

**PENERAPAN ALGORITMA *MULTILAYER PERCEPTRON*
PADA SENTIMEN PENGGUNA *TWEET* TERHADAP VAKSIN
COVID-19**

SKRIPSI

OLEH:

FORDINAND HALOMOAN PASARIBU

188160061



PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

MEDAN

2023

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 5/7/23

Access From (repository.uma.ac.id)5/7/23

**PENERAPAN ALGORITMA *MULTILAYER PERCEPTRON*
PADA SENTIMEN PENGGUNA *TWEET* TERHADAP VAKSIN
COVID-19**

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana di Fakultas Teknik
Universitas Medan Area



**PROGRAM STUDI TEKNIK INFROMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2023**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 5/7/23

Access From (repository.uma.ac.id)5/7/23

HALAMAN PENGESAHAN

Judul Skripsi : Penerapan Algoritma *Multilayer Perceptron* Pada Sentimen Pengguna *Tweet* Terhadap Vaksin Covid-19
Nama : Fordinand Halomoan Pasaribu
NPM : 188160061
Fakultas : Teknik



Tanggal Lulus : 19 Januari 2023

HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 19 Januari 2023



Fordinand Halomoan Pasaribu
188160061

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan di bawah ini:

Nama : Fordinand Halomoan Pasaribu
NPM : 188160061
Program Studi : Teknik Informatika
Fakultas : Teknik
Jenis karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul : **Penerapan Algoritma *Multilayer Perceptron* Pada Sentimen Pengguna *Tweet* Terhadap Vaksin COVID-19**

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Medan Area berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan
Pada tanggal : 19 Januari 2023
Yang menyatakan



(Fordinand Halomoan Pasaribu)

RIWAYAT HIDUP

Penulis dilahirkan di Pematang Siantar, pada tanggal 30 Maret 2000 dari ayah Herkules Pasaribu dan ibu Rabunga Nainggolan. Penulis merupakan putra ke-1 (pertama) dari 2 (dua) bersaudara.

Tahun 2018, Penulis lulus dari SMA Negeri 1 Bandar Perdagangan Simalungun dan di tahun yang sama juga terdaftar sebagai mahasiswa Fakultas Teknik Informatika di Universitas Medan Area.

Selama mengikuti perkuliahan, penulis ikut serta dalam perlombaan Komunitas Kompetisi Siber Indonesia (KCSI), serta aktif mengikuti Unit Kegiatan Mahasiswa (UKM) cabang olahraga Basket pada tahun ajaran 2018 dan 2019.

Penulis melaksanakan Program Magang di salah satu startup Indonesia, kemudian ikut pelaksanaan Bootcamp yang diadakan oleh Kominfo.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala karunia-Nya sehingga skripsi ini berhasil diselesaikan. Tema yang dipilih dalam penelitian ini adalah Analisis Sentimen dengan judul **PENERAPAN ALGORITMA MULTILAYER PERCEPTRON PADA SENTIMEN PENGGUNA TWEET TERHADAP VAKSIN COVID-19**. Terima kasih penulis sampaikan kepada:

1. Tuhan Yang Maha Esa, karena berkat-Nya yang melimpah sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
2. Kedua orang tua penulis, Bapak Herkules Pasaribu dan Rabunga Nainggolan yang selalu mendoakan dan selalu memberikan dukungan kepada penulis dalam menyelesaikan skripsi ini.
3. Bapak Dr. Rahmad Syah, S.Kom, M.Kom selaku Dekan Fakultas Teknik Universitas Medan Area.
4. Bapak Rizki Muliono, S.Kom, M.Kom selaku Kepala Program Studi Teknik Informatika.
5. Ibu Nurul Khairina, S.Kom, M.Kom selaku Dosen Pembimbing I dan Bapak Dr. Dian Noviandri, S.T, M.Kom selaku Dosen Pembimbing II yang telah memberikan arahan, bimbingan, dan motivasi kepada penulis sehingga skripsi ini bisa selesai dengan baik.
6. Seluruh dosen Fakultas Teknik Universitas Medan Area atas segala ilmu pengetahuan yang diberikan kepada penulis. Beserta seluruh staff akademik Fakultas Teknik Universitas Medan Area.

7. Kepada seluruh teman – teman Teknik Informatika angkatan 2018 yang telah memberikan semangat, bantuan, motivasi, dan doa kepada penulis agar dapat menyelesaikan skripsi ini dengan baik.
8. Kepada pengagum rahasia yaitu Yosepin Hutagaol yang memberikan kata mutiara “Jangan Pesimis, tapi Harus Optimis”.
9. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dalam menyelesaikan penulisan skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan, oleh karena itu kritis dan saran yang bersifat membangun sangat penulis harapkan demi kesempurnaan skripsi ini. Penulis berharap skripsi ini dapat bermanfaat baik untuk kalangan pendidikan maupun masyarakat. Akhir kata penulis ucapkan terima kasih.

Medan, 16 Januari 2023



Fordinand Halomoan Pasaribu
(188160061)

ABSTRAK

Organisasi Kesehatan Dunia (WHO) mengumumkan adanya kluster *pneumonia* di Wuhan, China pada akhir tahun 2019 yang kemudian diberi nama COVID-19 (*Corona Virus Disease 2019*). Bulan Maret 2020 WHO menyatakan COVID-19 menjadi pandemi dunia dikarenakan penyebarannya dengan begitu cepat serta menginfeksi orang di seluruh dunia. Kemudian pada bulan November 2020 vaksin COVID-19 dengan tingkat efikasi di atas 90% akhirnya ditemukan dan siap untuk digunakan. Munculnya vaksin COVID-19 telah mengakibatkan pro serta kontra pada masyarakat. Ada yang mendukung vaksin, ada yang mewaspadaai vaksin, bahkan meskipun pemerintah memberikan vaksin secara gratis, tetapi masih ada juga yang menolaknya, dan informasi tentang vaksin menyebar luas pada media sosial terutama *twitter*. Pengadaan vaksin corona menyebabkan timbulnya opini-opini yang beragam di masyarakat. Vaksin COVID-19 menjadi trending topik di media sosial *twitter*. Opini di *twitter* kemudian akan menjadi data untuk dilakukan analisis sentimen yang merupakan suatu proses bertujuan untuk mengetahui apakah polaritas suatu data berupa teks akan mengarah ke positif, negatif, atau netral. Metode yang digunakan pada penelitian ini yaitu pengumpulan data, *text preprocessing*, TF-IDF, algoritma *multilayer perceptron*, serta pengujian dengan *confusion matrix*. Dari jumlah data 228208 data opini positif, negatif, dan netral pada pengguna *tweet* mengenai vaksin COVID-19 dengan perbandingan *training* 90% dan *testing* 10%, model akan lebih banyak belajar menggunakan data *training* dengan jumlah yang besar. Hasil performa dari penelitian ini mendapatkan performa tertinggi pada akurasi 81.2%, presisi 83.8%, dan *recall* 71.2%. Hasil *wordcloud* pada opini positif terdapat 3 topik yaitu mengenai ketersediaan, berbayar, dan dosis. Opini negatif pengguna *tweet* terhadap vaksin COVID-19 mendapatkan 2 pokok permasalahan seperti efek samping vaksin dan kematian. Opini netral terdapat 3 topik seperti dosis, ketersediaan, umur, dan tanggal kadaluwarsa.

Kata kunci : Analisis Sentimen, *Twitter*, Vaksin COVID-19, Algoritma *Multilayer Perceptron*, TF-IDF

ABSTRACT

The World Health Organization (WHO) announced a pneumonia cluster in Wuhan, China at the end of 2019 which was later named COVID-19 (Corona Virus Disease 2019). In March 2020 WHO declared COVID-19 to be a world pandemic because it spread so quickly and infected people all over the world. Then in November 2020, a COVID-19 vaccine with an efficacy level above 90% was finally found and ready to be used. The emergence of the COVID-19 vaccine has raised pros and cons in society. Some support vaccines and some are wary of vaccines, even though the government provides vaccines for free, there are still those who reject them, and information about vaccines is spreading widely on social media, especially Twitter. Procurement of the corona vaccine has led to the emergence of various opinions in society. The COVID-19 vaccine has become a trending topic on Twitter and social media. Opinions on Twitter will then become data for sentiment analysis, which is a process that aims to find out whether the polarity of textual data is positive, negative, or neutral. The methods used in this study are data collection, text preprocessing, TF-IDF, multilayer perceptron algorithms, and testing with a fusion matrix. From the total data of 228208 positive, negative, and neutral opinion data on tweet users about the COVID-19 vaccine with a training ratio of 90% and testing 10%, the model will learn more using large amounts of training data. The performance results from this study obtained the highest performance at 81.2% accuracy, 83.8% precision, and 71.2% recall. Wordcloud results in positive opinions there are 3 topics, namely regarding availability, payment, and dosage. Negative opinions of tweet users about the COVID-19 vaccine get 2 main issues such as vaccine side effects and death. Neutral opinion on 3 topics such as dosage, availability, age, and expiration date.

Keywords: Sentiment Analysis, Twitter, COVID-19 Vaccine, Multilayer Perceptron Algorithm, TF-IDF

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN	iii
RIWAYAT HIDUP	v
KATA PENGANTAR	vi
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	4
1.3. Tujuan Penelitian.....	4
1.4. Batasan Masalah.....	4
1.5. Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA	7
2.1 <i>Data Mining</i>	7
2.2 <i>Natural Language Processing (NLP)</i>	9
2.3 COVID-19 (<i>Corona Virus Disease 2019</i>).....	10
2.4 Vaksin COVID-19.....	10
2.5 Analisis Sentimen.....	11
2.6 Sosial Media.....	12
2.7 <i>Text Preprocessing</i>	12
2.7.1 <i>Tokenization</i>	13
2.7.2 <i>Normalization</i>	13
2.7.3 <i>Stopwords</i>	14
2.7.4 <i>Lemmatization</i>	14
2.7.5 TF-IDF.....	15
2.8 <i>Google Colaboratory</i>	16
2.9 <i>Algoritma Multilayer Perceptron</i>	17
2.10 Penelitian Terdahulu.....	23

BAB III METODE PENELITIAN	26
3.1 Objek Penelitian	27
3.2 Teknik Pengolahan dan Analisis Data.....	27
3.2.1 Studi Literatur dan Pemahaman	27
3.2.2 Jenis Data	27
3.2.3 Pengumpulan Data	28
3.2.4 <i>Text Preprocessing</i>	29
3.3 Perhitungan Manual <i>Multilayer Perceptron</i>	38
BAB IV HASIL DAN PEMBAHASAN	44
4.1 Kebutuhan Perangkat Penelitian	44
4.1.1 Perangkat Keras	44
4.1.2 Perangkat Lunak.....	44
4.2 Hasil.....	44
4.2.1 Pengumpulan Data	45
4.2.2 <i>Text Preprocessing</i>	46
4.2.3 Algoritma <i>Multilayer Perceptron</i>	50
4.2.4 Evaluasi.....	55
4.3 Pembahasan	74
BAB V KESIMPULAN DAN SARAN	80
5.1 Kesimpulan.....	80
5.2 Saran.....	80
DAFTAR PUSTAKA	81
LAMPIRAN	86

DAFTAR GAMBAR

Gambar 2. 1 Tahapan Data Mining.....	9
Gambar 2. 2 Arsitektur Multilayer Perceptron	17
Gambar 3. 1 Metode Penelitian.....	26
Gambar 3. 2 Pengambilan Data dari Kaggle.....	28
Gambar 3. 3 Tahapan Text Preprocessing	30
Gambar 3. 4 Emoticon	32
Gambar 4. 1 Tampilan Dataset	45
Gambar 4. 2 Data Labeling Otomatis	46
Gambar 4. 3 Tahapan Tokenization	46
Gambar 4. 4 Tahapan Normalization	47
Gambar 4. 5 Tahapan Menghapus Emoticon.....	48
Gambar 4. 6 Tahapan Stopwords	48
Gambar 4. 7 Tahapan Lemmatization.....	49
Gambar 4. 8 Tahapan TF-IDF.....	50
Gambar 4. 9 Tahapan Pengujian 1	50
Gambar 4. 10 Hasil Performa Pengujian 1.....	52
Gambar 4. 11 Tahapan Pengujian 2	52
Gambar 4. 12 Hasil Performa Pengujian 2.....	53
Gambar 4. 13 Tahapan Pengujian 3	54
Gambar 4. 14 Hasil Performa Pengujian 3.....	55
Gambar 4. 15 Tampilan Rapidminer.....	57
Gambar 4. 16 Tampilan Split Data	57
Gambar 4. 17 Parameter Neural Network.....	58
Gambar 4. 18 Performa Akurasi	59
Gambar 4. 19 Performa Recall.....	59
Gambar 4. 20 Performa Presisi	59
Gambar 4. 21 Wordcloud Positif	68
Gambar 4. 22 Wordcloud Negatif.....	70
Gambar 4. 23 Wordcloud Netral.....	71
Gambar 4. 24 Persentase Opini.....	73

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	23
Tabel 3. 1 Data Latih.....	29
Tabel 3. 2 Tahap Tokenization.....	30
Tabel 3. 3 Tahap Normalization.....	31
Tabel 3. 4 Tahap Stopwords.....	32
Tabel 3. 5 Tahap Lemmatization	33
Tabel 3. 6 Term Frequency	34
Tabel 3. 7 Document Frequency	34
Tabel 3. 8 Inverse Document Frequency	35
Tabel 3. 9 Hasil Perhitungan TF-IDF	36
Tabel 3. 10 Hasil TF-IDF Berdasarkan Term	38
Tabel 4. 1 Perangkat Keras	44
Tabel 4. 2 Perangkat Lunak	44
Tabel 4. 3 Confusion Matrix Pengujian 1 Training	51
Tabel 4. 4 Confusion Matrix Pengujian 1 Testing	51
Tabel 4. 5 Confusion Matrix Pengujian 2 Training	53
Tabel 4. 6 Confusion Matrix Pengujian 2 Testing	53
Tabel 4. 7 Confusion Matrix Pengujian 3 Training	54
Tabel 4. 8 Confusion Matrix Pengujian 3 Testing	54
Tabel 4. 9 Pengujian dan Parameter.....	56
Tabel 4. 10 Hasil Performa	56
Tabel 4. 11 Confusion Matrix 3x3	60
Tabel 4. 12 Keterangan Kelas	60
Tabel 4. 13 Pengujian Confusion Matrix Training 1	60
Tabel 4. 14 Pengujian Confusion Matrix Testing 1	62
Tabel 4. 15 Pengujian Confusion Matrix Training 2	63
Tabel 4. 16 Pengujian Confusion Matrix Testing 2	64
Tabel 4. 17 Pengujian Confusion Matrix Training 3	65
Tabel 4. 18 Pengujian Confusion Matrix Testing 3	66
Tabel 4. 19 Asosiasi Kata Positif.....	68

