



**ANALISIS SENTIMEN PADA APLIKASI PEDULILINDUNGI
MENGUNAKAN METODE *LEXICON***

**NAMA : INA KRISTIANA
NPM : 18360004**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI INFORMASI
INSTITUT SAINS DAN TEKNOLOGI NASIONAL
JAKARTA
FEBRUARI 2022**



**ANALISIS SENTIMEN PADA APLIKASI PEDULILINDUNGI
MENGUNAKAN METODE *LEXICON***

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer

**NAMA : INA KRISTIANA
NPM : 18360004**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI INFORMASI
INSTITUT SAINS DAN TEKNOLOGI NASIONAL
JAKARTA
FEBRUARI 2022**

SURAT PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

NAMA : INA KRISTIANA

NPM : 18360004

TANGGAL :

TTD di atas MATERAI

HALAMAN PERNYATAAN NON PLAGIAT

Saya yang bertanda tangan di bawah ini :

Nama : Ina Kristiana

NPM : 18360004

Mahasiswa : Strata Satu (S1)

Tahun Akademik : 2018

Menyatakan bahwa saya tidak melakukan kegiatan plagiat dalam penulisan Skripsi yang berjudul “Analisis Sentimen Pada Aplikasi PeduliLindungi Menggunakan Metode *Lexicon*”.

Apabila suatu saat nanti terbukti saya melakukan plagiat, maka saya akan menerima sanksi yang telah ditetapkan.

Demikian Surat Pernyataan ini saya buat dengan sebenar-benarnya.

Jakarta,

TTD di atas Materai

Ina Kristiana

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Ina Kristiana
NPM : 18360004
Program Studi : Teknik Informatika
Judul Skripsi : Analisis Sentimen Aplikasi PeduliLindungi
Menggunakan Metode *Lexicon*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi Informasi, Institut Sains dan Teknologi Nasional

DEWAN PENGUJI

Pembimbing : Neny Rosmawarni, S.Kom., M.Kom
: NIDN. 0312018701 (.....)

Penguji : Ir. Andi Suprianto, M.Kom
: NIDN. 0327025904 (.....)

Penguji : Siti Nurmiati, S.Kom., M.Kom
: NIDN. 0402107703 (.....)

Penguji : Aryo Nur Utomo, S.T., M.Kom
: NIDN. 0319046803 (.....)

Ditetapkan di : Jakarta
Tanggal :

KATA PENGANTAR

Puji syukur saya panjatkan kepada Allah S.W.T. karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan proposal yang berjudul “Analisis Sentimen Pada Aplikasi Peduli Lindungi Menggunakan Metode *Lexicon*”. Penulisan proposal ini dilakukan dalam rangka memenuhi salah satu syarat memenuhi syarat akademis dalam menyelesaikan pendidikan tingkat Sarjana Program Studi Teknik Informatika pada Fakultas Sains dan Teknologi Informasi, Institut Sains dan Teknologi Nasional. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, terasa sangat sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, pada kesempatan ini saya mengucapkan terima kasih kepada semua pihak yang telah membantu saya dalam menyusun laporan ini, antara lain:

- a) Ibu Nyeny Rosmawarni, S.Kom., M.Kom selaku Dosen Pembimbing dan Ketua Program Studi Teknik Informatika, Fakultas Sains dan Teknologi Informasi, Institut Sains dan Teknologi Nasional yang telah banyak membantu, memberikan waktu luang, dan mengarahkan pada proses skripsi ini sampai selesai;
- b) Kedua orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
- c) Seluruh pihak yang telah berkontribusi, memberikan masukan dan motivasi serta kerja sama.

Akhir kata, saya berharap Allah S.W.T. berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga laporan proyek ini membawa manfaat bagi pengembangan ilmu pengetahuan.

Jakarta,

Penulis

Ina Kristiana

SURAT PERSYARATAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Institut Sains Dan Teknologi Nasional, saya yang bertanda tangan di bawah ini:

Nama : Ina Kristiana
NPM : 18360004
Program Studi : Teknik Informatika
Fakultas : Fakultas Sains Dan Teknologi Informasi
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Sains dan Teknologi Nasional **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul: **“Analisis Sentimen Pada Aplikasi PeduliLindungi Menggunakan Metode *Lexicon*”**.

Dengan Hak Bebas Royalti Noneksklusif ini Institut Sains dan Teknologi Nasional berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*) *soft copy & hard copy*, merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta
Pada Tanggal :

Yang menyatakan

(Ina Kristiana)

ABSTRAK

Nama : Ina Kristiana
Program Studi : Teknik Informatika
Judul : Analisis Sentimen Pada Aplikasi PeduliLindungi Menggunakan Metode *Lexicon*

Twitter merupakan platform sosial media yang memiliki pengguna cukup banyak. Dalam platform Twitter kita bebas berekspresi terkait hal apapun selama tidak melanggar ketentuan yang berlaku. Saat ini aplikasi PeduliLindungi sedang ramai diperbincangkan di Twitter maupun media social lainnya. Penelitian ini bertujuan untuk mengetahui sentimen terhadap data tanggapan pada aplikasi PeduliLindungi di Twitter. Peneliti melakukan pengambilan data menggunakan API Twitter dengan *library tweepy* yang kemudian akan dilakukan proses analisis sentimen menggunakan metode *lexicon* untuk menentukan hasil dari setiap tanggapan yang ada. Hasil yang diperoleh dalam penelitian ini yaitu tanggapan terhadap aplikasi PeduliLindungi bernilai negatif.

Kata kunci:

Analisis Sentimen, Twitter, PeduliLindungi, Keamanan Data, API, *Lexicon*

ABSTRACT

Name : Ina Kristiana
Study Program : Teknik Informatika
Title : Sentiment Analysis in PeduliLindungi Applications
Using the Lexicon Method

Twitter is a social media platform that has quite a lot of users. On the Twitter platform, we are free to express ourselves regarding any matter as long as it does not violate the applicable provisions. Currently, the PeduliLindungi application is being discussed on Twitter and other social media. This study aims to determine the sentiment towards the response data on the PeduliLindungi application on Twitter. The researcher collects data using the Twitter API with the tweepy library which will then carry out a sentiment analysis process using the lexicon method to determine the results of each response. The results obtained in this study are responses to the PeduliLindungi application that are worthy of being negative.

Keywords:

Sentiment Analysis, Twitter, PeduliLindungi, API, Lexicon

DAFTAR ISI

HALAMAN JUDUL.....	i
SURAT PERNYATAAN ORISINALITAS	ii
HALAMAN PERNYATAAN NON PLAGIAT	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
SURAT PERSYARATAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR PSEUDO CODE	xii
DAFTAR LAMPIRAN	xiii
1. PENDAHULUAN	xiv
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian.....	2
1.4. Manfaat Penelitian.....	3
1.5. Batasan Masalah.....	3
1.6. Sistematika Penulisan.....	4
2. TINJAUAN PUSTAKA	5
2.1. Analisis Sentimen.....	5
2.2. <i>Natural Language Processing</i> (NLP)	6
2.3. <i>Data Mining</i>	7
2.4. <i>Preprocessing</i>	7
2.5. <i>Application Programming Interface</i> (API)	8
2.6. Aplikasi PeduliLindungi	9
2.7. Bahasa Pemrograman Python.....	10
2.8. <i>Indonesia Sentiment Lexicon</i> (InSet).....	12
2.9. Penelitian Terdahulu	12
3. METODOLOGI PENELITIAN	15
3.1. Tahapan Penelitian	15
3.2. Tempat dan Waktu Penelitian	18
3.3. Instrumen Penelitian.....	18
3.4. Persiapan Alat Penelitian	18
3.5. Pengumpulan Data	23
3.6. Pra-Proses.....	25
3.7. Perhitungan Analisis Sentimen	25
4. HASIL DAN PEMBAHASAN	37
4.1. Hasil Analisis	37
4.2. Pembahasan.....	39
5. PENUTUP	42
5.1. Kesimpulan.....	42
5.2. Saran.....	42
DAFTAR PUSTAKA	43
LAMPIRAN	45

DAFTAR TABEL

Tabel 2.1. Perbandingan Penelitian Terdahulu	13
Tabel 3.1. <i>Tweet</i> Tanggapan Aplikasi PeduliLindungi	25
Tabel 3.2. <i>Tweet</i> Tanggapan Melalui Proses <i>Cleaning Text</i>	27
Tabel 3.3. <i>Tweet</i> Tanggapan Melalui Proses <i>Case Folding Text</i>	28
Tabel 3.4. <i>Tweet</i> Tanggapan Melalui Proses <i>Tokenizing Text</i>	29
Tabel 3.5. <i>Tweet</i> Tanggapan Melalui Proses <i>Filtering Text</i>	30
Tabel 3.6. <i>Tweet</i> Tanggapan Melalui Proses <i>Stemming Text</i>	32
Tabel 3.7. <i>Tweet</i> Tanggapan Setelah Melewati Tahap <i>Preprocessing</i>	33
Tabel 3.8. Perhitungan <i>Polarity Score</i>	35
Tabel 4.1. Jumlah Tanggapan Aplikasi PeduliLindungi	37
Tabel 4.2. Kata Tidak Baku	40

DAFTAR GAMBAR

Gambar 2.1. Produk <i>Natural Language Processing</i>	6
Gambar 2.2. Ilustrasi <i>Application Programming Interface</i>	8
Gambar 2.3. Aplikasi PeduliLindungi	9
Gambar 3.1. Tahapan Penelitian	15
Gambar 3.2. Contoh <i>Lexicon</i>	17
Gambar 3.3. Halaman Unduh Python	19
Gambar 3.4. Tampilan Awal Instalasi Python	19
Gambar 3.5. Tampilan Akhir Instalasi Python.....	20
Gambar 3.6. Tampilan <i>Command Prompt</i> Awal Instalasi <i>Jupyter Notebook</i>	21
Gambar 3.7. Tampilan <i>Jupyter Notebook</i>	21
Gambar 3.8. Tampilan <i>Command Prompt</i> Awal Instalasi <i>Library</i>	22
Gambar 3.9. Tampilan <i>Command Prompt</i> Selesai Instalasi <i>Library</i>	22
Gambar 3.10. Proses Pengumpulan Data	24
Gambar 3.11. Sampel Data Penelitian	24
Gambar 3.12. Alur Perhitungan <i>Polarity Score</i>	35
Gambar 4.1. Proporsi Tanggapan Aplikasi PeduliLindungi	36
Gambar 4.2. Proporsi Label	38
Gambar 4.3. <i>WordCloud</i> PeduliLindungi	39

DAFTAR PSEUDO CODE

Pseudo Code 3.1. Sintaks Proses <i>Cleaning Text</i>	26
Pseudo Code 3.2. Sintaks Proses <i>Case Folding Text</i>	28
Pseudo Code 3.3. Sintaks Proses <i>Tokenizing Text</i>	29
Pseudo Code 3.4. Sintaks Proses <i>Filtering Text</i>	30
Pseudo Code 3.5. Sintaks Proses <i>Stemming Text</i>	31
Pseudo Code 3.6. Sintaks Proses Pelabelan.....	33

DAFTAR LAMPIRAN

Lampiran 1. Lembar Konsultasi.....	45
Lampiran 2. Baris Kode	46

BAB 1

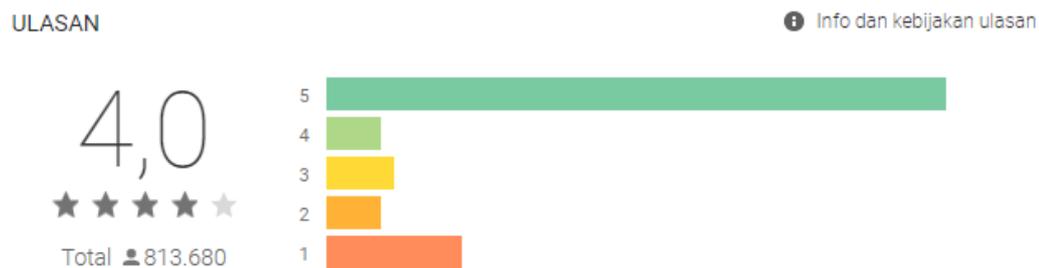
PENDAHULUAN

1.1. Latar Belakang

Dewasa ini, media sosial berkembang pesat di internet, salah satu yang banyak digemari adalah Twitter. Seperti media sosial lainnya, Twitter merupakan *platform* media sosial yang memungkinkan penggunaanya untuk berinteraksi secara personal ataupun terbuka. Berbagai topik ramai diperbincangkan di Twitter mulai dari politik, ekonomi, sosial, budaya, dan hukum. Melalui fitur *hashtag* para pengguna Twitter dapat mengetahui topik yang sedang dibahas secara real-time. Selain itu kata kunci pada Twitter dapat pula menjadi sumber perbincangan oleh pengguna. Tidak jarang sebelum suatu berita muncul sebagai *headline* di media *online* pembahasannya sudah terlebih dahulu dibahas di Twitter. Bahkan pembahasan di Twitter memicu suatu berita muncul. Saat ini penggunaan Twitter khususnya di Indonesia telah memberikan dampak besar pada sudut pandang topik tertentu.

