



**PENERAPAN ALGORITMA *NAÏVE BAYES* UNTUK MENDETEKSI
SENTIMEN PENGGUNA APLIKASI *E-LEARNING* (STUDI KASUS:
RUANGGURU, PAHAMIFY, MERDEKA MENGAJAR)**

TUGAS AKHIR



**Oleh :
MUHAMMAD ALFA FAWWAZ
18410100057**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
SURABAYA
2024**

**PENERAPAN ALGORITMA *NAÏVE BAYES* UNTUK MENDETEKSI
SENTIMEN PENGGUNA APLIKASI *E-LEARNING* (STUDI KASUS:
RUANGGURU, PAHAMIFY, MERDEKA MENGAJAR)**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana**



**UNIVERSITAS
Dinamika**

Oleh :

**Nama : Muhammad Alfa Fawwaz
NIM : 18410100057
Program Studi : S1 Sistem Informasi**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2024

Tugas Akhir

PENERAPAN ALGORITMA *NAÏVE BAYES* UNTUK MENDETEKSI SENTIMEN PENGGUNA APLIKASI *E-LEARNING* (STUDI KASUS: RUANGGURU, PAHAMIFY, MERDEKA MENGAJAR)

Dipersiapkan dan disusun oleh

Muhammad Alfa Fawwaz

NIM: 18410100057

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: Jum'at, 15 Maret 2024

Susunan Dewan Pembahas

Pembimbing:

I. Sulistiowati, S.Si., M.M.

NIDN: 0719016801


II. Vivine Nurcahyawati, M.Kom.

NIDN: 0723018101

Pembahas:

III. Tutut Wuriyanto, M.Kom.

NIDN. 0703056702



Digitally signed by
Vivine
Nurcahyawati
Date: 2024.03.19
12:14:15 +07'00'



Date:
2024.03.19
14:02:25
+07'00'

Tugas akhir ini telah diterima sebagai salah satu persyaratan untuk memperoleh gelar Sarjana



Digitally signed by

Anjik Sukmaaji

Date: 2024.03.20

14:20:54 +07'00'

Dr. Anjik Sukmaaji, S.Kom., M.Eng.

NIDN. 0731057301

Dekan Fakultas Teknologi dan Informasi
UNIVERSITAS DINAMIKA

"Setiap orang bisa mencapai kesuksesan, hanya saja beberapa perjalanannya lebih lama dari yang lain."

- Muhammad Alfa Fawwaz -



UNIVERSITAS
Dinamika



Ku persembahkan
Kepada Keluarga yang telah mendoakan
Bapak dan Ibu serta Dosen yang telah membimbingku
Serta semua teman-teman yang telah mensupport
tugas akhir ini hingga selesai.

UNIVERSITAS
Dinamika

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : Muhammad Alfa Fawwaz
NIM : 18.41010.0057
Program Studi : S1 Sistem Informasi
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **PENERAPAN ALGORITMA NAÏVE BAYES UNTUK MENDETEKSI SENTIMEN PENGGUNA APLIKASI E-LEARNING (STUDI KASUS: RUANGGURU, PAHAMIFY, MERDEKA MENGAJAR)**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut diatas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 15 Maret 2024
Yang menyatakan