Tabel 4. 20 Asosiasi Kata Negatif.....	70
Tabel 4. 21 Asosiasi Kata Netral.....	72
Tabel 4. 22 Permasalahan dan Solusi.....	74
Tabel 4. 23 Daftar Kata yang Tidak bisa di Normalkan	77



BAB I PENDAHULUAN

1.1. Latar Belakang

Organisasi Kesehatan Dunia (WHO) mengumumkan adanya kluster pneumonia di Wuhan, China pada akhir tahun 2019 yang kemudian diberi nama COVID-19 (Boon-Itt & Skunkan, 2020). COVID-19 menyebar dengan begitu cepat serta menginfeksi orang di seluruh dunia sehingga di bulan Maret 2020, WHO menyatakan COVID-19 (*Corona Virus Disease 2019*) menjadi pandemi dunia (World Health Organization, 2020). Satu tahun sesudah pengumumannya sebagai pandemi, hingga saat ini dunia masih belum bisa membaik dari COVID-19. Selama kurang lebih 11 bulan, vaksin COVID-19 dengan tingkat efikasi di atas 90% akhirnya ditemukan dan siap untuk digunakan. WHO menyatakan bahwa pada bulan November 2020 ada lebih dari 200 vaksin virus corona telah ditemukan atau bahkan dikembangkan dan sekitar 40 di antaranya berpeluang untuk masuk uji klinis. Vaksin yang dikembangkan oleh *Pfizer* di Jerman, *Moderna* di Amerika Serikat, dan *AstraZeneca* di Inggris mampu memberikan perlindungan maksimal terhadap virus COVID-19. Hingga saat ini, setidaknya ada 7 vaksin yang beredar setelah melalui tahap akhir atau fase III uji klinis, yaitu *Moderna*, *Pfizer/BioNTech*, *Sinovac*, *Sputnik V*, *Sinopharm*, *Covaxin*, dan *Oxford/AstraZeneca*. Munculnya vaksin COVID-19 telah mengakibatkan pro serta kontra pada masyarakat. Ada yang mendukung vaksin, ada yang mewaspadaai vaksin, bahkan meskipun pemerintah memberikan vaksin secara gratis, tetapi masih ada juga yang menolaknya (Putri, 2020). Informasi tentang vaksin menyebar luas pada media sosial (Puri dkk., 2020). Larangan pergerakan serta kewajiban untuk tetap di rumah

akibat pandemi COVID-19, sosial media seperti *twitter* sudah menjadi kawasan bagi penggunaannya untuk mengungkapkan kekhawatiran, pendapat, serta pandangan mereka pada COVID-19. Individu, Forum Kesehatan, serta pemerintah juga memakai *twitter* untuk berinteraksi tentang COVID-19 (Chandrasekaran dkk., 2020).

Perbincangan publik terkait oleh fasilitas kesehatan serta prosedur pemerintah di *twitter* bisa menjadi salah satu tolak ukur untuk mengetahui sentimen, khususnya untuk rumor terbaru seperti COVID-19. *Twitter* adalah layanan pengguna sosial media yang untuk memudahkan interaksi oleh pengguna lainnya dengan menyampaikan pesan yang mempunyai ukuran 280 karakter yang disebut *tweet* (Maulana, 2017). Keunggulan *twitter* dibandingkan media sosial lainnya adalah mendukung penyebaran berita lebih cepat yang akhirnya akan menjadi sebuah subjek perbincangan yang akan dibahas oleh para pemakainya. Hal ini mempermudah publik untuk memperoleh informasi terbaru karena beritanya yang di perbarui setiap waktu. *Twitter* juga mempermudah untuk terhubung oleh pengguna lain tanpa wajib saling *follow* ketika ingin mengetahui nama akun yang disebutkan (*mention*). Kebanyakan orang termakan informasi dari sosial media buat pertama kalinya, daripada sumber lama lainnya seperti TV, radio, dan surat kabar (Wong & Burkell, 2017). Masyarakat yakin apa yang mereka dapatkan pada media sosial, yang menyebabkan mereka sensitif atas isu serta informasi bohong. Begitu juga berita pandemi virus COVID-19 yang berlangsung di dunia termasuk Indonesia semenjak bulan Maret 2020 yang terus bermunculan pada aplikasi sosial media salah satunya *twitter*. Pengadaan vaksin *corona* menyebabkan timbulnya opini-opini yang beragam di masyarakat. Vaksin COVID-19 menjadi trending topik

di media sosial *twitter*. Opini di *twitter* kemudian akan menjadi data untuk dilakukan analisis sentimen.

Analisis sentimen adalah metode yang dirancang demi menentukan apakah polaritas data dalam bentuk teks (surat, kalimat, atau alinea) menuju ke positif, negatif, atau netral (Fauzi, 2018). Analisis Sentimen dapat digunakan untuk menentukan opini publik tentang suatu isu seperti korupsi, serta demonstrasi berdasarkan data tekstual. Sentimen analisis pada data *twitter* pernah dilakukan menggunakan metode RNN (*Recurrent Neural Network*) (Nemes & Kiss, 2021), penelitian ini mendapat akurasi sebesar 75% dari 1100 *dataset*. Hal ini disebabkan metode yang dikerjakan sekadar *tokenizing* serta *filtering* akibatnya masih banyak data ataupun teks yang tidak dibersihkan secara baik. Peneliti akan menganalisis data *tweet* yang diambil dari situs *kaggle.com* yang merupakan komunitas online *data science* dan *machine learning*. *Kaggle* mengizinkan pengguna agar mendapatkan serta mempublikasikan *dataset*, mengkaji dan membuat model dalam area *data science* berbasis web, beroperasi sebagai seorang *data scientist* serta *machine learning*, dan berpartisipasi dalam kompetisi untuk memecahkan tantangan *data science*. Adapun proses *text preprocessing* yang akan dilakukan oleh peneliti ialah *tokenization*, *normalization*, menghapus *emoticon*, *stopwords*, *lemmatization*, dan TF-IDF. Algoritma yang digunakan pada penelitian ini ialah *Multilayer Perceptron* (MLP) yang mana merupakan arsitektur *perceptron* yang paling banyak digunakan untuk jaringan saraf. Lapisan *perceptron* digabungkan dan membentuk arsitektur *multilayer*, hal ini memberikan kompleksitas yang diperlukan dari pemrosesan jaringan saraf. Dengan menambahkan lebih banyak lapisan dan lebih banyak neuron per lapisan, akan meningkatkan spesialisasi model

untuk melatih data (Ciaburro & Venkateswaran, 2017). MLP mempunyai satu atau lebih lembaran tersembunyi antara lapisan masukan serta lapisan keluaran. *Neuron* berpengaruh di tiap-tiap lapisan, hubungan antar *neuron* bersumber sejak lapisan bawah sampai ke lapisan atas, pada lapisan yang sama neuron tidak saling berkoneksi (Ramchoun, 2016). MLP adalah bagian dari arsitektur Jaringan Syaraf Tiruan (JST) yang mempunyai kerumitan rendah serta dapat memberikan hasil yang memenuhi (Rani, 2017).

1.2. Perumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan di atas, bahwa dapat dibuat rumusan masalah ialah bagaimana penerapan metode *multilayer perceptron* dan tingkat akurasi pengklasifikasian sentimen pengguna *tweet* terhadap vaksin COVID-19.

1.3. Tujuan Penelitian

Berdasarkan perumusan masalah diatas maka tujuan dilakukan penelitian ini ialah:

1. Menganalisis ketepatan hasil sentimen masyarakat pengguna *tweet* terhadap vaksin COVID-19.
2. Menguji tingkat akurasi, presisi, dan *recall* dalam menggunakan metode *Multilayer Perceptron* terhadap sentimen pengguna *tweet*.

1.4. Batasan Masalah

Adapun batasan masalah pada penelitian ini ialah:

1. Data yang diperlukan untuk menganalisis sentimen pengguna *tweet* berasal dari situs <https://www.kaggle.com/gpreda/all-covid19-vaccines-tweets>.
2. Menggunakan algoritma *Multilayer Perceptron*.
3. Evaluasi pengujian menggunakan akurasi, presisi, dan *recall*.
4. Jumlah data yang digunakan sebanyak 228208.
5. Data diambil dari bulan Desember 2020 sampai dengan bulan November 2021.
6. Melakukan ekstraksi data penelitian ini menggunakan tahapan *text preprocessing* diantaranya *tokenization*, *normalization*, menghapus *emoticon*, *stopwords*, *lemmatization*, dan TF-IDF.
7. Membagi opini menjadi 3 jenis klasifikasi yaitu positif, negatif, dan netral.
8. Melakukan pengujian dengan proporsi data *train* 70% dan *test* 30%, *train* 80% dan *test* 20%, serta *train* 90% dan *test* 10%.
9. Parameter yang digunakan adalah 1 *hidden layer* dengan jumlah *neuron* 61, fungsi aktivasi '*sigmoid biner*', dan *alpha* 0.2.
10. Visualisasi kata terbanyak yang digunakan dengan *WordCloud*.
11. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *Python* melalui *Google Colaboratory* dan *rapidminer* sebagai *tools* perbandingan performa.

1.5. Manfaat Penelitian

Adapun manfaat yang diharapkan dalam penelitian ialah:

1. Membantu pemerintah untuk mengetahui persepsi pengguna sosial media terhadap vaksin COVID-19, agar masyarakat tidak perlu khawatir dalam melakukan vaksinasi.
2. Mengedukasi masyarakat terutama pengguna sosial media terkait vaksin COVID-19.
3. Diharapkan menjadi referensi bagi penelitian berikutnya mengenai sentimen masyarakat terhadap vaksin COVID-19.



BAB II TINJAUAN PUSTAKA

2.1 *Data Mining*

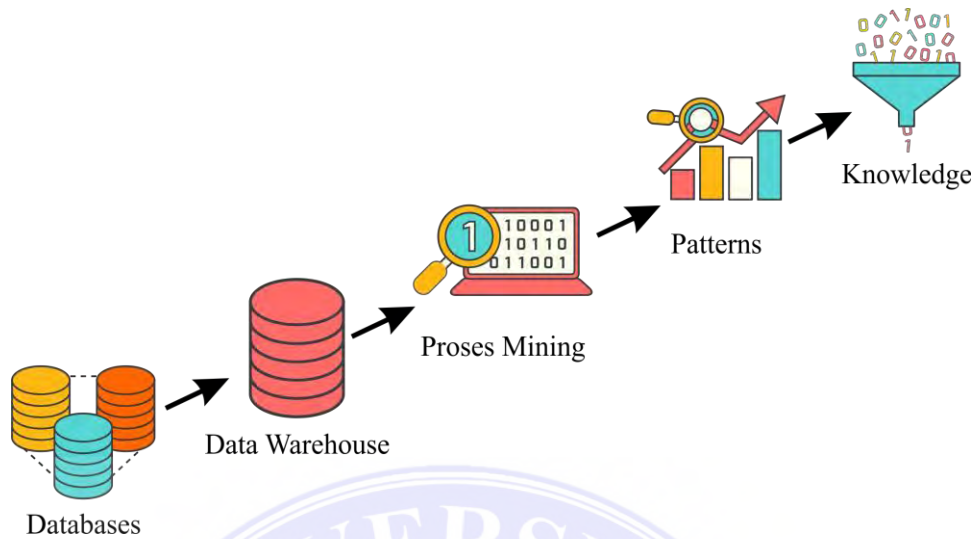
Data Mining (Penambang Data) ialah sebagian bidang ilmu komputer yang berfokus pada *machine learning* (Muliono dkk., 2019). Penambang data mengandung pencarian trend serta model yang dibutuhkan dalam basis data yang banyak demi mendukung membuat kesimpulan pada masa depan. Model-model ini dapat diidentifikasi bagi perangkat – perangkat khusus bahwa mampu membagikan satu penguraian data yang bermakna serta mendalam yang akhirnya bisa diamati secara lebih menyeluruh menggunakan perangkat pendukung keputusan yang lainnya.

Tingkatan pada penambang data terbagi dalam beberapa bagian (Muslim dkk., 2019) yaitu:

1. Pembersihan Data (*Data Cleaning*) merupakan proses yang dilakukan untuk menghilangkan *noise* dari data yang tidak konsisten atau yang bisa disebut dengan data asing. Tidak semua data yang diperoleh dari *database* perusahaan dan hasil eksperimen yang ada memiliki entri yang sempurna, seperti data yang hilang, data yang tidak valid. Data yang tidak relevan dapat disiasati dengan membuang atau biasa dikenal dengan proses pembersihan. Proses pembersihan dapat mempengaruhi kinerja teknik *data mining*.
2. Integrasi Data (*Data Integration*) ialah proses menggabungkan data dari berbagai *database* menjadi *database* baru. Data yang dibutuhkan dalam proses

data mining tidak hanya berasal dari satu *database*, tetapi juga dari beberapa *database*.

3. Seleksi Data (*Data Selection*) yaitu tidak semua data yang terdapat dalam *database* akan digunakan karena hanya data yang sesuai yang akan dianalisis dan diambil dari *database*.
4. Transformasi Data (*Data Transformation*) adalah proses mengubah dan menggabungkan data ke dalam format tertentu. *Data mining* memerlukan format data khusus sebelum dapat diterapkan.
5. Proses *Mining* adalah proses utama menggunakan metode untuk menemukan pengetahuan berharga yang tersembunyi dalam data.
6. Evaluasi Pola (*Pattern Evaluation*) bertugas untuk mengidentifikasi pola yang menarik berdasarkan pengetahuan. Tahap ini, tipikal model klasifikasi dievaluasi untuk menilai apakah asumsi yang ada memang terpenuhi. Jika ternyata hasil yang diperoleh tidak sesuai dengan hipotesis, ada beberapa alternatif yang dapat diambil, seperti *feed back* untuk memperbaiki proses *data mining*, atau mencoba metode *data mining* lain yang lebih tepat.
7. Presentasi Pengetahuan (*Knowledge Presentation*) adalah visualisasi serta penyajian pengetahuan tentang proses yang dipakai untuk mendapatkan pemahaman atau informasi yang telah digali oleh pengguna. Alur terakhir dari proses *data mining* yaitu menyusun ketetapan dari hasil analisis yang didapat.