Aplikasi PeduliLindungi merupakan aplikasi yang dibangun untuk membantu pemerintah dalam melakukan *tracing* dan *tracking* terhadap kasus COVID-19 yang berlangsung di Indonesia. Dari data yang dihasilkan oleh aplikasi PeduliLindungi itu kemudian dimanfaatkan dan diolah oleh pemerintah Indonesia dalam menentukan sebuah kebijakan dalam mengatur persebaran COVID-19 itu sendiri. Pada aplikasi PeduliLindungi juga terdapat fitur yang berkaitan dengan vaksin yaitu melihat jadwal vaksin pengguna dan sertifikat vaksin pengguna. Selain manfaat yang dihasilkan oleh aplikasi tersebut, nyatanya banyak pengguna PeduliLindungi yang meragukan dari aplikasi tersebut dan menuai kontroversi.

Salah satu topik yang sedang *trend* belakangan ini di Twitter adalah terkait aplikasi PeduliLindungi itu sendiri. Banyak pengguna yang mengomentari terkait kegunaan dan keamanan aplikasi PeduliLindungi. Selain itu peneliti juga melihat bahwa ulasan aplikasi PeduliLindungi pada Google PlayStore cenderung tidak terlalu sempurna yaitu ada di angka 4.0/5.0 dari 813.680 orang yang melakukan ulasan dan hal ini ditunjukkan pada **Gambar 1.1**.



Gambar 1.1. Ulasan Aplikasi PeduliLindungi

Analisis sentimen ialah 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, Touahria dan Tari, 2019).

Analisis sentimen dilakukan guna menilai opini dan kecenderungan sebuah opini terhadap suatu topik baik negatif maupun positif (Rozi, Pramono dan Dahlan, 2012). 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.

Terdapat dua pendekatan analisis sentimen yang dapat dilakukan yaitu *machine learning based* dan *lexicon based*. *Machine learning based* merupakan pendekatan analisis sentimen yang dibangun berdasarkan data lampau yang memiliki label dan dilakukan klasifikasi menggunakan algoritma. *Lexicon based* merupakan pendekatan analisis sentimen yang dilakukan perhitungan *polarity score* berdasarkan *score* setiap katanya yang ditentukan berdasarkan *lexicon positive* dan *lexicon negative*.

Untuk menyikapi permasalahan dan kontroversi yang menuai aplikasi PeduliLindungi, salah satu cara yang perlu dilakukan adalah untuk mengetahui sentimen yang melekat pada tanggapan pengguna PeduliLindungi dan masyarakat Indonesia yaitu dengan cara analisis sentimen terhadap data cuitan yang terdapat pada *platform* di Twitter. Setelah diketahui dari tanggapan terhadap data aplikasi PeduliLindungi yang dihasilkan dari proses analisis sentimen dapat digunakan untuk proses bisnis yang berkaitan dengan PeduliLindungi selanjutnya.

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang masalah di atas dapat dibuat rumusan masalah sebagai berikut:

1. Bagaimana proses analisis sentimen menggunakan metode *lexicon*?
2. Bagaimana pandangan masyarakat Indonesia terhadap aplikasi PeduliLindungi?

1.3. Tujuan Penelitian

Dalam penelitian ini penulis memberikan batasan masalah pada:

1. Mengetahui proses analisis sentimen menggunakan metode *lexicon*.
2. Mengetahui sentimen terhadap data tanggapan pada aplikasi PeduliLindungi.

1.4. Manfaat Penelitian

Manfaat yang diperoleh dalam penelitian ini adalah:

1. Dengan mengetahui proses analisis sentimen menggunakan metode *lexicon* diharapkan dapat pandangan masyarakat Indonesia terhadap aplikasi PeduliLindungi.
2. Sebagai tugas akhir penulis untuk menyelesaikan program sarjana.

1.5. Batasan Masalah

Dalam melakukan penelitian ini terdapat beberapa hal yang menjadi batasan masalah agar tercipta penelitian yang komprehensif dan tepat sasaran. Adapun batasan masalah tersebut adalah sebagai berikut:

1. Terbatas pada dataset dengan kata kunci PeduliLindungi yang di ambil pada bulan Februari 2022.
2. Terbatas pada proses analisis sentimen menggunakan metode *lexicon*.

1.6. Hipotesis Penelitian

Hipotesis dalam penelitian ini adalah sebagai berikut:

1. Proses analisis sentimen menggunakan metode *lexicon* berjalan dengan baik dan dapat menentukan kategori tanggapan.

2. Sentimen terhadap tanggapan aplikasi PeduliLindungi adalah negatif.

1.7. Sistematika Penulisan

- BAB I** **PENDAHULUAN**
Membahas tentang latar belakang masalah, maksud, dan tujuan, ruang lingkup, serta sistematika penulisan.
- BAB II** **LANDASAN TEORI**
Berisi uraian teori-teori yang menunjang penelitian yang mencakup pengertian dari analisis sentimen, *data mining*, *natural language processing* (NLP), dan *Application Programming Interface* (API).
- BAB III** **METODOLOGI PENELITIAN**
Berisi uraian tentang penelitian yang dilakukan meliputi penjelasan mengenai data yang akan digunakan, tahapan penelitian, dan hasil penelitian yang akan dilakukan.
- BAB IV** **HASIL DAN PEMBAHASAN**
Tentang uraian hasil analisis dari penelitian dan pembahasan terkait proses analisis sentimen menggunakan metode *lexicon*.
- BAB V** **PENUTUP**
Tentang uraian dari hasil penelitian yang ditarik dari hasil analisis pada bab sebelumnya serta saran yang diharapkan dapat memberi masukan untuk penelitian lanjutan.

BAB 2

TINJAUAN PUSTAKA

2.1. Analisis Sentimen

Sentiment analysis atau *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik, dan *text mining* yang bertujuan menganalisa pendapat, sentimen, evaluasi, sikap, penilaian, dan emosi seseorang apakah pembicara atau penulis berkenan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu. (Ardiani et al., 2020).

Analisis sentimen ialah 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 et al., 2021).

Berdasarkan sumber datanya *sentiment analysis* terbagi menjadi 2 kelompok yaitu, *coarsed-grained sentiment analysis*, *sentiment analysis* yang dilakukan pada level dokumen, secara garis besar fokus utama dari *sentiment analysis* jenis ini adalah menganggap seluruh isi dokumen sebagai sebuah sentiment positif dan sentiment negative. *Fined-grained sentiment analysis* adalah *sentiment analysis* yang dilakukan pada level kalimat, fokus utamanya adalah menentukan sentimen pada setiap kalimat. *Sentiment analysis* dapat diklasifikasikan ke dalam kelas sentimen bersifat positif, negatif, dan netral (Ardiani et al., 2020).

Terdapat dua pendekatan analisis sentimen yaitu dengan menggunakan *machine learning* dan menggunakan *lexicon*. Proses *sentiment analysis* yang dilakukan oleh penulis adalah menggunakan metode *lexicon*. *Lexicon* itu sendiri merupakan kamus yang berisikan kata-kata yang telah dilakukan pembobotan oleh penelitian sebelumnya. *Lexicon* dibagi menjadi dua yaitu *lexicon* positif dan *lexicon* negatif. Sehingga setelah proses *sentiment analysis* itu dilakukan, maka dihasilkan sebuah data yang memiliki label positif, negatif, dan netral.

1. Sentimen Positif : Menurut Kamus Besar Bahasa Indonesia (KBBI) sentimen positif merupakan reaksi atau sikap yang meningkatkan nilai seseorang atau sesuatu (Ardiani et al., 2020).
2. Sentimen Negatif : Menurut Kamus Besar Bahasa Indonesia (KBBI) sentimen negative merupakan reaksi atau sikap yang menurunkan nilai

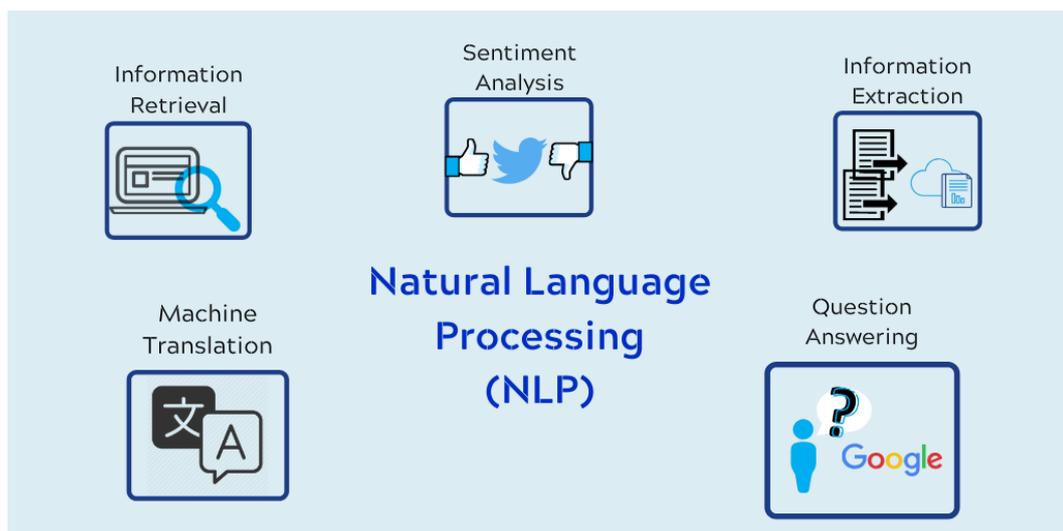
seseorang atau sesuatu, jadi kalimat yang memiliki sentiment negative akan menyebabkan penyurunan nilai pandang terhadap sesuatu, sehingga membentuk tren menurun (Ardiani et al., 2020).

3. Sentimen Netral : Kata netral sendiri dalam Kamus Besar Bahasa Indonesia (KBBI) berarti tidak berpihak, kalimat bersentimen netral merupakan ekspresi kalimat yang tidak bersifat positif maupun negatif (Ardiani et al., 2020).

2.2. *Natural Language Processing (NLP)*

NLP adalah disiplin ilmu komputer yang bertujuan untuk memahami konsep dan maksud dari bahasa manusia. Sementara manusia cukup mahir memahami sintaks linguistic dan tata bahasa serta hubungan spasial tersirat, komputer memiliki kesulitan besar pengolahan query bahasa alami (Huda, 2021).

Contoh dari produk *natural language processing (NLP)* adalah *information retrieval*, *sentimen analysis*, *information extraction*, *machine translation* dan *question answering*. Hal ini dapat ditunjukkan oleh **Gambar 2.1**.



(Sumber: <https://github.com/CSolitaire>)

Gambar 2.1. Produk *Natural Language Processing*

Natural language processing adalah metode yang memproses input teks menjadi kata-kata kunci jawaban *user*. Pengolahan bahasa alami merupakan cabang

dari ilmu komputer dan linguistik yang membahas tentang interaksi antara manusia dengan komputer menggunakan bahasa manusia (Aditama, 2020).

2.3. Data Mining

Data Mining adalah proses yang memperkerjakan satu atau lebih Teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstraksi pengetahuan (*knowledge*) secara otomatis (Sikumbang, 2018).

Data mining merupakan proses *iterative* dan interaktif untuk menemukan pola atau model baru yang sempurna, bermanfaat, dan dapat dimengerti dalam suatu *database* yang sangat besar (*massive database*). *Data mining* berisi pencarian tren atau pola yang diinginkan dalam *database* besar untuk membantu pengambilan keputusan diwaktu yang akan datang, pola-pola ini dikenali perangkat tertentu yang dapat memberikan suatu analisis data yang berguna dan berwawasan yang kemudian dapat dipelajari dengan lebih teliti, yang mungkin saja menggunakan perangkat pendukung keputusan yang lain (Sikumbang, 2018).

2.4. Preprocessing

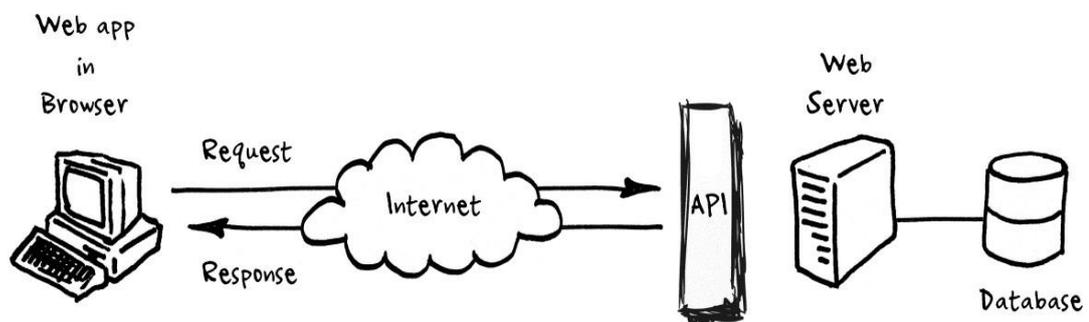
Preprocessing merupakan tahap manipulasi data, penghapusan data, dan modifikasi data sebelum data tersebut digunakan untuk meningkatkan kinerja performa pada Langkah selanjutnya dalam proses klasifikasi menggunakan *machine learning*. *Preprocessing* pada analisis sentimen diantaranya meliputi:

1. *Cleaning Text* : Merupakan proses penghilangan komponen tertentu yang terdapat dalam *tweet* yakni dapat berupa *URL*, *username*, *retweet*, *symbol*, dan *hashtag*.
2. *Case Folding Text* : Merupakan proses perubahan teks dalam dokumen menjadi bentuk huruf kecil atau *lowercase*.
3. *Tokenizing Text* : Merupakan proses pemecahan teks menjadi kata dengan batasan spasi.
4. *Filtering Text* : Merupakan proses penghapusan kata berdasarkan kata yang terdapat dalam daftar *stoplist* yang merupakan kata yang tidak memiliki makna sentimen.

