

BAB III

METODOLOGI PENELITIAN

3.1 Gambaran Umum Objek Penelitian

Objek pada penelitian ini adalah ulasan pengguna pada aplikasi PeduliLindungi. PeduliLindungi merupakan aplikasi yang dikembangkan untuk membantu instansi pemerintah dalam pelacakan guna mencegah penyebaran COVID-19 [4]. Sejak September 2021, PeduliLindungi merupakan aplikasi yang wajib digunakan pada beberapa aktivitas dan akses publik. Aplikasi tersebut juga memiliki banyak penambahan fitur atau menu di antaranya akses sertifikat vaksin, *scan QR* untuk *check-in* atau *check-out* tempat, dokumen perjalanan, e-HAC, dan masih banyak lagi.

Dari ulasan pengguna akan diperoleh kelas sentimen dari setiap aspek yang diulas. Adapun ulasan pengguna terhadap aplikasi PeduliLindungi pada penelitian ini akan diteliti dari versi aplikasi yang diulas. Penelitian ini menggunakan 2 jenis ulasan versi aplikasi yaitu versi 4.0.2 dan 4.0.5. Hal ini bertujuan untuk dapat membandingkan sentimen pada aspek PeduliLindungi dari 2 versi tersebut, apakah terdapat perubahan sentimen pada aspek aplikasi terhadap pembaruan yang dilakukan. Adapun pemilihan versi aplikasi didasarkan pada jumlah data ulasan yang memadai untuk penelitian ini, dan menyesuaikan periode pengambilan data.

3.1.1 PeduliLindungi Versi 4.0.2

PeduliLindungi versi 4.0.2 diperbarui pada tanggal 19 Oktober 2021. Berikut daftar dari menu yang dapat diakses oleh pengguna pada versi ini:

a. *Vaccine Certificate*

Menu ini berguna agar pengguna dapat mengakses sertifikat vaksin. Adapun sertifikat vaksin dapat muncul jika pengguna sudah melakukan vaksinasi sebelumnya.

b. *Covid-19 Test Results*

Menu ini dipakai untuk mengakses hasil tes COVID-19 yang dilakukan pada laboratorium yang berafiliasi dengan Kementerian Kesehatan.

c. *E-Hac*

E-Hac (*Electrononic Health Card*) merupakan sistem yang berfungsi untuk memantau perjalanan jarak jauh pengguna baik domestik maupun internasional dengan menggunakan kapal, pesawat, atau bus. Pada PeduliLindungi, pengguna dapat melakukan pembuatan e-Hac serta menampilkan e-Hac yang sudah dibuat.

d. *Scan QR Code*

Menu ini berfungsi untuk memindai *barcode* sebagai konfirmasi *check-in* ke tempat publik yang menyediakan *barcode*. Setelahnya pengguna dapat melihat kapasitas pengunjung yang berada di tempat tersebut. Dalam melakukan *check-out* pengguna perlu memindai kembali atau dapat langsung melakukan *check-out* dengan menekan tombol pada aplikasi.

e. *Check-In History*

Menu ini berfungsi untuk menampilkan riwayat *check-in* yang dilakukan oleh pengguna selama dua minggu terakhir.

f. *Travel Regulations*

Menu ini berisikan informasi dan regulasi dalam melakukan perjalanan.

g. *Telemedicine*

Menu ini menjembatani pengguna dengan berbagai *platform* yang dapat membantu dalam pemeriksaan kesehatan mandiri, dan juga *platform* kesehatan lain untuk melakukan konsultasi dengan dokter spesialis.

h. *Healthcare Facility*

Menu ini dapat diakses jika pengguna sudah mengaktifkan lokasi pada perangkat untuk menampilkan fasilitas kesehatan yang berada di sekitar wilayah pengguna berada.

i. *COVID-19 Statistic*

Menu ini dapat diakses jika pengguna mengaktifkan lokasi pada perangkat untuk menampilkan statistik status COVID-19 di wilayah pengguna

berada. Adapun PeduliLindungi juga akan memberikan notifikasi atau peringatan jika pengguna berpergian atau berada di yang berzona resiko.

j. *Get Vaccine*

Menu ini digunakan untuk melakukan pendaftaran vaksin dan pengguna perlu mengisi formulir yang disediakan.

k. *Account*

Menu ini menampilkan beberapa tombol yang mengarah pada menu yaitu perubahan akun, sertifikat vaksin, QR code, pengecekan status vaksinasi dan hasil dari tes COVID, FAQ, tentang PeduliLindungi, dan preferensi bahasa pada aplikasi.