Gambar 2. 1 Tahapan *Data Mining*

Sumber: (Muslim dkk., 2019)

2.2 *Natural Language Processing (NLP)*

NLP adalah pemrosesan bahasa alami, seperti kata lisan dan tulisan yang digunakan manusia untuk perbincangan sehari-hari via *computer*. Sistem komputer memproses bahasa wajib diwakili oleh serangkaian ikon/lambang yang sesuai dengan ketentuan khusus. Sementara metodenya, NLP akan mengatur *computer* untuk mengetahui tiap-tiap instruksi ataupun bahasa standar yang biasanya ditulis maupun dikerjakan oleh manusia. Tugas NLP meliputi *tokenization* dan *parsing*, *lemmatization/stemming*, *part-of-speech tagging*, *language detection*, dan *semantic relationship recognition*. Secara umum, tugas NLP adalah menguraikan bahasa menjadi blok-blok penyusunnya, kemudian memahami hubungan antara blok-blok tersebut, dan mengeksplorasi bagaimana blok-blok tersebut bekerja sama untuk menciptakan makna. (Rosyadi dkk., 2020).

2.3 COVID-19 (*Corona Virus Disease 2019*)

Coronavirus termasuk virus yang menyerang saluran pernapasan. Virus yang berhubungan dengan infeksi pada saluran pernapasan akan menggunakan sel epitel dan mukosa saluran napas sebagai target awal dan menyebabkan infeksi pada saluran pernapasan atau kerusakan organ (Susilo dkk., 2020). *Coronavirus* pada umumnya menyerang hewan khususnya kelelawar dan unta. *Coronavirus* mempunyai sampul (*enveloped*), dengan partikel bulat dan seringkali berbentuk pleomorfik. Dinding *coronavirus* dilapisi oleh protein S sebagai protein antigenik utama yang dapat berikatan dengan reseptor yang ada di tubuh hostnya. Terdapat enam jenis *coronavirus* yang ditemukan di saluran napas pada manusia yaitu 229E, NL63 dari genus *Polyonum*, OC43 dan HPU dari genus beta, *Middle East Respiratory Syndrome-associated Coronavirus* (MERS-CoV), and *Severe Acute Respiratory Syndrome-associated Coronavirus* (SARS-CoV). *Coronavirus* jenis baru atau SARS-CoV2 penyebab Covid-19 dapat diklasifikasikan dalam kelompok betacoronavirus yang menyerupai SARS-CoV dan MERS-CoV tetapi tidak sama persis (Wang dkk., 2020).

2.4 Vaksin COVID-19

The emergency committee telah menyatakan bahwa penyebaran COVID-19 dapat dihentikan jika dilakukan proteksi, deteksi dini, isolasi, dan perawatan yang cepat agar tercipta implementasi sistem yang kuat untuk menghentikan penyebaran COVID-19 (Sun dkk., 2020). Mengingat hal ini, sebagai upaya proteksi terhadap COVID-19, berbagai negara dari seluruh dunia telah berkomitmen bersama dengan melibatkan pemerintah, perusahaan bioteknologi, ilmuwan, dan akademisi untuk

dapat menciptakan vaksin COVID-19. Sejauh ini telah banyak kandidat vaksin yang diluncurkan untuk melawan virus SARS-CoV-2, penyebab COVID-19 (Prompetchara dkk., 2020).

Vaksin merupakan unsur ataupun campuran yang menghasilkan ketahanan fisik akan satu penyakit. Vaksin mengandung beraneka ragam macam serta kandungan, yang tiap-tiapnya mampu membuat pertahanan akan bermacam-macam penyakit serius. Vaksin dapat disuntikan dalam bentuk cair atau bahkan diminum (Putra, 2021). Vaksinasi adalah proses pemberian vaksin melalui disuntikkan maupun diteteskan ke dalam mulut untuk meningkatkan produksi antibody guna menangkal penyakit tertentu (Iskak dkk., 2021).

2.5 Analisis Sentimen

Analisis sentimen ialah termasuk dari penambang teks akan pengelompokkan kontradiksi (polaritas) dari teks, klasifikasi terbilang dikerjakan demi memprediksi bagaimana kontradiksi dari suatu tulisan apakah yang dibagikan bersifat positif, negatif, atau netral (Fanissa dkk., 2018). Analisis sentimen adalah bagian keahlian dari penambang teks, pemrograman bahasa alami, serta kecerdasan buatan. Analisis sentimen merupakan proses mengekstraksi, mengolah dan memahami data berupa teks yang tidak terstruktur secara otomatis guna mengambil informasi sentimen yang terdapat pada sebuah kalimat pendapat atau opini (Brahimi dkk., 2021). Analisis sentimen dapat diterapkan pada opini semua bidang seperti ekonomi, politik, sosial dan hukum. Media sosial Twitter ini membuka jendela bagi para peneliti untuk mempelajari emosi, suasana hati, dan pendapat publik melalui analisis sentimen (Qiu dkk., 2019).

2.6 Sosial Media

Pemakaian sosial media pada publik telah berkembang sebagian besar. Munculnya teknologi serta penerapan yang modern dalam aktivitas publik saat ini membantu tumbuhnya hubungan sosial lewat media sosial (Hao dkk., 2016). Berdasarkan kemenkominfo.go.id *user* internet di Indonesia memiliki 63 juta *user* internet. Sebagian besar pengguna mendapatkan 63 juta orang yang mengkoneksikan sosial media. Salah satu media sosial terbesar di dunia ialah *Twitter* yang profitnya mendapatkan USD 145 juta. Media sosial *Twitter* sering dimanfaatkan demi penyampaian pesan berbentuk pandangan maupun penilaian publik terkait topik-topik khusus maupun diskusi yang merupakan topik baru yang menarik.

Sosial Media telah banyak mengubah dunia, memutarbalikkan banyak pemikiran dan teori yang dimiliki. Level komunikasi digabungkan dalam suatu wadah yang disebut jejaring sosial. Dampak yang kelihatan pula harus berhati-hati, yang artinya sosial media selalu membuat peluang kepada setiap orang yang berpartisipasi dalam mengekspresikan pendapat mereka secara bebas. Seperti media sosial lainnya, *twitter* dapat pula menjadi sumber perbincangan oleh pengguna. Suatu berita muncul sebagai headline di media online pembahasannya sudah terlebih dahulu dibahas di *twitter*. Bahkan pembahasan di *twitter* memicu suatu berita muncul (Priyatno dkk., 2019).

2.7 Text Preprocessing

Text preprocessing teks ini diterapkan pada data yang akan diperlukan berisi prosedur analisis sentimen, dan data yang kita proses akan mengambil data yang

terdapat di dalamnya berdasarkan sentimen pengarangnya ialah positif, negatif atau netral. Untuk memudahkan dalam mengelola data, kita perlu memberikan analisis sentimen secara menurut panduan beserta melafalkan tujuan dari wacana yang muncul di dalam sentimen tersebut, kemudian bisa dibuat evaluasi sehingga sentimen tersebut adalah sentimen positif, negatif ataupun netral.

Fungsi utama dari pra-pemrosesan teks ialah buat membersihkan data, menghapus ataupun menangani data kosong serta menangani data yang kurang sempurna. Adapun maksud dari tahapan ini ialah agar hasil kalkulasi menjadi lebih maksimal (Putra, 2021).

2.7.1 *Tokenization*

Tokenisasi merupakan proses penguraian string teks menjadi term atau kata. Tujuannya yaitu memisahkan kata-kata dalam sebuah paragraf, kalimat atau halaman ke dalam kata tunggal (Najjichah dkk., 2019). Misalnya, tokenisasi kalimat "aku ingin makan nasi goreng bakso" menghasilkan enam token: "aku", "ingin", "makan", "nasi", "goreng", "bakso". Umumnya, pembatas antar token dikutip sebagai spasi dan tanda baca.

2.7.2 *Normalization*

Menghapus yang mengandung *link* ataupun *url* dalam sepotong kalimat. Kasus yang diambil pada teks akan diselesaikan untuk kategorisasi pendapat positif, negatif, dan netral. Format teks *link* ataupun *url* maupun huruf besar tidak akan menurunkan akibat apapun pada proses uraian, apalagi bisa mengganggu hasil final, akibatnya *link* ataupun *url* mampu dibuang, sedangkan *text* yang mengandung campuran huruf besar dapat diubah menjadi huruf kecil semua. Sedangkan

pengertian lain normalisasi merupakan proses perubahan data menjadi bentuk normal. Proses ini dilakukan penskalaan terhadap data menjadi dalam rentang nilai tertentu (Haryati dkk., 2016).

2.7.3 *Stopwords*

Stopwords merupakan langkah mengumpulkan kata yang penting dari hasil token/bagian melalui penggunaan proses menghilangkan *words* yang tidak penting. *Stopwords* ialah kata-kata normal yang sering ditemukan dalam jumlah banyak serta dipandang tidak mempunyai manfaat. Ilustrasi *stopwords* di bahasa Indonesia ialah “dan”, “yang”, “dari”, “di”, dll. Manfaat penerapan *stopwords* adalah dengan menghilangkan *words* yang mempunyai keterangan rendah dari suatu teks, kita bisa fokus terhadap *words* yang bermakna sebagai penggantinya (Najjichah dkk., 2019).

2.7.4 *Lemmatization*

Lemmatization merupakan proses mendapatkan format dasar dari sebuah kata. Muncul konsep yang menguraikan bahwa lematisasi ialah proses yang bertujuan untuk menormalkan teks ataupun kata-kata berlandaskan bentuk dasarnya (yaitu format morfologis), morfologi merupakan bidang yang mempelajari seluk-beluk bentuk kata serta pengaruh perubahan-perubahan bentuk kata terhadap golongan dan arti kata. Normalisasi di sini mengandung arti mengartikan serta menghilangkan awalan dan akhiran dari suatu kata. *Lemmatization* digunakan untuk menemukan bentuk dasar dari suatu kata sehingga dapat dimanfaatkan untuk memeriksa kebenaran dari penggunaan ejaan pada suatu kata (Yusup dkk., 2019).

2.7.5 TF-IDF

Pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) merupakan proses ekstraksi fitur lewat cara memberi nilai pembobotan di setiap kata yang terkandung dalam suatu arsip, ataupun bisa dibayangkan mengonversi kata-kata menjadi angka. Manfaat dari langkah ini ialah buat menaksir pentingnya suatu kalimat di dokumen. Upaya menentukan kualitas TF-IDF ialah seperti berikut ini (Pravina dkk., 2019).

Tahap untuk mengerjakan pembobotan TF-IDF ialah:

1. Tentukan kata kunci yang akan dipakai.
2. Tentukan nilai TF bisa memakai TF *biner*, ialah misalnya satu kata ditemukan dalam sebuah arsip akan diberi nilai 1 (satu), misalnya tidak akan diberi nilai 0 (nol).
3. Tentukan nilai *df* dengan cara menaksir jumlah arsip yang tercantum kata khusus.
4. Menaksir nilai *Inverse Document Frequency* (IDF) yang bermanfaat untuk memahami seberapa pentingnya kata dalam sebuah dokumen/arsip. IDF dirumuskan sebagai berikut.

$$IDF = \log \frac{D}{df} \quad 2.1$$

Keterangan:

df = banyak dokumen yang mengandung *term*

D = banyak seluruh dokumen.

5. Sesudah itu menaksir TF-IDF dengan memadukan kalkulasi TF dengan IDF sebagai berikut.

$$W_{ij} = tf \times IDF \quad 2.2$$

$$W_{ij} = tf \times \log \frac{D}{df} \quad 2.3$$

Keterangan:

W_{ij} = bobot kata dalam tiap-tiap dokumen

Tf = banyak kemunculan kata dalam dokumen

D = banyak seluruh dokumen

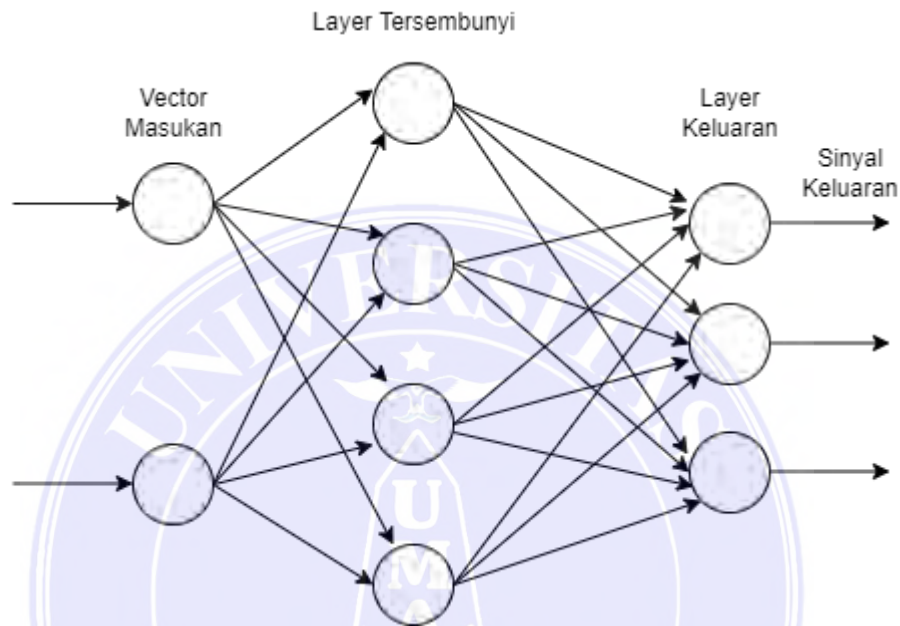
6. Sesudah itu jumlahkan tiap-tiap bobot kata kunci pada masing-masing dokumen.

2.8 Google Colaboratory

Google Colaboratory/Google Colab adalah salah satu produk *Google* yang mereplikasi *Jupiter Notebook* tetapi berbasis *cloud Google*. Perbedaannya adalah *Google Colab* dibuat khusus untuk para *programmer* atau *researcher* yang mungkin kesulitan untuk mendapatkan akses *computer* dengan spek tinggi. *Google colab* memungkinkan pengguna menjalankan kode Python tanpa perlu melakukan proses instalasi terlebih dahulu dan proses *set up* lainnya. Oleh karena itu, *software* ini merupakan yang terbaik bagi *programmer* yang ingin mengasah pengetahuan mengenai Python. Python merupakan salah satu bahasa pemrograman yang populer

saat ini yang dibuat oleh Guido van Rossum pada tahun 1990 di Belanda tepatnya di *Centrum Wiskunde & Informatica* (Wibowo dkk., 2020).