5. *Stemming Text* : Merupakan proses penguraian kata menjadi bentuk kata dasar.

2.5. *Application Programming Interface (API)*

API bekerja dengan cara membantu aplikasi berinteraksi dengan *library* dengan mengikuti serangkaian aturan yang ditentukan sebelumnya oleh API itu sendiri. Pendekatan ini memudahkan *developer* untuk membuat aplikasi yang berkomunikasi dengan berbagai *library* tanpa harus memikirkan kembali strategi yang digunakan selama semua *library* mengikut API yang sama. Kelebihan lain dari metode ini menunjukkan betapa mudahnya menggunakan *library* yang sama dengan bahasa pemrograman yang berbeda. Salah satu contoh pemanfaatan API adalah API dari Twitter, di mana API tersebut menyediakan akses *read* dan *write* data dengan mengintegrasikan Twitter ke dalam aplikasi kita sendiri. Ilustrasi dari proses kerja API ditunjukkan pada **Gambar 2.2**.



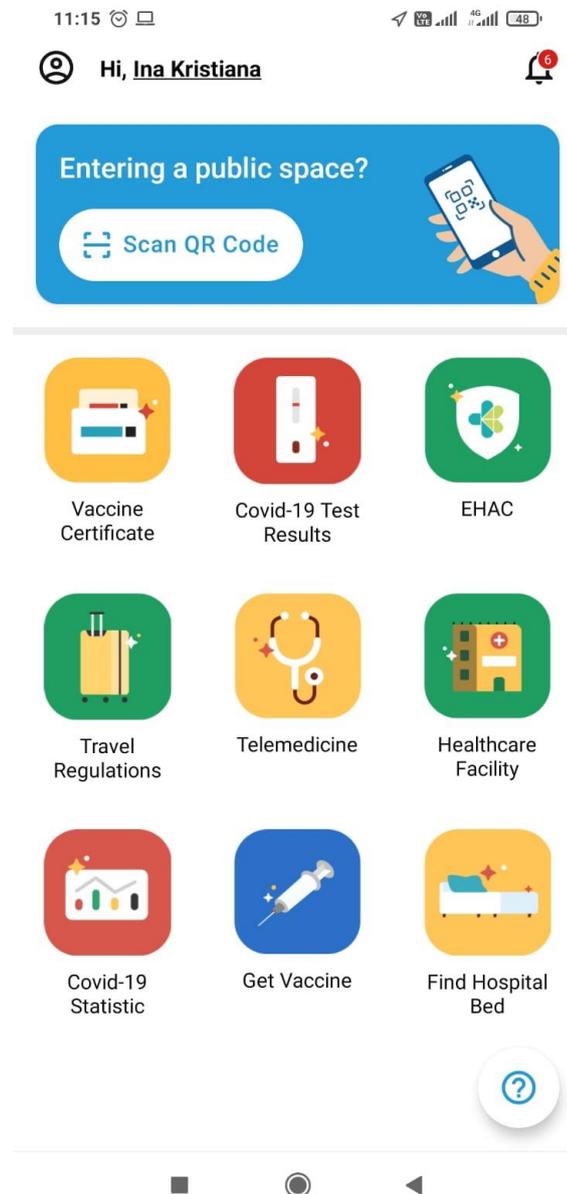
(Sumber: Tech Funnel)

Gambar 2.2. Ilustrasi Application Programming Interface

Twitter API juga dapat memiliki 3 bagian, yakni *Search* API yang dapat memudahkan *user* dalam mengelola *query search* pada konten twitter. *Keyword* dapat dicari dengan berdasar pada kata khusus atau mencari tweet yang lebih spesifik berdasarkan *username* Twitter. Selanjutnya, *Representational State Transfer* (REST) API yang digunakan untuk membangun sebuah aplikasi Twitter yang lebih kompleks dan memerlukan inti dari Twitter. 8 Dan terakhir adalah *Streaming* API yang digunakan *developer* untuk kebutuhan yang lebih intensif. Selain itu, streaming API dapat menghasilkan aplikasi yang mampu mengetahui tweet dan pengikut dari sebuah akun.

2.6. Aplikasi PeduliLindungi

Aplikasi PeduliLindungi adalah aplikasi yang dikembangkan untuk membantu instansi pemerintah terkait dalam melakukan pelacakan untuk menghentikan penyebaran *Coronavirus Disease* (COVID-19). Tampilan aplikasi ini ditunjukkan pada **Gambar 2.3**.



Gambar 2.3. Aplikasi PeduliLindungi

Aplikasi ini mengandalkan partisipasi masyarakat untuk saling membagikan data lokasinya saat bepergian agar penelusuran riwayat kontak dengan penderita COVID-19 dapat dilakukan. Pengguna aplikasi ini juga akan mendapatkan

notifikasi jika berada di keramaian atau berada di zona merah, yaitu area atau kelurahan yang sudah terdata bahwa ada orang yang terinfeksi COVID-19 positif atau ada Pasien Dalam Pengawasan.

2.7. Bahasa Pemrograman Python

Python merupakan bahasa pemrograman tingkat tinggi yang dibuat oleh Guido Van Rossum dan rilis pertama kali pada tahun 1991. Python merupakan bahasa pemrograman yang mudah dibaca, pemformatannya tidak berantakan secara visual, dan beberapa dari komponen maupun fungsi Python menggunakan Bahasa Inggris. Python menggunakan indentasi spasi dibandingkan menggunakan tanda kurung kurawal untuk membatasi setiap blok kode.

Python juga memiliki komunitas yang sangat luas dan besar. Python juga memiliki berbagai macam *library* yang *powerful*. Dalam dunia *data mining*, Python merupakan bahasa pemrograman yang populer digunakan karena *library* yang dimiliki Python untuk mengolah data sangat banyak. *Library* untuk mengolah data yang sering digunakan adalah Pandas, Matplotlib, dan NumPy.

2.7.1. Pandas

Pandas merupakan *library* yang cepat, bertenaga, fleksibel, dan mudah digunakan dalam mengolah serta memanipulasi data. Pandas dibangun di atas bahasa pemrograman Python. Pandas juga merupakan *library* yang *open source* sehingga pengguna tidak perlu membayar untuk dapat menggunakan *library* Pandas ini.

2.7.2. Matplotlib

Matplotlib merupakan *library* yang digunakan untuk membuat visualisasi statis, visualisasi beranimasi, dan visualisasi yang interaktif. Matplotlib dibangun di atas bahasa pemrograman Python. Matplotlib juga merupakan *library* yang *open source* sehingga pengguna tidak perlu membayar untuk dapat menggunakan *library* Matplotlib ini. Dengan Matplotlib kita dapat membuat visualisasi mulai dari *bar chart*, *line chart*, *pie chart*, maupun *scatter plot* untuk melihat hubungan antar data.

2.7.3. Tweepy

Tweepy merupakan *library* yang digunakan untuk mengakses Twitter API dalam bahasa pemrograman Python. Tweepy juga merupakan *library* penghubung Twitter API yang populer dan banyak digunakan oleh pelajar, peneliti, dan pengembang. Tweepy juga merupakan *library* yang *open source* sehingga pengguna tidak perlu membayar untuk dapat menggunakan *library* Tweepy ini. Salah satu kegunaan dari *library* ini adalah kita bisa mengambil data cuitan yang ada pada media sosial Twitter.

2.7.4. Natural Language Toolkit (NLTK)

Natural language toolkit (NLTK) merupakan *library* yang digunakan untuk mempelajari bahasa manusia dalam menerapkan bidang ilmu *natural language processing* (NLP). NLTK mengandung fungsi untuk mengolah data teks seperti *tokenizing*, *parsing*, *classification*, *stemming*, *tagging*, dan *semantic reasoning*. NLTK merupakan *library* yang dibangun di atas bahasa pemrograman Python dan bersifat *open source* sehingga tidak perlu membayarnya untuk dapat menggunakan *library* ini.

2.7.5. Sastrawi

Sama halnya dengan NLTK, sastrawi merupakan *library* yang digunakan untuk mempelajari bahasa manusia dalam menerapkan bidang ilmu *natural language processing* (NLP). Akan tetapi sastrawi merupakan *library* NLP yang lebih berfokus pada bahasa manusia Bahasa Indonesia. Sastrawi memiliki berbagai fungsi contohnya adalah *stemming* dan *filtering* dengan menggunakan Bahasa Indonesia.

2.7.6. WordCloud

WordCloud merupakan *library* yang digunakan untuk membuat visualisasi data yang merepresentasikan data teks dalam bentuk kata yang mewakili frekuensi kemunculan kata itu sendiri. *WordCloud* juga merupakan *library* yang dibangun di atas bahasa pemrograman Python dan bersifat *open*

source sehingga pengguna tidak perlu membayar untuk menggunakan *library* ini.

2.8. Indonesia Sentiment Lexicon (InSet)

Indonesia sentiment lexicon (InSet) merupakan *lexicon* Berbahasa Indonesia yang dibangun untuk mengidentifikasi opini dan mengkategorikannya dalam bentuk opini positif dan opini negatif dalam berbagai bentuk tanggapan seperti sebuah topik, acara, maupun produk. InSet dibangun secara manual dengan menimbang setiap kata dan ditingkatkan dengan menambahkan kata dan sinonim (Koto & Rahmaningtyas, 2017). InSet berisikan 3609 kata positif dan 6609 kata negatif dengan rentang *polarity score* dari -5 hingga +5.

2.9. Penelitian Terdahulu

Felix Fridom Mailo, Lutfan Lazuardi (2019) melakukan penelitian tentang Analisis Sentimen Data Twitter Menggunakan Metode Text Mining Tentang Masalah Obesitas di Indonesia. Pada penelitian ini dilakukan klasifikasi terhadap masalah obesitas di Indonesia menggunakan algoritma *naïve bayes*. Didapati hasil dari akurasi klasifikasi algoritma *naïve bayes* sebesar 94%.

Brata Mas Pintoko, Kemas Muslim L. (2018) melakukan penelitian tentang Analisis Sentimen Jasa Transportasi *Online* pada Twitter Menggunakan Metode Naïve Bayes Classifier. Pada penelitian ini dilakukan klasifikasi terhadap tanggapan jasa transportasi *online* menggunakan algoritma *naïve bayes*. Didapati hasil dari akurasi klasifikasi algoritma *naïve bayes* sebesar 86,80%.

Rani Puspita, Agus Widodo (2020) melakukan penelitian tentang Perbandingan Metode KNN, Decision Tree, dan Naïve Bayes Terhadap Analisis Sentimen Pengguna Layanan BPJS. Pada penelitian ini dilakukan klasifikasi terhadap pengguna layanan BPJS menggunakan algoritma KNN, *decision tree*, dan *naïve bayes*. Didapati hasil dari akurasi klasifikasi algoritma KNN sebesar 95.58%, algoritma *decision tree* sebesar 95.58%, dan algoritma *naïve bayes* sebesar 89.14%.

Desi Musfiroh, Ulfa Khaira, Pradita Eko Prasetyo Utomo, Tri Suratno (2021) melakukan penelitian tentang Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon. Pada penelitian ini

dilakukan proses analisis sentimen dengan menggunakan metode *lexicon based*. Dengan output berupa kategori sentimen positif, negatif, dan netral. Penelitian ini juga menghasilkan sebuah *wordcloud* untuk melihat kata terbanyak yang terdapat pada data yang dimiliki.