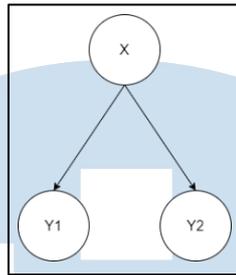
3.1.2 PeduliLindungi Versi 4.0.5

PeduliLindungi versi 4.0.5 diperbarui pada tanggal 19 November 2021. Terdapat beberapa hal yang diperbarui pada versi ini yang dicantumkan pada halaman PeduliLindungi di Google Play di antaranya:

- a. Perubahan pada tampilan (UI/UX) aplikasi.
- b. Penambahan opsi bahasa Mandarin, Jepang, Rusia, Korea, dan Spanyol.
- c. Perbaikan alur dari menu E-Hac.
- d. Penambahan CAPTCHA dalam melakukan klaim sertifikat secara pribadi.
- e. Penambahan FAQ mengenai status warna zonasi.
- f. Pembasmian *bugs (error)* pada aplikasi.

3.2 Variabel Penelitian

Variabel penelitian dibagi menjadi dua yaitu variabel independen dan variabel dependen. Pada penelitian ini, terdapat 1 variabel independen dan 2 variabel dependen yang dapat dilihat pada Gambar 3.1. Variabel independen pada penelitian ini adalah ulasan aplikasi PeduliLindungi (X), sedangkan variabel dependennya adalah aspek (Y1) dan sentimen (Y2) dari ulasan.



Gambar 3.1 Variabel Penelitian

3.3 Metode Penelitian

Metode penelitian dibagi menjadi metode penyelesaian dan alur penelitian sebagai berikut:

3.2.1 Metode Penyelesaian

Metode yang dipakai untuk melakukan *Aspect-Based Sentiment Analysis* (ABSA) pada penelitian ini adalah metode klasifikasi aspek dan sentimen. Survei [39] menyebutkan bahwa model *machine learning* dan/atau *deep learning* yang banyak dipakai dalam melakukan ABSA dan memiliki performa yang menjanjikan di antaranya *Convolutional Neural Network* (CNN), *Support Vector Machine* (SVM), dan *Naïve Bayes*. Tabel 2.1 juga menunjukkan bahwa ketiga model algoritma tersebut sering dipakai dalam tugas ABSA pada penelitian terdahulu. Berikut tabel perbandingan ketiga model klasifikasi tersebut:

Tabel 3.1 Perbandingan Model Klasifikasi CNN, SVM, dan *Naïve Bayes*

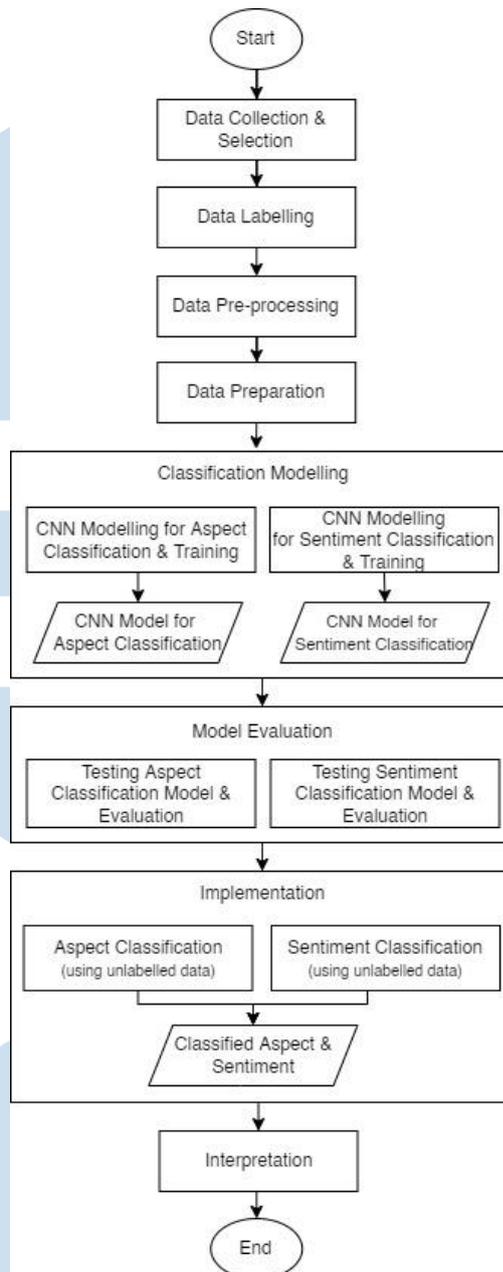
Kategori	CNN	SVM	<i>Naïve Bayes</i>
<i>Use-case</i> umum	CNN adalah pengklasifikasi non-linear yang baik bekerja pada tugas pengolahan citra dan mulai banyak dipakai dalam tugas NLP karena kemampuan dalam mengekstraksi fitur di dalam teks.	SVM adalah pengklasifikasi linear yang baik bekerja pada masalah klasifikasi pada umumnya, namun juga dipertimbangkan sebagai pengklasifikasi non-linear dengan memanfaatkan fungsi <i>kernel</i> .	<i>Naïve Bayes</i> adalah pengklasifikasi statistik yang berasumsi atas independensi setiap pasangan fitur untuk memprediksi probabilitas suatu kelas.
Kemampuan konfigurabilitas	CNN dapat disesuaikan (peningkatan atau pengurangan) kompleksitas modelnya dengan penambahan	SVM tidak memiliki kemampuan untuk menyesuaikan model terhadap kompleksitas.	<i>Naïve Bayes</i> tidak memiliki kemampuan untuk menyesuaikan model

Kategori	CNN	SVM	Naïve Bayes
	<i>layer</i> dan pemilihan parameter pada <i>layer</i> .		terhadap kompleksitas
Kemampuan kerja pada ukuran data yang besar	CNN baik digunakan pada data masukan yang berukuran besar, karena membutuhkan pelatihan pada banyak data.	SVM kurang dapat bekerja baik pada data yang berukuran besar dan memiliki banyak <i>noise</i> .	Naïve Bayes kurang dapat bekerja baik pada data yang berukuran besar dan memiliki banyak <i>noise</i> .
<i>Running time</i> pelatihan atau eksekusi model	CNN membutuhkan waktu yang lama dalam pelatihan atau eksekusi model karena kompleksitas modelnya.	Eksekusi dan pelatihan model SVM dapat dilakukan dengan cepat pada data yang kecil, namun akan meningkat seiring banyaknya jumlah data yang dipakai.	Eksekusi maupun pelatihan model Naïve Bayes dapat dilakukan dengan cepat.
Proses pembelajaran klasifikasi	Melalui proses iterasi yang dilakukan pembaruan <i>weight</i> dan <i>bias</i> pada tiap <i>neuron</i> hingga didapatkan nilai performa terbaik.	Melalui proses pencarian <i>hyperplane</i> terbaik dalam ruang dimensi untuk mengklasifikasikan titik data.	Melalui proses menemukan probabilitas suatu peristiwa terjadi dengan kemungkinan peristiwa sama yang telah terjadi sebelumnya.

Berdasarkan Tabel 3.1, CNN akan dipakai sebagai model utama pada penelitian ini untuk melakukan klasifikasi teks pada tugas ABSA dengan mempertimbangkan kemampuan CNN dalam penyesuaian kompleksitas model, ekstraksi fitur dan mendeteksi fitur penting, kemampuan pengolahan untuk *dataset* yang besar, dan proses pembelajaran iterasi yang memungkinkan pencapaian performa hingga yang paling baik. Di samping itu, juga akan dilakukan perbandingan performa klasifikasi dengan model SVM maupun Naïve Bayes.