2.9 Algoritma *Multilayer Perceptron*



Gambar 2. 2 Arsitektur *Multilayer Perceptron*

Sumber: (Guntoro dkk., 2019)

Multilayer Perceptron merupakan ANN (*Artificial Neural Network*) yang diturunkan dari Perceptron, berbentuk ANN umpan balik (*feedforward*) dengan satu ataupun lebih lapisan tersembunyi (*hidden layer*). Umumnya, jaringan terdiri atas lapisan input, setidaknya satu lapisan *neuron* komputasi di tengah (tersembunyi), dan sebuah lapisan *neuron* komputasi *output*.

ANN umumnya terdiri atas 3 ataupun justru 4 lapisan termasuk 1 ataupun 2 *hidden layer*. Tiap-tiap lapisan dapat berisikan 10 sampai 1000 *neuron*. ANN terbukti dapat memiliki 5 ataupun justru 6 lapisan (3 ataupun 4 *hidden layer*) di mana tiap-tiap *layer* menetapkan jutaan *neuron*, namun di sebagian besar

pemakaian menetapkan 2 lapisan (1 *hidden layer*) sebab setiap pengembangan 1 lapisan akan meningkatkan beban komputasi secara berulang-ulang (Syadid, 2019).

Jaringan saraf merupakan model prediktif yang berdasarkan perilaku *neuron* biologis. Suatu jaringan saraf tiruan khas buatan mungkin memiliki seratus *neuron*. Seperti analogi, pola saraf manusia dipercaya mempunyai kurang lebih 3×10^{10} *neuron*.

Pola *Perceptron* yang dikembangkan oleh Frank Rosenbalt mempunyai 3 layer yaitu (Rusma, 2019):

1. Memasukkan “retina” yang ditetapkan untuk *layer* kedua.
2. “*Asosiasi unit*” yang memadukan *input* dengan bobot serta memicu manfaat ambang tahap yang menarik ke *layer output*.
3. Lapisan keluaran yang memadukan nilai-nilai.

Lapisan-lapisan dari *Multilayer Perceptron* ialah sebagai berikut:

1. Lapisan masukan (*input layer*)

Terjadi dari *neuron* yang memperoleh data inputan dari variabel X. Semua *neuron* di lapisan ini bisa berkorelasi ke *neuron* pada *hidden layer* ataupun langsung ke lapisan luaran jika jaringan tidak menetapkan *hidden layer*.

2. Lapisan tersembunyi (*hidden layer*)

ialah unit-unit yang berada di dalam serta tertanam yang mana keluarannya tidak bisa secara langsung dipelajari.

3. Lapisan keluaran (*output layer*)

Ialah untuk mendapatkan satu set nilai bobot yang akan membuat *output* dari jaringan saraf agar berbanding terhadap nilai-nilai target yang seharusnya sedekat mungkin. Terdapat kurang lebih rumor yang terlibat dalam menyiapkan dan membentuk jaringan *multilayer perceptron*:

1. Pilih berapa banyak *hidden layer* yang akan diperlukan dalam jaringan.
2. Tentukan berapa banyak *neuron* yang akan diperlukan pada tiap-tiap *hidden layer*.
3. Temukan penyelesaian maksimum global yang mencegah minimum lokal.
4. Konvergensi ke penyelesaian maksimum pada masa durasi yang normal.
5. Validasi jaringan saraf untuk mengevaluasi *overfitting*.

Pada dasarnya sebuah *neuron* memiliki 2 koneksi, yaitu :

1. *Inhibitory relationship*, ialah korelasi yang menginformasikan neuron untuk tidak meningkatkan sinyal *input*.
2. *Excitatory relationship*, ialah korelasi yang menginformasikan kepada neuron untuk meningkatkan sinyal masukan.

Gambar 2.2 terletak 1 lapisan tersembunyi dengan 4 *neuron*, serta satu lapisan keluaran dengan 3 *neuron*. Berikut adalah persamaan dari algoritma *Multilayer Perceptron* (Guntoro dkk., 2019) :

1. Inialisasi bobot dengan bilangan acak kecil.
2. Jika kondisi penghentian belum terpenuhi, maka lakukan langkah 2 – 8.
3. Untuk setiap pasang data pelatihan, lakukan langkah 3 – 8.

4. Tiap unit masukan menerima sinyal dan meneruskan ke unit tersembunyi di atasnya.
5. Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$z_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad 2.4$$

$$z_i = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}} \quad 2.5$$

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

$$y_net_k = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad 2.6$$

$$y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \quad 2.7$$

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

$$\delta_k = (t_k - y_k) f'(y_net_k) = (t_k - y_k) y_k (1 - y_k), \quad t_k = \text{Target} \quad 2.8$$

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

$$\Delta w_{kj} = \alpha \delta_k z_j, \quad k = 1, 2, \dots, m; j = 1, 2, \dots, p \quad 2.9$$

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

$$\delta_net_j = \sum_{k=1}^m \delta_k w_{kj} \quad 2.10$$

Faktor δ unit tersembunyi.

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \quad 2.11$$

Hitung suku perubahan v_{ji}

$$\Delta v_{ji} = \alpha \delta_j x_i, \quad j = 1, 2, \dots, p; i = 1, 2, \dots, n \quad 2.12$$

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

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj}, \quad 2.13$$

$$(k = 1, 2, \dots, m; j = 1, 2, \dots, p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi, yaitu:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji}, \quad 2.14$$

$$(j = 1, 2, \dots, p; i = 1, 2, \dots, n)$$

Fungsi aktivasi merupakan fungsi yang digunakan untuk mengolah inputan informasi. Beberapa fungsi aktivasi yang digunakan pada ANN (Ryandhi, 2017) adalah :

1. Fungsi Aktivasi Linear

Menghasilkan nilai output yang sama dengan nilai input. Dapat digunakan dengan rumus :

$$f(x) = x \quad 2.15$$

2. Fungsi Aktivasi *Sigmoid Biner*

Menghasilkan nilai output pada interval 0 hingga 1. Rumus fungsi ini adalah:

$$y = f(x) = \frac{1}{1 + \exp(-x)} \quad 2.16$$

3. Fungsi aktivasi *Rectified Linear Unit* (ReLU)

Fungsi ReLU pada dasarnya melakukan threshold dari 0 hingga infinity. ReLU dapat menutupi kelemahan yang dimiliki oleh Sigmoid dan Tanh. Rumusnya adalah:

$$y(x) = \max(0, x) \quad 2.17$$

4. Fungsi *Hyperbolic Tangent*

Fungsi Tanh mempunyai rentang -1 hingga 1. Rumusnya adalah:

$$y(x) = \frac{2}{1 + e^{(-2x)}} \quad 2.18$$

2.10 Penelitian Terdahulu

Tabel 2. 1 Penelitian Terdahulu

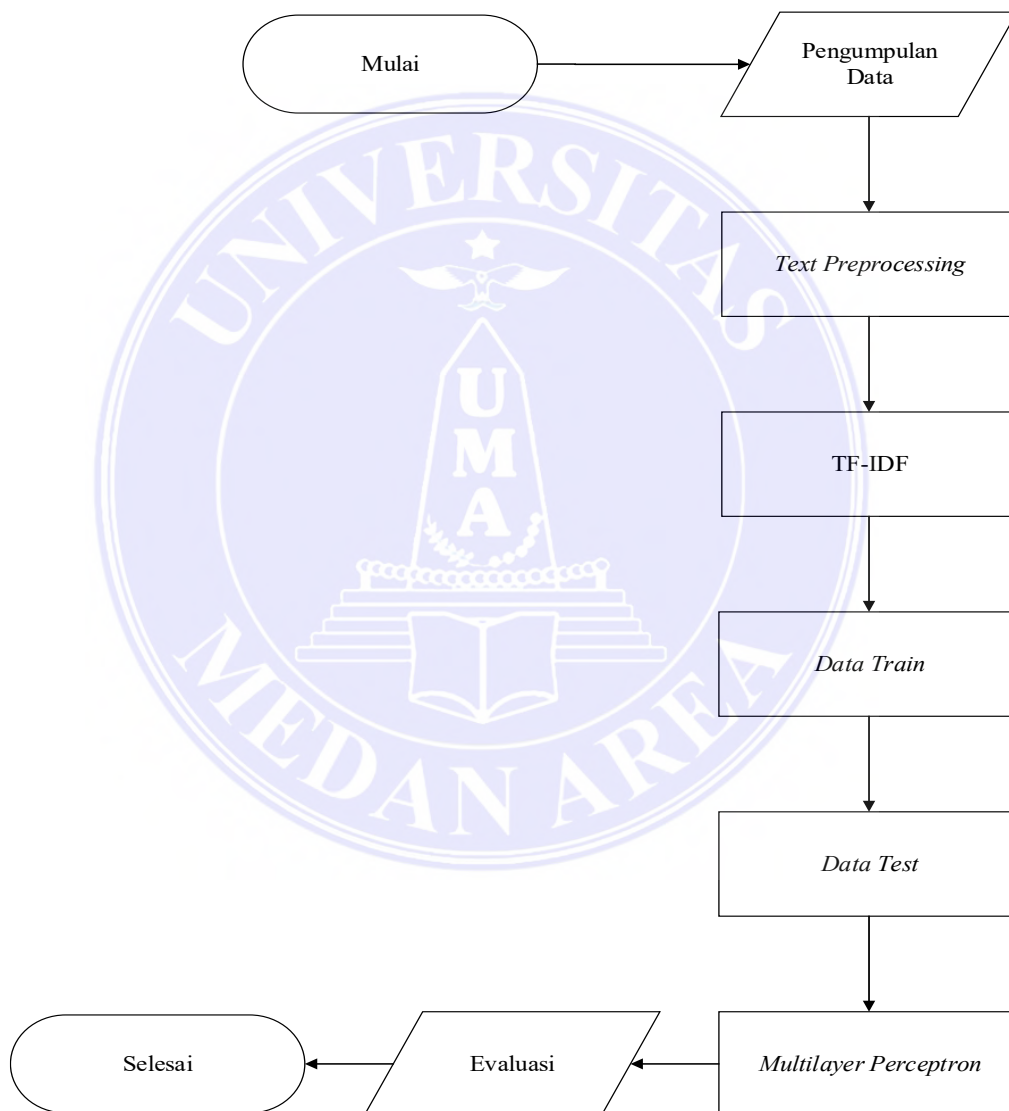
No	Judul	Penulis	Metode	Parameter	Hasil Penelitian
1	Analisis Sentimen Twitter Menggunakan Ekstraksi <i>TF-IDF</i> (<i>Term Frequency and Inverse Document Frequency</i>) dan Metode <i>Multilayer Perceptron</i> Pada PILKADA Kota Medan 2020. (2021)	Muntaqim Asbuch	<i>Multilayer Perceptron</i>	<ol style="list-style-type: none"> 1.) <i>Input</i> dan <i>output layer</i> berjumlah 1. 2.) Memiliki 4 <i>Hidden Layer</i>(HL) diantaranya HL ke-1 berjumlah 30 <i>neuron</i>, HL ke-2 berjumlah 15 <i>neuron</i>, HL ke-3 berjumlah 19 <i>neuron</i>, dan HL ke-4 berjumlah 39. 3.) Menggunakan fungsi aktivasi 'relu'. 4.) Menggunakan fungsi optimasi 'adam'. 5.) Menggunakan <i>random state</i> 45. 6.) <i>Epoch</i> sebanyak 30. 	Penelitian ini menggunakan data <i>tweet</i> berjumlah 630 data. Label yang diberikan hanya berupa positif dan negatif dengan jumlah masing-masing 50%. Pengujian klasifikasi ini dilakukan 30 kali pengujian berulang menggunakan ekstraksi fitur <i>tf-idf</i> dan algoritma <i>multilayer perceptron</i> didapatkan akurasi 92.733%, presisi 88%, serta <i>recall</i> 95%.
2	Analisis Sentimen Komentar Netizen Terhadap Calon Presiden Indonesia 2019 dari <i>Twitter</i> Menggunakan Algoritma <i>Term Frequency-Inverse Document</i>	Faqi Syadid	<i>TF-IDF</i> dan <i>Multilayer Perceptron</i>	<ol style="list-style-type: none"> 1.) 1 <i>Input layer</i> sebanyak 300 <i>neuron</i>. 2.) Memiliki 2 <i>hidden layer</i> dengan masing – masing berjumlah 117 <i>neuron</i>. 3.) <i>Output layer</i> sebanyak 1 <i>neuron</i>. 4.) Menggunakan fungsi aktivasi 'sigmoid biner'. 	Penelitian ini menggunakan 3 skenario dalam percobaan klasifikasi, yaitu skenario ke-1 menggunakan jumlah data latih sebanyak 700 data, skenario ke-2 sebanyak 800 data, serta skenario ke-3 sebanyak 900 data. Mendapatkan akurasi tertinggi sebesar 88% pada skenario ke-3.

	<i>Frequency (TF-IDF) dan Metode Multi Layer Perceptron (MLP) Neural Network.</i> (2019)			5.) Menggunakan <i>learning rate</i> 0.05. 6.) <i>Epoch</i> sebanyak 3.	
3	Analisis Sentimen Pada <i>Tweet</i> Dengan Tagar #Mahkamahakonstitusi Menggunakan Metode <i>Multilayer Perceptron</i> (2021).	Asif Ummatul Khaira	<i>Multilayer Perceptron</i>	1.) 1 <i>input layer</i> . 2.) 1 <i>hidden layer</i> . 3.) 1 <i>output layer</i> . 4.) Menggunakan fungsi aktivasi ‘sigmoid biner’. 5.) Menggunakan <i>learning rate</i> 0.1. 6.) <i>Epoch</i> sebanyak 3.	Penelitian ini menggunakan data sejumlah 300 dengan membagi kategori positif dan negatif. Hasilnya mendapatkan akurasi sebesar 89%.
4	Natural Language Processing untuk Analisis Sentimen Presiden Jokowi Menggunakan <i>Multilayer Perceptron</i> . (2020).	Nico Munasatya dan Sendi Novianto	<i>Multilayer Perceptron</i>	1.) <i>Hidden layer</i> yaitu 32, 64, dan 128 <i>neuron</i> . 2.) <i>Learning rate</i> yaitu 0.001, 0.005, 0.1, dan 0.0005. 3.) <i>Data suffling</i> yaitu True dan False. 4.) <i>Alpha</i> yaitu 0.0001, 0.001, 0.01, dan 0.1. 5.) <i>Epoch</i> sebanyak 10.	Penelitian ini menggunakan 3686 data dengan berisi label positif dan negatif. Pengujian menggunakan 10 tahap skenario dan mendapatkan akurasi tertinggi pada skenario ke-7 sebesar 93.26%. dan <i>recall</i> terbaik sebesar 94.21%.
5	Analisis Sentimen Pada Media Sosial <i>Twitter</i> terhadap Penanganan Bencana Banjir di Jawa Barat dengan	Aurell Layalia Safara Az-Zahra Gunawan, Jondri, dan	Jaringan Saraf Tiruan model <i>Multilayer Perceptron</i> (MLP)	1.) <i>Input layer</i> sebanyak 3000 <i>neuron</i> . 2.) <i>Hidden layer</i> dengan 64 dan 128 <i>neuron</i> dengan aktivasi ‘relu’.	Penelitian ini menggunakan 509 <i>tweet</i> untuk label positif dan 91 untuk label negatif. Model yang dibangun memiliki 10 skenario dan menghasilkan akurasi tertinggi sebesar 73.83%, presisi 77.79%, dan <i>recall</i> 90.00% pada skenario ke-8.