Muhammad Alfa Fawwaz
NIM: 18.41010.0057

ABSTRAK

Kegiatan belajar mandiri secara *online* dengan menggunakan aplikasi *e-learning* meningkat di tiap periodenya. Mengingat banyaknya kebutuhan materi yang harus dihadapi siswa, sehingga menjadi salah satu alasan perilaku pengguna memilih belajar mandiri secara *online* menggunakan aplikasi *e-learning* serta fleksibilitas waktu dan tempat juga memungkinkan bagi pengguna untuk belajar kapanpun dengan mengatur waktu belajar mereka sendiri dan di manapun lokasinya berada. Dilansir dari *my-best.id* didapatkan informasi 10 rekomendasi aplikasi belajar *online*, dan dalam penelitian ini dipilih tiga aplikasi yaitu Ruangguru, Pahamify, Merdeka Mengajar karena aplikasi tersebut berada pada peringkat teratas laman web yang dilansir dan memiliki ruang lingkup di Indonesia serta banyak memiliki ulasan pada kolom komentar *playstore*. Maka penelitian ini menggunakan data ulasan yang berisi tanggapan pengguna pada tiga aplikasi *e-learning* yang dijadikan objek untuk dilakukan prediksi sentimen negatif atau positif yang dapat digunakan sebagai evaluasi atau rencana strategis untuk dijadikan tolak ukur *e-learning* dalam melakukan penilaian dari *e-learning* itu sendiri di masa yang akan datang. Penerapan algoritma *naive bayes* menghasilkan nilai akurasi sebesar 89% pada aplikasi Ruangguru, 86% pada aplikasi Pahamify, dan 92% pada aplikasi Merdeka Mengajar. Hasil penelitian tersebut telah dievaluasi menggunakan *confusion matrix*.

Kata Kunci: analisis sentimen, *naive bayes*, *review e-learning*.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang telah memberikan rahmat dan hidayahnya sehingga penulis dapat melaksanakan dan menyelesaikan Laporan Tugas Akhir berjudul “Penerapan Algoritma *Naïve Bayes* Untuk Mendeteksi Sentimen Pengguna Aplikasi *E-Learning* (Studi Kasus: Ruangguru, Pahamify, Merdeka Mengajar)”.

Penyelesaian Laporan Tugas Akhir ini tidak terlepas dari bantuan berbagai pihak yang telah memberikan banyak masukan, nasihat, saran, kritik dan dukungan moral maupun *masteril* kepada penulis. Oleh karena itu, penulis menyampaikan rasa terimakasih atas bantuannya kepada:

1. Bapak dan Ibu tercinta yang selalu memberikan support dan semangat.
2. Bapak Prof. Dr. Budi Jatmiko, M.Pd., selaku Rektor Universitas Dinamika Surabaya.
3. Bapak Julianto Lemantara, S.Kom., M.Eng selaku Ketua Program Studi S1 Sistem Informasi Universitas Dinamika
4. Ibu Sulistiowati, S.Si., M.M. selaku Dosen Pembimbing 1 yang selalu membimbing, memberikan motivasi, dan mendukung penulis dalam menyelesaikan Tugas Akhir ini.
5. Ibu Vivine Nurcahyawati, M.Kom.,OCP. selaku Dosen Pembimbing 2 yang selalu membimbing, memberikan motivasi, dan mendukung penulis dalam menyelesaikan Tugas Akhir ini.
6. Bapak Tutut Wuriyanto, M.Kom. selaku dosen pembahas yang telah menyempurnakan tugas akhir ini.
7. Teman-teman yang telah memberikan bantuan dan dukungan dalam menyusun serta menyelesaikan laporan Tugas Akhir ini.
8. Pihak-pihak lain yang tidak dapat disebutkan satu-persatu yang telah memberikan bantuan dan dukungan kepada penulis untuk bisa menyelesaikan laporan Tugas Akhir ini.

Semoga diberikan balasan oleh Allah SWT atas bantuan yang telah diberikan dalam bimbingan serta nasihat dalam proses pengerjaan laporan Tugas Akhir ini. Dapat disadari bahwa dengan keterbatasan yang ada, laporan Tugas Akhir ini masih jauh dari kata sempurna. Kritik dan saran sangat diperlukan untuk sempurnanya laporan Tugas Akhir ini ke depan. Akhir dari laporan Tugas Akhir ini semoga dapat bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, 15 Maret 2024