3.2.2 Alur Penelitian

Alur penelitian yang digunakan untuk pada penelitian ini merupakan adaptasi dari jurnal penelitian [9], [10] dengan sejumlah penyesuaian. Berikut *flowchart* yang menggambarkan alur penelitian ini:



Gambar 3.2 Alur Penelitian

Berdasarkan Gambar 3.2, tahapan beserta penjelasan yang dilakukan pada penelitian ini adalah sebagai berikut:

1. *Data Collection and Selection*

Sebagai tahap awal, dilakukan pengumpulan data ulasan aplikasi PeduliLindungi yang berasal dari Google Play Store. Pengumpulan data menggunakan *library google_play_scraper* yang ada pada Python.

Setelah *scraping* data selesai, dilakukan pemilihan data sesuai versi yang akan diteliti yaitu ulasan aplikasi PeduliLindungi pada versi 4.0.2 dan 4.0.5. Ulasan aplikasi versi 4.0.2 akan dilakukan pelabelan data dan dipakai sebagai data *train*, *validation*, dan *test* pada model, sedangkan ulasan aplikasi versi 4.0.5 akan dipakai sebagai *unlabelled* data, yang akan dilakukan klasifikasi menggunakan model yang dibangun. Adapun pemilihan ulasan versi 4.0.2 dan 4.0.5 didasarkan pada kesamaan fitur dan menu yang dimiliki sehingga aspek yang diteliti tidak akan mengalami perubahan.

2. Data Labelling

Ulasan aplikasi PeduliLindungi versi 4.0.2 dilakukan pelabelan pada kolom aspek dan sentimen secara manual berdasarkan teks ulasannya. Adapun pelabelan dilakukan oleh 3 orang mahasiswa yang aktif menggunakan aplikasi PeduliLindungi dan juga diberikan kamus *labelling* yang berisikan kecenderungan kata-kata pada setiap kategori sentimen dan aspek dalam penelitian ini. Penentuan label akan mengikuti suara terbanyak dari tiap ulasannya.

Pemilihan kategori aspek sangatlah penting dalam proses pelabelan. Untuk mengatasi ini, penelitian ini mengikuti serangkaian kategori aspek sesuai dengan standar tertulis oleh Android di antaranya *Visual Experience* (UI/UX), *Functionality*, *Performance and Stability*, serta *Privacy and Security* [40]. Kategori aspek ini juga dipakai oleh penelitian [41] di mana kategori aspek yang digunakan dengan sedikit penyesuaian adalah *User Interface*, *User Experience*, *Functionality and Performance*, *Security*, dan *Support and Updates*.

Untuk menyesuaikan dengan kebutuhan penelitian, poin *Functionality* akan dikembangkan menjadi fitur atau menu yang menjadi aktivitas utama pada aplikasi PeduliLindungi. Dengan itu, kolom aspek memiliki 8 target label yang dapat dilihat pada Tabel 3.2. Dan pada kolom sentimen, terdapat

2 target label yaitu sentimen positif dan negatif yang dapat dilihat pada Tabel 3.3.

Tabel 3.2 Kategori Aspek

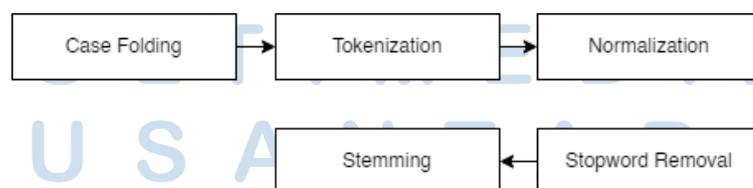
No	Aspek	Keterangan
1	<i>Visual experience (UI/UX)</i>	Berkaitan dengan tampilan atau visual aplikasi, interaksi atau <i>journey</i> pengguna dengan aplikasi, aksesibilitas konten, navigasi, serta notifikasi.
2	<i>Functionality – Scan QR, Check-in/out</i>	Berkaitan dengan fungsi untuk melakukan <i>check-in</i> atau <i>check-out</i> tempat menggunakan <i>scan QR</i> , serta riwayat dari <i>check-in</i> . Selain itu juga berkenaan dengan fungsi pendeteksian lokasi serta keakuratan dalam pendeteksian dalam melakukan fungsi scan.
3	<i>Functionality – Sertifikat Vaksin</i>	Berkaitan dengan fungsi yang ada pada menu sertifikat vaksin serta status vaksinasi pengguna.
4	<i>Functionality – e-Hac</i>	Berkaitan dengan fungsi yang ada pada menu e-Hac, seperti pembuatan dokumen perjalanan, melihat dokumen, serta akurasi dari dokumen yang dibuat dan ditampilkan.
5	<i>Functionality – Test COVID-19</i>	Berkaitan dengan fungsi yang ada pada menu COVID-19 <i>test results</i> , dalam menampilkan hasil tes COVID-19.
6	<i>Functionality – Register / Login</i>	Berkaitan dengan kemampuan aplikasi dalam menjalankan fungsi pendaftaran, maupun login ke dalam aplikasi.
7	<i>Performance and stability</i>	Berkaitan dengan stabilitas (apakah aplikasi mengalami <i>crash</i> , merespon dengan baik), performa aplikasi, serta penggunaan baterai.
8	<i>Privacy, Data, and Security</i>	Berkaitan dengan akses atau <i>permission</i> untuk mendukung jalannya aplikasi, seperti <i>permission</i> lokasi, pengaksesan data sensitif, penyimpanan data, serta kemampuan dalam menampilkan data yang sesuai dan terpercaya.