Tabel 2.1. Perbandingan Penelitian Terdahulu

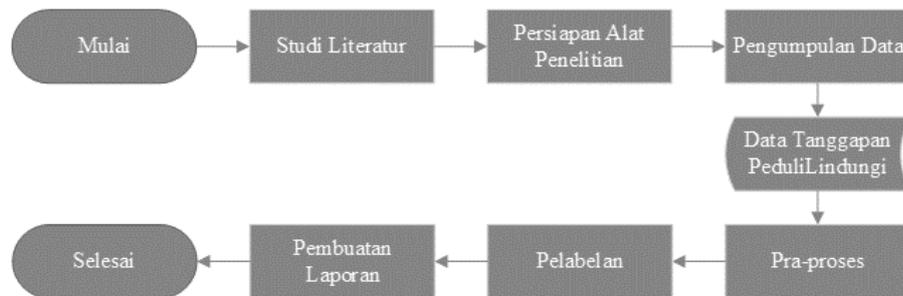
No	Penulis	Judul Penelitian	Objek	Metode Analisis Sentimen
1	Felix Fridom Mailo, Lutfan Lazuardi (2019)	Analisis Sentimen Data Twitter Menggunakan Metode Text Mining Tentang Masalah Obesitas di Indonesia	Masalah Obesitas di Indonesia	Naïve Bayes (<i>Machine Learning Based</i>)
2	Brata Mas Pintoko, Kemas Muslim L. (2018)	Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naïve Bayes Classifier	Jasa Transportasi Online	Naïve Bayes (<i>Machine Learning Based</i>)
3	Rani Puspita, Agus Widodo (2020)	Perbandingan Metode KNN, Decision Tree, dan Naïve Bayes Terhadap Analisis	Pengguna Layanan BPJS	KNN, Decision Tree, Naïve Bayes (<i>Machine Learning Based</i>)

		Sentimen Pengguna Layanan BPJS		
4	Desi Musfiroh, Ulfa Khaira, Pradita Eko Prasetyo Utomo, Tri Suratno (2021)	Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon	Perkuliahan Daring di Indonesia	<i>Lexicon dan Machine Learning (Hybrid Based)</i>

BAB 3 METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Tahapan penelitian dibentuk agar penelitian berjalan secara baik dan terstruktur. Penelitian meliputi beberapa tahapan yaitu studi literatur, pengumpulan data, pra-proses, pelabelan, dan pembuatan laporan. Tahapan penelitian ini ditunjukkan pada **Gambar 3.1**.



Gambar 3.1. Tahapan Penelitian

3.1.1. Studi Literatur

Tahapan ini dilakukan untuk memahami dan melakukan observasi terhadap teori dan penelitian yang telah dilakukan sebelumnya yang diambil dari jurnal ataupun tugas akhir. Tahapan ini dilakukan bertujuan untuk memahami teori secara mendetail terkait komponen dan lingkungan yang akan diteliti. Kemudian hasil dari studi literatur ini akan dituliskan ke dalam bab 2 sebagai acuan dasar teori bagi pembaca.

3.1.2. Pengumpulan Data

Tahapan ini peneliti melakukan pengumpulan data terkait aplikasi PeduliLindungi untuk dilakukan analisis sentimen yang terkandung dalam setiap tanggapannya. Tujuan dari pengumpulan data ini agar peneliti dapat menghasilkan sebuah informasi dan mengolah informasi tersebut lebih lanjut agar dapat menjawab rumusan masalah penelitian ini. Kemudian peneliti mengambil data dari tanggapan dari *platform social media* Twitter dengan menggunakan API Twitter dan bantuan *library* Tweepy.

3.1.3. Pra-Proses

Tahapan ini merupakan tahapan yang sangat penting. Dalam pemrosesan teks diperlukan pra-proses untuk mengolah beberapa bagian data agar dapat diproses oleh bahasa pemrograman dan dapat memberikan informasi yang diharapkan. Pada tahap ini terdiri dari beberapa bagian sebagai berikut:

1. *Cleaning Text*

Cleaning text merupakan proses untuk menghapus tanda *mention*, *hashtag*, *retweet*, *link*, angka, spasi, dan tanda baca yang ada pada seluruh tanggapan atau data yang dimiliki.

2. *Case Folding Text*

Case folding text merupakan proses untuk mengubah seluruh bentuk huruf ke dalam bentuk kecil atau *lower case*. Peneliti menggunakan fungsi *.lower()* untuk melakukan proses ini.

3. *Tokenizing Text*

Tokenizing text merupakan proses untuk memecah bentuk kalimat ke dalam bentuk kata per kata atau *token* dalam sebuah *list*. Peneliti menggunakan fungsi *word_tokenize* dari *library natural language toolkit*.

4. *Filtering Text*

Filtering text merupakan proses untuk menghapus kata yang tidak memiliki makna atau nilai sentimen. Peneliti menggunakan fungsi *StopWordRemoverFactory()* dari *library Sastrawi*.

5. *Stemming Text*

Stemming text merupakan proses untuk menghapus imbuhan dari seluruh kata. Peneliti menggunakan fungsi *StemmerFactory()* dari *library Sastrawi*.

3.1.4. Analisis Sentimen

Pada tahap ini peneliti melakukan pelabelan atau analisis sentimen dari seluruh tanggapan atau data yang telah diambil. Tahapan ini bertujuan untuk menghitung nilai *polarity* dari setiap kalimat atau tanggapan untuk ditentukan sentimen berdasarkan nilai *polarity* tersebut.

Proses perhitungan nilai *polarity* tersebut menggunakan metode *lexicon*. *Lexicon* itu sendiri merupakan sebuah kamus yang terdiri dari kata dan nilai *polarity* per kata. *Lexicon* terdiri dari *lexicon positive* dan *lexicon negative*. Kemudian dari data tanggapan yang kita miliki dilakukan pengecekan satu per satu, apabila sebuah kata pada data tanggapan tercantum dalam kamus *lexicon* maka nilai *polarity* akan ditambahkan sesuai dengan nilai *polarity* yang terkandung dalam *lexicon*. Contoh dari *lexicon* ditunjukkan pada **Gambar 3.2**.

(a) Lexicon Negatif		(b) Lexicon Positif	
mencederai	-4	hi	1
mengingkari	-4	dibantu	2
maaf	-3	makasih	4
mengkhianat	-4	halo	1
mencelakai	-5	thanks	3
mulu	-1	pengembangan	3
ngga	-2	diva	2
borong	-1	punya	3
lever	-2	tidak segan	2
kasian	-3	detailnya	1
gamau	-4	tak segan	2
doang	-1	aktivasi	2
pulas	-1	asih	3
abis	-2	kasih sayang	5
coba	-1	kekaguman	4
kangen	-3	kehangatan	4
kalau	-1	afeksi	2
maunya	-1	renjana	2
seandainya	-1	amor	2

Gambar 3.2. Contoh *Lexicon*

3.1.5. Pembuatan Laporan

Setelah proses analisis sentimen selesai, selanjutnya dilakukan proses pembuatan laporan untuk keperluan akademis dan pengembangan ilmu pengetahuan selanjutnya. Dalam laporan ini juga dituangkan detail dari proses penelitian tersebut sehingga pembaca dapat memahami apa saja yang terkandung dari penelitian ini.

3.2. Tempat dan Waktu Penelitian

Penelitian ini dilakukan secara mandiri menyesuaikan tempat peneliti dengan memanfaatkan jurnal referensi, buku dan fasilitas internet yang tersedia. Waktu yang digunakan untuk penelitian ini dimulai dari bulan September 2021 hingga bulan Februari 2021.

3.3. Instrumen Penelitian

Adapun instrumen penelitian yang digunakan yaitu:

- 1) Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut:
 - a. Intel Core-i5
 - b. Ram 4GB DDR3
 - c. HDD 500GB
- 2) Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:
 - a. Python 3
 - b. Jupyter Notebook
 - c. Microsoft Word

3.4. Persiapan Alat Penelitian

Pada tahap persiapan alat ini dilakukan pemasangan bahasa pemrograman Python, aplikasi Python *interpreter* Jupyter Notebook, dan *library* yang digunakan seperti Pandas, Matplotlib, Tweepy, Sastrawi, dan *Natural Language Toolkit*. Persiapan alat ini merupakan Langkah awal untuk melakukan pengumpulan data dan proses analisis sentimen.

3.5.1. Instalasi Python

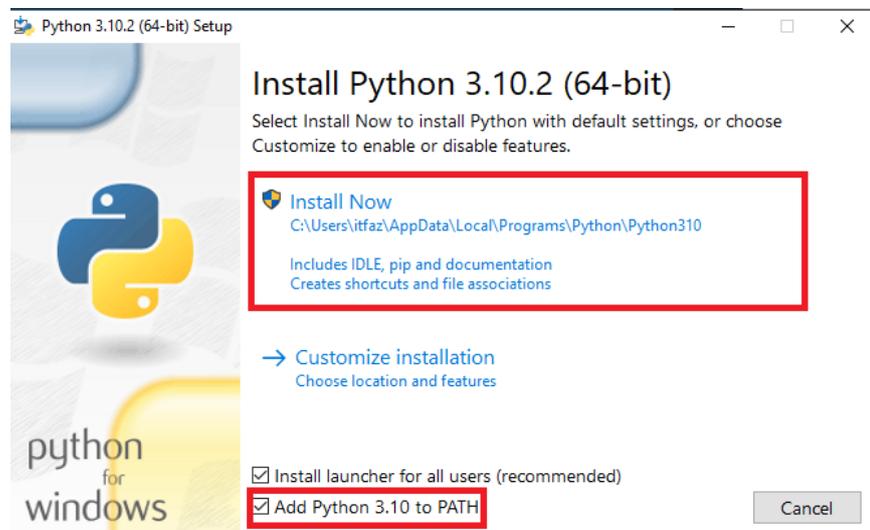
Instalasi Python dimulai dari mengunduh paket instalasi di *website* resmi Python yaitu Python.org. Pilih versi Python sesuai dengan yang diinginkan, peneliti merekomendasi versi Python terbaru yaitu Python versi 3. Proses instalasi Python ditunjukkan sebagai berikut:

1. Unduh paket instalasi Python di Python.org sesuai dengan versi yang akan digunakan. Proses ini ditunjukkan pada **Gambar 3.3.**



Gambar 3.3. Halaman Unduh Python

2. Aktifkan *option Add Python 3.10 to PATH* hingga muncul tanda ceklis kemudian klik *Install Now* hingga proses instalasi selesai. Proses ini ditunjukkan pada **Gambar 3.4.**



Gambar 3.4. Tampilan Awal Instalasi Python

3. Apabila sudah tampil tulisan *Setup was Successful* ataupun *Complete the Python Installer* artinya proses instalasi Python telah selesai dan siap untuk digunakan. Proses ini ditunjukkan pada **Gambar 3.5**.

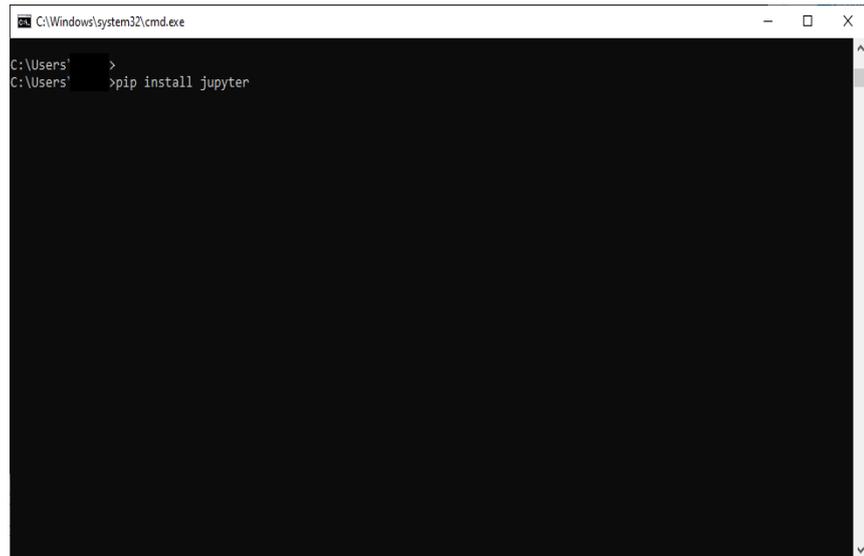


Gambar 3.5. Tahap Akhir Instalasi Python

3.5.2. Instalasi Jupyter Notebook

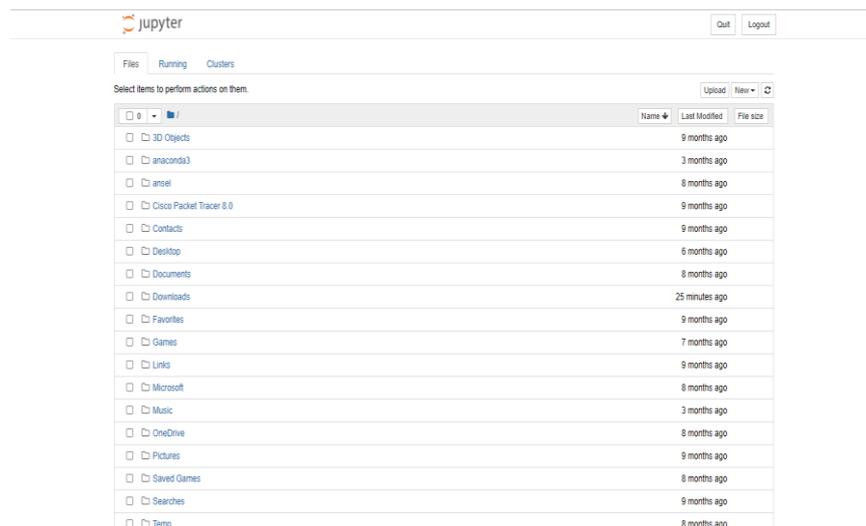
Jupyter notebook digunakan untuk menjalankan bahas pemrograman Python dalam bentuk *interpreter* artinya setiap baris kode dijalankan per *cell* tanpa harus melakukan proses *compile* terlebih dahulu. *Jupyter notebook* juga biasa disebut sebagai *tool* untuk menjalankan bahasa pemrograman Python secara interaktif. Proses instalasi *jupyter notebook* dapat ditunjukkan sebagai berikut:

1. Jalankan *windows command prompt* dengan menggunakan *shortcut* tombol *windows + R* kemudian ketik *cmd*. Setelah terbuka jendela seperti **Gambar 3.6** selanjutnya ketik perintah *pip install jupyter notebook*.