Penulis

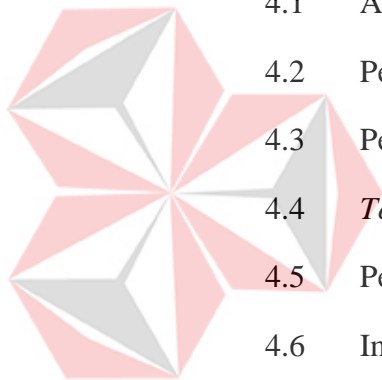


UNIVERSITAS
Dinamika

DAFTAR ISI

| | Halaman |
|-------------------------------------|---------|
| ABSTRAK | vii |
| KATA PENGANTAR | viii |
| DAFTAR ISI..... | x |
| DAFTAR GAMBAR | xii |
| DAFTAR TABEL..... | xiii |
| DAFTAR LAMPIRAN..... | xv |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah..... | 3 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan | 4 |
| 1.5 Manfaat | 4 |
| BAB II LANDASAN TEORI..... | 5 |
| 2.1 Analisis Sentimen | 6 |
| 2.2 <i>Python</i> | 6 |
| 2.3 <i>Text Preprocessing</i> | 7 |
| 2.4 <i>Naïve Bayes</i> | 9 |
| 2.5 <i>Confusion Matrix</i> | 10 |
| 2.6 Visualisasi | 11 |
| BAB III METODOLOGI PENELITIAN..... | 12 |
| 3.1 Tahap Awal..... | 12 |
| 3.1.1 Studi Literatur | 13 |
| 3.1.2 Observasi | 13 |
| 3.2 Analisis Data..... | 13 |

| | | |
|--|---|-----------|
| 3.2.1 | Pengambilan Data | 13 |
| 3.2.2 | Pelabelan Data | 14 |
| 3.2.3 | <i>Text Preprocessing</i> | 14 |
| 3.2.4 | Pembagian Data | 17 |
| 3.2.5 | Klasifikasi <i>Naïve Bayes</i> | 17 |
| 3.2.6 | Evaluasi Data | 21 |
| 3.2.7 | Visualisasi | 22 |
| 3.3 | Tahap Akhir | 22 |
| 3.3.1 | Kesimpulan dan Saran | 22 |
| BAB IV HASIL DAN PEMBAHASAN | | 23 |
| 4.1 | Analisis | 23 |
| 4.2 | Pengambilan Data | 23 |
| 4.3 | Pelabelan Data | 28 |
| 4.4 | <i>Text Pre-Processing</i> | 29 |
| 4.5 | Pembagian Data | 37 |
| 4.6 | Implementasi Klasifikasi <i>Naïve Bayes</i> | 38 |
| 4.7 | Evaluasi Data | 39 |
| 4.8 | Visualisasi | 42 |
| 4.9 | Hasil dan Pembahasan | 44 |
| BAB V KESIMPULAN DAN SARAN..... | | 46 |
| 5.1 | Kesimpulan | 46 |
| 5.2 | Saran | 46 |
| DAFTAR PUSTAKA | | 47 |



DAFTAR GAMBAR

| | Halaman |
|---|---------|
| Gambar 3. 1 Diagram Alir Metode Penelitian..... | 12 |
| Gambar 3. 2 Diagram Alir Tahap Awal | 12 |
| Gambar 3. 3 Diagram Alir Analisis Data | 13 |
| Gambar 3. 4 Diagram Alir Tahap Akhir..... | 22 |
| Gambar 4. 1 Hasil Pengambilan Data | 25 |
| Gambar 4. 2 Hasil Pengkategorian 4 Data..... | 26 |
| Gambar 4. 3 Hasil Pengurutan Data Terbaru..... | 27 |
| Gambar 4. 4 Hasil Proses Penyisahan Kolom | 27 |
| Gambar 4. 5 Hasil Pelabelan Data | 28 |
| Gambar 4. 6 Hasil Menyimpan Data | 29 |
| Gambar 4. 7 Pengambilan dari Cloud..... | 30 |
| Gambar 4. 8 Hasil Case Folding Text Pre-Processing | 32 |
| Gambar 4. 9 Hasil Stopword Removal Text Pre-Processing..... | 33 |
| Gambar 4. 10 Hasil Tokenizing Text Pre-Processing | 35 |
| Gambar 4. 11 Hasil Stemming Text Pre-Processing | 37 |
| Gambar 4. 13 Hasil Perhitungan Confusion Matrix Ruangguru | 41 |
| Gambar 4. 14 Hasil Perhitungan Confusion Matrix Pahamify..... | 41 |
| Gambar 4. 15 Hasil Perhitungan Confusion Matrix Merdeka Mengajar..... | 42 |
| Gambar 4. 16 Wordcloud Ruangguru..... | 43 |
| Gambar 4. 17 Wordcloud Pahamify | 43 |
| Gambar 4. 18 Wordcloud Merdeka Mengajar | 44 |