	Metode Jaringan Saraf Tiruan. (2021).	Kemas Muslim Lhaksamana		<p>3.) <i>Output layer</i> memiliki 1 <i>neuron</i> dengan aktivasi 'sigmoid'.</p> <p>4.) Menggunakan optimasi 'adam'.</p> <p>5.) Menggunakan <i>learning rate</i> 0.001, 0.005, dan 0.1.</p> <p>6.) <i>Alpha</i> 0.2</p> <p>7.) <i>Epoch</i> sebanyak 5.</p> <p>8.) <i>Data suffling</i> yaitu <i>True</i> dan <i>False</i>.</p>	
6	Analisis Sentimen Mahasiswa Terhadap Fasilitas Universitas Telkom Menggunakan Metode Jaringan Saraf Tiruan dan TF-IDF. (2019).	Muhammad Farhan Muzakki, Jondri, Rian Febrian Umbara.	Jaringan Saraf Tiruan dengan model <i>Multilayer Perceptron</i>	<p>1.) <i>Hidden layer</i> yaitu 64 dan 128 <i>neuron</i>.</p> <p>2.) 1 <i>output layer</i>.</p> <p>3.) Menggunakan <i>learning rate</i> 0.0005, 0.001, 0.005, 0.01, dan 0.1.</p> <p>4.) <i>Data suffling</i> yaitu <i>True</i> dan <i>False</i>.</p> <p>5.) <i>Alpha</i> 0.2.</p> <p>6.) <i>Epoch</i> sebanyak 10.</p>	Penelitian ini menggunakan 10000 data hasil kuisioner yang berisi opini positif dan negatif. Pengujian model terdapat 10 skenario, di mana skenario ke-6 memiliki akurasi tertinggi sebesar 91.23%, <i>recall</i> 93.95%, dan presisi 94.27%.

BAB III METODE PENELITIAN

Gambar 3.1 terdapat prosedur yang dikerjakan dalam penelitian ini, ialah pengumpulan data, *text preprocessing*, pembobotan TF-IDF, algoritma *multilayer perceptron*, serta evaluasi.



Gambar 3. 1 Metode Penelitian

3.1 Objek Penelitian

Objek yang diteliti adalah opini masyarakat mengenai virus *corona* melalui jejaring sosial *twitter*. Data yang digunakan yaitu berupa *tweet* (*hashtags*, *retweets*, *favorites* dan *text*) dengan kata kunci *corona* sejak tanggal 12 Desember 2020 sampai 23 November 2021 sebanyak 228208 *record data* seluruh dunia dengan didominasi oleh negara India, Kanada, dan Amerika Serikat.

3.2 Teknik Pengolahan dan Analisis Data

Teknik pengolahan dan analisis data menggunakan teknik *text mining* atau juga dikenal sebagai data *mining* teks. *Text mining* merupakan suatu kegiatan menggali informasi, kegiatan tersebut dilakukan oleh seorang pengguna yang berhubungan pada sekumpulan arsip menetapkan peralatan penguraian (Imam & Fajtriab, 2015). Penambang teks dipakai untuk memperoleh data yang bermanfaat dari sekumpulan teks yang ada di *twitter*. Adapun langkah-langkah yang dilaksanakan dalam penelitian ini ialah:

3.2.1 Studi Literatur dan Pemahaman

Tahapan ini peneliti mencari berbagai sumber berupa artikel ilmiah, jurnal, serta *textbook* yang berkaitan pada penelitian. Peneliti menggunakan 6 penelitian terdahulu untuk menjadikan acuan agar mengetahui kelebihan dan kekurangan dari masing-masing penelitian terdahulu.

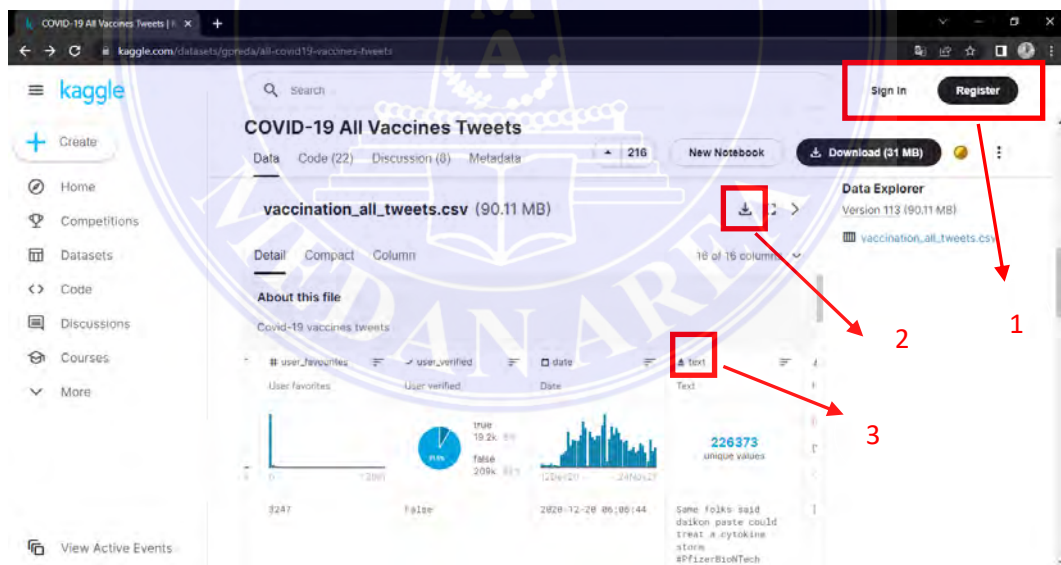
3.2.2 Jenis Data

Penelitian ini, jenis data yang dipakai ialah data sekunder, dikarenakan data tersebut dikumpulkan atau sudah tersedia dari pihak lain. Ketersediaan data

sekunder sangat bertumpu pada ketentuan peneliti akan mempublikasi data tersebut ke publik atau tidak.

3.2.3 Pengumpulan Data

Pengumpulan data diambil melalui situs <https://www.kaggle.com/gpreda/all-covid19-vaccines-tweets> berupa *tweet* (*hashtags, retweets, favorites*) sejak tanggal 12 Desember 2020 sampai 23 November 2021 sebanyak 228208 *record data* seluruh dunia dengan didominasi oleh negara India, Kanada, dan Amerika Serikat. Semua *tweet* yang ditampilkan kemudian di simpan untuk di analisis. Berikut adalah pengambilan data pada situs *Kaggle*.



Gambar 3. 2 Pengambilan Data dari *Kaggle*

Sumber: <https://www.kaggle.com>

Gambar 3.2 terlebih dahulu mendaftar/masuk pada situs kaggle berikut sebelum mengambil data, pastikan akun email aktif untuk mendapatkan 6 *digit code*

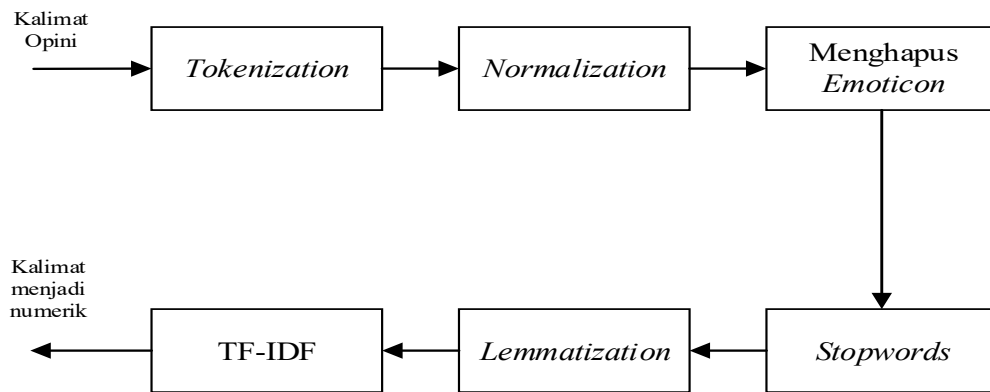
verifikasi akun(dapat dilihat pada nomor 1). Kemudian data bisa di *download* (dapat dilihat pada nomor 2) dan kolom yang akan digunakan untuk pada penelitian ini yaitu kolom *text* (pada nomor 3) yang berisi opini pengguna *tweet* dari seluruh dunia. Adapun diantaranya peneliti mengambil beberapa data latih sebagai berikut:

Tabel 3. 1 Data Latih

No	Teks
1	Kasus Covid-19 terkendali bukan berarti kita abai. Vaksinasi dan prokes juga tetap harus dilakukan, terlebih dahulu ada varian baru. Jangan abai, tetap waspada demi keselamatan diri sendiri dan orang yang kita sayangi.
2	Vaksin untuk virus Covid-19 ini sangat membantu sekali.
3	Vaksin Covid-19 mengandung microchip dan magnet.

3.2.4 Text Preprocessing

Pada pra-pemrosesan teks mencakup proses untuk menyiapkan data yang akan dipakai pada *text mining*. Sehingga *dataset* telah bersih dan siap untuk digunakan. Tahap *text preprocessing* pada penelitian ini menggunakan beberapa tahapan, yaitu: *tokenization*, *normalization*, menghapus *emoticon*, *stopwords*, *lemmatization*, dan TF-IDF. Gambaran umum tahapan pra-pemrosesan teks dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Tahapan *Text Preprocessing*

Sumber: (Ina & Ifani, 2021)

1. *Tokenization*

Merupakan proses membagi teks (bisa berbentuk kalimat, alinea ataupun arsip) sebagai token/bagian khusus. Prosedur membagi kalimat menjadi kata-kata umumnya ditetapkan oleh pembatas. Pembatas atas *delimiter* dari tiap-tiap kata adalah spasi. Tokenisasi dikerjakan untuk mempermudah prosedur klasifikasi dengan membagi kalimat menjadi kata per kata. Menurut data latih di Tabel 3.1 hasil prosedur tokenisasi bisa dilihat pada Tabel 3.2 berikut.

Tabel 3. 2 Tahap *Tokenization*

No	Teks	Hasil <i>Tokenizing</i>
1	Kasus Covid-19 terkendali bukan berarti kita abai. Vaksinasi dan prokes juga tetap harus dilakukan, terlebih dahulu ada varian baru. Jangan abai, tetap waspada demi keselamatan diri sendiri dan orang yang kita sayangi.	“Kasus”, “Covid-19”, “terkendali”, “bukan”, “berarti”, “kita”, “abai”, “.”, “Vaksinasi”, “dan”, “prokes”, “juga”, “tetap”, “harus”, “dilakukan”, “,”, “terlebih”, “dahulu”, “ada”, “varian”, “baru”, “.”, “Jangan”, “abai”, “,”, “tetap”, “waspada”, “demi”, “keselamatan”, “diri”,

		“sendiri”, “dan”, “orang”, “yang”, “kita”, “sayangi”, “.”
2	Vaksin untuk virus Covid-19 ini sangat membantu sekali.	“Vaksin”, “untuk”, “virus”, “Covid-19”, “ini”, “sangat”, “membantu”, “sekali”, “.”
3	Vaksin Covid-19 mengandung microchip dan magnet.	“Vaksin”, “Covid-19”, “mengandung”, “microchip”, “dan”, “magnet”, “.”

2. Normalization

Bertujuan untuk menormalkan kalimat sehingga kalimat yang mempunyai huruf kapital akan diubah menjadi huruf kecil. Kemudian menghilangkan simbol, *link*, dan lain sebagainya.

Tabel 3. 3 Tahap *Normalization*

No	Hasil <i>Tokenizing</i>	Hasil <i>Normalization</i>
1	“Kasus”, “Covid-19”, “terkendali”, “bukan”, “berarti”, “kita”, “abai”, “.”, “Vaksinasi”, “dan”, “prokes”, “juga”, “tetap”, “harus”, “dilakukan”, “.”, “terlebih”, “dahulu”, “ada”, “varian”, “baru”, “.”, “Jangan”, “abai”, “.”, “tetap”, “waspada”, “demi”, “keselamatan”, “diri”, “sendiri”, “dan”, “orang”, “yang”, “kita”, “sayangi”, “.”	“kasus”, “covid”, “terkendali”, “bukan”, “berarti”, “kita”, “tidak”, “peduli”, “vaksinasi”, “dan”, “prokes”, “juga”, “tetap”, “harus”, “dilakukan”, “terlebih”, “dahulu”, “ada”, “varian”, “baru”, “jangan”, “tidak”, “peduli”, “tetap”, “waspada”, “demi”, “keselamatan”, “diri”, “sendiri”, “dan”, “orang”, “yang”, “kita”, “sayangi”
2	“Vaksin”, “untuk”, “virus”, “Covid-19”, “ini”, “sangat”, “membantu”, “sekali”, “.”	“vaksin”, “untuk”, “virus”, “covid”, “ini”, “sangat”, “membantu”, “sekali”
3	“Vaksin”, “Covid-19”, “mengandung”, “microchip”, “dan”, “magnet”, “.”	“vaksin”, “covid”, “mengandung”, “microchip”, “dan”, “magnet”

3. Menghapus *emoticon*

Saat menulis status (*tweeting*) terkadang seseorang menggunakan emoji yang salah atau tidak tepat, baik disengaja maupun tidak. Contoh, mereka cuman bisa menghujat sebab tidak menemukan fakta buruk :) , opini adalah fitnah tapi

emojinya senyum :), jadi emoji akan mengganggu proses analisis sentimen *tweet*. Sementara *emoticon, feeling and sentiment* bisa dilihat pada Gambar 3.4.

Emoticon	Feeling	Sentimen
:) :-)	Happy	Positif
:(:-)	Sad	Negatif
:D :-D	Very Happy!	Positif
D: D=	Very Sad	Negatif
* * * * *	Fascinated	Positif

Gambar 3. 4 *Emoticon*

Sumber: (Ina and Ifani 2021)

4. *Stopwords*

Proses *Stopwords* pada penelitian ini adalah penghilangan kata-kata yang paling umum seperti kata “atau”, “dengan”, “dan”, “di”, dan lain-lain.