Gambar 3.6. Tampilan *Command Prompt* Awal Instalasi *Jupyter Notebook*

2. Setelah proses instalasi selesai, *jupyter notebook* dapat digunakan dengan cara mengetikkan perintah *jupyter notebook* pada jendela *command prompt* dan tampil jendela *jupyter notebook* seperti pada **Gambar 3.7.**

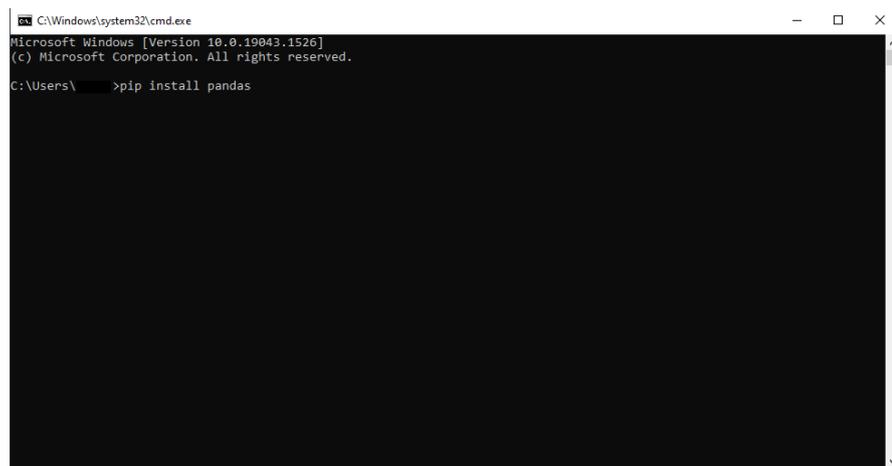


Gambar 3.7. Tampilan *Jupyter Notebook*

3.5.3. Instalasi Library

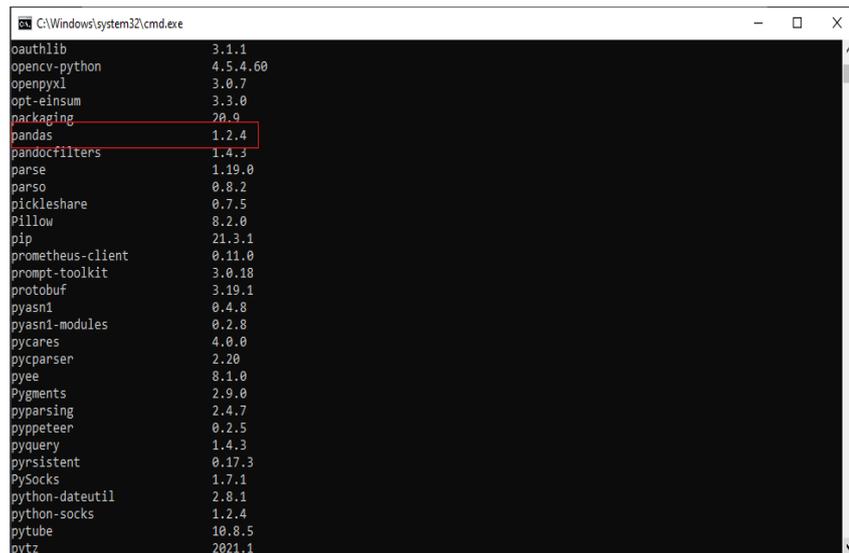
Library yang digunakan pada penelitian ini antara lain adalah Pandas, Matplotlib, Tweepy, Sastrawi, dan *Natural Language Toolkit*. Seluruh *library* tersebut perlu dipasang agar dapat digunakan pada bahasa pemrograman Python. Proses instalasi *library* dapat ditunjukkan sebagai berikut:

1. Jalankan *windows command prompt* dengan menggunakan *shortcut* tombol *windows + R* kemudian ketik *cmd*. Setelah terbuka jendela seperti **Gambar 3.8** selanjutnya ketik perintah *pip install <nama library>*. Sebagai contoh yang pertama adalah proses instalasi *library* Pandas dengan mengetikkan perintah *pip install pandas*.



Gambar 3.8. Tampilan *Command Prompt* Awal Instalasi *Library*

2. Tunggu proses instalasi hingga selesai sampai proses dalam *command prompt* tidak lagi jalan atau apabila sudah dapat mengetikkan perintah *pip list* seperti pada **Gambar 3.9**. Kemudian lakukan instalasi untuk *library* lainnya yang dibutuhkan dengan mengulang proses nomor 1.



```

C:\Windows\system32\cmd.exe
oauthlib 3.1.1
opencv-python 4.5.4.60
openpyxl 3.0.7
opt-einsum 3.3.0
packaging 20.9
pandas 1.2.4
pandocfilters 1.4.3
parse 1.19.0
parso 0.8.2
pickleshare 0.7.5
Pillow 8.2.0
pip 21.3.1
prometheus-client 0.11.0
prompt-toolkit 3.0.18
protobuf 3.19.1
pyasn1 0.4.8
pyasn1-modules 0.2.8
pycparser 4.0.0
pyparser 2.20
pyee 8.1.0
Pygments 2.9.0
pyparsing 2.4.7
pyppeteer 0.2.5
pyquery 1.4.3
pysistent 0.17.3
PySocks 1.7.1
python-dateutil 2.8.1
python-socks 1.2.4
pytube 10.8.5
pytz 2021.1

```

Gambar 3.9. Tampilan *Command Prompt* Selesai Instalasi *Library*

3.5. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data sekunder yang diambil menggunakan API Twitter dengan *library tweepy* yang tersedia pada Bahasa Pemrograman Python. *Query* yang digunakan untuk mengambil data penelitian ini adalah dengan kata kunci PeduliLindungi dengan Bahasa Indonesia. Data diambil pada bulan Februari 2022. Contoh dari data yang diambil menggunakan *library tweepy* dapat dilihat pada **Gambar 3.10**.

```

[{'text': 'RT @tamerlane9470: Kalo reuni 212 aja bisa dihujat gegara dianggap mengganggu kepentingan publik, gimana yak kalo de mo tolak wajib vaksin,...',
  'favourite_count': 0,
  'retweet_count': 12,
  'created_at': datetime.datetime(2021, 12, 1, 15, 6, 2, tzinfo=datetime.timezone.utc)},
 {'text': '@nazmiring Punya masnya tadi sempat muncul di pedulilindungi?\\n\\nSaya udah hubungi klinik tempat tes, mereka bilang sudah diinput. \\n\\nMana flightnya pagi bener saya mas.',
  'favourite_count': 0,
  'retweet_count': 0,
  'created_at': datetime.datetime(2021, 12, 1, 15, 1, 46, tzinfo=datetime.timezone.utc)},
 {'text': 'RT @polres_pemalang: Tips Menghindari Oversharing\\n\\nSumber : @CCICPolri\\n\\n@ariwibowo202\\n@DivHumas_Polri\\n@poldaja teng_\\n@CCICPolri\\n\\n#pemalang #...',
  'favourite_count': 0,
  'retweet_count': 4,
  'created_at': datetime.datetime(2021, 12, 1, 14, 58, 17, tzinfo=datetime.timezone.utc)},
 {'text': '@nazmiring Halo mas, sorry nimbrung.\\nSaya juga mengalami problem yang sama. Ga tau gimana check in ini.\\n\\nKalo tib a2 barcodenya muncul lagi di pedulilindungi, kabar2 ya mas.',
  'favourite_count': 0,
  'retweet_count': 0,
  'created_at': datetime.datetime(2021, 12, 1, 14, 54, 25, tzinfo=datetime.timezone.utc)},
 {'text': 'RT @polres_pemalang: Tips Menghindari Oversharing\\n\\nSumber : @CCICPolri\\n\\n@ariwibowo202\\n@DivHumas_Polri\\n@poldaja teng_\\n@CCICPolri\\n\\n#pemalang #...',
  'favourite_count': 0,
  'retweet_count': 4,
  'created_at': datetime.datetime(2021, 12, 1, 14, 54, tzinfo=datetime.timezone.utc)},
 {'text': '@nctjaefs tanda peduli tuh scan barcode di pedulilindungi :)',
  'favourite_count': 0,
  'retweet_count': 0,
  'created_at': datetime.datetime(2021, 12, 1, 14, 48, 42, tzinfo=datetime.timezone.utc)},

```

Gambar 3.10. Proses Pengumpulan Data

Proses pengambilan data ini dimulai dari mendeklarasikan kode otentikasi berupa *consumer_key*, *consumer_secret*, *access_token*, dan *access_token_secret*. Kode tersebut diambil pada portal *developer* API Twitter dan hanya dapat digunakan oleh pemilik akun Twitter tersebut saja dan dilarang untuk disebar luaskan. Setelah kode otentikasi sudah dideklarasikan, selanjutnya kode tersebut digunakan untuk otentikasi dalam proses menghubungkan program dengan API Twitter. Kemudian tentukan *query* atau kata kunci yang menjadi target untuk mengumpulkan data tanggapan yang memiliki kata kunci tersebut.

Terkumpul data sebanyak 1504 baris data yang mengandung kata kunci PeduliLindungi. Kemudian data tersebut dibuat dalam bentuk *dataframe* untuk dapat diproses lebih lanjut. Terdapat 4 kolom yaitu *text*, *favourite_count*, *retweet_count*, dan *created_at*. Sampel data tersebut ditunjukkan pada **Gambar 3.11**.

	<i>text</i>	<i>favourite_count</i>	<i>retweet_count</i>	<i>created_at</i>
0	RT @PemkotMalang: Wali Kota Sutiaji se usai men...	0	1	2022-02-07 07:30:16+00:00
1	Wali Kota Sutiaji se usai meninjau menyampaikan...	0	1	2022-02-07 07:29:34+00:00
2	RT @VVYND: Penasaran kenapa aplikasi PeduliLin...	0	10164	2022-02-07 07:26:54+00:00
3	Gegara Tes Covid-19 Naik jadi 500 Ribu per Har...	0	0	2022-02-07 07:23:38+00:00
4	RT @VVYND: Penasaran kenapa aplikasi PeduliLin...	0	10164	2022-02-07 07:08:48+00:00

Gambar 3.11. Sampel Data Penelitian

Kolom *text* merupakan kolom yang mengandung data tanggapan dari pengguna yang menyinggung kata PeduliLindungi. Kolom *favourite_count* merupakan kolom yang mengandung berapa kali tanggapan tersebut difavoritkan oleh pengguna lainnya. Kolom *retweet_count* merupakan kolom yang mengandung berapa kali tanggapan tersebut di-*retweet*. Kemudian kolom *created_at* merupakan kolom tanggal dari diunggahnya tanggapan tersebut ke dalam Twitter.

3.6. Pra-proses

Dari data yang telah peneliti kumpulkan sebelumnya, selanjutnya data tersebut harus diproses agar dapat menghasilkan sebuah informasi yang dituju. Karena data yang peneliti kumpulkan merupakan data yang tidak terstruktur, oleh karena itu sebelum dilakukan analisis sentimen, data tersebut perlu melewati tahap

pra-proses agar data tersebut menjadi data yang terstruktur dan dapat diolah oleh bahasa pemrograman. Pada tahap pra-proses ini setidaknya ada lima tahapan utama yaitu *cleaning text*, *case folding text*, *tokenizing text*, *filtering text*, dan *stemming text*.

Tabel 3.1. *Tweet* Tanggapan Aplikasi PeduliLindungi

No	<i>Tweet</i> Tanggapan
1	Tampilan Baru Aplikasi Peduli Lindungi Aplikasi Peduli Lindungi mengalami beberapa perubahan pada tampilan layar lebih sederhana. Aplikasi tidak lagi melacak terus-menerus, diketahui dari opsi ""Allow all the time"" sudah dihilangkan. https://t.co/PLxRhTbMuc
2	RT: Pelayanan satpas patuhi protokol kesehatan dan gunakan aplikasi peduli lindungi. #ntmcpolri #rtmcjatim #pedulilindungi #patuhiprokes @pidressitubondo @RTMCJatim https://t.co/38F478Geyv
3	@PLindungi min adikku udah vaksin tapi pas cek peduli lindungi keterangannya belum vaksin padahal 2 kali vaksin. gimana solusinya? masa harus vaksin 4 kali sih :(

Sumber : Hasil Rancangan 2022

Pada **Tabel 3.1** menunjukkan sampel dari *tweet* tanggapan mengenai aplikasi PeduliLindungi yang diperoleh menggunakan API *Tweepy*. Selanjutnya data *tweet* tersebut melewati tahap *preprocessing* yang meliputi *cleaning text*, *case folding text*, *tokenizing text*, *filtering text*, dan *stemming text*.

