENDAHULUAN

Bab ini berisi tentang bagaimana kondisi terkini terkait topik analisis sentimen. Disampaikan juga mengenai banyaknya opini yang diproduksi oleh sosial media terhadap topik permasalahan tertentu. Kemudian dipaparkan tentang permasalahan dan hasil yang ingin dicapai pada penelitian ini.

BAB II GAMBARAN UMUM PERANGKAT LUNAK YANG DIKEMBANGKAN

Bab ini menjelaskan fenomena mengenai permasalahan analisis sentimen, dan banyaknya opini yang diproduksi Twitter yang mengandung sentimen terhadap suatu topik permasalahan tertentu, serta deskripsi singkat mengenai sistem yang dibangun.

BAB III TINJAUAN PUSTAKA

Bab ini menjelaskan beberapa teori yang berhubungan dengan penelitian ini, serta berisi perbandingan antara penelitian saat ini dan penelitian yang telah dilakukan sebelumnya.

BAB IV METODE PENELITIAN

Bab ini menggambarkan bagaimana tahapan yang akan dilakukan untuk menyelesaikan penelitian dari awal hingga akhir. Mulai dari bagaimana proses pengumpulan data dilakukan, kemudian akan dilakukan *pre-processing* untuk selanjutnya diproses dengan TF IDF sebelum dilakukan permodelan dengan *Backpropagation*.

BAB V HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil pengujian yang diperoleh terhadap penggunaan metode *Backpropagation* untuk klasifikasi cuitan tentang bakal calon Presiden Republik Indonesia 2024 berdasarkan

kata kunci yang digunakan. Hasil dari klasifikasi akan dipaparkan berdasarkan persentase polaritas.

BAB VI PENUTUP

Bab ini memberikan kesimpulan dari semua pembahasan pada bab-bab sebelumnya serta memberikan saran yang bermanfaat dalam pengembangan penelitian ke depannya.

BAB II

GAMBARAN UMUM PERANGKAT LUNAK YANG DIKEMBANGKAN

2.1 Fenomena Perangkat Lunak yang Dikembangkan

Seiring dengan berjalannya waktu, peningkatan jumlah pengguna jejaring sosial semakin bertambah. Terutama Twitter, jejaring sosial ini menjadi media favorit masyarakat Indonesia untuk melakukan pertukaran informasi serta media komunikasi antar masyarakat dengan masyarakat, perusahaan dengan pelanggan, dan juga bagi pemerintah dengan masyarakat. Pemerintah menggunakan Twitter sebagai sarana komunikasi publik secara langsung dengan masyarakat terkait suatu keluhan, hal ini dibuktikan dengan beberapa pejabat pemerintah yang aktif menanggapi serta bertindak apabila ada masyarakat pengguna Twitter yang melakukan laporan mengenai keluhan tertentu.

Banyaknya pejabat pemerintah yang menggunakan Twitter sebagai media pelayanan publik memberikan dampak positif bagi masyarakat. Dampak positif yang paling jelas adalah masyarakat pengguna Twitter bisa mengetahui kinerja pejabat pemerintah secara terbuka. Hal tersebut dapat dijadikan para pengguna sebagai referensi mengenai pejabat tersebut, bahkan juga sebagai perspektif akan dukungan terhadap bakal calon Presiden Republik Indonesia 2024.

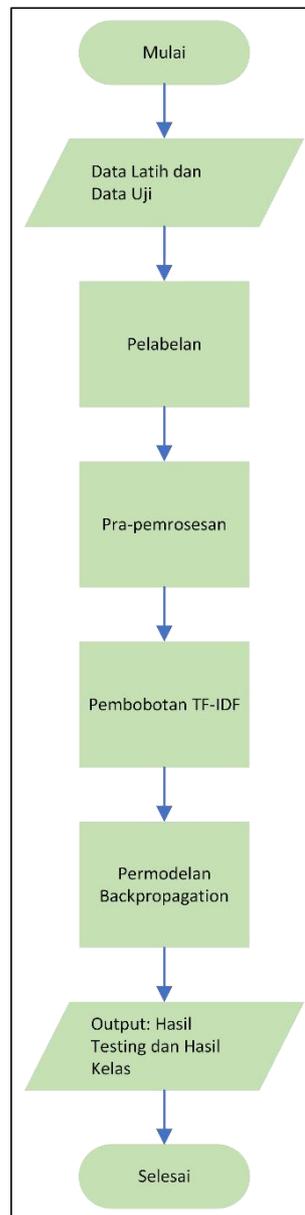
Pemilu Presiden Republik Indonesia akan dilaksanakan pada tahun 2024. Namun, beberapa lembaga survei telah mengeluarkan hasil survei

mengenai elektabilitas dari para bakal calon Presiden. Beberapa bakal calon yang memiliki elektabilitas teratas adalah Prabowo Subianto, Ganjar Pranowo, dan Anies Baswedan. Beberapa nama yang telah disebutkan sebelumnya merupakan pejabat yang aktif di jejaring sosial Twitter. Terdapat banyak dukungan dari pengguna Twitter terhadap beberapa pejabat tersebut untuk menjadi bakal calon Presiden Republik Indonesia 2024. Dukungan tersebut berupa opini yang memiliki sentimen positif. Tidak hanya itu, pengguna Twitter juga terkadang membuat opini yang mengandung sentimen netral bahkan negatif terhadap pejabat bakal calon Presiden Republik Indonesia 2024.

Analisis sentimen tentang opini bakal calon Presiden Republik Indonesia 2024 pada Twitter dengan algoritme *Backpropagation* adalah suatu sistem yang dikembangkan untuk membantu menganalisis opini-opini masyarakat pengguna Twitter yang mengandung sentimen positif, negatif, dan netral. Data yang digunakan pada penelitian ini berupa data yang di *crawling* dari Twitter.

Gambaran umum sistem yang akan diimplementasikan, akan diawali dengan pengumpulan data, kemudian data yang sudah terkumpul akan dilakukan pelabelan yang kemudian akan dilakukan tahap *pre-processing*. Setelah itu akan didapatkan data hasil dari *pre-processing* yang akan dilakukan pembobotan TF-IDF. Hasil dari perhitungan TF-IDF akan masuk pada proses klasifikasi menggunakan algoritme *Backpropagation* dan akan

memberikan hasil klasifikasi sentimen yang menunjukkan apakah sebuah data akan masuk ke dalam kelas positif, negatif, atau netral.



Gambar 2.1 Gambaran Umum Sistem

BAB III

TINJAUAN PUSTAKA

3.1 Teori Pendukung

3.1.1 Data Mining

Data *mining* merupakan suatu proses untuk menggali informasi dan pola yang bermanfaat dari data yang sangat besar. Data *mining* mencakup semua proses mulai dari pengumpulan data, ekstraksi data, analisis data, dan statistik data. Data *mining* juga memiliki beberapa istilah diantaranya dikenal sebagai *knowledge discovery*, *knowledge extraction*, *pattern analysis*, *information harvesting*, dan lain-lain.

Data *mining* juga merupakan proses logis untuk menemukan informasi yang berguna. Setelah ditemukannya pola dan informasi maka hal tersebut dapat digunakan sebagai alat pendukung dalam pengambilan keputusan dalam mengembangkan bisnis. Data *mining* juga dapat digunakan untuk meramalkan tren masa depan yang memungkinkan pebisnis membuat keputusan yang efektif, proaktif, dan dinamis. Data-data yang diolah dengan menggunakan teknik data *mining* akan mampu menghasilkan pengetahuan yang sesuai dengan harapan.

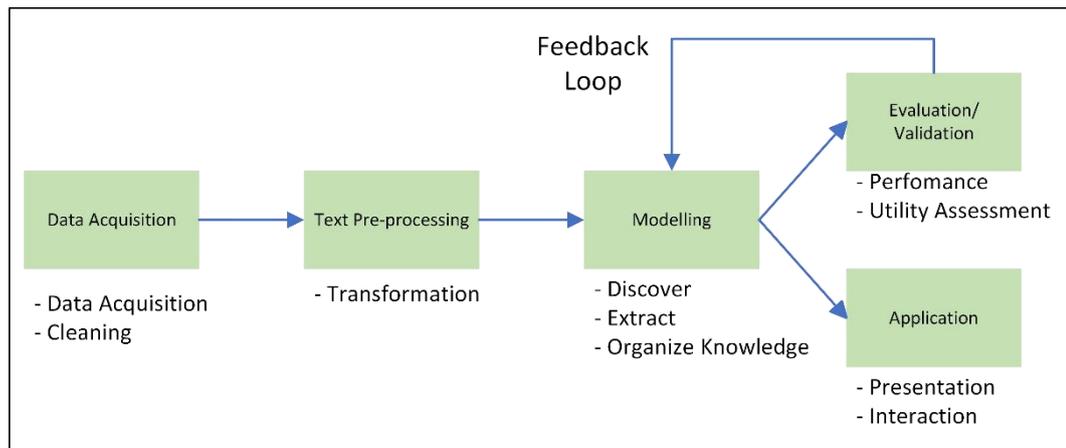
Menurut (Arhami & Nasir, 2020) data *mining* adalah suatu proses menganalisis pola data yang tersembunyi dari berbagai perspektif untuk dilakukan proses kategorisasi sehingga

menghasilkan informasi yang berguna, yang dikumpulkan di area umum, data *warehouse* untuk analisis yang efisien, serta memfasilitasi pengambilan keputusan bisnis, dan informasi lainnya.

3.1.2 *Text Mining*

Text mining adalah bagian dari bidang ilmu data *mining* yang memberikan sebuah set metodologi dan alat untuk dapat membantu dalam menemukan (*discovering*), memvisualisasikan (*presenting*), mengevaluasi pengetahuan dari kumpulan data teks yang sangat besar (Purbo, 2019).

Text mining biasa disebut juga dengan istilah data *text mining*, merupakan suatu proses yang dilakukan dalam analisis teks untuk memperoleh informasi yang berkualitas tinggi. Informasi berkualitas tinggi biasanya dapat diperoleh dengan melihat pola dan tren seperti dengan mempelajari pola statistik. *Text mining* biasanya melibatkan proses penataan teks seperti *parsing*, penambahan fitur kata, dan penghapusan kata yang tidak diperlukan. Menurut (Purbo, 2019) proses dasar *text mining* dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Proses Dasar *Text Mining*

Secara umum, proses *text mining* perlu mengikuti empat alur tahapan yang harus dilakukan yaitu:

1. Pengumpulan data, proses ini dilakukan bertujuan untuk mendapatkan data dari berbagai sumber, salah satu sumber data yang banyak digunakan adalah Twitter.
2. Penyiapan data atau *text pre-processing*, data yang telah didapatkan dari proses sebelumnya merupakan data mentah, sehingga perlu dilakukan proses pembersihan, dan normalisasi.
3. Permodelan, proses ini biasanya dilakukan dengan menggunakan algoritme tertentu untuk didapatkan hasil sehingga dapat dilakukan evaluasi.
4. Presentasi dan interaksi, biasanya dilakukan untuk memvisualisasikan hasil permodelan yang telah dilakukan.

3.1.3 Analisis Sentimen

Analisis sentimen merupakan suatu proses yang dilakukan untuk memahami dokumen tekstual secara otomatis untuk mendapatkan informasi yang terkandung dalam suatu kalimat yang berupa opini (Mejova, 2009) . Tujuan dilakukannya analisis sentimen adalah untuk mengetahui pendapat serta pandangan yang memiliki sentimen tertentu dan berkaitan terhadap suatu topik permasalahan tertentu.

Menurut (Liu, 2016) analisis sentimen atau biasa juga disebut dengan *opinion mining* adalah bidang studi yang melakukan analisis opini, sentimen, evaluasi, penilaian, sikap, dan emosi masyarakat terhadap suatu entitas. Analisis sentimen terbagi menjadi dua proses yaitu *Sentiment Classification* dan *Sentiment Extraction*. *Sentiment Extraction* merupakan proses mengekstraksi sentimen yang telah dievaluasi, sedangkan *Sentiment Classification* merupakan proses penentuan opini mengenai suatu aspek yang bernilai positif, netral, atau negatif.

3.1.4 Twitter

Twitter adalah salah satu jejaring sosial yang paling populer di dunia bahkan di Indonesia sendiri pengguna aktif Twitter mencapai 19,5 juta pengguna. Twitter menyediakan fitur *tweet* yang memungkinkan pengguna untuk dapat mengekspresikan opini dan perasaan mereka terhadap suatu topik permasalahan (Russell &

Klassen, 2019). Selain dengan menggunakan fitur *tweet* pengguna juga bisa menggunakan fitur *retweet* dan *comment* untuk menanggapi opini dari pengguna lain.

3.1.5 *Term Frequency-Inverse Document Frequency* TF-IDF

Menurut (Russell & Klassen, 2019) , Pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) merupakan proses yang dilakukan untuk mengekstraksi fitur dari teks dengan cara memberikan nilai pada setiap kata yang ada dalam dataset, kemudian akan dilakukan perhitungan untuk menentukan bobot dari setiap kata.

TF-IDF bertujuan untuk mengetahui seberapa penting setiap kata mewakili dari sebuah kalimat. Penentuan skor atau bobot dalam TF-IDF dilakukan berdasarkan frekuensi munculnya kata dalam dokumen. Nilai TF-IDF dapat ditemukan pada Persamaan (3.1).

$$TFIDF = TF \times IDF = TF \times \log_e \frac{|D|}{DF} \dots \dots \dots \quad (3.1)$$

Di mana,

- TF = *term frequency*, banyak sebuah kata atau istilah muncul.
- DF = *document frequency*, jumlah dokumen di mana kata atau istilah tersebut muncul. Minimal satu dokumen.

Dapat disimpulkan, apabila TF tinggi maka TFIDF akan naik yang berarti kata atau istilah itu penting. DF tinggi artinya TFIDF menjadi rendah artinya kata atau istilah itu tidak penting.

3.1.6 Klasifikasi

Menurut (Pierson, 2017), klasifikasi merupakan suatu proses untuk mendapatkan sejumlah model ataupun fungsi yang menggambarkan, mengenali serta membedakan kelas data atau konsep yang bertujuan untuk menggunakan model tersebut agar dapat memprediksi kelas objek yang label kelasnya belum diketahui. Klasifikasi didasarkan pada empat komponen dasar yaitu:

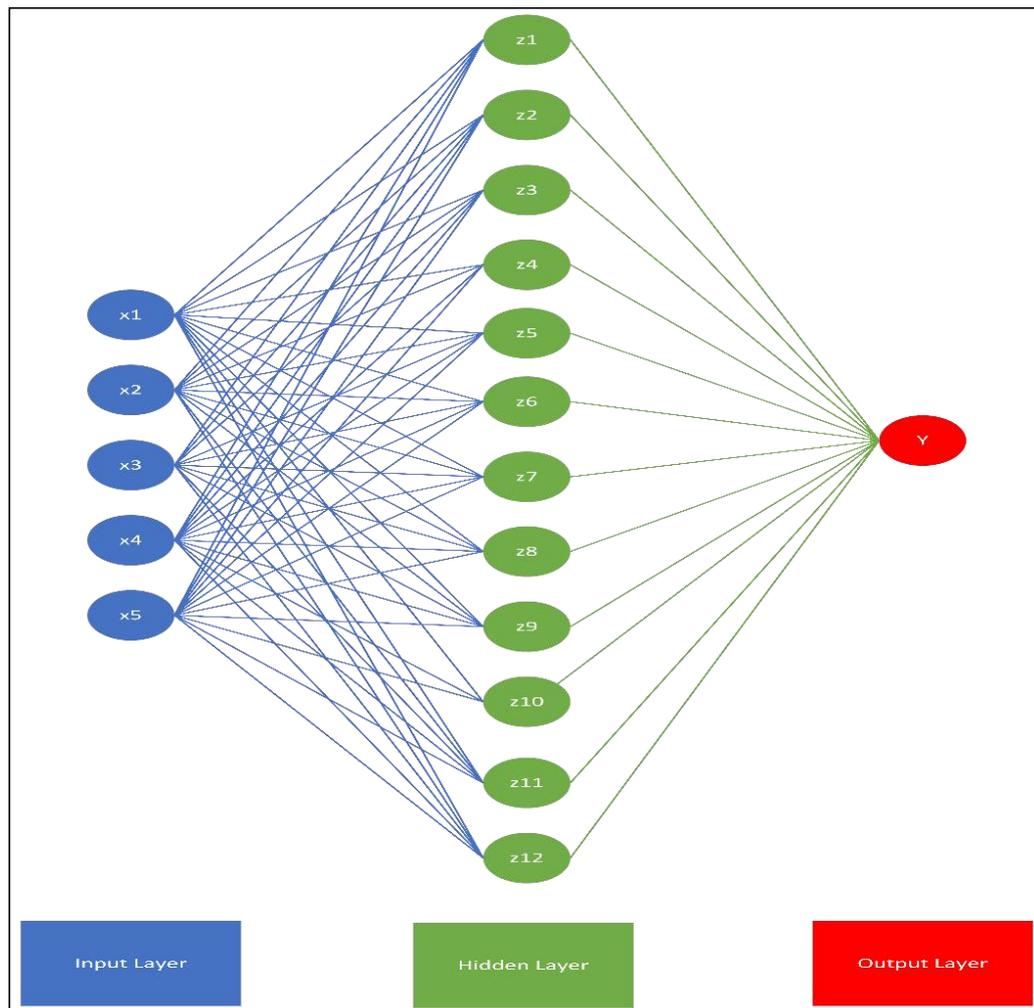
- a. Kelas atau label, merupakan variabel dependen dari model yang berupa variabel kategori untuk menjelaskan sebuah label pada objek setelah dilakukan proses klasifikasi.
- b. Prediktor atau atribut, merupakan variabel independen dari sebuah model yang diwakili oleh karakteristik dari data yang akan diklasifikasikan dan berdasarkan klasifikasi yang dibuat.
- c. *Tranining set* atau dataset latihan, merupakan kumpulan data yang berupa baris nilai-nilai untuk dua komponen sebelumnya (kelas dan prediktor) dapat berupa variabel kontinu maupun kategoris, dan digunakan untuk melakukan pelatihan atau pembangunan model yang nantinya akan menyesuaikan kelas berdasarkan prediktor yang tersedia.