Tabel 3. 4 Tahap *Stopwords*

No	Hasil Normalization	Hasil Stopwords
1	“kasus”, “covid”, “terkendali”, “bukan”, “berarti”, “kita”, “tidak”, “peduli”, “vaksinasi”, “dan”, “prokes”, “juga”, “tetap”, “harus”, “dilakukan”, “terlebih”, “dahulu”, “ada”, “varian”, “baru”, “jangan”, “tidak”, “peduli”, “tetap”, “waspada”, “demi”, “keselamatan”, “diri”, “sendiri”, “dan”, “orang”, “yang”, “kita”, “sayangi”	“kasus”, “covid”, “terkendali”, “berarti”, “kita”, “peduli”, “vaksinasi”, “prokes”, “tetap”, “harus”, “dilakukan”, “terlebih”, “dahulu”, “varian”, “baru”, “peduli”, “tetap”, “waspada”, “keselamatan”, “diri”, “sendiri”, “orang”, “kita”, “sayangi”
2	“vaksin”, “untuk”, “virus”, “covid”, “ini”, “sangat”, “membantu”, “sekali”	“vaksin”, “virus”, “covid”, “sangat”, “membantu”, “sekali”
3	“vaksin”, “covid”, “mengandung”, “microchip”, “dan”, “magnet”	“vaksin”, “covid”, “mengandung”, “microchip”, “magnet”

5. *Lemmatization*

Lemmatization merupakan proses menemukan bentuk dasar suatu kata.

Bertujuan untuk menormalkan teks atau kata berdasarkan bentuk dasar dari bentuk

lemma-nya. Berikut dibawah ini daftar imbuhan di tiap tiap tahapan yang dapat dilihat pada Tabel 3.6.

Tabel 3. 5 Tahap *Lemmatization*

No	Hasil <i>Stopwords</i>	Hasil <i>Lemmatization</i>
1	“kasus”, “covid”, “terkendali”, “berarti”, “kita”, “peduli”, “vaksinasi”, “prokes”, “tetap”, “harus”, “dilakukan”, “terlebih”, “dahulu”, “varian”, “baru”, “jangan”, “peduli”, “tetap”, “waspada”, “keselamatan”, “diri”, “sendiri”, “orang”, “kita”, “sayangi”	“kasus”, “covid”, “kendali”, “arti”, “kita”, “peduli”, “vaksinasi”, “prokes”, “tetap”, “harus”, “laku”, “lebih”, “dahulu”, “varian”, “baru”, “peduli”, “tetap”, “waspada”, “selamat”, “diri”, “sendiri”, “orang”, “kita”, “sayang”
2	“vaksin”, “virus”, “covid”, “sangat”, “membantu”, “sekali”	“vaksin”, “virus”, “covid”, “sangat”, “bantu”, “sekali”
3	“vaksin”, “covid”, “mengandung”, “microchip”, “magnet”	“vaksin”, “covid”, “kandung”, “microchip”, “magnet”

6. TF-IDF

Secara umum, TF-IDF merupakan proses perubahan teks menjadi angka, sebab arsitektur *machine learning* serta *deep learning* tidak dapat menganalisis data *input* berbentuk *text*, maka dari itu diperlukan angka sebagai *input*. Proses TF-IDF dari penelitian ini sebagai berikut:

Term Frequency (TF) ialah jumlah frekuensi kemunculan *term* (t) di dokumen (D), berdasarkan hasil tahap *lemmatization* pada Tabel 3.4 maka dapat dihitung TF sebagai berikut.

- 1.) “kasus”, “covid”, “kendali”, “arti”, “kita”, “peduli”, “vaksinasi”, “prokes”, “tetap”, “harus”, “laku”, “lebih”, “dahulu”, “varian”, “baru”, “peduli”, “tetap”, “waspada”, “selamat”, “diri”, “sendiri”, “orang”, “kita”, “sayang”
- 2.) “vaksin”, “virus”, “covid”, “sangat”, “bantu”, “sekali”
- 3.) “vaksin”, “covid”, “kandung”, “microchip”, “magnet”

Tabel 3. 6 *Term Frequency*

Term (t)	D1	D2	D3
kasus	1	0	0
covid	1	1	1
kendali	1	0	0
arti	1	0	0
kita	2	0	0
peduli	2	0	0
vaksinasi	1	0	0
proses	1	0	0
tetap	2	0	0
harus	1	0	0
laku	1	0	0
lebih	1	0	0
dahulu	1	0	0
varian	1	0	0
baru	1	0	0
waspada	1	0	0
selamat	1	0	0
diri	1	0	0
sendiri	1	0	0
orang	1	0	0
sayang	1	0	0
vaksin	0	1	1
virus	0	1	0
sangat	0	1	0
bantu	0	1	0
sekali	0	1	0
kandung	0	0	1
microchip	0	0	1
magnet	0	0	1

Selanjutnya menghitung *Document Frequency* (DF) yaitu banyaknya dokumen dimana suatu *term* muncul, proses menghitung DF dapat dilihat pada Tabel 3.7.

Tabel 3. 7 *Document Frequency*

Term (t)	df
kasus	1

covid	3
kendali	1
arti	1
kita	1
peduli	1
vaksinasi	1
proses	1
tetap	1
harus	1
laku	1
lebih	1
dahulu	1
varian	1
baru	1
waspada	1
selamat	1
diri	1
sendiri	1
orang	1
sayang	1
vaksin	2
virus	1
sangat	1
bantu	1
sekali	1
kandung	1
microchip	1
magnet	1

Menghitung *Inverse Document Frequency* (IDF) bermanfaat untuk memahami seberapa pentingnya kata dalam sebuah dokumen. IDF dirumuskan dengan menggunakan persamaan (2.1).

Tabel 3. 8 *Inverse Document Frequency*

Term (t)	df	idf
kasus	1	$\log(3/1) = 0.477$
covid	3	$\log(3/3) = 0$
kendali	1	$\log(3/1) = 0.477$
arti	1	$\log(3/1) = 0.477$
kita	1	$\log(3/1) = 0.477$
peduli	1	$\log(3/1) = 0.477$

vaksinasi	1	$\log(3/1) = 0.477$
proses	1	$\log(3/1) = 0.477$
tetap	1	$\log(3/1) = 0.477$
harus	1	$\log(3/1) = 0.477$
laku	1	$\log(3/1) = 0.477$
lebih	1	$\log(3/1) = 0.477$
dahulu	1	$\log(3/1) = 0.477$
varian	1	$\log(3/1) = 0.477$
baru	1	$\log(3/1) = 0.477$
waspada	1	$\log(3/1) = 0.477$
selamat	1	$\log(3/1) = 0.477$
diri	1	$\log(3/1) = 0.477$
sendiri	1	$\log(3/1) = 0.477$
orang	1	$\log(3/1) = 0.477$
sayang	1	$\log(3/1) = 0.477$
vaksin	2	$\log(3/2) = 0.176$
virus	1	$\log(3/1) = 0.477$
sangat	1	$\log(3/1) = 0.477$
bantu	1	$\log(3/1) = 0.477$
sekali	1	$\log(3/1) = 0.477$
kandung	1	$\log(3/1) = 0.477$
microchip	1	$\log(3/1) = 0.477$
magnet	1	$\log(3/1) = 0.477$

Menghitung TF-IDF dengan menggabungkan perhitungan TF dengan IDF.

Maka hasil dari TF-IDF adalah sebagai berikut:

Tabel 3. 9 Hasil Perhitungan TF-IDF

Term (t)	D1	D2	D3	idf	tf.idf		
					D1	D2	D3
kasus	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
covid	1	1	1	$\log(3/3) = 0$	0	0	0
kendali	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
arti	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
kita	2	0	0	$\log(3/1) = 0.477$	0.954	0	0
peduli	2	0	0	$\log(3/1) = 0.477$	0.954	0	0
vaksinasi	1	0	0	$\log(3/1) = 0.477$	0.477	0	0

proses	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
tetap	2	0	0	$\log(3/1) = 0.477$	0.954	0	0
harus	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
laku	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
lebih	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
dahulu	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
varian	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
baru	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
waspada	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
selamat	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
diri	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
sendiri	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
orang	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
sayang	1	0	0	$\log(3/1) = 0.477$	0.477	0	0
vaksin	0	1	1	$\log(3/2) = 0.176$	0	0.176	0.176
virus	0	1	0	$\log(3/1) = 0.477$	0	0.477	0
sangat	0	1	0	$\log(3/1) = 0.477$	0	0.477	0
bantu	0	1	0	$\log(3/1) = 0.477$	0	0.477	0
sekali	0	1	0	$\log(3/1) = 0.477$	0	0.477	0
kandung	0	0	1	$\log(3/1) = 0.477$	0	0	0.477
microchip	0	0	1	$\log(3/1) = 0.477$	0	0	0.477
magnet	0	0	1	$\log(3/1) = 0.477$	0	0	0.477

3.3 Perhitungan Manual *Multilayer Perceptron*

Berikut adalah perhitungan manual dari algoritma *Multilayer Perceptron*.

Diketahui nilai TF-IDF sebagai berikut:

Tabel 3. 10 Hasil TF-IDF Berdasarkan Term

Term (t)	Nilai TF-IDF
kasus	0.477
covid	0
kendali	0.477
arti	0.477
kita	0.954
peduli	0.954
vaksinasi	0.477
proses	0.477
tetap	0.954
harus	0.477
laku	0.477
lebih	0.477
dahulu	0.477
varian	0.477
baru	0.477
waspada	0.477
selamat	0.477
diri	0.477
sendiri	0.477
orang	0.477
sayang	0.477
vaksin	0.176
virus	0.477
sangat	0.477
bantu	0.477
sekali	0.477
kandung	0.477
microchip	0.477
magnet	0.477

- 1.) Inisialisasi semua bobot dengan bilangan acak kecil. Dilakukan dengan cara memberikan nilai random pada masing-masing sisi yang menghubungkan antar neuron.

- 2.) Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 – 8.
- 3.) Untuk setiap pasang data pelatihan. Lakukan langkah 3 – 8.
- 4.) Tiap unit masukan menerima sinyal dan meneruskan ke unit tersembunyi.
- 5.) Menghitung semua keluaran di unit tersembunyi z_j ($j= 1, 2, \dots, p$) dengan persamaan (2.6).

$$z_{net1} = (0.477) (0.001) + (0) (0.03) + (0.477) (0.2) + (0.477) (-0.002) + (0.954) (0.04) + (0.954) (0.007) + (0.477) (0.06) + (0.477) (0.03) + (0.954) (-0.03) + (0.477) (0.5) + (0.477) (0.01) + (0.477) (-0.1) + (0.477) (-0.001) + (0.477) (0.05) + (0.477) (0.007) + (0.477) (-0.009) + (0.477) (-0.005) + (0.477) (-0.01) + (0.477) (0.004) + (0.477) (0.006) + (0.477) (-0.09) + (0.176) (0.2) + (0.477) (0.014) + (0.477) (0.004) + (0.477) (0.4) + (0.477) (0.325) + (0.477) (-0.022) + (0.477) (0.012) + (0.477) (-0.123) + (0.02) = 0.672$$

$$z_{net2} = (0.477) (0.022) + (0) (0.63) + (0.477) (0.27) + (0.477) (0.2) + (0.954) (0.064) + (0.954) (-0.007) + (0.477) (-0.6) + (0.477) (-0.03) + (0.954) (-0.093) + (0.477) (-0.5) + (0.477) (-0.01) + (0.477) (-0.91) + (0.477) (-0.1) + (0.477) (0.5) + (0.477) (-0.27) + (0.477) (-0.09) + (0.477) (-0.95) + (0.477) (-0.001) + (0.477) (0.4) + (0.477) (-0.60) + (0.477) (0.34) + (0.176) (-0.42) + (0.477) (0.01) + (0.477) (0.04) + (0.477) (-0.4) + (0.477) (0.350) + (0.477) (-0.02) + (0.477) (-0.12) + (0.477) (-0.173) + (-0.22) = -1.588$$

$$z_{net3} = (0.477) (-0.1) + (0) (0.02) + (0.477) (0.007) + (0.477) (-0.276) + (0.954) (0.03) + (0.954) (-0.7) + (0.477) (-0.676) + (0.477) (0.5) + (0.954) (0.55) + (0.477) (0.34) + (0.477) (0.006) + (0.477) (0.009) + (0.477) (-0.123) + (0.477) (0.50) + (0.477) (0.206) + (0.477) (-0.09) + (0.477) (-0.95) + (0.477) (-0.101) + (0.477) (0.4) + (0.477) (-0.60) + (0.477) (0.34) + (0.176) (-0.72) + (0.477) (-0.301) + (0.477) (0.4) + (0.477) (0.094) + (0.477) (0.530) + (0.477) (0.2) + (0.477) (0.002) + (0.477) (0.130) + (0.3) = 0.272$$

Menghitung sigmoid biner:

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

$$f(z_{net_1}) = \frac{1}{1 + e^{-0.672}} = 0.662$$

$$f(z_{net_2}) = \frac{1}{1 + e^{1.588}} = 0.169$$

$$f(z_{net_3}) = \frac{1}{1 + e^{-0.272}} = 0.567$$

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

$$y_{net_j} = w_{k0} + \sum_{j=1}^p z_j w_{kj} = 0.051 + (0.169)(0.021) + (0.567)(-0.021)$$

$$= 0.042$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-0.042}} = 0.510$$

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

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

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

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

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

$$= (1 - 0.510) 0.510 (1 - 0.510) = 0.122$$

$$\Delta w_{10} = \alpha \delta_k z_0 = 0.2 (0.122) (1) = 0.024$$

$$\Delta w_{11} = \alpha \delta_k z_1 = 0.2 (0.122) (0.662) = 0.016$$

$$\Delta w_{12} = \alpha \delta_k z_2 = 0.2 (0.122) (0.169) = 0.005$$

$$\Delta w_{13} = \alpha \delta_k z_3 = 0.2 (0.122) (0.567) = 0.014$$

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

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

$$\delta_1 = (0.122)(0.02) = 0.024$$

$$\delta_2 = (0.122)(-0.37) = -0.045$$

$$\delta_3 = (0.122)(0.17) = 0.020$$

Faktor δ unit tersembunyi:

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

$$\delta_1 = \delta_{net_{z_1}} (1 - z_1) = (0.024)(0.662)(1 - 0.662) = 0.005$$

$$\delta_2 = \delta_{net_{z_2}} (1 - z_2) = (-0.045)(0.169)(1 - 0.169) = -0.006$$

$$\delta_3 = \delta_{net_{z_3}} (1 - z_3) = (0.020)(0.567)(1 - 0.567) = 0.004$$

Hitung suku perubahan bobot:

$$\Delta v_{ji} = \delta \delta_j x_i$$

$$\Delta v_{10} = \alpha \delta_1 = (0.3)(0.005)(1) = 0.001$$

$$\Delta v_{20} = \alpha \delta_1 = (0.3)(-0.006)(1) = -0.001$$

$$\Delta v_{30} = \alpha \delta_1 = (0.3)(0.004)(1) = 0.001$$

$$\Delta v_{11} = \alpha \delta_1 = (0.3)(0.005)(0.477) = 0.001$$

$$\Delta v_{21} = \alpha \delta_2 = (0.3)(-0.006)(0.477) = -0.001$$

$$\Delta v_{31} = \alpha \delta_3 = (0.3)(0.004)(0.477) = 0.001$$

$$\Delta v_{12} = \alpha \delta_1 = (0.3)(0.005)(0.954) = 0.001$$

$$\Delta v_{22} = \alpha \delta_2 = (0.3)(-0.006)(0.954) = -0.002$$

$$\Delta v_{32} = \alpha \delta_3 = (0.3)(0.004)(0.954) = 0.001$$

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

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

$$w_{10}(\text{baru}) = w_{10}(\text{lama}) + \Delta w_{10} = -0.1 + 0.024 = -0.076$$

$$w_{11}(\text{baru}) = w_{11}(\text{lama}) + \Delta w_{11} = 0.5 + 0.016 = 0.516$$

$$w_{12}(\text{baru}) = w_{12}(\text{lama}) + \Delta w_{12} = -0.4 + 0.005 = -0.395$$

$$w_{13}(\text{baru}) = w_{13}(\text{lama}) + \Delta w_{13} = -0.2 + 0.014 = -0.186$$

Perubahan bobot garis yang menuju ke unit tersembunyi yaitu:

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

$$v_{10}(\text{baru}) = v_{10}(\text{lama}) + \Delta v_{10} = -0.01 + 0.001 = -0.009$$

$$v_{20}(\text{baru}) = v_{20}(\text{lama}) + \Delta v_{20} = 0.01 + (-0.001) = 0.009$$

$$v_{30}(\text{baru}) = v_{30}(\text{lama}) + \Delta v_{30} = -0.01 + 0.001 = -0.009$$

$$v_{11}(\text{baru}) = v_{11}(\text{lama}) + \Delta v_{11} = 0.02 + 0.001 = 0.021$$

$$v_{21}(\text{baru}) = v_{21}(\text{lama}) + \Delta v_{21} = 0.03 + (-0.001) = 0.029$$

$$v_{31}(\text{baru}) = v_{31}(\text{lama}) + \Delta v_{31} = -0.01 + 0.001 = -0.009$$

$$v_{12}(\text{baru}) = v_{12}(\text{lama}) + \Delta v_{12} = 0.03 + 0.001 = 0.031$$

$$v_{22}(\text{baru}) = v_{22}(\text{lama}) + \Delta v_{22} = 0.01 + 0.002 = 0.012$$

$$v_{32}(\text{baru}) = v_{32}(\text{lama}) + \Delta v_{32} = 0.03 + 0.001 = 0.031$$

Berdasarkan perhitungan manual dari algoritma *multilayer perceptron* untuk menentukan label positif (jika lebih besar dari 0), negatif (jika lebih kecil dari 0), dan netral (jika sama dengan 0), maka dapat dilihat dari tahapan ke-9 pada perubahan bobot garis yang menuju ke unit tersembunyi dimana ditentukan dari $v_{ji}(\text{baru})$ yaitu :

1. $V_{10}(\text{baru}) = -0.009$, $V_{11}(\text{baru}) = 0.021$, dan $V_{12}(\text{baru}) = 0.031$ akan ditentukan berdasarkan hasil yang paling dominan, maka data latih pada nomor 1 memiliki label positif.
2. $V_{20}(\text{baru}) = 0.009$, $V_{21}(\text{baru}) = 0.029$, dan $V_{22}(\text{baru}) = 0.012$ akan ditentukan berdasarkan hasil yang paling dominan, maka data latih pada nomor 2 memiliki label positif.
3. $V_{30}(\text{baru}) = -0.009$, $V_{31}(\text{baru}) = -0.009$, dan $V_{32}(\text{baru}) = 0.031$ akan ditentukan berdasarkan hasil yang paling dominan, maka data latih pada nomor 3 memiliki label negatif.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil yang telah dilakukan pada penelitian ini dapat ditarik kesimpulan yaitu penerapan algoritma *Multilayer Perceptron* dan tingkat akurasi pengklasifikasian sentimen pengguna *tweet* terhadap vaksin COVID-19 mendapatkan akurasi terbaik pada model MLP3 sebesar 81.2%, presisi 83.8%, dan *recall* 71.2%. Visualisasi kata terbanyak pada sentimen positif terdapat 3 topik yaitu mengenai ketersediaan, berbayar, dan dosis. Selanjutnya sentimen negatif terdapat 2 pokok masalah seperti efek samping vaksin dan kematian. Terakhir sentimen netral terdapat 4 topik seperti dosis, ketersediaan, umur, dan tanggal kadaluwarsa.

5.2 Saran

Adapun saran yang dapat dilakukan pada penelitian selanjutnya adalah sebagai berikut:

1. Membuat kamus khusus pada 1 file csv yang di dalamnya terdapat kata non-baku menjadi baku dalam *text preprocessing* untuk menormalkan kalimat atau kata yang disingkat pada sebuah *dataset*.
2. Menambah jumlah data yang lebih banyak dari data yang digunakan pada penelitian ini dan menggunakan fungsi aktivasi lain seperti *identity*, *tanh*, dan *relu*.
3. Membangun sebuah sistem dengan berbasis desktop atau website.
4. Menguji dengan metode lainnya seperti *K-Nearest Neighbors*, *C.45*, *Random Forest*, dan *Support Vector Machine*.

DAFTAR PUSTAKA

- Asbuch, M. (2021). *Analisis Sentimen Twitter Menggunakan Ekstraksi TF-IDF (Term Frequency and Inverse Document Frequency) dan Metode Multilayer Perceptron Pada PILKADA Kota Medan 2020*. Universitas Sumatera Utara.
- Boon-Itt, S., & Skunkan, Y. (2020). JMIR Public Health and Surveillance. *Public Perception of the COVID-19 Pandemic on Twitter: Sentiment Analysis and Topic Modeling Study*, 6.
- Brahimi, B. ... Tari, A. (2021). Improving sentiment analysis in Arabic: A combined approach. *Journal of King Saud University - Computer and Information Sciences*, 33(10), 1242–1250. <https://doi.org/10.1016/j.jksuci.2019.07.011>
- Chandrasekaran, R. ... Moustakas, E. (2020). Journal of Medical Internet Research. *Topics, Trends, and Sentiments of Tweets About the COVID-19 Pandemic: Temporal Inveillance Study*, 22.
- Ciaburro, G., & Venkateswaran, B. (2017). *Neural Network with R*.
- Fanissa, S. ... Adinugroho, S. (2018). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. *Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query Expansion Ranking*.
- Fauzi, M. A. (2018). Random Forest Approach for Sentiment Analysis in Indonesian Language. *Indonesian Journal of Electrical Engineering and Computer Science*, 12(1), 46. <https://doi.org/10.11591/ijeecs.v12.i1.pp46-50>
- Guntoro, G. ... Lisnawita, L. (2019). Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer. *Prediksi Jumlah Kendaraan di Provinsi Riau Menggunakan Metode Backpropagation.*, 51–57.
- Hao, X. ... Fan, W. (2016). Online Information Review. *How to Strengthen the Sosial Media Interactivity of E-Government: Evidence From China*, 79–96.
- Haryati, D. Hadiana, A. . (2016). Klasifikasi Jenis Batubara Menggunakan

- Jaringan Syaraf Tiruan Dengan Menggunakan Algoritma Backpropagation. *Seminar Nasional Teknologi Informasi dan Komunikasi*, 18–19.
- Haumeni J, W., & Elon, Y. (2022). Tingkat Kecemasan Remaja Dalam Penerimaan Vaksin COVID-19. *Community of Publishing in Nursing*, 10.
- Imam, A., & Fajtriab, H. (2015). Prosiding Seminar Nasional Matematika dan Pendidikan Matematika UMS 2015. *Implementasi Text Mining pada Mesin Pencarian Twitter untuk Menganalisis Topik - Topik Terkait “KPK dan Jokowi.”*
- Ina, N., & Ifani, H. (2021). *Sentimen Analisis COVID-19 dengan Metode Probabilistic Neural Network dan TF-IDF*, 3, 100–111.
- Iskak, I. ... Ahmad, W. R. (2021). Meningkatkan Kesadaran Masyarakat Tentang Pentingnya Vaksinasi Di Masjid Al – Ikhlas, Jakarta Barat. *Jurnal PADMA: Pengabdian Dharma Masyarakat*, 1(3).
<https://doi.org/10.32493/jpdm.v1i3.11431>
- Khaira Ummatul, A. (2021). *Analisis Sentimen Pada Tweet Dengan Tagar #Mahkamahakonstitusi Menggunakan Metode Multilayer Perceptron* [Universitas Islam Riau Pekanbaru]. <https://repository.uir.ac.id/3724/1/AsifUmmatulKhaira.pdf>
- Maulana, A. (2017). *Cuitan di Twitter kini bisa 280 karakter*. Retrieved Desember 1, 2020,. from <https://www.cnnindonesia.com>.
- Muliono, R. ... Harahap, M. K. (2019). Analysis of Frequent Itemsets Mining Algorithm Against Models of Different Datasets. *Journal of Physics: Conference Series*, 1361(1), 012036. <https://doi.org/10.1088/1742-6596/1361/1/012036>
- Muslim, M. ... Nurzahputra, A. (2019). *Data Mining Algoritma C4.5* (E. Listiana & N. Cahyani (ed.)).
- Najjichah, H. ... Subagyo, H. (2019). Pengaruh Text Preprocessing Dan Kombinasinya Pada Peringkasan Dokumen Otomatis Teks Berbahasa Indonesia. *Jurnal Teknologi Informasi*, XV(1), 1–11.

- Nemes, L., & Kiss, A. (2021). Social media sentiment analysis based on COVID-19. *Journal of Information and Telecommunication*, 5(1), 1–15. <https://doi.org/10.1080/24751839.2020.1790793>
- Pravina, A. M. ... Adikara, P. P. (2019). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. *Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM)*.
- Priyatno, A. . . . A.Z, A. (2019). Deteksi Bot Spammer Twitter Berbasis Time Interval Entropy dan Global Vectors for Word Representations Tweet's Hashtag. *Jurnal Ilmiah Teknologi Sistem Informasi*, 37.
- Prompetchara, E. ... Palaga, T. (2020). Immune responses in COVID-19 and potential vaccines: Lessons learned from SARS and MERS epidemic. *Asian Pacific Journal of Allergy and Immunology*. <https://doi.org/10.12932/AP-200220-0772>
- Puri, N. ... Gunaratne, K. (2020). Human Vaccines & Immunotherapeutics. *Social Media and Vaccine Hesitancy: New Updates for the Era of COVID-19 and Globalized Infectious Diseases*, 1–8.
- Putra, A. (2021). *Implementasi Metode Support Vector Machine Dalam Analisis Sentimen Pada Data Ulasan Twitter Vaksin COVID-19* [Universitas Islam Indonesia]. <https://repository.dinamika.ac.id/id/eprint/5917/1/17410100190-2021-UNIVERSITASDINAMIKA.pdf>
- Putri, G. S. (2020). Internet. Keraguan pada Vaksin Covid-19, Bagaimana Masyarakat Harus Bersikap? <https://www.kompas.com/sains/read/2020/12/23/160000023/keraguan-pada-vaksin-covid-19-bagaimana-masyarakat-harus-bersikap?page=all>.
- Qiu, J. ... Shuai, Q. (2019). Investigating the opinions distribution in the controversy on social media. *Information Sciences*, 489, 274–288. <https://doi.org/10.1016/j.ins.2019.03.041>
- Ramchoun, H. (2016). *International Journal of Interactive Multimedia and*

Artificial Intelligence. *Multilayer Perceptron: Architecture Optimization and Training*.

Rani, K. H. (2017). International Journal on Recent and Innovation Trends in Computing and Communication. *Advancements in Multi-Layer Perceptron Training to Improve Classification Accuracy*.

Rosyadi, H. E. ... Affandi, R. R. (2020). Rancang Bangun Chatbot Informasi Lowongan Pekerjaan Berbasis Whatsapp dengan Metode NLP (Natural Language Processing). *BRILIANT: Jurnal Riset dan Konseptual*, 5(1), 619–626.

Rusma, I. N. (2019). *Pemilihan Model Prediksi Indeks Harga Saham yang Dikembangkan Berdasarkan Algoritma Support Vector Machine(SVM) atau Multilayer Perceptron(MLP) Studi Kasus : Saham PT Telekomunikasi Indonesia TBK*, 5.

Ryandhi, R. (2017). *Penerapan Metode Artificial Neural Network (ANN) Untuk Peramalan Inflasi di Indonesia*. Institut Teknologi Sepuluh November.

Sun, P. ... Pan, B. (2020). Understanding of COVID-19 based on current evidence. *Journal of Medical Virology*, 92(6), 548–551. <https://doi.org/10.1002/jmv.25722>

Susilo, A. ... Herikurniawan, H. (2020). Coronavirus Disease 2019: Tinjauan Literatur Terkini. *J Penyakit Dalam Indonesia*.

Syadid, F. (2019). *Analisis Sentimen Komentar Netizen Terhadap Calon Presiden Indonesia 2019 dari Twitter Menggunakan Algoritma Term Frequency-Inverse Document Frequency (TF-IDF) dan Metode Multi Layer Perceptron (MLP) Neural Network* [Universitas Islam Negeri Syarif Hidayatullah Jakarta].

[https://repository.uinjkt.ac.id/dspace/bitstream/123456789/48184/1/FAQI SYADID-FST.pdf](https://repository.uinjkt.ac.id/dspace/bitstream/123456789/48184/1/FAQI%20SYADID-FST.pdf)

Wang, Z. ... Ke, H. (2020). A Handbook of 2019-nCoV Pneumonia Control and Prevention. *Hubei Sci Technol Press*, 1–108.

Wibowo, W. ... Azies, H. (2020). *Belajar Pemrograman Bahasa Python*. Fakultas Vokasi ITS PRESS.