3.5.1. *Cleaning Text*

Pada proses *cleaning text*, seluruh simbol *mention* dan *hashtag* dihapus yang ditunjukkan dengan karakter yang diberi warna biru, seluruh tanda *retweet* dihapus yang ditunjukkan dengan karakter yang diberi warna hijau, seluruh *link* dihapus yang ditunjukkan dengan karakter yang diberi warna merah, seluruh angka dihapus yang ditunjukkan dengan karakter yang diberi warna kuning, dan seluruh tanda baca dihapus yang ditunjukkan dengan karakter yang diberi warna merah muda. Sintaks yang digunakan dalam proses ini ditunjukkan pada **Pseudo Code 3.1**.

Pseudo Code 3.1. Sintaks Proses *Cleaning Text*

```

DEFINE FUNCTION cleaningText(text):
  SET text TO re.sub(r'@[A-Za-z0-9]+', '', text)
  SET text TO re.sub(r'#[A-Za-z0-9]+', '', text)
  SET text TO re.sub(r'RT[\s]', '', text)
  SET text TO re.sub(r"http\S+", '', text)
  SET text TO re.sub(r'[0-9]+', '', text)
  SET text TO text.replace('\n', ' ')
  SET text TO text.translate(str.maketrans('', '', string.punctuation))
  SET text TO text.strip(' ')

```

Terlihat pada **Tabel 3.2** data tanggapan sebelum melalui proses *cleaning text* terdapat *symbol*, *link*, angka, tanda baca, RT dan *hashtag*. Dalam proses *cleaning text*, semuanya dibersihkan secara otomatis menggunakan *script* di setiap baris datanya kemudian hasil dari pada proses *cleaning text* terdapat pada kolom setelah melalui proses.

Tabel 3.2. *Tweet* Tanggapan Melalui Proses *Cleaning Text*

No	Sebelum Melalui Proses	Setelah Melalui Proses
1	Tampilan Baru Aplikasi Peduli Lindungi Aplikasi Peduli Lindungi mengalami beberapa perubahan pada tampilan layar lebih sederhana. Aplikasi tidak lagi melacak terus-menerus. diketahui dari opsi "Allow all the time" sudah dihilangkan. https://t.co/PLxRhTbMuc	Tampilan Baru Aplikasi Peduli Lindungi Aplikasi Peduli Lindungi mengalami beberapa perubahan pada tampilan layar lebih sederhana Aplikasi tidak lagi melacak terus menerus diketahui dari opsi <i>Allow all the time</i> sudah dihilangkan

No	Sebelum Melalui Proses	Setelah Melalui Proses
2	<p>RT: Pelayanan satpas patuhi protokol kesehatan dan gunakan aplikasi peduli lindungi #ntmcpolri #rtmcjatim #pedulilindungi #patuhiprokes @pidressitubondo @RTMCJatim https://t.co/38F478Geyv</p>	Pelayanan satpas patuhi protokol kesehatan dan gunakan aplikasi peduli lindungi
3	<p>@PLindungi min adikku udah vaksin tapi pas cek peduli lindungi keterangannya belum vaksin padahal 2 kali vaksin gimana solusinya? masa harus vaksin 4 kali sih :(</p>	min adikku udah vaksin tapi pas cek peduli lindungi keterangannya belum vaksin padahal kali vaksin gimana solusinya masa harus vaksin kali sih

Sumber : Hasil Rancangan 2022

3.5.2. Case Folding Text

Pada proses *case folding text*, seluruh huruf kapital dirubah bentuk menjadi huruf kecil atau *lower case*. Sintaks yang digunakan dalam proses ini ditunjukkan pada **Pseudo Code 3.2**. Kemudian hasil yang telah melalui proses *case folding text* ditunjukkan pada **Tabel 3.3**.

Pseudo Code 3.2. Sintaks Proses *Case Folding Text*

```

DEFINE FUNCTION casefoldingText(text):
    SET text TO text.lower()
    RETURN text

```

Terlihat pada **Tabel 3.3** ata tanggapan sebelum melalui proses *case folding text* terdapat huruf yang masih bercetak tebal ditandai dengan label berwarna kuning. Dalam proses *case folding text*, semuanya dilalukan perubahan ke dalam bentuk huruf kecil secara otomatis menggunakan *script*

di setiap baris datanya kemudian hasil dari pada proses *case folding text* terdapat pada kolom setelah melalui proses.

Tabel 3.3. *Tweet* Tanggapan Melalui Proses *Case Folding Text*

No	Sebelum Melalui Proses	Setelah Melalui Proses
1	Tampilan Baru Aplikasi Peduli Lindungi Aplikasi Peduli Lindungi mengalami beberapa perubahan pada tampilan layar lebih sederhana Aplikasi tidak lagi melacak terus menerus diketahui dari opsi <i>Allow all the time</i> sudah dihilangkan	tampilan baru aplikasi peduli lindungi aplikasi peduli lindungi mengalami beberapa perubahan pada tampilan layar lebih sederhana aplikasi tidak lagi melacak terus menerus diketahui dari opsi <i>allow all the time</i> sudah dihilangkan
2	Pelayanan satpas patuhi protokol kesehatan dan gunakan aplikasi peduli lindungi	pelayanan satpas patuhi protokol kesehatan dan gunakan aplikasi peduli lindungi
3	min adikku udah vaksin tapi pas cek peduli lindungi keterangannya belum vaksin padahal kali vaksin gimana solusinya masa harus vaksin kali sih	min adikku udah vaksin tapi pas cek peduli lindungi keterangannya belum vaksin padahal kali vaksin gimana solusinya masa harus vaksin kali sih

Sumber : Hasil Rancangan 2022

3.5.3. *Tokenizing Text*

Pada proses *tokenizing text*, seluruh kalimat dipecah menjadi sebuah kata yang dimasukkan kedalam sebuah *list*. Sintaks yang digunakan dalam proses ini ditunjukkan pada **Pseudo Code 3.3**. Kemudian hasil yang telah melalui proses *tokenizing text* ditunjukkan pada **Tabel 3.4**.

Pseudo Code 3.3. Sintaks Proses *Tokenizing Text*

```
DEFINE FUNCTION tokenizingText(text):
    SET text TO word_tokenize(text)
```

RETURN text

Terlihat pada **Tabel 3.4** data tanggapan sebelum melalui proses *tokenizing text* masih dalam bentuk kalimat. Dalam proses *tokenizing text*, semua kalimat dipecah menjadi dalam bentuk kata per kata secara otomatis menggunakan *script* di setiap baris datanya kemudian hasil dari pada proses *tokenizing text* terdapat pada kolom setelah melalui proses.

Tabel 3.4. *Tweet* Tanggapan Melalui Proses *Tokenizing Text*

No	Sebelum Melalui Proses	Setelah Melalui Proses
1	tampilan baru aplikasi peduli lindungi aplikasi peduli lindungi mengalami beberapa perubahan pada tampilan layar lebih sederhana aplikasi tidak lagi melacak terus menerus diketahui dari opsi <i>allow all the time</i> sudah dihilangkan	[tampilan, baru, aplikasi, peduli, melindungi, peduli, melindungi, mengalami, beberapa, perubahan, pada, tampilan, layer, lebih, sederhana, aplikasi, tidak, lagi, melacak, terus, menerus, diketahui, dari, opsi, <i>allow, all, the, time</i> , sudah, dihilangkan]
2	pelayanan satpas patuhi protokol kesehatan dan gunakan aplikasi peduli melindungi	[pelayanan, satpas, patuhi, protokol, kesehatan, dan, gunakan, aplikasi, peduli, melindungi]
3	min adikku udah vaksin tapi pas cek peduli melindungi keterangannya belum vaksin padahal kali vaksin gimana solusinya masa harus vaksin kali sih	[min, adikku, udah, vaksin, tapi, pas, cek, peduli, melindungi, keterangannya, belum, vaksin, padahal, kali, vaksin, gimana, solusinya, masa, harus, vaksin, kali, sih]

Sumber : Hasil Rancangan 2022

3.5.4. *Filtering Text*

Pada proses *filtering text*, seluruh kata yang tidak memiliki makna sentimen dihapus karena tidak berpengaruh terhadap perhitungan *polarity*

score. Sintaks yang digunakan dalam proses ini ditunjukkan pada **Pseudo Code 3.4**. Kemudian hasil yang telah melalui proses *filtering text* ditunjukkan pada **Tabel 3.5**.

Pseudo Code 3.4. Sintaks Proses *Filtering Text*

```

DEFINE FUNCTION filteringText(text):
  SET listStopwords TO set(stopwords.words('indonesian'))
  SET filtered TO []
  FOR txt IN text:
    IF txt not IN listStopwords:
      filtered.append(txt)
  SET text TO filtered
  RETURN text

```

Terlihat pada **Tabel 3.5** data tanggapan sebelum melalui proses *filtering text* kata yang tidak memiliki makna seperti beberapa, pada, terus, serta kata dan yang ditandai dengan *highlight* berwarna kuning. Dalam proses *filtering text*, semua kata yang tidak memiliki makna dibersihkan secara otomatis menggunakan *script* di setiap baris datanya kemudian hasil dari pada proses *filtering text* terdapat pada kolom setelah melalui proses.

Tabel 3.5. *Tweet* Tanggapan Melalui Proses *Filtering Text*

No	Sebelum Melalui Proses	Setelah Melalui Proses
1	[tampilan, baru, aplikasi, peduli, lindungi, peduli, lindungi, mengalami, beberapa, perubahan, pada, tampilan, layar, lebih, sederhana, aplikasi, tidak, lagi, melacak, terus, menerus, diketahui, dari, opsi,	[tampilan, baru, aplikasi, peduli, lindungi, peduli, lindungi, perubahan, tampilan, layar, lebih, sederhana, aplikasi, tidak, lagi, melacak, opsi, allow, all, the, time, sudah, dihilangkan]

No	Sebelum Melalui Proses	Setelah Melalui Proses
	<i>allow, all, the, time,</i> sudah, dihilangkan]	
2	[pelayanan, satpas, patuhi, protokol, kesehatan, dan , gunakan, aplikasi, peduli, lindungi]	[pelayanan, satpas, patuhi, protokol, kesehatan, dan, aplikasi, peduli, lindungi]
3	[min, adikku, udah, vaksin, tapi , pas, cek, peduli, lindungi, keterangannya, belum, vaksin, padahal, kali, vaksin, gimana, solusinya, masa, harus, vaksin, kali, sih]	[min, adikku, udah, vaksin, pas, cek, peduli, lindungi, keterangannya, belum, vaksin, padahal, kali, vaksin, gimana, solusinya, masa, harus, vaksin, kali, sih]

Sumber : Hasil Rancangan 2022

3.5.5. Stemming Text

Pada proses *stemming text*, seluruh imbuhan yang terdapat pada sebuah kata dihapus yang ditunjukkan dengan karakter yang diberi warna biru. Sintaks yang digunakan dalam proses ini ditunjukkan pada **Pseudo Code 3.5**. Kemudian hasil yang telah melalui proses *stemming text* ditunjukkan pada **Tabel 3.6**.

Pseudo Code 3.5. Sintaks Proses *Stemming Text*

```

DEFINE FUNCTION stemmingText(text):
    SET factory TO StemmerFactory()
    SET stemmer TO factory.create_stemmer()
    SET text TO [stemmer.stem(word) FOR word IN text]
    RETURN text

```

Terlihat pada **Tabel 3.6** data tanggapan sebelum melalui proses *stemming text* terdapat kata imbuhan seperti -an, per-, di-, -kan, -I, dan lain sebagainya seperti yang ditandai dengan *highlight* berwarna biru. Dalam

proses *stemming text*, seluruh kata imbuhan dibersihkan secara otomatis menggunakan *script* di setiap baris datanya kemudian hasil dari pada proses *stemming text* terdapat pada kolom setelah melalui proses.

Tabel 3.6. *Tweet* Tanggapan Melalui Proses *Stemming Text*

No	Sebelum Melalui Proses	Setelah Melalui Proses
1	[tampilan, baru, aplikasi, peduli, lindungi, peduli, lindungi, perubahan, tampilan, layar, lebih, sederhana, aplikasi, tidak, lagi, melacak, opsi, <i>allow</i> , <i>all</i> , <i>the</i> , <i>time</i> , sudah, dihilangkan]	[tampil, baru, aplikasi, peduli, lindungi, peduli, lindungi, ubah, tampil, layar, lebih, sederhana, aplikasi, tidak, lagi, melacak, opsi, <i>allow</i> , <i>all</i> , <i>the</i> , <i>time</i> , sudah, hilang]
2	[pelayanan, satpas, patuhi, protokol, kesehatan, dan, aplikasi, peduli, lindungi]	[layanan, satpas, patuh, protokol, sehat, dan, aplikasi, peduli, lindungi]
3	[min, adikku, udah, vaksin, pas, cek, peduli, lindungi, keterangannya, belum, vaksin, padahal, kali, vaksin, gimana, solusinya, masa, harus, vaksin, kali, sih]	[min, adik, udah, vaksin, pas, cek, peduli, lindungi, keterangan, belum, vaksin, padahal, kali, vaksin, gimana, solusi, masa, harus, vaksin, kali, sih]

Sumber : Hasil Rancangan 2022

Kemudian hasil dari tahapan praproses yang terdapat pada *tweet* setelah melalui proses *cleaning text*, *case folding text*, *tokenizing text*, *filtering text*, dan *stemming text* ditunjukkan pada **Tabel 3.7** tersebut telah memiliki struktur yang lebih baik, sehingga dapat diproses dan dihitung *polarity score* pada tahap analisis sentimen nantinya.