DAFTAR TABEL

| | Halaman |
|---|---------|
| Tabel 2. 1 Penelitian Terdahulu dengan metode yang sama | 5 |
| Tabel 2. 2 Tahap Cleansing | 8 |
| Tabel 2. 3 Case Folding | 8 |
| Tabel 2. 4 Stopword Removal | 9 |
| Tabel 2. 5 Tokenizing | 9 |
| Tabel 2. 6 Stemming | 9 |
| Tabel 2. 7 Confusion Matrix | 10 |
| Tabel 3. 1 Identifikasi Masalah Saat Ini | 14 |
| Tabel 3. 2 Hasil Cleansing | 15 |
| Tabel 3. 3 Hasil Case Folding | 15 |
| Tabel 3. 4 Hasil Stopword Removal | 16 |
| Tabel 3. 5 Hasil Tokenizing | 16 |
| Tabel 3. 6 Hasil Stemming | 17 |
| Tabel 3. 7 Data Hasil Text Preprocessing | 18 |
| Tabel 3. 8 Contoh Frekuensi Kata Uji pada Data Training | 18 |
| Tabel 3. 9 Contoh Hasil Probabilitas Kata | 19 |
| Tabel 3. 10 Contoh Confusion Matrix | 21 |
| Tabel 4. 1 Instalasi Package Library | 23 |
| Tabel 4. 2 Pengambilan Data | 24 |
| Tabel 4. 3 Pengambilan Ulasan Review | 25 |
| Tabel 4. 4 Mengkategorikan Data | 26 |
| Tabel 4. 5 Mengurutkan Data Terbaru | 26 |
| Tabel 4. 6 Menggunakan dan Memilah Kolom | 27 |
| Tabel 4. 7 Pelabelan Data | 28 |
| Tabel 4. 8 Simpan Data | 29 |
| Tabel 4. 9 Mengambil Data dari Penyimpanan Cloud Colab | 30 |
| Tabel 4. 10 Cleansing | 31 |
| Tabel 4. 11 Case Folding | 32 |
| Tabel 4. 12 Stopword Removal | 33 |
| Tabel 4. 13 Tokenizing | 34 |

| | |
|-------------------------------------|----|
| Tabel 4. 14 Instalasi Library | 35 |
| Tabel 4. 15 Stemming | 36 |
| Tabel 4. 16 Pembagian Data | 38 |
| Tabel 4. 17 Naive Bayes | 39 |
| Tabel 4. 18 Confusion Matrix..... | 40 |
| Tabel 4. 19 Hasil Akhir | 44 |



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

| | |
|---------------------------------|----|
| Lampiran 1 Hasil Turnitin | 49 |
| Lampiran 2 Form Bimbingan | 51 |
| Lampiran 3 Biodata Penulis..... | 52 |