- d. Dataset pengujian, merupakan data baru yang akan dilakukan percobaan klasifikasi berdasarkan model yang telah dibangun untuk mengukur tingkat akurasi, sehingga model klasifikasi dapat dievaluasi.

3.1.7 *Backpropagation*

Metode *Backpropagation* (propagasi balik) merupakan suatu metode *Neural Networks*, termasuk dalam algoritme pembelajaran tingkat lanjut dari hasil pengembangan aturan algoritme sebelumnya yaitu *perceptron*. Hal yang memiliki kesamaan antara BNN dan *perceptron* adalah pada tahapan dalam algoritme jaringan. Tahapan metode *Backpropagation* terdiri dari *feedforward* yang merupakan hasil tiruan dari *perceptron* dan ditambahkan dengan tahap *backpropagation error* untuk optimalisasi bobot baru berdasarkan nilai *error* dari proses sebelumnya (Windarto et al., 2020).

Salah satu hal yang membedakan antara *Backpropagation* dengan *perceptron* terdapat pada arsitektur jaringan yang dimiliki. *Perceptron* memiliki jaringan lapis tunggal (*single layer perceptron*), sedangkan metode *Backpropagation* memiliki jaringan lapis jamak (*multi layer perceptron*). Menurut (Supardi & Utami, 2014) arsitektur *Backpropagation* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Arsitektur *Backpropagation Neural Network*

Gambar 3.2 menunjukkan arsitektur dari *Backpropagation Neural Network*. Lapisan masukan (*input layer*) ditunjukkan dengan unit - unit x_i , sementara lapisan keluaran (*output layer*) ditunjukkan dengan unit Y . Lapisan tersembunyi (*hidden layer*) ditunjukkan dengan unit-unit z_i . Unit-unit tersembunyi pada *Backpropagation Neural Network* juga biasanya memiliki bias.

Sinyal keluaran dari setiap *neuron* pada lapisan masukan (*input layer*) merupakan sinyal masukan bagi *neuron* pada lapisan

tersembunyi (*hidden layer*), dan sinyal keluaran dari *neuron* pada lapisan tersembunyi (*hidden layer*) merupakan sinyal masukan bagi *neuron* pada lapisan keluaran (*output layer*). Jumlah lapisan tersembunyi (*hidden layer*) pada metode *Backpropagation Neural Network* ditentukan berdasarkan percobaan. Semakin banyak jumlah lapisan tersembunyi (*hidden layer*) maka performa jaringan akan memberikan hasil yang lebih akurat, tetapi proses pelatihannya akan membutuhkan waktu yang lama. Penentuan banyaknya jumlah *neuron* pada lapisan tersembunyi (*hidden layer*) tergantung jumlah *neuron* pada lapisan masukan (*input layer*).

Pada *Backpropagation Neural Network* diperlukan fungsi aktivasi untuk menentukan nilai keluaran berdasarkan dari nilai total masukan pada *neuron*. Sebuah fungsi aktivasi digunakan *neuron* untuk menyalakan atau memadamkan penjalaran sinyal dari *neuron* tersebut. Fungsi aktivasi pada metode *Backpropagation Neural Network* ini tidak hanya menggunakan sebuah fungsi aktivasi, akan tetapi turunan dari fungsi tersebut juga dapat digunakan. *Backpropagation* dapat menggunakan fungsi aktivasi seperti *sigmoid biner* maupun *sigmoid bipolar* juga turunan fungsinya.

Berikut langkah-langkah algoritme *Backpropagation*:

1. Inisiasi bobot (akan ditentukan bilangan acak pada rentang antara 0 sampai dengan 1).

2. Selama kondisi berhenti tidak terpenuhi lakukan langkah 3 sampai dengan langkah 10.
3. Untuk setiap pasangan vektor pelatihan lakukan langkah 3 sampai dengan langkah 8.

Feedforward

4. Setiap *neuron* pada lapisan masukan (X_i , $i=1,2, \dots, n$) menerima sinyal masukan x_i dan menjalankan sinyal tersebut ke semua *neuron* pada lapisan selanjutnya (dalam hal ini adalah lapisan tersembunyi).
5. Untuk setiap *neuron* dalam lapisan tersembunyi (Z_j , $j=1,2, \dots, p$) jumlahkan bobotnya dengan sinyal masukannya masing-masing.

$$Z_{inj} = v_{0j} + \sum_{i=1}^n x_i v_j \dots \dots \dots \quad (3.2)$$

Terapkan fungsi aktivasi untuk menghitung nilai sinyal keluaran.

$$Z_j = f(Z_{inj}) \dots \dots \dots \quad (3.3)$$

Kemudian kirimkan sinyal ini ke semua *neuron* pada lapisan berikutnya (dalam hal ini adalah lapisan keluaran).

6. Untuk setiap *neuron* pada lapisan keluaran (Y_k , $k=1, 2, \dots, m$) jumlahkan bobotnya dengan sinyal masukannya masing-masing.

$$Y_{ink} = w_{0j} + \sum_{i=1}^p z_j w_k \dots \dots \dots \quad (3.4)$$

Terapkan fungsi aktivasi untuk menghitung nilai sinyal keluaran

$$Y_k = f(y_{ink}) \dots \dots \dots \quad (3.5)$$

Back propagation of error

7. Setiap *neuron* pada lapisan keluaran (Y_k , $k=1, 2, \dots, m$) menerima sebuah pola target yang berhubungan dengan pola masukan pelatihan kemudian hitung kesalahannya.

$$\delta_k = (t_k - y_k) \cdot f'(y_{in}) \dots \dots \dots (3.6)$$

Hitung perubahan bobotnya (digunakan nanti untuk mengubah nilai w_{jk}).

$$\Delta w_{jk} = a \cdot \delta_k \cdot z_k \dots \dots \dots (3.7)$$

Hitung perubahan biasnya (digunakan nanti untuk mengubah nilai w_{0k}).

$$\Delta w_{0k} = a \cdot \delta_k \dots \dots \dots (3.8)$$

8. Untuk setiap *neuron* pada lapisan tersembunyi (Z_j , $j=1, 2, \dots, p$):
Jumlahkan nilai *delta* masukannya (dari *neuron* pada lapisan di atasnya).

$$\delta_{inj} = \sum_{k=1}^m \delta_k w_{jk} \dots \dots \dots (3.9)$$

Kalikan dengan turunan aktivasinya untuk menghitung nilai kesalahannya

$$\delta_j = \delta_{inj} \cdot f'(z_{inj}) \dots \dots \dots (3.10)$$

Hitung perubahan bobotnya (digunakan untuk mengubah nilai v_{ij})

$$\Delta v_{ij} = a \cdot \delta_j x_i \dots \dots \dots (3.11)$$

Kemudian hitung perubahan biasnya (digunakan nanti untuk mengubah nilai v_{0j})

$$\Delta v_{0j} = a \cdot \delta_j \dots \dots \dots \quad (3.12)$$

Memperbarui Bobot dan Bias

9. Untuk setiap *neuron* pada lapisan keluaran (Y_k , $k=1, 2, \dots, m$) ganti nilai bobot dan biasnya ($j=0, 1, 2, \dots, p$)

$$w_{jk(\text{baru})} = w_{jk(\text{lama})} + \Delta w_{jk} \dots \dots \dots \quad (3.13)$$

Untuk setiap *neuron* pada lapisan tersembunyi (z_j , $j=1, 2, \dots, p$) ganti nilai bobot dan biasnya ($i=0, 1, 2, \dots, n$)

$$v_{ij(\text{baru})} = v_{ij(\text{lama})} + \Delta v_{ij} \dots \dots \dots \quad (3.14)$$

Uji/periksa kondisi berhenti.

3.1.8 Confusion Matrix

Confusion matrix adalah teknik yang digunakan untuk mengevaluasi dari model klasifikasi yang dibangun. Berdasarkan hasil dari klasifikasi model, *confusion matrix* akan memperkirakan apakah suatu objek itu benar atau salah. Proses kerja dari *confusion matrix* adalah dengan membandingkan hasil prediksi klasifikasi dengan kelas asli yang berupa informasi aktual dari klasifikasi. Setelah model yang dibangun berhasil melakukan klasifikasi data cuitan, maka akan dibutuhkan suatu ukuran untuk menentukan apakah klasifikasi yang dilakukan oleh model tersebut memiliki akurasi yang valid atau tidak (Geron, 2017). Tabel 3.1 merupakan *confusion matrix*.

Tabel 3. 1 Confusion Matrix

<i>Classification</i>	<i>Prdicted Positives</i>	<i>Predicted Negatives</i>
<i>Actual Posotive Cases</i>	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
<i>Actual Negative Cases</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

Berdasarkan Tabel 3.1 dapat dijelaskan:

- *True Positive (TP)* adalah banyaknya jumlah dokumen yang prediksi kelasnya memiliki nilai positif dan kelas aktualnya bernilai positif.
- *False Negative (FN)* adalah banyaknya jumlah dokumen yang diprediksi menjadi kelas negatif oleh sistem, namun kelas aktualnya memiliki nilai positif
- *False Positive (FP)* adalah banyaknya jumlah dokumen yang diprediksi sebagai kelas positif oleh sistem, namun kelas aktualnya adalah negatif.
- *True Negative (TN)* adalah banyaknya jumlah dokumen kelas yang diberikan oleh sistem sama dengan kelas aktualnya, yaitu negatif.

Selanjutnya, terdapat beberapa perhitungan yang harus dilakukan berdasarkan hasil dari tabel *confusion matrix* di atas seperti *accuracy*, *precision*, *recall*, *f-measure* yang dapat dilihat pada Persamaan (3.15), (3.16), (3.17), dan (3.18).

$$Accuracy = \frac{TN+TP}{TN+TP+FP+FN} \dots\dots\dots (3.15)$$

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (3.16)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (3.17)$$

$$F-measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \dots\dots\dots (3.18)$$

Dengan keterangan sebagai berikut.

- *Accuracy*, merupakan perhitungan berdasarkan seluruh keadaan yang diprediksikan dengan nilai benar terhadap seluruh keadaan yang diprediksi.
- *Precision*, merupakan perhitungan berdasarkan kondisi benar, yaitu kelas aktual dan kelas prediksi yang sama (positif) dan berdasarkan seluruh kondisi yang diprediksi positif.
- *Racall*, merupakan perhitungan berdasarkan kondisi benar, yaitu memiliki kelas positif berdasarkan seluruh kondisi aktual yang bernilai positif.
- *F-Measure*, merupakan perhitungan berdasarkan hasil dari perhitungan *precision* dan *recall* untuk kemudian dicari nilai tengah terhadap kedua evaluasi tersebut.

3.1.9 *Text Preprocessing*

Tahap *preprocessing* bertujuan untuk mempersiapkan dataset yang berupa dokumen teks yang tidak terstruktur menjadi dokumen terstruktur untuk siap digunakan pada proses *text mining* selanjutnya

(Bengfort et al., 2018) . Berikut ini tahapan-tahapan dari *text preprocessing*:

a. *Cleaning*

Cleaning merupakan tahapan awal setelah tahap pengumpulan data, yang dilakukan untuk membersihkan data teks yang akan diolah. Data teks mentah yang masih banyak mengandung *emoji*, simbol yang tidak diperlukan serta *@username_pengguna* ini perlu dilakukan penghapusan.

b. *Case Folding*

Case folding merupakan tahapan yang dilakukan untuk menyamaratakan karakter dari setiap kalimat. Terkadang data yang didapatkan dari sosial media memiliki pola penulisan yang jauh dari aturan baku, ada yang kapitalisasi, *uppercase* dan juga ada yang *lowercase*. *Case folding* bertujuan membuat sebuah kalimat menjadi huruf kecil semua, hal ini agar mudah untuk memproses tahapan selanjutnya.

c. *Tokenizing*

Tokenizing merupakan proses memecah kalimat menjadi kata-kata. Proses *tokenizing* biasanya mencakup penghapusan angka, tanda baca, simbol serta spasi yang tidak diperlukan.

d. *Normalization*

Normalization atau Normalisasi merupakan tahapan yang dilakukan untuk mengubah kata yang tidak sesuai dengan aturan baku menjadi bentuk baku sesuai ejaan Bahasa Indonesia.

e. *Filtering*

Filtering merupakan tahapan yang dilakukan setelah tahap *normalization*, tujuannya adalah untuk mengambil kata-kata yang dianggap penting. Biasanya tahap ini memisahkan antara kata hubung dan kata benda atau yang biasa disebut dengan penghapusan *stopwords*.

f. *Stemming*

Stemming merupakan tahapan yang dilakukan untuk mengubah kata yang memiliki imbuhan menjadi kata dasar. Biasanya kata kerja yang memiliki imbuhan seperti meng-ambil akan diubah menjadi kata dasarnya yaitu ambil.

3.1.10 *Python*

Python merupakan bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido Van Rossum dengan menggunakan pendekatan interpreter, dalam artian *python* akan berjalan tanpa harus melakukan kompilasi seluruh kode menjadi bahasa mesin terlebih dahulu. *Python* merupakan bahasa pemrograman yang sangat populer saat ini karena kesederhanaanya dalam penulisan

kode, serta banyaknya pustaka untuk berbagai macam aplikasi contohnya untuk pemrosesan bahasa alami NLP (Muller & Guido, 2017).

3.1.11 *Natural Language Processing (NLTK)*

NLTK adalah pustaka *python* yang dikembangkan oleh Steven Bird dari departemen ilmu komputer dan informasi *University of Pennsylvania* yang dibuat khusus untuk pemrosesan dan klasifikasi teks. Fitur yang disediakan oleh pustaka ini beberapa diantaranya adalah untuk *tokenization*, *tagging*, *filtering*, manipulasi teks, *parsing*, serta fungsi untuk pemrosesan bahasa alami lainnya (Kulkarni & Shivananda, 2019).