Tabel 3.7. *Tweet* Tanggapan Setelah Melewati Tahap *Preprocessing*

No	<i>Tweet</i> Tanggapan Terstruktur
1	[tampil, baru, aplikasi, peduli, lindungi, peduli, lindungi, ubah, tampil, layar, lebih, sederhana, aplikasi, tidak, lagi, melacak, opsi, <i>allow</i> , <i>all</i> , <i>the</i> , <i>time</i> , sudah, hilang]

No	<i>Tweet</i> Tanggapan Terstruktur
2	[layanan, satpas, patuh, protokol, sehat, dan, aplikasi, peduli, lindungi]
3	[min, adik, udah, vaksin, pas, cek, peduli, lindungi, keterangan, belum, vaksin, padahal, kali, vaksin, gimana, solusi, masa, harus, vaksin, kali, sih]

Sumber : Hasil Rancangan 2022

3.7. Perhitungan Analisis Sentimen

Dalam penelitian ini penulis melakukan pelabelan dengan menggunakan metode *lexicon based* dimana setiap katanya akan dihitung bobot sentimen yang dinamakan *polarity score*, berdasarkan *polarity score* tersebut penulis mendapatkan kategori tanggapan yang dibagi menjadi tiga kategori yaitu positif, negatif, dan netral. Sintaks yang digunakan dalam proses ini ditunjukkan pada **Pseudo Code 3.6**.

Pseudo Code 3.6. Sintaks Proses Pelabelan

```

SET lexicon_positive TO dict()
IMPORT csv
with open('positive.tsv') as csvfile:
    SET reader TO csv.reader(csvfile, delimiter='t')
    FOR row IN reader:
        IF 'word' not IN row and 'weight' not IN row:
            SET lexicon_positive[row[0]] TO int(row[1])

SET lexicon_negative TO dict()
IMPORT csv
with open('negative.tsv') as csvfile:
    SET reader TO csv.reader(csvfile, delimiter='t')
    FOR row IN reader:
        IF 'word' not IN row and 'weight' not IN row:
            SET lexicon_negative[row[0]] TO int(row[1])

DEFINE FUNCTION sentiment_analysis_lexicon_indonesia(text):
    SET score TO 0

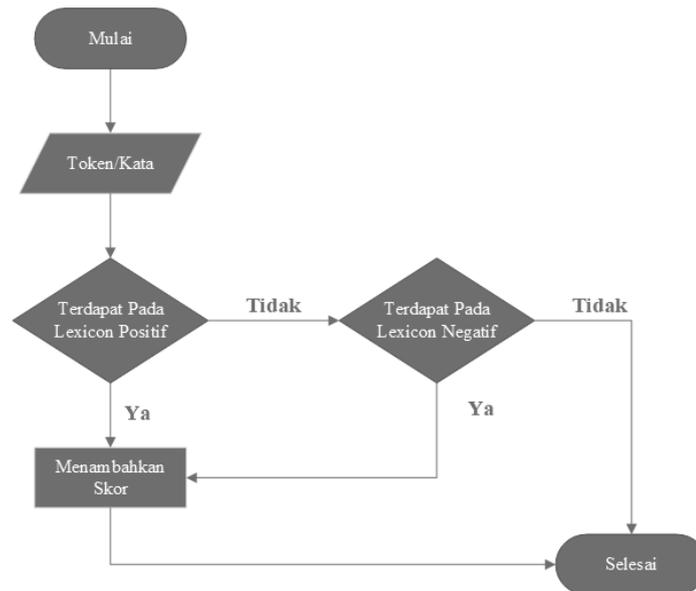
```

```
FOR word IN ast.literal_eval(text):
    IF (word IN lexicon_positive):
        SET score TO score + lexicon_positive[word]

FOR word IN ast.literal_eval(text):
    IF (word IN lexicon_negative):
        SET score TO score + lexicon_negative[word]

polarity=""
IF (score > 0):
    SET polarity TO 'positive'
ELSEIF (score < 0):
    SET polarity TO 'negative'
ELSE:
    SET polarity TO 'neutral'
RETURN score, polarity
```

Proses perhitungan *polarity score* bertujuan untuk menentukan sentimen dari setiap tanggapannya. Untuk menentukan sentimen, peneliti melakukan perhitungan *polarity score* dengan cara mengecek satu per satu setiap kata pada setiap tanggapan apakah kata tersebut ada pada daftar *lexicon* positif atau ada pada daftar *lexicon* negatif. Proses ini dapat ditunjukkan melalui *flowchart* pada **Gambar 3.12**.



Gambar 3.12. Alur Perhitungan *Polarity Score*

Dalam proses ini peneliti contohkan menggunakan tanggapan sebagai berikut :

[layanan, satpas, patuh, protokol, sehat, aplikasi]

polarity score > 0, maka sentimen positif

polarity score < 0, maka sentimen negatif

polarity score = 0, maka sentimen netral

Tabel 3.8. Perhitungan *Polarity Score*

No	Teks	<i>Lexicon Positif</i>	<i>Lexicon Negatif</i>	Skor
1	layanan	Tercantum	-	3
2	satpas	-	-	0
3	patuh	Tercantum	-	4
4	protokol	-	-	0
5	sehat	Tercantum	-	4
6	aplikasi	-	Tercantum	-4
Total				7 (Sentimen Positif)

Sumber : Hasil Rancangan 2022

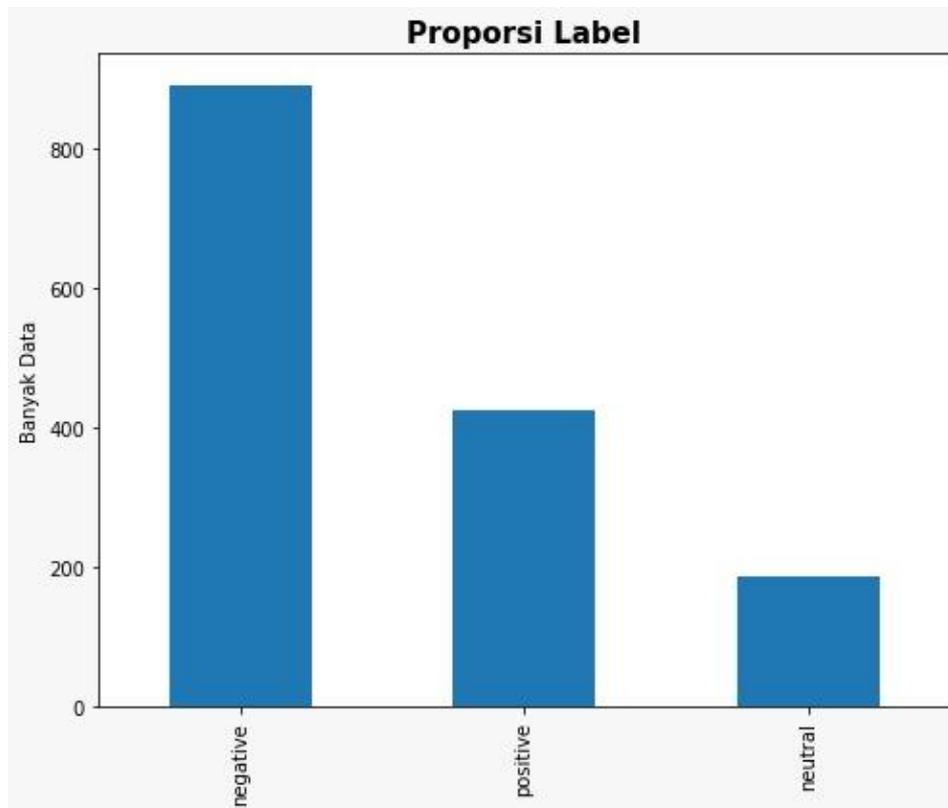
Berdasarkan **Tabel 3.8** bahwa contoh tanggapan diatas setelah melewati proses perhitungan *polarity score* menggunakan metode *lexicon* dengan total skor sebesar 7 maka tanggapan tersebut masuk ke dalam kategori positif. Apabila tanggapan memiliki skor yang bernilai 0 maka tanggapan tersebut masuk ke dalam kategori netral, dan apabila tanggapan memiliki skor minus maka tanggapan tersebut masuk ke dalam kategori negatif.

BAB 4

HASIL DAN PEMBAHASAN

4.1. Hasil Analisis

Analisis sentimen umumnya dikategorikan dalam 3 kelas yaitu *positive*, *negative*, dan *neutral*. Untuk menentukan kategori tersebut setiap tanggapan harus dihitung *polarity score* berdasarkan *lexicon*. Pada tahap sebelumnya peneliti telah melakukan proses perhitungan *polarity* terhadap data tanggapan yang diambil. Dari total data mentah sebanyak 2659 baris kemudian dilakukan penghapusan data yang duplikat sehingga menjadi 1504. Secara keseluruhan, gambaran mengenai tanggapan yang diperoleh berdasarkan masing-masing kategori ditunjukkan oleh **Gambar 4.1**.



Gambar 4.1. Jumlah Tanggapan Aplikasi PeduliLindungi

Berdasarkan **Tabel 4.1** diketahui bahwa dari total tanggapan yang diperoleh yaitu sebanyak 1504 tanggapan, sebanyak 426 tanggapan merupakan tanggapan positif yaitu tanggapan yang mendukung dan menilai positif terhadap aplikasi

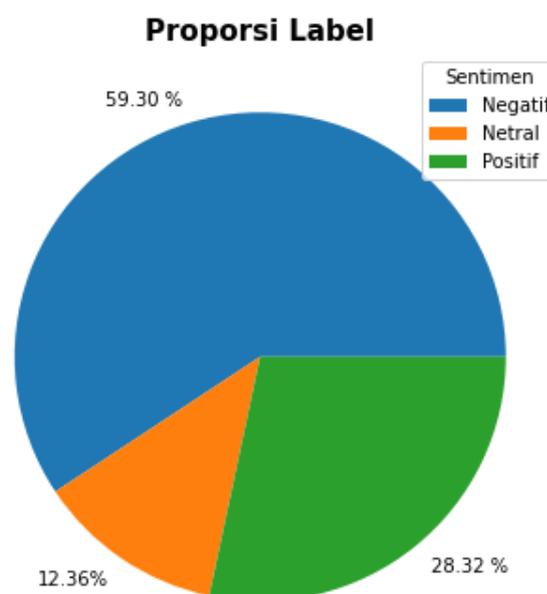
PeduliLindungi, sebanyak 892 tanggapan merupakan tanggapan negatif yaitu tanggapan yang tidak mendukung dan menilai negatif terhadap aplikasi PeduliLindungi, dan sebanyak 186 tanggapan merupakan tanggapan netral yaitu tanggapan yang menilai netral terhadap aplikasi PeduliLindungi. Secara garis besar jumlah tanggapan yang diperoleh dengan kategori negatif lebih banyak dibandingkan tanggapan yang diperoleh dengan kategori positif dan netral.

Tabel 4.1. Jumlah Tanggapan Aplikasi PeduliLindungi

No	Sentimen	Jumlah Data	Dalam Persentase
1	Negatif	892	59.30%
2	Positif	426	12.36%
3	Netral	186	28.32%

Sumber : Hasil Rancangan 2022

Berdasarkan **Gambar 4.2** bahwa tanggapan yang memiliki sentimen negatif memiliki proporsi sebesar 59.30% yang ditandai dengan bagian berwarna biru, tanggapan yang memiliki sentimen positif memiliki proporsi sebesar 28.32% yang ditandai dengan bagian berwarna hijau, dan tanggapan yang memiliki sentimen netral memiliki proporsi sebesar 12.36% yang ditandai dengan bagian berwarna kuning.



Gambar 4.2. Proporsi Label

4.2. Pembahasan

4.2.1. Pembahasan Analisis Sentimen

PeduliLindungi merupakan aplikasi *tracking* and *tracing* COVID-19 yang digunakan untuk membantu instansi pemerintah Indonesia terkait pelacakan untuk menghentikan penyebaran kasus COVID-19 di Indonesia itu sendiri. Akan tetapi dari banyaknya manfaat pada aplikasi tersebut, banyak juga tanggapan yang menuai kontroversi masyarakat terhadap aplikasi tersebut. Berdasarkan rumusan masalah yang menjadi pertanyaan dilaksanakannya penelitian yaitu ingin mengetahui sentimen masyarakat Indonesia yang menggunakan *platform* Twitter terhadap aplikasi PeduliLindungi.

Telah dilakukan penelitian terhadap data tanggapan aplikasi PeduliLindungi yang diambil dari *platform* media sosial Twitter pada bulan Februari 2022. Kemudian telah dilakukan proses analisis sentimen untuk mengkategorikan sentimen setiap kalimat atau tanggapan yang diperoleh.