Tabel 3.3 Kategori Sentimen

No	Sentimen	Keterangan
1	Positif	Ulasan yang diberikan mengandung kata-kata baik, emosi positif dan mendukung baik secara implisit maupun eksplisit.
2	Negatif	Ulasan yang diberikan mengandung kata-kata buruk, emosi negatif dan tidak mendukung baik secara implisit maupun eksplisit.

3. Data Pre-processing

Pada tahap ini, data ulasan aplikasi pada versi 4.0.2 dan 4.0.5 dilakukan pembersihan dan perubahan data teks yang berformat tidak terstruktur menjadi terstruktur. Gambar 3.3 menunjukkan alur pada tahap ini.



Gambar 3.3 Alur Tahapan *Data Pre-processing*

a. *Case Folding*

Seluruh teks pada ulasan aplikasi akan diubah ke huruf yang sama yaitu huruf kecil, menggunakan fungsi *lower()* pada Python. Selain itu juga dilakukan penghapusan angka, *whitespaces*, tanda baca, dan *single* karakter menggunakan modul yang tersedia pada Python.

b. *Tokenization*

Dalam pemecahan kalimat menjadi kata, dilakukan proses tokenisasi menggunakan fungsi *split()* pada Python.

c. *Normalization*

Teks ulasan yang sudah ditokenisasi kemudian dilakukan normalisasi yaitu pengubahan pengubahan kata *slang* atau bahasa gaul, dan kata-kata singkat menjadi kata baku sesuai dengan KBBI. Pengubahan kata *slang* tersebut menggunakan *dataset* kamus berisikan 1.500 kata.

d. *Stopword removal*

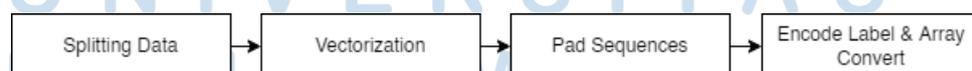
Teks ulasan dilakukan penghapusan *stopword* (kata-kata yang tidak menyimpan informasi). Penghapusan *stopword* dilakukan dengan menggunakan *library* Sastrawi pada Python.

e. *Stemming*

Proses *stemming* atau mentransformasi kata yang memiliki imbuhan maupun akhiran menjadi kata dasar dilakukan menggunakan *library* Sastrawi pada Python.

4. *Data Preparation*

Tahap ini dimaksudkan untuk mempersiapkan data teks menjadi *input* yang diterima oleh *neural network*. Terdapat empat tahapan utama di antaranya:



Gambar 3.4 Alur *Data Preparation*

a. *Splitting Data*

Proses ini untuk membagi data ulasan menjadi data *train* dan data *test* dengan rasio, dan *random state* tertentu. Ulasan aplikasi versi 4.0.2 tersebut dibagi menjadi data *train* (70%), data *validation* (15%), dan data *test* (15%). Adapun data *train* dipakai selama proses pelatihan model, data *validation* dipakai untuk mengevaluasi model yang sudah dilatih, serta data *test* dipakai untuk mengukur performa model dalam melakukan tugasnya.

b. *Vectorization*

Proses ini bertujuan untuk mengonversi teks menjadi bentuk *integer* unik dari *vocabulary* yang ada pada data, agar bisa dimasukkan ke dalam *neural network*.

c. *Pad sequences*

Neural network mengharuskan *input* dalam bentuk dan ukuran yang sama. Oleh karena itu, diperlukan fungsi *pad sequence* yang mentransformasikan setiap input ke dalam panjang yang sama. Pada penelitian ini, akan digunakan argumen *padding* yaitu *post* untuk mengisi *value* 0 di akhir tiap *sequences* agar panjangnya sama.