3.1.12 *Pandas*

Pandas adalah pustaka yang dikembangkan oleh Wes McKinney diatas bahasa pemrograman *python* yang dapat digunakan untuk analisis dan manipulasi data. Struktur data yang banyak digunakan pada pustaka ini adalah *dataframe* yang merupakan representasi dari *index* dan *record* dari sekumpulan data yang dapat bersifat heterogen (Mckinney, 2018).

3.1.13 *Preference value*

Menurut (Haga et al., 2018) *preference value* adalah suatu teknik pengukuran yang digunakan untuk mengetahui jumlah respon positif pengguna Twitter. Dalam penelitian ini *preference*

value digunakan untuk mengetahui trending bakal calon Presiden Republik Indonesia 2024 di Twitter. Semakin tinggi nilai *preference value* yang dihasilkan maka dapat diartikan semakin banyak respon positif pengguna Twitter terhadap bakal calon Presiden tersebut. Rumus *preference value* dapat dilihat pada Persamaan (3.19).

$$Preference\ value = \frac{sp}{sp + sr + sn} * akurasi \dots\dots\dots (3.19)$$

Keterangan Persamaan 3.19:

Sp: sentimen positif

Sn: sentimen negatif

Sr: sentimen netral

3.1.14 *Scikit Learn*

Scikit learn merupakan *machine learning library* yang dikembangkan diatas bahasa pemrograman *Python*. Terdapat banyak sekali modul yang berhubungan dengan *machine learning* yang dapat digunakan, seperti permodelan algoritme, *pre processing* data, modul untuk melakukan ekstraksi fitur hingga modul untuk melakukan pengujian model yang sedang dibangun (Swamynathan, 2017).

3.1.15 *Matplotlib*

Menurut (Morgan, 2016) *matplotlib* adalah pustaka 2D *plotting* pada *python* yang memiliki fungsi untuk memvisualisasikan grafik. Dengan menggunakan pustaka *matplotlib* kita bisa memvisualisasikan data dengan berbagai macam diagram, seperti bar diagram, pie diagram, histogram, dan masih banyak grafik diagram lainnya.

3.1.16 Sastrawi

Sastrawi adalah pustaka yang dikembangkan di atas bahasa pemrograman *python* yang dapat digunakan untuk proses *stemming* Bahasa Indonesia. Proses kerja dari sastrawi *stemmer* adalah mengubah kata yang memiliki imbuhan menjadi kata dasarnya, seperti kata berimbuhan *mem-buat* diubah menjadi kata dasarnya yaitu *buat*.

3.1.17 Fungsi Aktivasi *ReLU*

Menurut (Swamynathan, 2017) , *ReLU* (Rectified Linier Unit) adalah suatu fungsi aktivasi non linier yang populer digunakan untuk menghasilkan pembatas pada bilangan nol, artinya apabila $x \leq 0$ maka $x = 0$ dan apabila $x > 0$ maka $x = x$. Fungsi aktivasi *ReLU* juga dapat mempercepat proses konvergensi jika dibanding fungsi aktivasi sigmoid atau *tanh*. Perhitungan *ReLU* dapat dilihat pada Persamaan 3.20.

$$f(x) = \max(0, x) \dots \dots \dots (3.20)$$

3.2 Hasil Penelitian Terdahulu

Penelitian yang telah dilakukan sebelumnya dapat dijadikan referensi untuk penulis dalam memperkaya teori dan juga sebagai acuan yang digunakan dalam mengkaji penelitian ini. Berikut merupakan beberapa penelitian terdahulu yang dijadikan sebagai referensi.

Tabel 3. 2 Penelitian Terdahulu

No	Judul	Penulis/Tahun	Hasil
1.	Analisis Mengenai Calon Presiden Indonesia 2019 di Twitter Menggunakan Metode <i>Backpropagation</i> .	(Rozi et al., 2020).	Pada penelitian ini, kelas pada dataset dibagi menjadi 3 polaritas yaitu positif, negatif, dan netral. sedangkan nilai akurasi pada pengujian <i>hidden layer</i> memiliki nilai akurasi sebesar 89.6%, nilai <i>precision</i> 90.2%, nilai <i>recall</i> 83% pada jumlah <i>hidden layer</i> 6.
2.	Analisis Sentimen Data Pada BPJS Kesehatan Dengan Metode <i>Backpropagation Neural Network</i> .	(Ricky et al., 2020).	Pada penelitian ini, data opini mengenai BPJS kesehatan dibagi menjadi dua kelas polaritas yaitu positif dan negatif. Klasifikasi dilakukan dengan metode <i>Backpropagation Neural Networks</i> dengan <i>accuracy</i> sebesar 87.14%, <i>precision</i> sebesar 86.97%, <i>recall</i> sebesar 87.14%, dan <i>F1 score</i> sebesar 87.05%.
3.	Analisis Sentimen Data Twitter Mengenai Program Vaksinasi Di Indonesia	(Wicaksono, 2022).	Pada penelitian ini, membahas mengenai program vaksinasi di Indonesia menggunakan dataset yang

	Menggunakan Algoritme <i>Backpropagation</i> .		berasal dari Twitter. Klasifikasi dilakukan menggunakan metode <i>Backpropagation</i> dengan menambahkan metode <i>anova</i> sebagai seleksi fitur. Hasil akurasi akhir yang diperoleh sebesar 78,00%.
--	--	--	--

Tabel 3. 2 Penelitian Terdahulu (Lanjutan 1)

No	Judul	Penulis/Tahun	Hasil
4.	Analisis Sentimen Calon Presiden Indonesia 2019 Berdasarkan Komentar Publik di Facebook.	(Santoso & Nugroho, 2019)	Pada penelitian ini, opini masyarakat dikumpulkan melalui facebook sejak tanggal 17 April 2019 sampai 22 Mei 2019, polaritas kelas yang digunakan berupa positif dan negatif. Hasil dari penelitian ini Joko Widodo diperoleh 85% sentimen positif, dan 15% untuk sentimen negatif. Sedangkan Prabowo Subianto diperoleh 76% sentimen positif, dan 24% untuk sentimen negatif.
5.	<i>Prediction of Indonesia Presidential Election Results for the 2019-2024 Period Using Twitter Sentiment Analysis.</i>	(Kristiyanti et al., 2019)	Pada penelitian ini, menggunakan algoritme <i>Support Vector Machine</i> dengan tambahan <i>Particle Swarm Optimization</i> dan Algoritme Genetika sebagai seleksi fitur. Data yang digunakan merupakan data dari Twitter dengan beberapa keyword seperti

No	Judul	Penulis/Tahun	Hasil
			#jokowi2periode, #2019tetapjokowi, #2019prabowosandi dan #2019gantipresiden dengan total data sekitar 4000 <i>tweet</i> . Hasil dari penelitian ini memprediksi Prabowo dan Sandiaga Uno yang menang dengan jumlah sentimen positif terbanyak yakni 830 dari 1000 <i>tweet</i> . Akurasi yang dihasilkan dari SVM dengan PSO adalah 86.20% dan nilai AUC sebesar 0.93.

Berdasarkan penelitian terdahulu yang terlampir pada Tabel 3.2, terdapat beberapa perbedaan. Perbedaan penelitian terletak pada dataset yang digunakan serta beberapa penelitian menggunakan algoritme yang berbeda, perbedaan juga terletak pada objek yang diteliti. Pada penelitian pertama objek yang dibahas adalah calon Presiden 2019. Data yang diambil dari Twitter merupakan data cuitan pengguna pada tahun 2019 dengan kata kunci terkait. Polaritas pada penelitian tersebut terbagi menjadi tiga kelas yaitu positif, negatif, dan netral. Pada penelitian kedua *Backpropagation* digunakan untuk mengklasifikasikan data dengan objek yang berbeda dengan penelitian sekarang yaitu BPJS kesehatan. Pada penelitian ketiga *Backpropagation* digunakan untuk klasifikasi program vaksinasi di Indonesia. Pada penelitian keempat dataset dikumpulkan dari jejaring sosial Facebook, metode klasifikasi yang digunakan adalah *Naïve Bayes*

Classifier. Pada penelitian kelima objek yang diteliti mengenai prediksi presiden 2019-2024 algoritme klasifikasi yang digunakan adalah *Support Vector Machine* dengan tambahan *Particle Swarm Optimization*, dan Algoritme Genetika sebagai seleksi fitur.

BAB IV

METODE PENELITIAN

4.1 Lokasi dan Waktu Penelitian

4.1.1 Lokasi

Tempat pengambilan data pada penelitian ini adalah melalui jejaring sosial Twitter, dengan otorisasi API (*Application Programming Interface*) yang dapat diakses pada link <https://developer.twitter.com>.

4.1.2 Waktu Penelitian

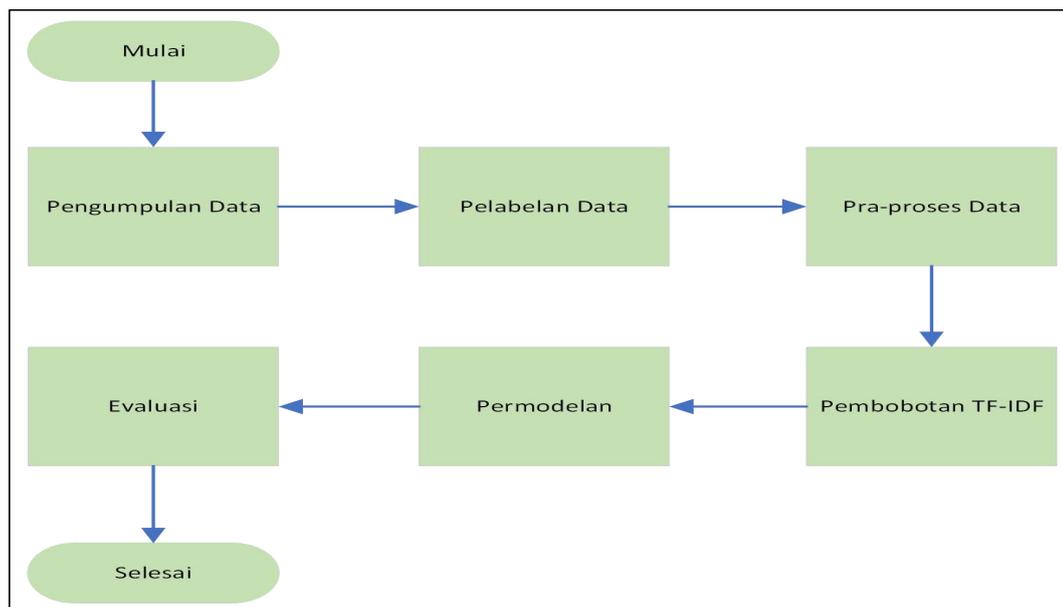
Penelitian ini dimulai dari bulan April 2022 sampai pada bulan Juli 2022. Jadwal lengkap penelitian dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Jadwal Penelitian

No	Kegiatan	Bulan			
		April 2022	Mei 2022	Juni 2022	Juli 2022
1.	Studi Literatur	■	■	■	
2.	Pengumpulan dan Pengolahan Data		■	■	
3.	Perancangan Sistem		■	■	
4.	Implementasi Sistem		■	■	■
5.	Analisis dan Pengujian				■
6.	Penulisan Skripsi	■	■	■	■

4.2 Alur Penelitian

Penelitian diawali dengan melakukan pengumpulan data yang dilakukan dengan cara *crawling* data dari jejaring sosial Twitter untuk mendapatkan dataset berdasarkan kata kunci, @prabowo, @aniesbaswedan, @ganjarpranowo, #capres2024, yang berupa cuitan dari pengguna. Dataset yang telah terkumpul akan dilakukan pelabelan terlebih dahulu untuk menentukan polaritas dari cuitan tersebut apakah positif, netral, atau negatif. Proses selanjutnya yang dilakukan adalah tahap *pre-processing* yang terdiri dari *cleaning*, *case folding*, *tokenizing*, *normalization*, *filtering*, dan *stemming*. Data cuitan yang sudah bersih, selanjutnya akan digunakan untuk proses pembobotan TF-IDF sehingga didapatkan bobot dari setiap katanya. Setelah itu akan dilakukan pelatihan dengan algoritme *Backpropagation* untuk membuat suatu model analisis sentimen. Model yang telah berhasil dilatih akan dilakukan evaluasi untuk menghasilkan model yang terbaik. Alur tahapan penelitian dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Alur Penelitian

4.2.1 Pengumpulan Data

Penelitian ini menggunakan data dari cuitan pengguna Twitter dengan kata kunci @prabowo, @aniesbaswedan, @ganjarpranowo, #capres2024. Data cuitan didapatkan menggunakan API (*Application Programming Interface*) Twitter melalui proses *crawling*. *Crawling* data cuitan dilakukan menggunakan bantuan pustaka *tweepy* yang merupakan salah satu pustaka yang dikembangkan di atas bahasa pemrograman *python*. Data yang dihasilkan dari hasil *crawling* merupakan data dengan format JSON (*Javascript Object Notation*), kemudian dilakukan konversi menjadi data dengan format CSV yang diolah dengan bantuan pustaka *pandas dataframe* (Mitchell, 2018).

4.2.2 Pelabelan Data

Data cuitan pengguna yang didapatkan dari hasil *crawling* akan dilakukan proses pelabelan terlebih dahulu. Proses pelabelan dilakukan secara manual dengan cara memberi kelas pada dataset dengan polaritas positif, netral, atau negatif.

4.2.3 Text Preprocessing

a. *Cleaning*

Cleaning membersihkan data teks yang akan diolah. Data teks mentah yang masih banyak mengandung *emoji*, *@username_pengguna* akan dilakukan penghapusan.

b. *Case Folding*

Case folding bertujuan membuat sebuah kalimat menjadi huruf kecil semua, hal ini agar mudah untuk memproses tahapan selanjutnya.

c. *Tokenizing*

Tokenizing merupakan proses memecah kalimat menjadi kata-kata. Proses *tokenizing* biasanya mencakup penghapusan angka, tanda baca, simbol serta spasi yang tidak diperlukan.

d. *Normalization*

Normalization atau Normalisasi merupakan tahapan yang dilakukan untuk mengubah kata yang tidak sesuai dengan aturan baku menjadi bentuk baku sesuai ejaan bahasa Indonesia.

e. *Filtering*

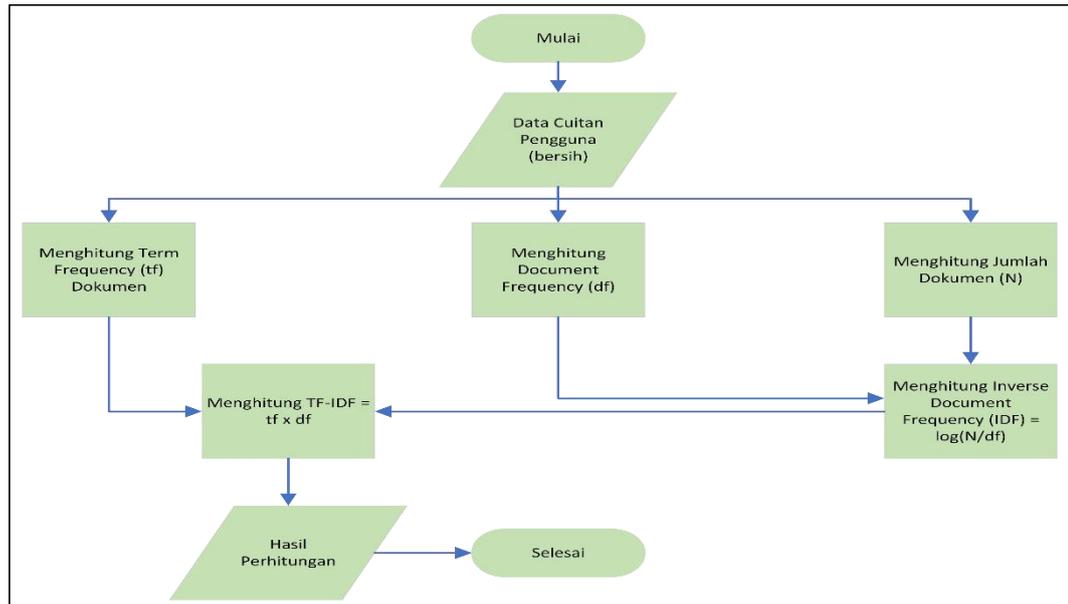
Filtering bertujuan untuk memisahkan antara kata hubung dan kata benda atau yang biasa disebut dengan penghapusan *stopwords*.

f. *Stemming*

Stemming merupakan tahapan yang dilakukan untuk mengubah kata yang memiliki imbuhan menjadi kata dasar. Biasanya kata kerja yang memiliki imbuhan seperti meng-ambil akan diubah menjadi kata dasarnya yaitu ambil.