Dari data tanggapan tersebut terdapat 426 data tanggapan yang bernilai positif, 892 data tanggapan yang bernilai negatif, dan 186 data tanggapan yang bernilai netral. Berdasarkan hasil penelitian tersebut didapati bahwa tanggapan masyarakat terhadap aplikasi PeduliLindungi adalah negatif ditandai dengan dominannya tanggapan yang bernilai negatif.



Gambar 4.3. WordCloud PeduliLindungi

Kemudian peneliti juga membuat *wordcloud* untuk mendapatkan kata terbanyak yang ada pada data tanggapan. Berdasarkan **Gambar 4.3** bahwa kata terbanyak yang muncul adalah PeduliLindungi dan diikuti beberapa kata lainnya seperti mengganggu, kepentingan, dianggap, dihujat, dan wajib. Berdasarkan beberapa kata yang sering muncul tersebut juga meyakinkan bahwa sentimen yang dimiliki pada data tanggapan PeduliLindungi memiliki sentimen yang negatif.

4.2.2. Permasalahan Analisis Sentimen

Dalam proses analisis sentimen dengan menggunakan metode *lexicon* ada masalah yang ditimbulkan. Mulai dari jumlah data yang didapati tidak terlalu banyak, banyaknya kalimat atau kata yang tidak baku atau tidak sesuai dengan KBBI, dan timbul kata yang tidak memiliki makna baru diluar dari *library* yang digunakan untuk melakukan proses *filtering*.

Data yang diambil pada bulan Februari 2022 hanya mendapatkan data sebanyak 2659 baris dari 15,7 juta pengguna *platform* Twitter di Indonesia. Oleh karena itu data tanggapan terkait aplikasi PeduliLindungi masih cenderung sedikit untuk mewakili sentimen masyarakat terhadap aplikasi PeduliLindungi itu sendiri.

Banyaknya kalimat dan kata yang tidak baku muncul pada data tanggapan. Contohnya adalah penggunaan seperti yang ditunjukkan pada **Tabel 4.2** yaitu kata ‘gk’ yang seharusnya ‘tidak’, kata ‘donlot’ yang seharusnya ‘unduh’, kata ‘kek’ yang seharusnya ‘seperti’ dan beberapa kata tidak baku lainnya. Hal ini dapat mempengaruhi perhitungan *polarity score* dari setiap tanggapan karena kata tersebut tidak terdapat pada *lexicon* positif maupun negatif sehingga kata tersebut dianggap netral atau 0 (tidak memiliki *polarity score*).

Tabel 4.2. Kata Tidak Baku

Bentuk Tidak Baku	Bentuk Baku
gk	Tidak
donlot	Unduh
kek	Seperti

Sumber : Hasil Rancangan 2022

Kemudian banyak kata tidak bermakna yang baru atau diluar dari *database library* Sastrawi. Ini dikarenakan topik yang diangkat merupakan topik yang baru. Banyak kalimat baru yang tidak bermakna seperti ‘vaksin’, ‘aplikasi’, ‘sertifikat’, dan lain sebagainya yang merupakan bagian dari proses *filtering*.

BAB 5

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian dapat diambil beberapa kesimpulan sebagai berikut :

1. Proses analisis sentimen dengan menggunakan metode *lexicon* berjalan dengan baik dan sesuai dengan target yaitu menghasilkan kategori tanggapan yang bernilai positif, negative, dan netral.
2. Sentimen dari data yang diperoleh menunjukkan hasil yang positif dengan proporsi label tanggapan positif sebanyak 892 data, label tanggapan negatif sebanyak 426 data, dan label tanggapan netral sebanyak 186 data.

5.2. Saran

Berdasarkan penelitian yang dilakukan, dapat ditarik saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Dikarenakan data aplikasi PeduliLindungi itu tergolong banyak yang tidak baku bentuk katanya, maka diharapkan untuk mengimplementasikan sebuah proses untuk mengubah tanggapan atau kalimat yang tidak baku menjadi baku.
2. Dikarenakan penelitian ini hanya menggunakan satu metode analisis sentimen, maka diharapkan untuk mencoba metode analisis sentimen lainnya.

DAFTAR PUSTAKA

- Aditama, K. (2020). Pemanfaatan Natural Language Processing Dan Pattern Matching Dalam Pembelajaran Melalui Guru Virtual. *Elkom*, 13(1), 121–133. <http://ejurnal.stekom.ac.id/index.php/home/page121>
- Ardiani, L., Sujaini, H., & Tursina, T. (2020). Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, 8(2), 44–51.
- Brahimi, B., Touahria, M., & 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>
- Huda, I. (2021). Implementasi Natural Language Processing (NLP) Untuk Aplikasi Pencarian Lokasi. *Jurnal Nasional Teknologi Terapan (JNTT)*, 3(2), 15. <https://doi.org/10.22146/jntt.35036>
- Kementerian Komunikasi dan Informatika. (2021). *Tentang Apa Itu PeduliLindungi?* <https://www.pedulilindungi.id/>
- Koto, F., & Rahmaningtyas, G. Y. (2017). Inset Lexicon: Evaluation of a word list for Indonesian sentiment analysis in microblogs. *International Conference on Asian Language Processing (IALP)*, 391–394. <https://doi.org/10.1109/IALP.2017.8300625>
- Lee, D. H., Yang, J. K., Lee, C. H., & Kim, K. J. (2019). A Data Driven Approach to Selection of Critical Process Steps in the Semiconductor Manufacturing Process Considering Missing and Imbalanced Data. *Journal of Manufacturing Systems*, 52(1), 146–156. <https://doi.org/10.1016/j.jmsy.2019.07.001>
- Mailo, F. F., & Lazuardi, L. (2019). Analisis Sentimen Data Twitter Menggunakan

- Metode Text Mining Tentang Masalah Obesitas di Indonesia. *Journal of Information Systems for Public Health*, 4(1), 28–36.
- Musfiroh, D., Khaira, U., Utomo, P. E. P., & Suratno, T. (2021). Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon. *Indonesian Journal of Machine Learning and Computer Science*, 1(1), 24–33.
- Pintoko, B. M., & L, K. M. (2018). Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naive Bayes Classifier. *E-Proceeding of Engineering*, 5(3), 8121–8130.
- Puspita, R., & Widodo, A. (2020). Perbandingan Metode KNN, Decision Tree, dan Naive Bayes Terhadap Analisis Sentimen Pengguna Layanan BPJS. *Jurnal Informatika Universitas Pamulang*, 5(4), 646–654.
- Roihan, A., Sunarya, P. A., & Rafika, A. S. (2020). Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 5(1), 75–82. <https://doi.org/10.31294/ijcit.v5i1.7951>
- Sikumbang, E. D. (2018). Penerapan Data Mining Penjualan Sepatu Menggunakan Metode Algoritma Apriori. *Jurnal Teknik Komputer*, 4(1), 6.
- Zoghi, Z., & Serpen, G. (2021). UNSW - NB15 Computer Security Dataset: Analysis through Visualization. *ArXiv, abs/2101.0*.

LAMPIRAN 1 : LEMBAR KONSULTASI

LAMPIRAN 2 : BARIS KODE**#General**

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

#TweetGather

```
import tweepy as tw
```

#Preprocessing & Sentiment Analysis

```
import re
import string
from nltk.tokenize import word_tokenize
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from wordcloud import WordCloud
```

#A1 -> Identitas Otentikasi ke API Twitter

```
consumer_key = 'aFAVbyJEL3wNyuIC1hIqlyPn2'
consumer_secret =
'53FZKXHusWYxZtAjm3L9UiTD5fXf0BqtP6dHGzhyNvhCkOTEZV'
access_token = '1105520701174308864-A6IgsqwrXrpIOhvZUzCSDSnPU0ncIp'
access_token_secret =
'iPvLe2nigWZPgxW1y444W4vVsMVVUPAr911BJYuEdahEg'
```

#A2 -> Proses Otentikasi ke API Twitter

```

auth = tw.OAuthHandler(consumer_key, consumer_secret) #Set User
auth.set_access_token(access_token, access_token_secret) #Set User
api = tw.API(auth, wait_on_rate_limit=True) #Connect ke API Twitter

```

#A3 -> Set Query

```

tweet_list = tw.Cursor(api.search_tweets, q="pedulilindungi",
tweet_mode='extended', lang='id').items()

```

#A4 -> Proses Pengambilan Data Tweet

```

output = []

```

```

for tweet in tweet_list:

```

```

    text = tweet._json["full_text"]

```

```

    print(text)

```

```

    favourite_count = tweet.favorite_count

```

```

    retweet_count = tweet.retweet_count

```

```

    created_at = tweet.created_at

```

```

    line = {'text' : text, 'favourite_count' : favourite_count, 'retweet_count' :
retweet_count, 'created_at' : created_at}

```

```

    output.append(line)

```

#A5 -> Pengecekan Data Tweet

```

Output

```

#A6 -> Perubahan dari List ke DataFrame agar dapat diproses lebih lanjut

```

df = pd.DataFrame(output)

```

```

df.to_csv(r'peduli_output_data_contoh.csv', index = False, header = True)

```

```

df = pd.read_csv('peduli_output_data_contoh.csv')

```

df

```
df = pd.read_csv('peduli.csv')
```

```
df.head(5)
```

#A7 -> Pilih Kolom Tweet Saja

```
tweets = df[['description']]
```

#A8 -> Tahap Pembuatan Fungsi Preprocessing

```
def cleaningText(text):
```

```
    text = re.sub(r'@[A-Za-z0-9]+', '', text) # remove mentions
```

```
    text = re.sub(r'#[A-Za-z0-9]+', '', text) # remove hashtag
```

```
    text = re.sub(r'RT[\s]', '', text) # remove RT
```

```
    text = re.sub(r"http\S+", '', text) # remove link
```

```
    text = re.sub(r'[0-9]+', '', text) # remove numbers
```

```
    text = text.replace('\n', ' ') # replace new line into space
```

```
    text = text.translate(str.maketrans("", "", string.punctuation)) # remove all
punctuations
```

```
    text = text.strip(' ') # remove characters space from both left and right text
```

```
    return text
```

```
def casefoldingText(text): # Converting all the characters in a text into lower case
```

```
    text = text.lower()
```

```
    return text
```

```
def tokenizingText(text): # Tokenizing or splitting a string, text into a list of tokens
```

```
    text = word_tokenize(text)
```

```
    return text
```

```
def filteringText(text): # Remove stopwords in a text
    listStopwords = set(stopwords.words('indonesian'))
    filtered = []
    for txt in text:
        if txt not in listStopwords:
            filtered.append(txt)
    text = filtered
    return text
```

```
def stemmingText(text): # Reducing a word to its word stem that affixes to suffixes
and prefixes or to the roots of words
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    text = [stemmer.stem(word) for word in text]
    return text
```

#A9 -> Proses Pengaplikasian Fungsi Preprocessing

```
tweets['text_clean'] = tweets['description'].apply(cleaningText)
tweets['text_clean'] = tweets['text_clean'].apply(casefoldingText)
tweets.drop(['description'], axis = 1, inplace = True)

tweets['text_preprocessed'] = tweets['text_clean'].apply(tokenizingText)
tweets['text_preprocessed'] = tweets['text_preprocessed'].apply(filteringText)
tweets['text_preprocessed'] = tweets['text_preprocessed'].apply(stemmingText)

#drop duplicates/spams tweets
tweets.drop_duplicates(subset = 'text_clean', inplace = True)
```

#A10 -> Cek Hasil Preprocessing

Tweets

#A11 -> Proses Analisis Sentimen Dengan Lexicon

```
lexicon_positive = dict()
```

```
import csv
```

```
with open('positive.tsv', 'r') as csvfile:
```

```
    reader = csv.reader(csvfile, delimiter='\t')
```

```
    for row in reader:
```

```
        if "word" not in row and "weight" not in row:
```

```
            lexicon_positive[row[0]] = int(row[1])
```

```
lexicon_negative = dict()
```

```
import csv
```

```
with open('negative.tsv', 'r') as csvfile:
```

```
    reader = csv.reader(csvfile, delimiter='\t')
```

```
    for row in reader:
```

```
        if "word" not in row and "weight" not in row:
```

```
            lexicon_negative[row[0]] = int(row[1])
```

```
# Function to determine sentiment polarity of tweets
```

```
def sentiment_analysis_lexicon_indonesia(text):
```

```
    #for word in text:
```

```
        score = 0
```

```
        for word in text:
```

```
            if (word in lexicon_positive):
```

```
                score = score + lexicon_positive[word]
```

```
        for word in text:
```

```
    if (word in lexicon_negative):
        score = score + lexicon_negative[word]
polarity=""
if (score > 0):
    polarity = 'positive'
elif (score < 0):
    polarity = 'negative'
else:
    polarity = 'neutral'
return score, polarity
```

#A12 -> Merapihkan DataFrame Setelah Analisis Sentimen

```
results = tweets['text_preprocessed'].apply(sentiment_analysis_lexicon_indonesia)
results = list(zip(*results))
tweets['polarity_score'] = results[0]
tweets['polarity'] = results[1]
print(tweets['polarity'].value_counts())
tweets
tweets.to_csv(r'peduli_contoh_labeled.csv', index = False, header =
True,index_label=None)
```