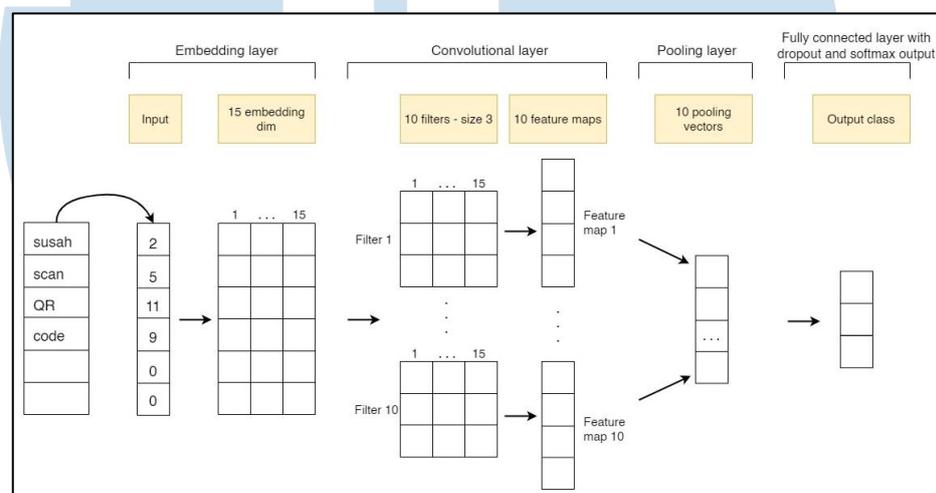
d. *Encode label dan Array Convert*

Proses *encode label* bertujuan untuk mengonversi kelas label aspek maupun sentimen menjadi bentuk *integer* yang unik. Adapun pada klasifikasi aspek yang merupakan *multi-class classification*, akan ditambahkan fungsi *to_categorical_function* dari Keras yang akan mengonversi kelas label ke dalam *matrix binary* atau *array*.

5. *Classification Modelling*

Tahap ini merupakan tahap pemodelan menggunakan data ulasan versi 4.0.2 yang sudah melalui tahap *data preparation*. Adapun pemodelan dibagi menjadi dua yaitu pemodelan untuk klasifikasi aspek dan klasifikasi sentimen. Pada keduanya akan dibuat model CNN tersendiri menggunakan parameter yang menunjukkan hasil akurasi terbaik dalam proses

hyperparameter tuning. Adapun *hyperparameter tuning* merupakan proses pemilihan parameter yang tepat untuk diimplementasikan pada model, dan mempengaruhi kinerja dari model [42]. Setelah menginisiasi model, akan dilakukan pelatihan menggunakan data *train*. Gambar 3.5 menunjukkan contoh struktur dari model CNN yang akan dibuat. Luaran tahap ini adalah model CNN untuk klasifikasi aspek dan model CNN untuk klasifikasi sentimen.



Gambar 3.5 Contoh Struktur Model CNN

6. Model Evaluation

Pada tahap ini, model dilakukan evaluasi terhadap akurasi dan *loss*-nya menggunakan data validasi. Kemudian dilakukan juga pengujian menggunakan data *test* yang akan dievaluasi performa klasifikasinya menggunakan *metric* yaitu *accuracy*, *precision*, *recall*, dan *F1 score*. Jika data memiliki label yang seimbang, *accuracy* akan menjadi metrik utama, sedangkan jika data memiliki label yang tidak seimbang maka *F1 score* akan menjadi metrik utama dalam evaluasi.

7. Implementation

Tahap ini merupakan tahap klasifikasi *unlabelled data* (ulasan versi 4.0.5) yang sudah dilakukan *pre-processing* sebelumnya, menggunakan model CNN untuk klasifikasi aspek dan klasifikasi sentimen yang dibangun

pada tahap keempat penelitian ini. Luaran tahap ini adalah ulasan versi 4.0.5 yang sudah diklasifikasikan aspek dan sentimennya.