4.2.4 Pembobotan *Term Frequency–Inverse Document Frequency*

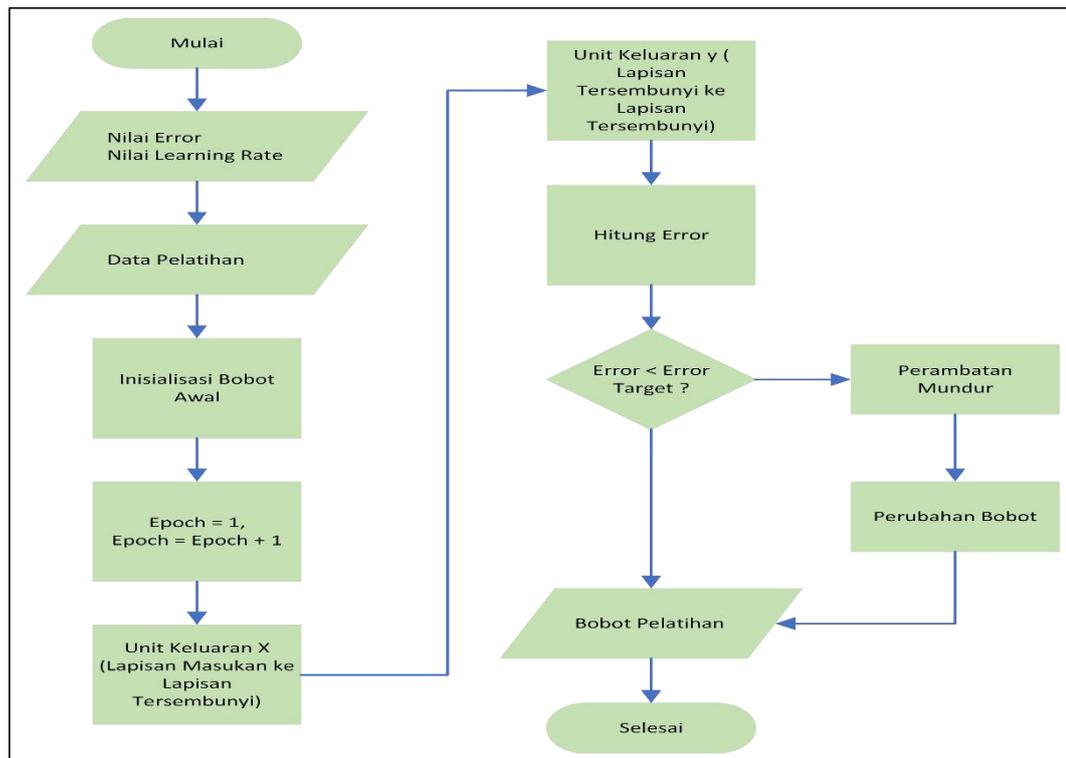
Proses yang dilakukan setelah data cuitan bersih adalah proses perhitungan TF-IDF. Pada proses pembobotan menggunakan TF-IDF ini akan dilakukan perhitungan bobot untuk setiap katanya. Bobot akan dihitung berdasarkan frekuensi kemunculan kata pada setiap dokumen. Alur tahapan pembobotan menggunakan TF-IDF dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Alur Pembobotan TF-IDF

4.2.5 Permodelan *Backpropagation*

Setelah dataset telah melalui proses pembobotan dengan metode TF-IDF, selanjutnya dataset akan dilakukan proses pelatihan dengan menggunakan algoritme *Backpropagation*. Proses pelatihan dengan algoritme *Backpropagation* akan membagi kelas menjadi 3 kelas polaritas yaitu positif, netral, dan negatif. Alur pelatihan menggunakan *Backpropagation* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Alur Proses Pelatihan *Backpropagation*

4.2.6 Evaluasi Model

Evaluasi hasil sentimen dilakukan untuk mengetahui performa klasifikasi dari *Backpropagation*. Perhitungan dilakukan dengan menggunakan *confusion matrix* dengan menghitung akurasi, presisi, *recall*, dan *F-measure*. Hasil dari evaluasi ini dapat dijadikan sebagai gambaran seberapa akurat algoritme *Backpropagation* dalam mengklasifikasi sentimen cuitan pengguna Twitter terhadap bakal calon Presiden Republik Indonesia 2024.

BAB V

HASIL DAN PEMBAHASAN

5.1 Gambaran Umum Aplikasi

Aplikasi analisis sentimen terhadap bakal calon Presiden Republik Indonesia 2024 dikembangkan di atas bahasa pemrograman *python*. Tahap *crawling* dilakukan dengan Twitter API menggunakan pustaka *python* yakni *tweepy* dengan beberapa *keyword* yang telah ditentukan seperti @prabowo, @aniesbaswedan, @ganjarpranowo, dan #capres2024. Data yang telah didapatkan dari hasil *crawling* akan dilakukan pelabelan secara manual dengan memberi kelas sentimen positif, negatif, dan netral. Dataset yang telah memiliki kelas akan dilakukan tahapan praproses dengan beberapa pustaka *python* yakni *regular expression* untuk melakukan *cleaning* berdasarkan pola tertentu seperti menghilangkan *prefix*, tanda baca, dan karakter yang tidak diperlukan. Dataset yang telah bersih akan diubah menjadi token kata perkata dengan bantuan pustaka NLTK yakni dengan modul *word_tokenize*. Data yang telah menjadi token akan dilakukan normalisasi untuk mengubah kata yang tidak baku menjadi kata baku. Data hasil dari normalisasi akan dilakukan penghapusan *stopwords* dengan bantuan dari pustaka NLTK yakni *nltk.corpus*. Tahap terakhir yang dilakukan pada praproses adalah *stemming*, proses ini dilakukan dengan bantuan pustaka sastrawi pada *python* dengan modul *StemmerFactory*. Data hasil dari praproses akan dilakukan pembobotan dengan TF-IDF, proses ini dilakukan dengan bantuan pustaka *scikit learn* yakni dengan modul

TfidfVectorizer. Permodelan *backpropagation* pada penelitian ini dilakukan dengan pustaka *scikit learn*, yakni dengan modul *MLPClassifier neural network* dengan membuat 3 model berdasarkan dataset bakal calon. Pengukuran akurasi dari model klasifikasi yang dibangun dilakukan dengan modul *ConfusionMatrixDisplay* dari pustaka *scikit learn*. Kemudian, akan dilakukan prediksi untuk dataset baru menggunakan model klasifikasi yang telah dibangun. Hasil dari prediksi beserta hasil dari perhitungan *preference value* akan divisualisasikan menggunakan *bar diagram*. Proses visualisasi dilakukan dengan bantuan pustaka *matplotlib* pada *python* yakni dengan menampilkan hasil perbandingan *preference value* bakal calon Presiden Republik Indonesia 2024.

5.2 Pengumpulan Data

Langkah pertama yang dilakukan dalam penelitian ini adalah melakukan pengumpulan data cuitan dari jejaring sosial Twitter. Pengumpulan data cuitan Twitter dilakukan dengan melakukan *crawling* menggunakan Twitter API, serta bantuan dari pustaka *python tweepy*. Kata kunci yang digunakan dalam melakukan *crawling* merupakan nama bakal calon Presiden Republik Indonesia 2024 yang didapatkan berdasarkan survei online yang dibatasi hanya 3 teratas saja yaitu Prabowo Subianto, Ganjar Pranowo, dan Anies Baswedan. Berdasarkan nama bakal calon Presiden Republik Indonesia 2024 tersebut, didapatkan kata kunci berupa, @prabowo, @aniesbaswedan, @ganjarpranowo, dan #capres2024. Untuk

dapat melakukan *crawling* diperlukan inisialisasi API yaitu seperti pada Gambar 5.1.

```
# import pustaka tweepy
import tweepy

# Inisialisasi API
consumer_key = 'EcruZdG9NathSk9XykyYkX8k1'
consumer_secret = 'QdW0kqw61H0mcPaISJHs2CguFm5a5oTxClr62P10pFoUHJHodX'
access_token = '1497020949337878530-Aa1xTs0gm7vCjcK181TFQCuu0SabD1'
access_token_secret = 'DYBxDyaE0g3M227QmTrnq21cOmf6dZLtXZPjB0xgT9kCT'

#Autentikasi
auth = tweepy.OAuth1UserHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
```

Gambar 5. 1 Inisialisasi Twitter API

Setelah melakukan inisialisasi Twitter API maka akan dilakukan *crawling* data cuitan berdasarkan kata kunci yang telah ditentukan sebelumnya. Untuk kode *crawling* data cuitan Twitter dapat dilihat pada Gambar 5.2.

```
# import pustaka pandas
import pandas as pd

keywords = '@ganjarpranowo' # @aniesbaswedan, @ganjarpranowo, #capres2024, #prabowocapres2024

# crawling data
tweets = tweepy.Cursor(api.search_tweets, q=keywords, since='2022-01-01', until='2022-06-02' count=100,
                        tweet_mode='extended').items(5000)

# inisialisasi dataframe
columns = ['User', 'Tweet', 'Tanggal tweet']
data = []

# loop data cuitan
for tweet in tweets:
    data.append([tweet.user.screen_name, tweet.full_text, tweet.created_at])

# membuat dataframe
df = pd.DataFrame(data, columns=columns)

df.to_csv('today_7_12_2022', index=False)
# print(df)
```

Gambar 5. 2 Crawling Data Cuitan Twitter

Setelah *crawling* dilakukan berdasarkan kata kunci yang telah ditentukan, hasil *crawling* dilakukan penghapusan data duplikat sehingga didapatkan dataset yang berjumlah 732 data cuitan Ganjar Pranowo, 727 data cuitan Anies Baswedan, dan 715 untuk Prabowo Subianto, total dataset berjumlah 2174 data.

5.3 Pelabelan Data

Berdasarkan hasil *crawling* terdapat beberapa penghapusan dataset karena hanya berisi *link*. Kemudian dilakukan pelabelan secara manual dengan memberi kelas positif, netral, dan negatif untuk tiap masing-masing bakal calon. Jumlah data yang diberikan label adalah 720 untuk dataset Anies Baswedan dengan jumlah sentimen positif 303, sentimen negatif 210, dan sentimen netral 207. Untuk dataset Ganjar Pranowo jumlah dataset berlabel sebesar 730 dengan jumlah sentimen positif 305, sentimen negatif 215, dan sentimen netral 210. Sedangkan untuk dataset Prabowo Subianto jumlah dataset berlabel sebesar 712 dengan jumlah sentimen positif 307, sentimen negatif 203, dan sentimen netral 202. Adapun contoh hasil dari pelabelan secara manual dataset dapat dilihat pada Tabel 5.1.

Tabel 5. 1 Contoh Hasil Pelabelan Secara Manual

Dataset	Sentimen
---------	----------

<p>Deklarasi Dukungan untuk bapak @prabowo menjadi Presiden RI 2024 di Rapat Kordinasi Cabang @dpc_gerindramedan.</p> <p>Seluruh pengurus DPC Gerindra Kota Medan, pimpinan anak cabang, ranting, meminta dengan hormat bapak H Prabowo Subianto agar bersedia menjadi Calon Presiden RI 2024.</p> <p>https://t.co/q3hPLIB13g</p>	Positif
<p>"@detikcom Sejarah fantastis yg terukir oleh anies baswedan. Nikmat mana lagi Jkt58, yg jalian dustakan! Prestasi ini membuat saya beralih mendukungnya utk jadi capres di neraka thn 2024 nanti."</p>	Negatif
<p>"@ganjarpranowo Entah Ganjar atau Anies yang RINegatif, yang jelas alumni #UGM yg menang He he he"</p>	Netral

5.4 *Text Preprocessing*

Pada tahap *text processing* dilakukan normalisasi dataset dengan melakukan beberapa proses sehingga didapatkan dataset yang siap untuk dilakukan pembobotan pada tahap selanjutnya. Untuk tahap *text processing* terdiri dari beberapa tahapan yaitu *cleaning*, *case folding*, *tokenizing*, *normalization*, *filtering*, dan *stemming*.

a. *Cleaning*

Pada tahap ini dilakukan penghapusan karakter, angka, tanda baca, spasi berlebih, serta beberapa komponen cuitan Twitter yakni berupa username (@username), karakter HTML, URL, *hashtag*, dan *emoticon*. Tahapan ini dilakukan karena beberapa hal tersebut tidak memiliki pengaruh apapun terhadap analisis sentimen, dan juga komponen tersebut hanya akan menjadi *noise* pada dataset. Adapun contoh data cuitan hasil dari tahap *cleaning* dapat dilihat pada Tabel 5.2.

Tabel 5.2 Hasil *Cleaning*

Teks Cuitan	Hasil <i>Cleaning</i>
Deklarasi Dukungan untuk bapak @prabowo menjadi Presiden RI 2024 di Rapat Kordinasi Cabang @dpc_gerindramedan. Seluruh pengurus DPC Gerindra Kota Medan, pimpinan anak cabang, ranting, meminta dengan hormat bapak H Prabowo Subianto agar bersedia menjadi Calon Presiden RI 2024. https://t.co/q3hPLIB13g	Deklarasi Dukungan untuk bapak menjadi Presiden RI di Rapat Kordinasi Cabang gerindramedan Seluruh pengurus DPC Gerindra Kota Medan pimpinan anak cabang ranting meminta dengan hormat bapak Prabowo Subianto agar bersedia menjadi Calon Presiden RI
@BungkusTukang @ganjarpranowo Siap kawal Pak Ganjar Presiden 2024	Siap kawal Pak Ganjar Presiden
RT @OposisiCerdas: Anies Baswedan Masuk Nominasi Capres 2024 dari PKS: Popularitas dan Elektabilitas Tinggi	RT Anies Baswedan Masuk Nominasi Capres dari PKS Popularitas dan Elektabilitas Tinggi

Proses *cleaning* menghapus non ASCII seperti emoticon, chinese word, mention, URL, angka, tanda baca, *multiple* spasi, dan juga *single* karakter. Kode program untuk melakukan proses *cleaning* dapat dilihat pada Gambar 5.3.

```
# cleaning menggunakan RegEx
def cleaning(text):
    # hapus tab, baris, back slice
    text = text.replace('\t', " ").replace('\n', " ").replace('\u', " ").replace('\\', "")

    # hapus non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')

    # hapus mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\w+/\w+)", " ", text).split())

    # hapus incomplete URL
    text = text.replace("http://", " ").replace("https://", " ")

    # hapus angka
    text = re.sub(r"\d+", "", text)

    # hapus tanda baca
    text = text.translate(str.maketrans("", "", string.punctuation))

    # hapus multiple spasi
    text = re.sub('\s+', ' ', text)

    # hapus single karakter
    text = re.sub(r"\b[a-zA-Z]\b", "", text)

    return text

df['tweet_cleaning'] = df['Tweet'].apply(cleaning)
```

Gambar 5.3 *Source Code Cleaning*

b. *Case Folding*

Case folding dilakukan untuk mengubah setiap karakter menjadi huruf kecil semua, hal ini dilakukan karena tidak semua cuitan yang dihasilkan Twitter itu memiliki penulisan yang sama, oleh sebab itu dilakukan *case folding*. Selain itu, *case folding* juga bertujuan untuk mempermudah proses pada tahap berikutnya. Adapun contoh data cuitan hasil dari tahap *case folding* dapat dilihat pada Tabel 5.3.