Wong, L. Y. C., & Burkell, J. (2017). Motivations for Sharing News on Social Media. *Proceedings of the 8th International Conference on Social Media & Society - #SMSociety17*, 1–5. <https://doi.org/10.1145/3097286.3097343>

World Health Organization. (2020, Maret 11). *WHO Director-General's opening remarks at the media briefing on COVID-19*.

Yusup, M. ... Renita, D. (2019). Pengaruh Text Preprocessing Dan Kombinasinya Pada Peringkat Dokumen Otomatis Teks Berbahasa Indonesia. *MIND Journal*, *XV*(1), 1–11.



LAMPIRAN

Lampiran Kode Program

```

1. # Read dataset
2. import pandas as pd
3. df =
   pd.read_csv('https://raw.githubusercontent.com/fordinand45/Vaksin/master/vaccination_all_tweets.csv')
4. df.head()
5.
6. # Cek baris dan kolom
7. df.shape
8.
9. df.dtypes
10.
11. # cek tanggal min dan max
12. df.date.min(), df.date.max()
13.
14. # cek negara yang paling dominan
15. df['user_location'].value_counts()
16.
17. # Cek missing value
18. df.isna().sum()
19.
20. col = ['id', 'user_name', 'user_location', 'user_description',
         'user_created', 'user_followers', 'user_friends',
         'user_favourites', 'user_verified', 'hashtags', 'source',
         'retweets', 'favorites', 'is_retweet']
21.
22. df = df.drop(col, axis=1)
23. df.head()
24.
25.
26. ## Labeling Otomatis
27.
28. # Mengubah format type data
29. df['date'] = pd.to_datetime(df['date'], format='%Y-%m-%d')
30. df.dtypes
31.
32. # Import library nltk
33. import nltk
34. nltk.download('vader_lexicon')
35. from nltk.sentiment.vader import SentimentIntensityAnalyzer
36.

```

```

37. sid = SentimentIntensityAnalyzer()
38. ps = lambda x : sid.polarity_scores(x)
39. sentiment_scores = df.text.apply(ps)
40. sentiment_scores.head()
41.
42. sentiment_df = pd.DataFrame(data = list(sentiment_scores))
43. sentiment_df.head()
44.
45. # Membagi 3 jenis kelas sentimen
46. labelize = lambda x : 0 if x==0 else(1 if x>0 else -1)
47. sentiment_df['klasifikasi'] = sentiment_df.compound.apply(labelize)
48. sentiment_df.head()
49.
50. # Menggabungkan data dan sentimen
51. data = df.join(sentiment_df.klasifikasi)
52. data.head()
53.
54. # Menampilkan Jumlah klasifikasi
55. data['klasifikasi'].value_counts()
56.
57. # Menampilkan visualisasi jumlah masing-masing klasifikasi
58. import matplotlib.pyplot as plt
59.
60. labels = ['Neutral', 'Positive', 'Negative']
61. quantity = [111153, 79913, 37141]
62. colors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']
63.
64. plt.title('Percentage of Positive, Negative and Neutral Opinions')
65. plt.pie(quantity, labels=labels, colors=colors,
66.         autopct='%1.1f%%', startangle=90)
67. plt.axis('equal')
68. plt.show()
69.
70.
71. ## Text Preprocessing
72. # Tokenization
73. import re
74. nltk.download('punkt')
75. from nltk.tokenize import word_tokenize
76.
77. def tokenization(text):
78.     text = word_tokenize(text)
79.     return text
80.
81. data['tokenization'] = data['text'].apply(lambda x: tokenization(x))
82. data.head()

```

```

83.
84. # Normalization
85. def normalization(text):
86.     text = re.sub(r'[\#\@\!]\w+', "", text) #removing # and @
87.     text = re.sub(r'[\d+]', "", text) #removing numbers
88.     text = re.sub(r'RT[\s]+', "", text)
89.     text = re.sub(r'http\S+|www\S+|https\S+', "", text) #removing urls
90.     text = re.sub(r'[\.\,\;\!\?\*\$\%\'\\"':]', "", text) #removing some non-sense
        characters
91.     text = text.lower()
92.     return text
93.
94. data['normalization'] = data['text'].transform(normalization).apply(lambda
        x: tokenization(x))
95. data.head()
96.
97. # Menghapus Emoticon
98. try:
99.     import emot
100.    except:
101.        !pip install emot
102.        import emot
103.
104.    from emot.emo_unicode import UNICODE_EMOJI
105.    emojis = list(UNICODE_EMOJI.keys())
106.
107.    def emoji(text):
108.        text = ''.join([word for word in text if word not in emojis])
        #removing emojis
109.        return text
110.
111.    data['menghapus_emoji'] =
        data['normalization'].transform(emoji).apply(lambda text :
        tokenization(text))
112.    data.head()
113.
114.    # Stopwords
115.    from nltk.corpus import stopwords
116.    from string import punctuation
117.    import string
118.    nltk.download('stopwords')
119.    stop = set(stopwords.words('english'))
120.    punctuation = list(string.punctuation)
121.    stop.update(punctuation)
122.
123.    # function to remove stopwords

```

```

124.     def stopWords(tweet):
125.         clean_tweet = tweet
126.         clean_tweet = " ".join(word for word in clean_tweet if word not
           in stop)
127.         return clean_tweet
128.
129.     data['stopwords'] =
           data['menghapus_emoji'].transform(stopWords).apply(lambda x:
           tokenization(x))
130.     data.head()
131.
132.     # Lemmatization
133.     nltk.download('wordnet')
134.     from nltk.stem import WordNetLemmatizer
135.     nltk.download('omw-1.4')
136.
137.     lemma = nltk.WordNetLemmatizer()
138.
139.     def lemmatizer(text):
140.         text = [lemma.lemmatize(word) for word in text]
141.         return text
142.
143.     data['lemmatization'] =
           data['stopwords'].transform(lemmatizer).apply(lambda x: lemmatizer(x))
144.     data.head()
145.
146.     # TF-IDF
147.     lemma = data['lemmatization'].astype(str)
148.
149.     from sklearn.feature_extraction.text import TfidfVectorizer
150.
151.     tfidf_vectorizer = TfidfVectorizer()
152.     X_input_tfidf = tfidf_vectorizer.fit_transform(lemma)
153.     print(tfidf_vectorizer.vocabulary_)
154.
155.     # Menampilkan hasil TF-IDF dari index 0
156.     print(lemma[0])
157.     print(X_input_tfidf[0])
158.
159.
160.     ## Membuat Model menggunakan Algoritma Multilayer
           Perceptron
161.     # PENGUJIAN 1 : TRAIN 70% : TEST 30%
162.     from sklearn.model_selection import train_test_split
163.     from sklearn.neural_network import MLPClassifier
164.     from sklearn import metrics

```

```

165.     from sklearn.metrics import accuracy_score, precision_score,
        recall_score
166.     from sklearn.metrics import confusion_matrix
167.     import numpy as np
168.
169.     # get X and y
170.     df_X = X_input_tfidf
171.     df_y = data['klasifikasi']
172.
173.     #hold out, dibagi menjadi training dan testing set
174.     X_train, X_test, y_train, y_test = train_test_split(df_X, df_y,
        test_size=0.3, random_state=0)
175.
176.
177.     # Model Learning
178.     mlp1= MLPClassifier(hidden_layer_sizes=(61), alpha=0.2,
        activation='logistic')
179.     mlp1.fit(X_train, y_train)
180.
181.     y_pred_tr = mlp1.predict(X_train)
182.     y_pred_ts = mlp1.predict(X_test)
183.
184.     # Performance training
185.     print('=== Multilayer Perceptron Performance Training ===\n')
186.     print('Accuracy :', metrics.accuracy_score(y_train, y_pred_tr))
187.     print('Precision :', metrics.precision_score(y_train, y_pred_tr,
        average='macro', labels=np.unique(y_pred_tr)))
188.     print('Recall :', metrics.recall_score(y_train, y_pred_tr,
        average='macro'))
189.     print('Confuse Matrix Train:\n', confusion_matrix(y_train,
        y_pred_tr))
190.     print('-----')
191.
192.     # Performance testing
193.     print('=== Multilayer Perceptron Performance Testing ===\n')
194.     print('Accuracy :', metrics.accuracy_score(y_test, y_pred_ts))
195.     print('Precision :', metrics.precision_score(y_test, y_pred_ts,
        average='macro', labels=np.unique(y_pred_ts)))
196.     print('Recall :', metrics.recall_score(y_test, y_pred_ts,
        average='macro'))
197.     print('Confuse Matrix Test:\n', confusion_matrix(y_test,
        y_pred_ts))
198.
199.     # PENGUJIAN 2 : TRAIN 80% : TEST 20%
200.     from sklearn.model_selection import train_test_split
201.     from sklearn.neural_network import MLPClassifier

```

```

202.     from sklearn import metrics
203.     from sklearn.metrics import accuracy_score, precision_score,
        recall_score
204.     from sklearn.metrics import confusion_matrix
205.     import numpy as np
206.
207.     # get X and y
208.     df_X = X_input_tfidf
209.     df_y = data['klasifikasi']
210.
211.     #hold out, dibagi menjadi training dan testing set
212.     X_train, X_test, y_train, y_test = train_test_split(df_X, df_y,
        test_size=0.2, random_state=0)
213.
214.
215.     # Model Learning
216.     mlp2= MLPClassifier(hidden_layer_sizes=(61), alpha=0.2,
        activation='logistic')
217.     mlp2.fit(X_train, y_train)
218.
219.     y_pred_tr = mlp2.predict(X_train)
220.     y_pred_ts = mlp2.predict(X_test)
221.
222.     # Performance training
223.     print('=== Multilayer Perceptron Performance Training ===\n')
224.     print('Accuracy :', metrics.accuracy_score(y_train, y_pred_tr))
225.     print('Precision :', metrics.precision_score(y_train, y_pred_tr,
        average='macro', labels=np.unique(y_pred_tr)))
226.     print('Recall :', metrics.recall_score(y_train, y_pred_tr,
        average='macro'))
227.     print('Confuse Matrix Train:\n', confusion_matrix(y_train,
        y_pred_tr))
228.     print('-----')
229.
230.     # Performance testing
231.     print('=== Multilayer Perceptron Performance Testing ===\n')
232.     print('Accuracy :', metrics.accuracy_score(y_test, y_pred_ts))
233.     print('Precision :', metrics.precision_score(y_test, y_pred_ts,
        average='macro', labels=np.unique(y_pred_ts)))
234.     print('Recall :', metrics.recall_score(y_test, y_pred_ts,
        average='macro'))
235.     print('Confuse Matrix Test:\n', confusion_matrix(y_test,
        y_pred_ts))
236.
237.     # PENGUJIAN 3 : TRAIN 90% : TEST 10%
238.     from sklearn.model_selection import train_test_split

```

```

239.     from sklearn.neural_network import MLPClassifier
240.     from sklearn import metrics
241.     from sklearn.metrics import accuracy_score, precision_score,
        recall_score
242.     from sklearn.metrics import confusion_matrix
243.     import numpy as np
244.
245.     # get X and y
246.     df_X = X_input_tfidf
247.     df_y = data['klasifikasi']
248.
249.     #hold out, dibagi menjadi training dan testing set
250.     X_train, X_test, y_train, y_test = train_test_split(df_X, df_y,
        test_size=0.1, random_state=0)
251.
252.
253.     # Model Learning
254.     mlp3= MLPClassifier(hidden_layer_sizes=(61), alpha=0.2,
        activation='logistic')
255.     mlp3.fit(X_train, y_train)
256.
257.     y_pred_tr = mlp3.predict(X_train)
258.     y_pred_ts = mlp3.predict(X_test)
259.
260.     # Performance training
261.     print('=== Multilayer Perceptron Performance Training ===\n')
262.     print('Accuracy :', metrics.accuracy_score(y_train, y_pred_tr))
263.     print('Precision :', metrics.precision_score(y_train, y_pred_tr,
        average='macro', labels=np.unique(y_pred_tr)))
264.     print('Recall :', metrics.recall_score(y_train, y_pred_tr,
        average='macro'))
265.     print('Confuse Matrix Train:\n', confusion_matrix(y_train,
        y_pred_tr))
266.     print('-----')
267.
268.     # Performance testing
269.     print('=== Multilayer Perceptron Performance Testing ===\n')
270.     print('Accuracy :', metrics.accuracy_score(y_test, y_pred_ts))
271.     print('Precision :', metrics.precision_score(y_test, y_pred_ts,
        average='macro', labels=np.unique(y_pred_ts)))
272.     print('Recall :', metrics.recall_score(y_test, y_pred_ts,
        average='macro'))
273.     print('Confuse Matrix Test:\n', confusion_matrix(y_test,
        y_pred_ts))
274.

```

```

275.     ## Membuat/Menampilkan Wordcloud dari masing-masing
         sentiment
276.     import matplotlib.pyplot as plt
277.     from wordcloud import WordCloud
278.
279.     # Wordcloud Sentimen Negatif
280.     neg_tweets = data[data.klasifikasi == -1]
281.     neg_string = []
282.     for t in neg_tweets['lemmatization'].astype(str):
283.         neg_string.append(t)
284.     neg_string = pd.Series(neg_string).str.cat(sep=' ')
285.     from wordcloud import WordCloud
286.     wordcloud = WordCloud(width=1000, height=1000,
        max_font_size=200, background_color='white').generate(neg_string)
287.     plt.figure(figsize=(8,8), dpi=100)
288.     plt.imshow(wordcloud, interpolation="bilinear")
289.     plt.axis("off")
290.     plt.show()
291.
292.     # Wordcloud Sentimen Positif
293.     pos_tweets = data[data.klasifikasi == 1]
294.     pos_string = []
295.     for t in pos_tweets['lemmatization'].astype(str):
296.         pos_string.append(t)
297.     pos_string = pd.Series(pos_string).str.cat(sep=' ')
298.     wordcloud = WordCloud(width=1000, height=1000,
        max_font_size=200, background_color='white').generate(pos_string)
299.     plt.figure(figsize=(8,8), dpi=100)
300.     plt.imshow(wordcloud, interpolation="bilinear")
301.     plt.axis("off")
302.     plt.show()
303.
304.     # Wordcloud Sentimen Netral
305.     net_tweets = data[data.klasifikasi == 0]
306.     net_string = []
307.     for t in net_tweets['lemmatization'].astype(str):
308.         net_string.append(t)
309.     net_string = pd.Series(net_string).str.cat(sep=' ')
310.     wordcloud = WordCloud(width=1000,
        height=1000, max_font_size=200,
        background_color='white').generate(net_string)
311.     plt.figure(figsize=(8,8), dpi=100)
312.     plt.imshow(wordcloud, interpolation="bilinear")
313.     plt.axis("off")
314.     plt.show()

```