8. *Interpretation*

Tahap ini bertujuan untuk menjelaskan hasil dari klasifikasi aspek dan sentimen yang dilakukan, serta melakukan perbandingan sentimen di setiap aspek pada versi 4.0.2 dengan versi 4.0.5. Perbandingan ini dimaksudkan untuk mengetahui apakah dengan pembaruan versi, sentimen yang ada pada setiap aspek mengalami perubahan.

3.4 Teknik Pengumpulan Data

Pengumpulan data teks ulasan aplikasi PeduliLindungi dilakukan dengan teknik *scraping*, di mana hasil ulasan diambil dari ulasan pengguna pada Google Play Store. Adapun *scraping* data dilakukan dengan menggunakan *library google_play_scraper* yang ada pada Python.

3.5 Tools

Jenis *tools* yang membantu dalam proses penelitian ini adalah *programming language* dan *software* sebagai berikut:

3.5.1 *Programming Language*

Penelitian ini menggunakan bahasa pemrograman yaitu Python dalam melakukan *scraping* data, pembersihan (*pre-processing*) data, pemodelan serta klasifikasi teks. Dalam konteks tugas klasifikasi teks, Python memiliki dukungan *library* yang lengkap serta kegunaannya dalam tugas *deep learning*.

Tabel 3.4 Perbandingan Bahasa Pemrograman

Kategori	Python	R
<i>Usability</i>	Python merupakan bahasa pemrograman yang terkenal untuk melakukan tugas <i>data science</i> , dan handal dalam tugas <i>machine learning</i> , <i>deep learning</i> , aplikasi berskala besar, khususnya dalam data analisis.	R merupakan bahasa pemrograman yang terkenal dalam melakukan tugas pembelajaran statistik dan penggunaan <i>dataset</i> yang terstruktur.

Kategori	Python	R
<i>Library</i>	Python memiliki banyak <i>library data science</i> yang dapat digunakan mulai dari eksplorasi data, <i>data modelling</i> , hingga <i>library</i> untuk melakukan komputasi algoritma <i>machine learning</i> .	R memiliki banyak <i>library data science</i> dengan kecenderungan pada <i>library</i> analisis statistik dan visualisasi. Dalam pemodelan data, seringkali harus menggunakan <i>package</i> di luar R.
<i>Data Collection</i>	Mendukung segala jenis format data, memiliki <i>library</i> yang memudahkan dalam <i>scraping</i> data dari <i>web</i> .	Format data yang didukung R terbatas, serta kurang serba guna dalam melakukan <i>scraping data</i> .

Pada penelitian ini, terdapat beberapa *library* utama yang dipakai di antaranya:

Tabel 3.5 *Library* Utama pada Penelitian

<i>Library</i>	Tujuan
Google_play_scraper	Menyediakan API yang mendukung tugas <i>scraping data</i> yang berasal dari Google Play.
NLTK	<i>Library</i> yang mendukung tugas dalam <i>Natural Language Processing</i> .
Sastrawi	<i>Library</i> yang mendukung tugas <i>stemming</i> kata, <i>stopwords</i> dalam bahasa Indonesia
Matplotlib	<i>Library</i> yang mendukung visualisasi data.
Sklearn	<i>Library</i> yang mendukung tugas <i>machine learning</i> .
Numpy	<i>Library</i> yang mendukung operasi matematika pada <i>array</i> .
Tensorflow	<i>Library</i> yang mendukung tugas pemodelan klasifikasi untuk pembuatan <i>neural network</i> dengan banyak <i>layer</i> .

3.5.2 Software

Berikut *software* yang digunakan pada penelitian ini, di antaranya:

- a. JupyterLab 3.0.14 sebagai *code editor* dalam penggunaan bahasa pemrograman Python.
- b. Anaconda Navigator 2.0.3 yang mengelola setiap *library* yang dipakai pada Python.
- c. Microsoft Excel 365 sebagai lembar kerja untuk penyimpanan data dan manajemen data.
- d. Tableau 2021.4.2 sebagai *tools* untuk melakukan data visualisasi.