Tabel 5. 3 Hasil Case Folding

Teks Cuitan	Hasil <i>Case Folding</i>
Deklarasi Dukungan untuk bapak menjadi Presiden RI di Rapat Koordinasi Cabang gerindramedan Seluruh pengurus DPC Gerindra Kota Medan pimpinan anak cabang ranting meminta dengan hormat bapak Prabowo Subianto agar bersedia menjadi Calon Presiden RI	deklarasi dukungan untuk bapak menjadi presiden ri di rapat koordinasi cabang gerindramedan seluruh pengurus dpc Gerindra kota medan pimpinan anak cabang ranting meminta dengan hormat bapak prabowo subianto agar bersedia menjadi calon presiden ri
Siap kawal Pak Ganjar Presiden	siap kawal pak ganjar presiden
RT Anies Baswedan Masuk Nominasi Capres dari PKS Popularitas dan Elektabilitas Tinggi	rt anies baswedan masuk nominasi capres dari pks popularitas dan elektabilitas tinggi

Proses *casefolding* dilakukan dengan menggunakan *built in method* pada *python* yaitu *lower()*. Kode program untuk melakukan proses *case folding* dapat dilihat pada Gambar 5.4.

```
# mengubah text menjadi huruf kecil
def caseFolding(text):
    text = text.lower()
    return text
df['tweet_case_folding'] = df['tweet_cleaning'].apply(caseFolding)
df
```

Gambar 5. 4 Source Code Case Folding

c. Tokenizing

Tokenizing dilakukan untuk mengubah dataset cuitan yang masih berupa kalimat ataupun paragraf menjadi kata tunggal, yakni dengan

memecah setiap kalimat menjadi potongan kata perkata. Hal lain yang dilakukan pada saat proses *tokenizing* adalah menghapus karakter tertentu yang dianggap sebagai tanda baca. Adapun contoh data cuitan hasil dari tahap *tokenizing* dapat dilihat pada Tabel 5.4.

Tabel 5. 4 Hasil *Tokenizing*

Teks Cuitan	Hasil <i>Tokenizing</i>
deklarasi dukungan untuk bapak menjadi presiden ri di rapat kordinasi cabang gerindramedan seluruh pengurus dpc gerindra kota medan pimpinan anak cabang ranting meminta dengan hormat bapak prabowo subianto agar bersedia menjadi calon presiden ri	['deklarasi', 'dukungan', 'untuk', 'bapak', 'menjadi', 'presiden', 'ri', 'di', 'rapat', 'kordinasi', 'cabang', 'gerindramedan', 'seluruh', 'pengurus', 'dpc', 'gerindra', 'kota', 'medan', 'pimpinan', 'anak', 'cabang', 'ranting', 'meminta', 'dengan', 'hormat', 'bapak', 'prabowo', 'subianto', 'agar', 'bersedia', 'menjadi', 'calon', 'presiden', 'ri']
siap kawal pak ganjar presiden	['siap', 'kawal', 'pak', 'ganjar', 'presiden']
rt anies baswedan masuk nominasi capres dari pks popularitas dan elektabilitas tinggi	['rt', 'anies', 'baswedan', 'masuk', 'nominasi', 'capres', 'dari', 'pks', 'popularitas', 'dan', 'elektabilitas', 'tinggi']

Proses *tokenizing* dilakukan dengan bantuan pustaka NLTK, yaitu menggunakan method *word_tokenize*. Kode program untuk melakukan proses *tokenizing* dapat dilihat pada Gambar 5.5.

```

# import pustaka NLTK
import nltk
from nltk.tokenize import word_tokenize

# tokenisasi
def tokenizing(text):
    return word_tokenize(text)
df['tokenisasi'] = df['tweet_case_folding_new'].apply(tokenizing)
df

```

Gambar 5.5 Source Code Tokenizing

d. Normalization

Normalisasi dilakukan untuk mengubah data yang tidak sesuai ejaan Bahasa Indonesia menjadi kata yang baku. Proses ini dilakukan dengan membuat kamus data yang menyesuaikan dengan kesalahan umum ejaan serta ditambahkan dengan kesalahan yang ada dalam dataset. Adapun contoh data cuitan hasil dari tahap normalisasi dapat dilihat pada Tabel 5.5.

Tabel 5.5 Cuitan Hasil Normalisasi

Teks Hasil Tokenizing	Teks Normalisasi
['krj', 'pak', 'krj', 'kampanye', 'teross', 'hemm']	['kerja', 'pak', 'kerja', 'kampanye', 'terus', 'hemm']
['ali', 'trs', 'menurutmu', 'siapa', 'yg', 'pantes', 'jadi', 'presiden']	['ali', 'terus', 'menurutmu', 'siapa', 'yang', 'pantas', 'menjadi', 'presiden']
['kata', 'yg', 'unik', 'ditujukan', 'untuk', 'padeh', 'ganjar', 'pranowo', 'namun', 'yg', 'pasti', 'jika', 'diperhatikan', 'gestur', 'tubuhnya', 'adalah', 'sosok', 'yg', 'mudah', 'berbaur', 'dan', 'bergaul', 'dgn', 'siapapun', 'pokoknya', 'friendly']	['kata', 'yang', 'unik', 'ditujukan', 'untuk', 'pak', 'ganjar', 'pranowo', 'namun', 'yang', 'pasti', 'jika', 'diperhatikan', 'gestur', 'tubuhnya', 'adalah', 'sosok', 'yang', 'mudah', 'berbaur', 'dan', 'bergaul', 'dengan', 'siapapun', 'pokoknya', 'friendly']

'lah']	'lah']
--------	--------

Normalisasi dilakukan dengan membuat logika yang menyamakan data kesalahan pada dataset dengan *corpus* yang berisi ejaan yang benar, sehingga ketika program menemukan kata yang tidak baku maka akan digantikan dengan kata yang baku. Kode program untuk melakukan proses normalisasi dapat dilihat pada Gambar 5.6.

```
# normalisasi data yang sudah diproses tokenisasi
# inisialisasi corpus data
normalizad_word = pd.read_excel("dataset/normalisasi.xlsx")
normalizad_word_dict = {}

# Loop data dari corpus file normalisasi.xlsx
# assign data ke variabel normalized_word_dict
for index, row in normalizad_word.iterrows():
    if row[0] not in normalizad_word_dict:
        normalizad_word_dict[row[0]] = row[1]

# inisialisasi fungsi normalisasi
# cek jika didalam dataset terdapat data tidak normal
# ubah data tersebut menjadi sesuai ketentuan corpus
def normalisasiTeks(document):
    return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in document]
df['tweet_normalisasi'] = df['tokenisasi'].apply(normalisasiTeks)
df
```

Gambar 5. 6 Source Code Normalisasi

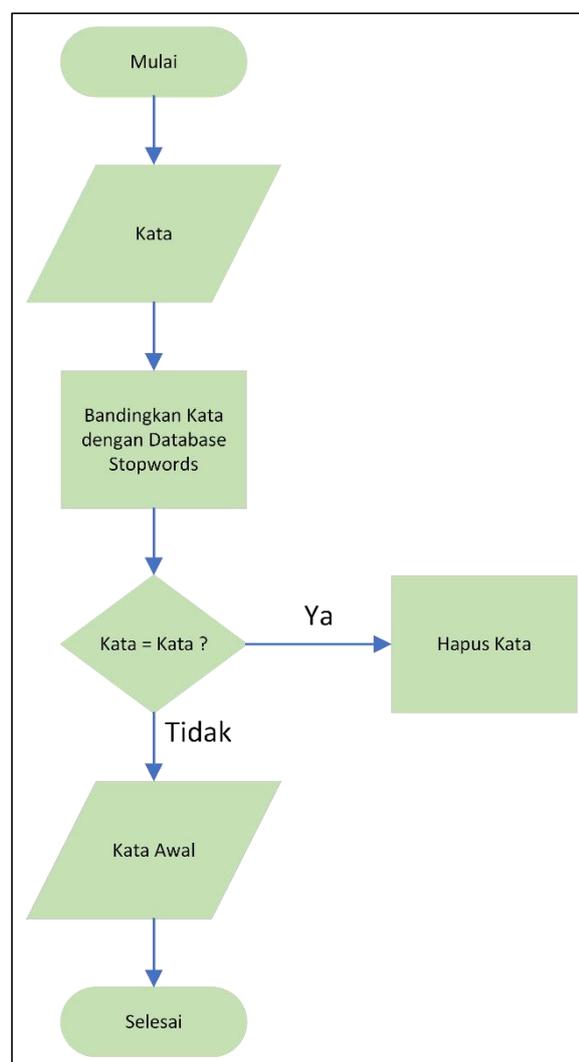
e. Filtering

Filtering (stopwords removal) adalah proses yang dilakukan untuk mengeliminasi dan menghapus kata yang kurang penting yang tidak memiliki pengaruh apapun pada saat klasifikasi. Kata yang tergolong pada *stopwords* adalah kata yang berupa kata keterangan, kata ganti, kata depan, dan kata hubung seperti ‘dan’, ‘atau’, ‘di’, ‘ke’, dan seterusnya. Adapun tahapan dalam melakukan *stopwords* yaitu:

1. Dataset yang telah dilakukan proses tokenisasi berupa kata perkata akan dijadikan sebagai masukan.

2. Setiap kata pada dataset akan dibandingkan dengan kata yang ada pada *corpus stopwords*.
3. Apabila dataset yang dibandingkan tersebut sama, maka kata tersebut akan dilakukan eliminasi atau penghapusan, jika tidak maka kata tersebut tidak dihapus.

Proses *filtering stopwords* dapat dilihat pada *flowchart* Gambar 5.7.



Gambar 5.7 Proses *Filtering Stopwords*

Adapun contoh data cuitan hasil dari tahap *filtering stopwords* dapat dilihat pada Tabel 5.6.

Tabel 5. 6 Hasil *Filtering Stopwords*

Teks Token	Teks <i>Filtering Stopwords</i>
['rt', 'alhamdulillah', 'semoga', 'terjalin', 'kesepahaman', 'antara', 'kedua', 'partai', 'saling', 'menguatkan', 'dan', 'sama', 'sama', 'mengusung', 'anies']	['terjalin', 'kesepahaman', 'partai', 'menguatkan', 'mengusung', 'anies']
['pks', 'bakal', 'segera', 'menggelar', 'pertemuan', 'dengan', 'partai', 'nasdem', 'seiring', 'dengan', 'masuknya', 'gubernur', 'dki', 'jakarta', 'anies', 'baswedan', 'dalam', 'bursa', 'bakal', 'calon', 'presiden', 'capres', 'rapimnas', 'pks']	['pks', 'menggelar', 'pertemuan', 'partai', 'nasdem', 'seiring', 'masuknya', 'gubernur', 'dki', 'jakarta', 'anies', 'baswedan', 'bursa', 'calon', 'presiden', 'capres', 'rapimnas', 'pks']
['kami', 'bersepakat', 'dan', 'bermufakat', 'mendukung', 'pak', 'anies', 'baswedan', 'sebagai', 'capres', 'di', 'pilpres', 'mendatang', 'menuju', 'indonesia', 'berkeadilan', 'dan', 'sejahtera', 'anies', 'ri']	['bersepakat', 'bermufakat', 'mendukung', 'anies', 'baswedan', 'capres', 'pilpres', 'indonesia', 'berkeadilan', 'sejahtera', 'anies']

Proses *filtering* dilakukan dengan bantuan pustaka NLTK dengan menggunakan modul *stopwords* Bahasa Indonesia yang disediakan. Kode program untuk melakukan proses *filtering stopwords* dapat dilihat pada Gambar 5.8.

```

# download stopwords dari NLTK
nltk.download('stopwords')

# import modul stopwords dari pustaka NLTK
from nltk.corpus import stopwords

# tahap filtering penghapusan stopword pada data
# init stopword bahasa Indonesia
list_stopwords = stopwords.words('indonesian')

# tambahkan stopword dari corpus lokal
txt_stopword = pd.read_csv("dataset/stopwords.txt", names= ["stopwords"], header = None)

# konversi to list
list_stopwords.extend(txt_stopword["stopwords"][0].split(' '))
list_stopwords = set(list_stopwords)

# inisialisasi fungsi filtering
# cek jika kata pada corpus ada pada dataset
# maka hapus kata tersebut
def filtering(words):
    return [word for word in words if word not in list_stopwords]
df['tweet_tokens_stopwords'] = df['tweet_normalisasi'].apply(filtering)
df

```

Gambar 5. 8 Source Code Filtering Stopwords

f. Stemming

Proses *stemming* mengubah kata yang berimbuhan menjadi kata dasar, dilakukan dengan bantuan pustaka *python* sastrawi, modul yang digunakan pada pustaka ini adalah *StemmerFactory*. Adapun contoh data cuitan hasil dari tahap *stemming* dapat dilihat pada Tabel 5.7.

Tabel 5. 7 Cuitan Hasil Stemming

Teks Token	Teks Hasil <i>Stemming</i>
['bertambah', 'masif', 'terstruktur', 'terencana', 'dana', 'berlimpah', 'capres']	['tambah', 'masif', 'struktur', 'rencana', 'dana', 'limpah', 'capres']
['survei', 'tipis', 'pemilih', 'calon', 'pemilih', 'partai', 'nasdem', 'mencalonkan', 'ganjar', 'pranowo', 'capres']	['survei', 'tipis', 'pilih', 'calon', 'pilih', 'partai', 'nasdem', 'calon', 'ganjar', 'pranowo', 'capres']

Tabel 5. 7 Cuitan Hasil *Stemming* (Lanjutan 1)

Teks Token	Teks Hasil <i>Stemming</i>
['anies', 'baswedan', 'mendulang', 'suara', 'partai', 'sp', 'jk', 'memainkan', 'politik', 'identitas', 'waspada', 'berani', 'taruhan', 'alphard', 'hasan', 'nasbi', 'anies', 'tiket', 'capres']	['anies', 'baswedan', 'dulang', 'suara', 'partai', 'sp', 'jk', 'main', 'politik', 'identitas', 'waspada', 'berani', 'taruh', 'alphard', 'hasan', 'nasbi', 'anies', 'tiket', 'capres']

Proses *stemming* dilakukan dengan menggunakan pustaka sastrawi dengan membuat objek *stemmer* yang melakukan perubahan terhadap kata yang memiliki imbuhan. Kode program untuk melakukan proses *stemming* dapat dilihat pada Gambar 5.9.

```
# import pustaka sastrawi dan swifter
import swifter
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# membuat stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# inialisasi fungsi stemmer
def stemmed_wrapper(term):
    return stemmer.stem(term)

# inialisasi dictionary
term_dict = {}

# loop setiap data dari tweet_normalisasi
# loop setiap value dari document
# cek jika term tidak ada pada term_dict
# tidak ada perubahan
for document in df['tweet_normalisasi']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ''

# loop setiap values dari term_dict
# assign values tersebut pada variabel term_dict
# dengan index berdasarkan term
for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term, ":", term_dict[term])

# proses stemming
def stemmingTeks(document):
    return [term_dict[term] for term in document]
df['tweet_tokens_stemmed'] = df['tweet_tokens_stopwords'].swifter.apply(stemmingTeks)
print(df['tweet_tokens_stemmed'])
```

Gambar 5. 9 Source Code *Stemming*

5.5 Ekstraksi Fitur TF-IDF

Pada tahap ekstraksi fitur dengan TF-IDF dilakukan perubahan terhadap dataset yang masih berupa *raw text* menjadi bobot-bobot berupa bilangan numerik yang akan diproses sebagai data latih dan data uji. Perubahan nilai *term* dilakukan berdasarkan jumlah dokumen teks pada dataset dikalikan banyaknya kata yang muncul pada satu dokumen, dengan syarat dataset sudah berbentuk token. Untuk contoh perhitungan TF-IDF secara manual berdasarkan dataset yang dipakai pada penelitian ini dapat dilihat pada Tabel 5.8.

D1 = “prabowo magang efektif moga presiden nkri belah”.

D2 = “keren hasil amat ips menhan oleh elektabilitas prabowo”.

D3 = “fakta prabowo goyah pasang capres cawapres indonesia ideal moga indonesia maju adil sejahtera”.

D4 = “sumsel benar yakin rakyat menhan prabowo sebab kerja serius fokus prabowo bangun tahan indonesia kuat segani”.

Berdasarkan dokumen D1, D2, D3, dan D4 yang telah dilakukan *preprocessing* dan *tokenizing*, maka akan dilakukan ekstraksi fitur menggunakan Persamaan (3.1). Hasil yang didapatkan dari ekstraksi fitur dengan TF-IDF dapat dilihat pada Tabel 5.8.

Tabel 5. 8 Pembobotan TF-IDF

Term	TF				df	D/df	IDF	Wdt = TFdt * IDFft			
	D1	D2	D3	D4				D1	D2	D3	D4

Term	TF				df	D/df	IDF	Wdt = TFdt * IDFft			
	D1	D2	D3	D4				D1	D2	D3	D4
adil	0	0	1	0	1	4	0.602	0	0	0.602	0
amat	0	1	0	0	1	4	0.602	0	0	0	0
bangun	0	0	0	1	1	4	0.602	0	0	0	0.602
benar	0	0	0	1	1	4	0.602	0	0	0	0.602
belah	1	0	0	0	1	4	0.602	0.602	0.602	0	0
capres	0	0	1	0	1	4	0.602	0	0	0.602	0
cawapres	0	0	1	0	1	4	0.602	0	0	0.602	0
efektif	1	0	0	0	1	4	0.602	0.602	0.602	0	0
elektabilitas	0	1	0	0	1	4	0.602	0	0	0	0
fokus	0	0	0	1	1	4	0.602	0	0	0	0.602
goyah	0	0	1	0	1	4	0.602	0	0	0.602	0
hasil	0	1	0	0	1	4	0.602	0	0	0	0
ideal	0	0	1	0	1	4	0.602	0	0	0.602	0
indonesia	0	0	2	1	2	2	0.301	0	0	0.602	0.301
ips	0	1	0	0	1	4	0.602	0	0	0	0
keren	0	1	0	0	1	4	0.602	0	0	0	0
kerja	0	0	0	1	1	4	0.602	0	0	0	0.602
kuat	0	0	0	1	1	4	0.602	0	0	0	0.602
magang	1	0	0	0	1	4	0.602	0.602	0.602	0	0
maju	0	0	1	0	1	4	0.602	0	0	0.602	0
menhan	0	1	0	1	2	2	0.301	0	0	0	0.301
moga	1	0	1	0	2	2	0.301	0.301	0.301	0.301	0
nkri	1	0	0	0	1	4	0.602	0.602	0.602	0	0
oleh	0	1	0	0	1	4	0.602	0	0	0	0
pasang	0	0	1	0	1	4	0.602	0	0	0.602	0
prabowo	1	1	1	2	4	1	0	0	0	0	0
presiden	1	0	0	0	1	4	0.602	0.602	0.602	0	0

Tabel 5. 8 Pembobotan TF-IDF (Lanjutan 1)

Term	TF				df	D/df	IDF	Wdt = TFdt * IDFft			
	D1	D2	D3	D4				D1	D2	D3	D4
rakyat	0	0	0	1	1	4	0.602	0	0	0	0.602
sebab	0	0	0	1	1	4	0.602	0	0	0	0.602
segani	0	0	0	1	1	4	0.602	0	0	0	0.602
sejahtera	0	0	1	0	1	4	0.602	0	0	0.602	0
serius	0	0	0	1	1	4	0.602	0	0	0	0.602
sumsel	0	0	0	1	1	4	0.602	0	0	0	0.602
tahan	0	0	0	1	1	4	0.602	0	0	0	0.602
yakin	0	0	0	1	1	4	0.602	0	0	0	0.602
Nilai rata-rata hasil ekstraksi fitur TF-IDF								0.095	0.095	0.163	0.224

Adapun pada penelitian ini pembobotan atau ekstraksi fitur dengan TF-IDF untuk seluruh dataset dilakukan dengan menggunakan pustaka *python* yakni menggunakan modul *TfidfVectorizer* pada *sklearn*. Kode program untuk melakukan proses TF-IDF dapat dilihat pada Gambar 5.10.

```
# import library
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

# baca dataset dari file .csv
df = pd.read_csv('dataset/prabowo/praProses-Prabowo-done.csv')

# inisialisasi fitur dan label
X_train = df['tweet_tokens_stemmed']
y = df['Sentimen']

# instansiasi objek TfidfVectorizer dari modul sklearn
vectorizer = TfidfVectorizer(analyzer='word', max_features=300, max_df=0.5, use_idf=True, norm='l2')

# Training vocab, return fitur dokumen sebagai matrix
tfidf = vectorizer.fit_transform(X_train)
```

Gambar 5. 10 Source Code TF-IDF

5.6 Permodelan *Backpropagation*

Hasil perhitungan bobot dari setiap kata yang telah didapatkan pada tahap ekstraksi fitur TF-IDF akan dijadikan sebagai *input* pada proses klasifikasi dengan algoritme *Backpropagation*. Data yang akan dijadikan *input* pada proses permodelan *Backpropagation* dapat dilihat pada Tabel 5.9.

Tabel 5. 9 Hasil TF-IDF untuk Klasifikasi *Backpropagation*

	X1	X2	X3	X4	X5	X6	ty
Dokumen 1	0,602	0,602	0,301	0,602	0,602	0,602	1
Dokumen 2	0,602	0,602	0,301	0,602	0,602	0,602	1

Keterangan Tabel 5.9.

$t(1)$ = positif

x = *input*

Dokumen = dokumen cuitan

Berdasarkan Tabel 5.9, data tersebut akan dilakukan klasifikasi menggunakan algoritme *Backpropagation* secara manual, untuk melakukan perhitungan secara manual diperlukan beberapa tahapan yaitu inisialisasi bobot, *feedforward*, *backpropagation of error*, dan penyesuaian bobot.

a. Inisialisasi bobot

Untuk contoh data yang digunakan pada proses perhitungan manual ini adalah dokumen 1 pada Tabel 5.9. Ditentukan nilai bias awal pada *hidden node* dan *output* seperti pada Tabel 5.10 dan Tabel 5.11 dengan nilai dari

masukan $x_1 = 0,602$ $x_2 = 0,602$ $x_3 = 0,301$ $x_4 = 0,602$ $x_5 = 0,602$ $x_6 = 0,602$ dan $t = 1$.

Tabel 5. 10 Inisialisasi Bobot dan Bias Awal

<i>Input</i>	<i>Z1 (hidden)</i>	<i>Z2 (hidden)</i>
X1	0.03	0.02
X2	0.01	0.3
X3	0.05	0.04
X4	0.07	0.08
X5	0.08	0.44
X6	0.63	0.97
Bias	0.89	0.06

Tabel 5. 11 Inisialisasi Bobot dan Bias Awal Output

<i>Y (output)</i>	
Z1	0.3
Z2	0.13
Bias	0.36

b. Feedforward

Langkah yang akan dilakukan selanjutnya adalah menghitung sinyal *input* dengan Persamaan (3.2). Untuk nilai yang akan dijadikan sebagai bobot adalah yang menghubungkan *node* dari *input* ke *node hidden*.

$$Z_{inj} = v_{0j} + \sum_{i=1}^n x_i v_j$$

$$Z_{in1} = 0.89 + 0.602(0.03) + 0.602(0.01) + 0.301(0.05) + 0.602(0.07) + 0.602(0.08) + 0.602(0.63) = 1.39869$$

$$Z_{in2} = 0.06 + 0.602(0.02) + 0.602(0.3) + 0.301(0.012) + 0.62(0.48) + 0.602(0.44) + 0.602(0.97) = 1.16166$$

Kemudian dilakukan penerapan fungsi aktivasi *relu*, untuk setiap masing-masing sinyal *input*.

$$Z_j = f(z_{inj}) = \max(0, x)$$

$$f(Z_{in1}) = \max(0, 1.39869) = 1.39869$$

$$f(Z_{in2}) = \max(0, 1.16166) = 1.16166$$

Untuk mengetahui nilai *output* sementara, dapat dihitung dengan menjumlahkan bobot dari *output* dengan unit hasil fungsi aktivasi.

$$Y_{ink} = w_{0j} + \sum_{i=1}^p z_j w_k$$

$$= 0.36 + (1.39869 \times 0.3) + (1.16166 \times 0.13) = 0.9306228$$

Terapkan fungsi aktivasi kembali berdasarkan hasil *output* yang telah didapatkan dari perhitungan *yk*.

$$Y_k = f(Y_{ink}) = \max(0, 0.9306228) = 0.9306228$$

Nilai *output* yang dihasilkan berdasarkan proses *feedforward* adalah sebesar 0.9306228, sedangkan target yang diinginkan adalah 1, maka proses akan dilanjutkan pada tahap *backpropagation of error* untuk

melakukan optimasi perubahan bobot dengan mencari faktor kesalahan, sehingga dapat tercapai nilai *output* $y = 1$ atau mendekati 1.

c. *Backpropagation*

Setelah proses *feedforward* selesai, langkah selanjutnya adalah menghitung nilai *factor* δ sebagai unit *output*.

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

$$\begin{aligned} \delta_k &= (1 - 0.9306228) (0.9306228) \\ &= 0.0645640041 \end{aligned}$$

Hasil *factor* yang telah didapatkan akan digunakan untuk menghitung perubahan bobot dari *output layer* dengan ketentuan nilai *learning rate* (a) = 0,2

$$\Delta w_{jk} = a \cdot \delta_k \cdot z_k$$

$$\Delta w_{11} = 0.2 (0.0645640041) \cdot 1.39869 = 0.01806100538$$

$$\Delta w_{21} = 0.2 (0.0645640041) \cdot 1.16166 = 0.0150002842$$

Hitung perubahan biasnya (digunakan nanti untuk mengubah nilai w_{0k})

$$\Delta w_{0k} = a \cdot \delta_k$$

$$\Delta w_{01} = 0.2 * 0.0645640041 = 0.01291280082$$

Menghitung *factor* δ berdasarkan kesalahan di setiap unit tersembunyi dengan Persamaan (3.9).

$$\delta_{inj} = \sum_{k=1}^m \delta_k w_{jk}$$

$$\delta_{in1} = (0.0645640041) (0.01806100538) = 0.001166090825$$

$$\delta_{in2} = (0.0645640041) (0.0150002842) = 0.0009684784106$$

$$\delta_j = \delta_{inj} \cdot f'(z_{inj})$$

$$\delta_1 = (0.001166090825) (1.39869) = 0.001630999576$$

$$\delta_2 = (0.0009684784106) (1.16166) = 0.00112504263$$

Kemudian dilakukan perhitungan suku perubahan bobot v_{ij} berdasarkan *factor* yang diperoleh dari masing-masing *input* pada *hidden layer*.

$$\Delta v_{ij} = a \cdot \delta_j x_i$$

$$\Delta v_1 = 0.2 (0.001630999576)(0.602) = 0.000196372349$$

$$\Delta v_2 = 0.2 (0.001630999576)(0.602) = 0.000196372349$$

$$\Delta v_3 = 0.2 (0.001630999576)(0.301) = 0.00009818617448$$

$$\Delta v_4 = 0.2 (0.001630999576)(0.602) = 0.000196372349$$

$$\Delta v_5 = 0.2 (0.001630999576)(0.602) = 0.000196372349$$

$$\Delta v_6 = 0.2 (0.001630999576)(0.602) = 0.000196372349$$

Kemudian hitung perubahan biasnya (digunakan nanti untuk mengubah nilai v_{0j}).

$$\Delta v_{0j} = a \cdot \delta_j$$

$$\Delta v_{01} = 0.2 (0.001630999576) = 0.0003261999152$$

$$\Delta v_{01} = 0.2 (0.00112504263) = 0.000225008526$$

Proses perhitungan bobot untuk setiap *node* pada *hidden layer* menggunakan perhitungan yang sama, sehingga hasil perubahan bobot dan bias baru secara lengkap dapat dilihat pada Tabel 5.12.

Tabel 5. 12 Perubahan Bobot (v_{ij})

<i>Input</i>	<i>Z₁ (hidden)</i>	<i>Z₂ (hidden)</i>
X1	0.000196372349	0.0001354551327
X2	0.000196372349	0.0001354551327
X3	0.000098186178	0.00006772756633
X4	0.000196372349	0.0001354551327
X5	0.000196372349	0.0001354551327
X6	0.000196372349	0.0001354551327
Bias	0.0003261999152	0.000225008526

d. Penyesuaian Bobot

Langkah terakhir yang dilakukan adalah mendapatkan nilai bobot dan bias baru untuk unit *output*.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

$$w_{10}(\text{baru}) = 0.36 + 0.01291280082 = 0.3129128008$$

$$w_{11}(\text{baru}) = 0.3 + 0.01806100538 = 0.3180610054$$

$$w_{12}(\text{baru}) = 0.13 + 0.0150002842 = 0.1450002842$$

Untuk setiap neuron pada lapisan tersembunyi ($z_j, j=1, 2, \dots, p$) ganti nilai bobot dan biasnya ($i=0, 1, 2, \dots, n$)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

$$v_{01}(\text{baru}) = 0.03 + 0.000196372349 = 0.03019637235$$

$$v_{02}(\text{baru}) = 0.01 + 0.000196372349 = 0.01019637235$$

$$v_{03}(\text{baru}) = 0.05 + 0,00008016929739 = 0.0500801693$$

$$v_{04}(\text{baru}) = 0.07 + 0.000196372349 = 0.07019637235$$

$$v_{05}(\text{baru}) = 0.08 + 0.000196372349 = 0.08019637235$$

$$v_{06}(\text{baru}) = 0.63 + 0.000196372349 = 0.6301963723$$

$$v_{07}(\text{baru}) = 0,89 + 0.0003261999152 = 0.8903261999$$

Perhitungan perubahan bobot pada *node* ke 2 sama dengan yang telah dituliskan pada perhitungan sebelumnya, sehingga diperoleh bobot dan bias baru yang digunakan untuk iterasi selanjutnya yang dapat dilihat pada Tabel 5.13 dan Tabel 5.14.

Tabel 5. 13 Inisialisasi Bobot *Input* Iterasi Kedua

<i>Input</i>	Z_1 (<i>hidden</i>)	Z_2 (<i>hidden</i>)
X1	0.03019637235	0.02013545513
X2	0.01019637235	0.3001354551
X3	0.0500801693	0.04006772757
X4	0.07019637235	0.08013545513
X5	0.08019637235	0.4401354551
X6	0.6301963723	0.9701354551
Bias	0.8903261999	0.06022500853

Tabel 5. 14 Inisialisasi Bobot *Output* Iterasi Kedua

Y (<i>output</i>)	
Z ₁	0.3180610054
Z ₂	0.1450002842
Bias	0.3129128008

Menghitung nilai *Mean Square Error* (MSE)

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (t_k - y_k)^2$$

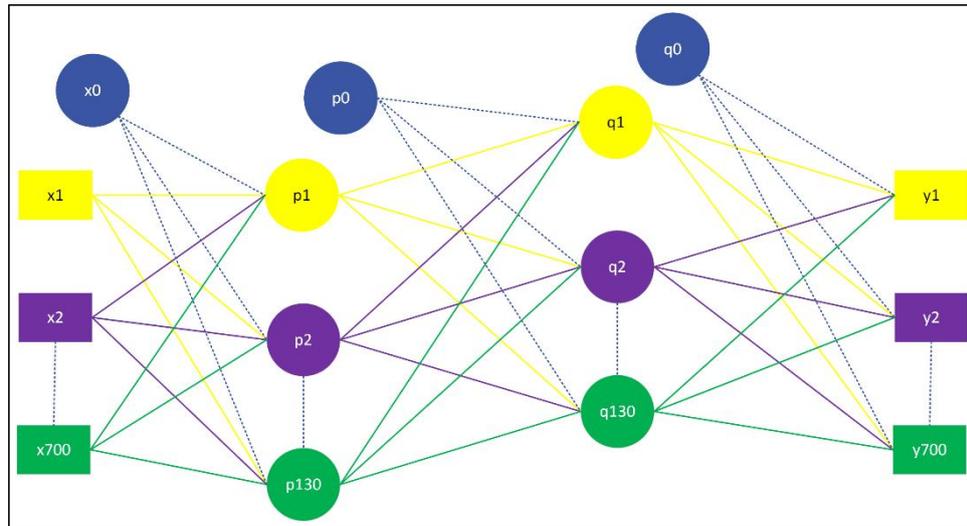
t_k = nilai *output* target

y_k = nilai *ouput* jaringan

N = jumlah *output* dari *neuron*

$$\text{MSE} = \frac{(1 - 0.9306228)^2}{1} = 0.00481319588$$

Proses perhitungan ini akan terus berlanjut sebanyak iterasi yang diinginkan hingga nilai y mencapai nilai target. Untuk seluruh proses iterasi dari masing masing dataset bakal calon Presiden Republik Indonesia 2024 pada penelitian ini dilakukan dengan menggunakan bantuan pustaka *python* yakni menggunakan modul *MLPClassifier neural network* pada *sklearn*. Arsitektur *neural network* yang digunakan dapat dilihat pada Gambar 5.11.



Gambar 5. 11 *Arsitektur Neural Network*

Pada Gambar 5.11 dapat dijelaskan bahwa yang menjadi *input layer* adalah variabel $x_1, x_2 - x_{700}$ yang merepresentasikan jumlah seluruh dataset cuitan yang telah diproses menjadi *vector* pada tahap ekstraksi fitur TF-IDF. Variabel $p_1, p_2 - p_{130}$ dijadikan sebagai *hidden layer* pertama yang memiliki *node* atau *neuron* sebanyak 130 *node*. *Hidden layer* kedua direpresentasikan dengan variabel $q_1, q_2 - q_{130}$ dengan jumlah *node* sebanyak 130 *node*, sedangkan x_0, p_0 , dan q_0 merupakan nilai bias untuk setiap *layer* yang dijadikan sebagai acuan agar tidak terjadi stagnan pada saat proses pelatihan. Kemudian hasil *output* dari *vector* q merupakan *vector output* dari y , dimana y merupakan hasil klasifikasi dari algoritme *Backpropagation* yang menentukan suatu kelas dari hasil analisis sentimen bakal calon Presiden Republik Indonesia 2024 berdasarkan dataset *input*.

5.6.1 Encode Label

Dataset yang telah dilakukan pelabelan secara manual pada tahapan sebelumnya yang menghasilkan sentimen positif, netral, dan negatif untuk tiap dataset perlu untuk dilakukan *encoding*, hal ini dilakukan karena program hanya mengenali kelas numerik. Untuk hasil dari *encoding* label yaitu negatif = 0, netral = 1, dan positif = 2. Adapun proses *encoding* label dilakukan dengan bantuan modul dari *sklearn* yaitu *LabelEncoder* yang dapat dilihat pada Gambar 5.12.

```
# import pustaka sklearn
from sklearn.preprocessing import LabelEncoder

# instansiasi objek LabelEncoder
label_encoder = LabelEncoder()

# encode Label ganjar
# negatif = 0, netral = 1, positif = 2
df_ganjar['Sentimen_encoded'] = label_encoder.fit_transform(df_ganjar['Sentimen'])

# encode Label Anies
# negatif = 0, netral = 1, positif = 2
df_anies['Sentimen_encoded'] = label_encoder.fit_transform(df_anies['Sentimen'])

# encode Label Prabowo
# negatif = 0, netral = 1, positif = 2
df_prabowo['Sentimen_encoded'] = label_encoder.fit_transform(df_prabowo['Sentimen'])
```

Gambar 5. 12 Label Encoding

5.6.2 Konfigurasi Model Pada Modul *Sklearn*

Berdasarkan pada Tabel 5.15 terdapat 1 *input layer* dan 1 *output layer* yang mana *node* pada *input layer* dan *node* pada *output layer* berjumlah sebanyak dataset cuitan pengguna Twitter untuk masing-masing bakal calon Presiden Indonesia yaitu 700 *node*. Pemilihan jumlah *node* pada setiap *hidden layer* dilakukan secara acak. Inisialisasi *learning rate* awal yaitu dengan nilai 0.001, bias dan bobot

pada konfigurasi model ditentukan secara acak oleh sistem. Fungsi aktivasi yang digunakan adalah relu, sedangkan maksimal *epoch* pelatihan adalah sebesar 300 *epoch*.

Tabel 5. 15 Konfigurasi Model Pada *Sklearn*

<i>Input Layer</i>	<i>Output Layer</i>	<i>Hidden Layer</i>		Fungsi Aktivasi
		HL1	HL2	
1	1	130	130	Relu

5.6.3 Pembagian Data *Training* dan Data *Testing*

Pembagian data *training* dan data *testing* untuk setiap dataset bakal calon Presiden Republik Indonesia 2024 dapat dilihat pada Tabel 5.16.

Tabel 5. 16 Pembagian Data *Training* dan Data *Testing*

Dataset	Data <i>Training</i>	Data <i>Testing</i>
Ganjar Pranowo	70%	30%
Prabowo Subianto	70%	30%
Anies Baswedan	70%	30%

Berdasarkan pada Tabel 5.16 data *training* dan data *testing* dibagi berdasarkan dataset masing-masing bakal calon Presiden Republik Indonesia 2024. Persentase pembagian antara data *training* dan data *testing* adalah sebesar 70% untuk data *training* dan 30% untuk data *testing*. Berdasarkan pembagian data *training* dan data *testing* akan dilakukan permodelan untuk setiap bakal calon yang

akan menghasilkan 3 model *Backpropagation*. Adapun kode program *Backpropagation* menggunakan modul *MLPClassifier* neural network pada *sklearn* dapat dilihat pada Gambar 5.13.

```
# import pustaka sklearn
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split

# split data training dan data testing dataset ganjar
X_train_ganjar, X_test_ganjar, y_train_ganjar, y_test_ganjar = train_test_split(X_ganjar,
                                                                              y_ganjar, test_size=0.3, random_state=0)
X_train_anies, X_test_anies, y_train_anies, y_test_anies = train_test_split(X_anies,
                                                                              y_anies, test_size=0.3, random_state=0)
X_train_prabowo, X_test_prabowo, y_train_prabowo, y_test_prabowo = train_test_split(X_prabowo,
                                                                              y_prabowo, test_size=0.3, random_state=0)

# instansiasi objek TFIDF
vectorizer = TfidfVectorizer(analyzer='word', max_features=300, max_df=0.5, use_idf=True, norm='l2')

# proses mengubah dataset ganjar text menjadi TF-IDF vector
tfidf_ganjar = vectorizer.fit_transform(X_train_ganjar)
X_test_tf_ganjar = vectorizer.transform(X_test_ganjar)
# anies
tfidf_anies = vectorizer.fit_transform(X_train_anies)
X_test_tf_anies = vectorizer.transform(X_test_anies)
#prabowo
tfidf_prabowo = vectorizer.fit_transform(X_train_prabowo)
X_test_tf_prabowo = vectorizer.transform(X_test_prabowo)

#konfigurasi model
learning_rate = 0.001
hidden_layer = (130, 130)
epoch = 300

# model ganjar
model_ganjar = MLPClassifier(hidden_layer_sizes=hidden_layer, max_iter = epoch,
                             learning_rate_init = learning_rate, activation='relu', solver='adam', verbose=True)

# model prabowo
model_prabowo = MLPClassifier(hidden_layer_sizes=hidden_layer, max_iter = epoch,
                              learning_rate_init = learning_rate, activation='relu', solver='adam', verbose=True)

# model anies
model_anies = MLPClassifier(hidden_layer_sizes=hidden_layer, max_iter=epoch,
                            learning_rate_init = learning_rate, activation='relu', solver='adam', verbose=True)

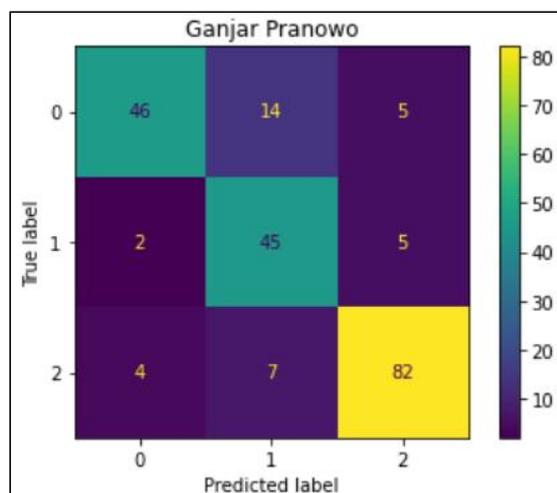
# training
model_ganjar.fit(tfidf_ganjar, y_train_ganjar)
model_anies.fit(tfidf_anies, y_train_anies)
model_prabowo.fit(tfidf_prabowo, y_train_prabowo)
```

Gambar 5. 13 Model BPNN Sklearn

5.7 Evaluasi Model

Agar dapat diketahui apakah model yang dibangun sudah baik, maka diperlukan suatu pengujian. Pengujian yang digunakan untuk mengevaluasi 3 model yang telah dibangun sebelumnya dilakukan menggunakan *confusion matrix*. Berdasarkan hasil dari *confusion matrix* maka akan didapatkan nilai *accuracy*, *precision*, *recall*, dan *f-measure*. Pengukuran *confusion matrix* dilakukan dengan bantuan modul

ConfusionMatrixDisplay pada *sklearn*. Hasil dari pengukuran *confusion matrix* untuk model Ganjar Pranowo dapat dilihat pada Gambar 5.14.



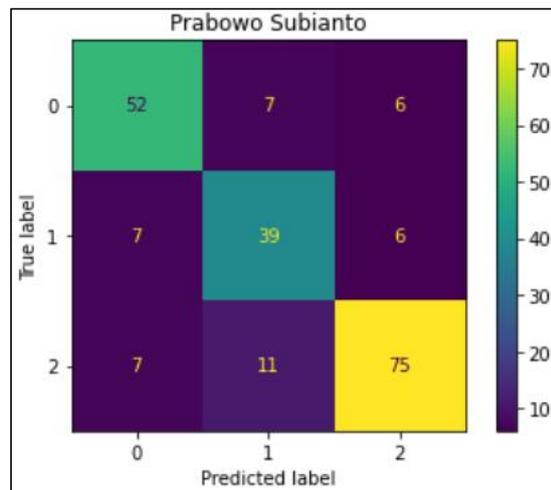
Gambar 5. 14 Confusion Matrix Ganjar Pranowo

Berdasarkan pada Gambar 5.14 dapat dijelaskan bahwa *true* label merupakan kelas sebenarnya dari dataset, sedangkan *predicted* label merupakan kelas hasil prediksi dari sistem. Total data yang ada pada tabel *confusion matrix* merupakan jumlah dari keseluruhan 30% data *testing* yang telah diinisialisasi pada tahap pembagian data *training* dan data *testing*. Nilai 82 yang ada pada kotak warna kuning merupakan total data yang diprediksi memiliki kelas positif oleh sistem sama dengan kelas aktualnya. Semakin besar nilai secara diagonal merepresentasikan bahwa model yang dibangun semakin baik. Hasil *accuracy*, *precision*, *recall*, dan *f-measure* berdasarkan Gambar 5.14 dapat dilihat pada Gambar 5.15.

	precision	recall	f1-score	support
0	0.88	0.71	0.79	65
1	0.68	0.87	0.76	52
2	0.89	0.88	0.89	93
accuracy			0.82	210
macro avg	0.82	0.82	0.81	210
weighted avg	0.84	0.82	0.82	210

Gambar 5. 15 Accuracy Confusion Matrix Ganjar Pranowo

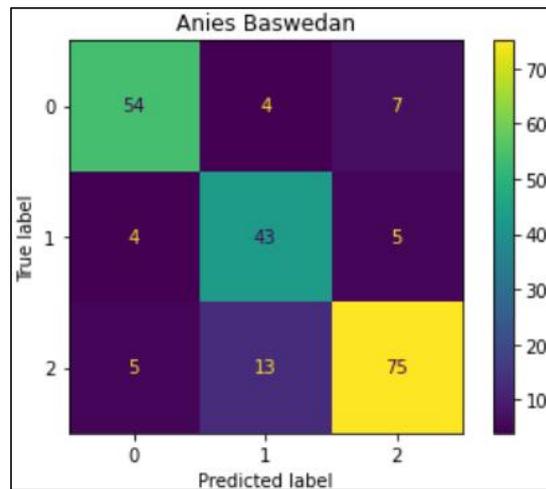
Adapun hasil pengujian *confusion matrix* untuk model Prabowo Subianto dan Anies Baswedan dapat dilihat pada Gambar 5.16, Gambar 5.17, Gambar 5.18, dan Gambar 5.19.



Gambar 5. 16 Confusion Matrix Model Prabowo Subianto

	precision	recall	f1-score	support
0	0.79	0.80	0.79	65
1	0.68	0.75	0.72	52
2	0.86	0.81	0.83	93
accuracy			0.79	210
macro avg	0.78	0.79	0.78	210
weighted avg	0.80	0.79	0.79	210

Gambar 5. 17 Accuracy Model Prabowo Subianto



Gambar 5. 18 Confusion Matrix Model Anies

	precision	recall	f1-score	support
0	0.88	0.71	0.79	65
1	0.68	0.87	0.76	52
2	0.89	0.88	0.89	93
accuracy			0.82	210
macro avg	0.82	0.82	0.81	210
weighted avg	0.84	0.82	0.82	210

Gambar 5. 19 Accuracy Model Anies

Berdasarkan hasil dari pengujian ketiga model menggunakan *confusion matrix* didapatkan *accuracy* yang cukup baik, rata-rata *accuracy* untuk tiap model sekitar 0,81 atau jika dipersentase nilainya mencapai 81%. Model tersebut sudah bisa digunakan untuk memprediksi dataset baru yang belum memiliki kelas atau label sentimen. Proses prediksi dataset baru berdasarkan masing-masing bakal calon akan dilakukan pada tahap selanjutnya yaitu untuk mencari nilai *preference value* dari bakal calon Presiden Republik Indonesia 2024.

5.8 Preference Value

Preference value bertujuan untuk mengetahui respon positif pengguna Twitter terhadap bakal calon Presiden Republik Indonesia 2024 yang dibahas pada penelitian ini. Skenario yang dilakukan untuk mencari *preference value* pada penelitian ini adalah dengan melakukan klasifikasi sentimen untuk data yang belum memiliki label menggunakan model-model yang telah dilakukan evaluasi pada tahapan sebelumnya. Dataset yang digunakan adalah dataset yang berbeda dengan data *training* dan data *testing* sebelumnya. Jumlah dataset yang digunakan untuk masing-masing bakal calon adalah 1000 data hasil *crawling* yang telah dilakukan tahap preprocessing. Adapun contoh hasil prediksi masing-masing model dapat dilihat pada Tabel 5.17, Tabel 5.18, dan Tabel 5.19.

Tabel 5. 17 Prediksi Sentimen Dataset Prabowo Subianto

Dataset Prabowo Subianto	Prediksi Sentimen
bengkulu Fren catet ya berdasar LSN mayoritas nahdliyin mulai menjadi mendukung Prabowo untuk ikut dalam pemilu 2024 mendatang	Positif
Kalo emang sengaja membiarkan prabowo akan semakin nyungsep di pilpres mendatang	Negatif
Iya juga kedekatan dengan Pak Jokowi menjadikan orang makin melirik Pak Prabowo untuk memimpin di Jokowi Prabowo Akrab Indonesia Tanpa Polarisasi Pemimpin Pemersatu Bangsa	Netral

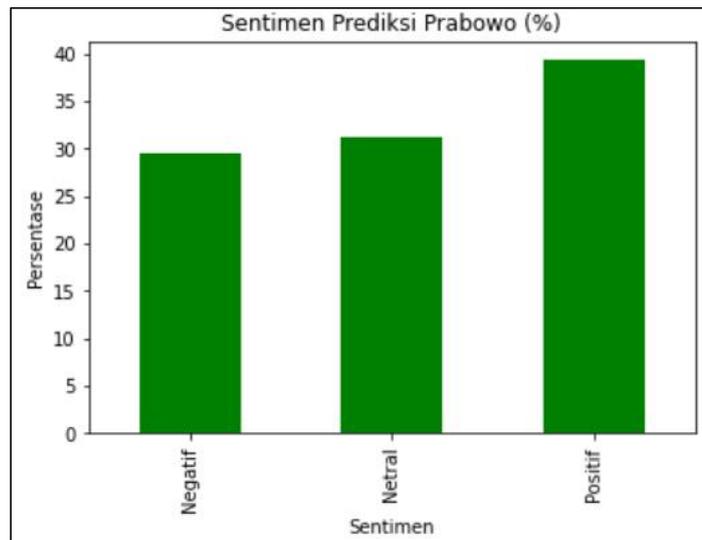
Tabel 5. 18 Prediksi Sentimen Dataset Ganjar Pranowo

Dataset Ganjar Pranowo	Prediksi Sentimen
Ganjar tidak berprestasi buat Rakyatini fakta cuma jago medsos	Negatif
Ganjar Pranowo presiden pilihan rakyat	Positif
Tanda pemilu sudah dekat	Netral

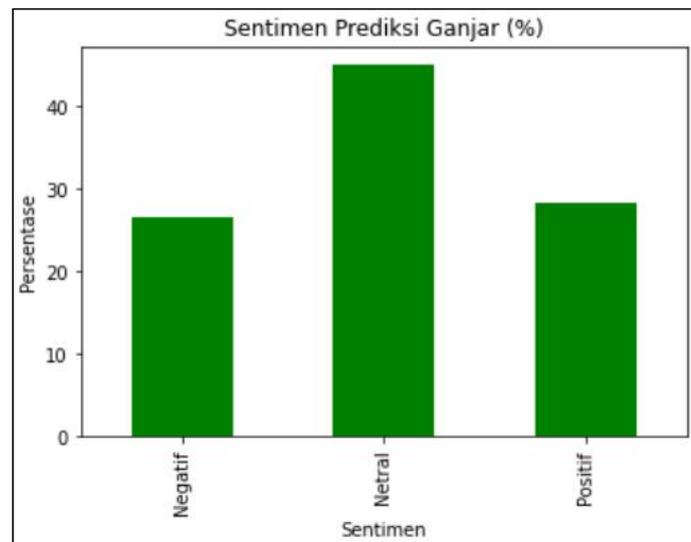
Tabel 5. 19 Prediksi Sentimen Anies Baswedan

Dataset Anies Baswedan	Prediksi Sentimen
sibuk dengan pencitraan nya Untuk capres Tapi jakarta di tinggalkan nya dengan banyak masalah Yg terparah adalah tingkat kemiskinan yg terus meningkat Padahal APBD sangat besar dgn jumlah penduduk yg tidak besar Lalu di kemanakan duit tsb Asmara	Negatif
bisa kerjanya juga dong warga jakarta mengakui	Positif
Berbahaya sekali Ada Indikasi pecah belah bangsa	Netral

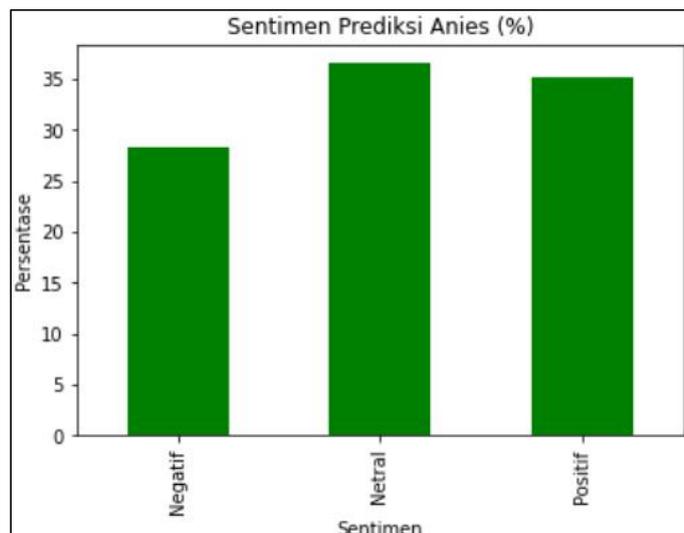
Adapun persentase polaritas sentimen untuk seluruh dataset dari masing-masing model dapat dilihat pada Gambar 5.20, Gambar 5.21, dan Gambar 5.22.



Gambar 5. 20 Persentase Sentimen Prabowo Subianto



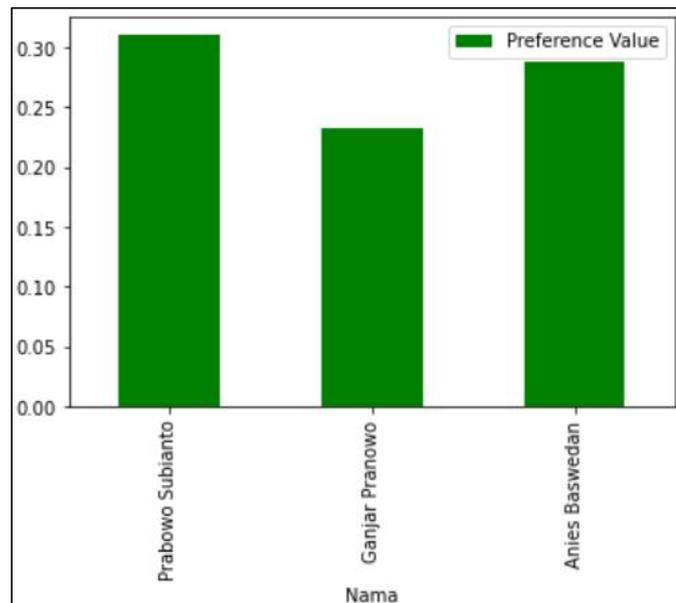
Gambar 5. 21 Persentase Sentimen Ganjar Pranowo



Gambar 5. 22 Persentase Sentimen Anies Baswedan

Berdasarkan Gambar 5.20, 5.21, dan 5.22 dapat dilihat bahwa persentase sentimen untuk dataset Prabowo Subianto adalah 39,3% sentimen positif, 31,1% sentimen netral, dan 29,5% sentimen negatif. Untuk dataset Ganjar Pranowo adalah 28,2% sentimen positif, 45% sentimen netral, dan 26,7% sentimen negatif. Sedangkan untuk dataset Anies Baswedan didapatkan 35,1% sentimen positif, 36,5% sentimen netral, dan 28,3% sentimen negatif. Persentase tersebut berdasarkan dari total masing-masing 1000 dataset.

Preference value bakal calon Presiden didapatkan berdasarkan jumlah sentimen masing-masing dataset yang dilakukan perhitungan dengan Persamaan (3.19), sehingga didapatkan nilai perbandingan *preference value* dari masing-masing bakal calon. Hasil grafik perbandingan *preference value* untuk bakal calon Presiden Republik Indonesia 2024 dapat dilihat pada Gambar 5.23.



Gambar 5. 23 Perbandingan *Preference Value*

Berdasarkan Gambar 5.23 dapat dilihat bahwa Prabowo Subianto mendapatkan respon positif paling tinggi yaitu dengan nilai *preference value* sebesar 0,31 atau 31%. Diikuti dengan Anies Baswedan dengan nilai *preference value* sebesar 0,29 atau 29%, kemudian Ganjar Pranowo mendapatkan nilai *preference value* paling rendah yaitu sebesar 0,23 atau 23%. Sehingga dapat disimpulkan bahwa Prabowo Subianto merupakan bakal calon Presiden Republik Indonesia 2024 yang memiliki respon positif dan elektabilitas paling baik berdasarkan data cuitan yang dikumpulkan pada bulan Juli 2022.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka dapat ditarik beberapa kesimpulan yaitu:

1. Algoritme *Backpropagation* dapat digunakan untuk model klasifikasi sentimen cuitan pengguna Twitter dengan 70% data *training* dan 30% data *testing* didapatkan rata-rata akurasi 81%.
2. Hasil pengujian masing-masing model dengan menggunakan *confusion matrix* didapatkan rata-rata *accuracy* sebesar 81%, *precision* 80,3%, *recall* 81%, dan *f-measure* 80%.
3. Berdasarkan perhitungan nilai *preference value* didapatkan hasil Prabowo Subianto 0,31 atau 31%, Anies Baswedan 0,29 atau 29%, Ganjar Pranowo 0,23 atau 23%. Sehingga dapat diketahui Prabowo Subianto memiliki elektabilitas paling baik berdasarkan cuitan Twitter bulan Juli 2022.

6.2 Saran

Adapun saran mengenai penelitian sejenis untuk lebih memperhatikan tahapan *preprocessing*, sebab data yang dihasilkan cuitan Twitter masih banyak yang tidak baku sehingga perlu dilakukan normalisasi. Selain itu, untuk mendapatkan model yang bagus dataset yang juga bisa lebih diperbanyak lagi.

DAFTAR PUSTAKA

- Arhami, M., & Nasir, M. 2020. *Data Mining, Algoritma dan Implementasi*. Penerbit ANDI. Sleman.
- Aslam, S. 2021. *Twitter by the Numbers: Stats, Demographics & Fun Facts*. (<https://www.omnicoreagency.com/twitter-statistics>). Diakses pada tanggal 30 Maret 2022).
- Assuja, M. A., & Saniati. 2016. *Analisis Sentimen Tweet Menggunakan Backpropagation Neural Network*. Jurnal Teknoinfo, vol. 10(2), pp. 23–28.
- Bengfort, B., Bilbro, R., & Ojeda, T. 2018. *Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning*. O'Reilly Media. Sebastopol.
- Géron, A. 2017. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media. Sebastopol.
- Haga, S. G., Lhaksmana, M. K., & Murdiansyah, D. T. 2018. *Klasifikasi Sentimen Terhadap Bakal Calon Gubernur Jawa Barat 2018 di Twitter Menggunakan Naive Bayes*. E-Proceeding of Engineering, vol. 5(1).
- Kristiyanti, D. A., Normah, & Umam, A. H. 2019. *Prediction of Indonesia Presidential Election Results for the 2019-2024 Period Using Twitter Sentiment Analysis*. 2019 5th International Conference on New Media Studies.
- Kulkarni, A., & Shivananda, A. 2019. *Natural Language Processing Recipes, Unlocking Text Data with Machine Learning and Deep Learning using Python*. Apress. Bangalore.
- Liu, B. 2016. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers. Chicago.
- Mckinney, W. 2018. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython* (2nd ed.). O'Reilly Media. Sebastopol.
- Mejova, Y. 2009. *Sentiment Analysis: An Overview*. Computer Science Department, University of Iowa.
- Mitchell, R. E. 2018. *Web Scraping with Python: Collecting More Data from the Modern Web* (2nd ed.). O'Reilly Media. Sebastopol.
- Morgan, P. 2016. *Data Analysis From Scratch with Python*. AI Sciences. Tallinn.
- Müller, A. C., & Guido, S. 2017. *Introduction to Machine Learning with Python*. O'Reilly Media. Sebastopol.
- Pierson, L. 2017. *Data Science For Dummies, 2nd Edition* (2nd ed.). John Wiley & Sons. Hoboken.

- Purbo, O. W. 2019. *Text Mining - Analisis MedSos, Kekuatan Brand & Intelejen di Internet*. Andi Publisher. Yogyakarta.
- Ricky, B., Irawan, B., & Setianingsih, C. 2020. *Analisis Sentimen Data Pada BPJS Kesehatan Dengan Metode Backpropagation Neural Network*. Teknik Komputer, Universitas Telkom.
- Rozi, I. F., Pramitarini, Y., & Puspitasari, N. 2020. *Analisis Mengenai Calon Presiden Indonesia 2019 Di Twitter Menggunakan Metode Backpropagation*. JIP (Jurnal Informatika Polinema).
- Russell, M. A., & Klassen, M. 2019. *Mining the Social Web* (3rd ed.). O'Reilly Media. Sebastopol.
- Santoso, E. B., & Nugroho, A. 2019. *Analisis Sentimen Calon Presiden Indonesia 2019 Berdasarkan Komentar Publik Di Facebook*. Eksplora Informatika, vol. 9(1), pp. 60–69.
- Supardi, J., & Utami, A. S. 2014. *Development of Artificial Neural Network Architecture for Face Recognition in Real Time*. International Journal of Machine Learning and Computing, vol. 4(1).
- Swamynathan, M. 2017. *Mastering Machine Learning with Python in Six Steps*. Apress. Bangalore.
- Wicaksono, H. Y. 2022. *Analisis Sentimen Data Twitter Mengenai Program Vaksinasi Di Indonesia Menggunakan Algoritma Backpropagation*. Jurnal Ilmiah Matematika, vol. 09.
- Windarto, A. P., Nasution, D., Tambunan, A. W. F., Hasibuan, M. S., Siregar, M. N. H., Solikhun, M. R. L., Fadhilah, Y., & Nofriansyah, D. 2020. *Jaringan Saraf Tiruan: Algoritma Prediksi dan Implementasi*. Yayasan Kita Menulis. Medan.

Listing Code

```
In [ ]: import matplotlib.pyplot as plt
import joblib
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, classification_report
```

```
In [ ]: # Load model dan data test prabowo
file1_prabowo = 'model/valid/test/prabowo_model_new.pkl'
model_prabowo = joblib.load(file1_prabowo)

# Load data test
file2_prabowo = 'model/valid/test/X_test_tf_prabowo.pkl'
X_test_tf_prabowo = joblib.load(file2_prabowo)

# Load data test y
file3_prabowo = 'model/valid/test/y_test_prabowo.pkl'
y_test_prabowo = joblib.load(file3_prabowo)
```

```
In [ ]: # Load model dan data test ganjar
file1_ganjar = 'model/valid/ganjar/ganjar_model_valid_acc82%.pkl'
model_ganjar = joblib.load(file1_ganjar)

# Load data test
file2_ganjar = 'model/valid/ganjar/X_test_tf_ganjar.pkl'
X_test_tf_ganjar = joblib.load(file2_ganjar)

# Load data test y
file3_ganjar = 'model/valid/ganjar/y_test_ganjar.pkl'
y_test_ganjar = joblib.load(file3_ganjar)
```

```
In [ ]: # Load model dan data test anies
file1_anies = 'model/valid/anies/anies_model_acc_82%.pkl'
model_anies = joblib.load(file1_anies)

# Load data test
file2_anies = 'model/valid/anies/X_test_tf_anies.pkl'
X_test_tf_anies = joblib.load(file2_anies)

# Load data test y
file3_anies = 'model/valid/anies/y_test_anies.pkl'
y_test_anies = joblib.load(file3_anies)
```

```
In [ ]: # confusion matrix model anies
result_anies = model_anies.predict(X_test_tf_anies)
result_anies
con_mat_anies = confusion_matrix(y_test_anies, result_anies)
con_mat_anies
print(classification_report(y_test_anies, result_anies))
```

```
In [ ]: # display hasil dari confusion matrix anies
disp_anies = ConfusionMatrixDisplay(confusion_matrix=con_mat_anies,
                                     display_labels=model_anies.classes_)
disp_anies.plot()
plt.title('Anies Baswedan')
plt.show()
```

```
In [ ]: # confusion matrix model Ganjar
result_ganjar = model_ganjar.predict(X_test_tf_ganjar)
result_ganjar
con_mat_ganjar = confusion_matrix(y_test_ganjar, result_ganjar)
con_mat_ganjar
print(classification_report(y_test_ganjar, result_ganjar))
```

```
In [ ]: # display hasil dari confusion matrix ganjar
disp_ganjar = ConfusionMatrixDisplay(confusion_matrix=con_mat_ganjar,
                                     display_labels=model_ganjar.classes_)
disp_ganjar.plot()
plt.title('Ganjar Pranowo')
plt.show()
```

```
In [ ]: # prabowo model predict
result_prabowo = model_prabowo.predict(X_test_tf_prabowo)
result_prabowo
con_mat_prabowo = confusion_matrix(y_test_prabowo, result_prabowo)
con_mat_prabowo
print(classification_report(y_test_prabowo, result_prabowo))
```

```
In [ ]: # display hasil dari confusion matrix prabowo
disp_prabowo = ConfusionMatrixDisplay(confusion_matrix=con_mat_prabowo,
                                     display_labels=model_prabowo.classes_)
disp_prabowo.plot()
plt.title('Prabowo Subianto')
plt.show()
```