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Banyaknya aplikasi saat ini menunjukkan perkembangan teknologi dan internet yang sangat cepat. Sampai akhir tahun 2020, diperkirakan terdapat sekitar 196,7 juta individu yang menggunakan internet di Indonesia, berdasarkan informasi yang diberikan oleh Asosiasi Penyelenggara Jasa Internet Indonesia. Aplikasi yang telah ada bisa dengan mudah diakses dan diunduh pada perangkat *mobile* yang berbasis *android* melalui *Google Play Store*. Menurut databoks.katadata.co.id, jumlah unduhan aplikasi di seluruh dunia meningkat sebesar 8,7% pada kuartal pertama tahun 2021 dibandingkan periode yang sama tahun 2020. Dengan 3,8 miliar unduhan, *Google Play Store* menjadi penyedia unduhan terbesar (Hendriyanto, Ridha, & Enri, 2022).

Aplikasi *e-learning* adalah salah satu yang tersedia di *Google Playstore*. Kegiatan belajar mandiri secara *online* dengan menggunakan aplikasi *e-learning* meningkat di tiap periodenya. Sedangkan proses kegiatan belajar mengajar yang ada pada tempat lembaga belajar mengajar masih bersifat terpusat kepada guru dan kurangnya interaksi keterlibatan siswa secara langsung pada kegiatan belajar mengajar serta belum memanfaatkan media pembelajaran secara optimal dianggap kurang efektif dalam mencapai tujuan pembelajaran dan harus disesuaikan. Mengingat banyaknya kebutuhan materi yang harus dihadapi siswa. Oleh karena itu, hal ini menjadi salah satu faktor yang mendorong perilaku pengguna untuk memilih untuk belajar secara mandiri melalui *platform online* menggunakan aplikasi *e-learning* serta fleksibilitas waktu dan tempat juga memungkinkan bagi pengguna untuk belajar kapanpun dengan mengatur waktu belajar mereka sendiri dan di manapun lokasinya berada (Utami, 2016).

Dilansir dari *my-best.id* didapatkan informasi 10 rekomendasi aplikasi belajar *online* yaitu zenius, ruangguru, duolingo, checkmath, udey, merdeka mengajar, pahamify, kelaskita, aplikasi belajar TK-PAUD, belajar al-quran. Dalam penelitian ini dipilih tiga aplikasi yaitu Ruangguru, Pahamify, Merdeka Mengajar karena aplikasi tersebut berada pada peringkat teratas laman *web* yang dilansir dan

memiliki ruang lingkup di Indonesia serta banyak memiliki ulasan pada kolom komentar *playstore*.

Pada tiga aplikasi yang dijadikan objek terdapat berbagai ulasan komentar yang positif maupun negatif, agar pandangan pengguna terhadap tiga aplikasi yang dijadikan objek penelitian bisa lebih baik dari sebelumnya, Untuk mengetahui pendapat pengguna saat ini, diperlukan pertimbangan kondisi dan pendapat pengguna, oleh karena itu untuk mengetahui pendapat publik perlu dilakukan penelitian yang melibatkan pemrosesan pada teks data yang ada pada ulasan komentar dikarenakan bentuk data opini atau ulasan pengguna yang belum terstruktur.

Oleh karena itu tiga aplikasi tersebut seharusnya dapat memenuhi ekspektasi dari penggunaannya untuk meningkatkan layanannya agar pengguna tetap setia menggunakan layanan dari aplikasi itu sendiri dan dapat bersaing dengan aplikasi lainnya, karena masih banyak ulasan pengguna dari tiga aplikasi itu sendiri yang puas ataupun tidak puas dengan performa atau layanan yang diberikan oleh tiga aplikasi tersebut.

Dalam sistem dari ketiga aplikasi itu sendiri, ulasan pengguna merupakan salah satu aspek penting dalam kelangsungan proses bisnisnya yang sangat bergantung kepada pengguna. Maka dari itu mempertahankan pengguna adalah salah satu upaya untuk meningkatkan reputasi aplikasi (Anindira & Imran, 2021). Oleh karena itu ulasan dari pengguna yang ada di sosial media perlu dilakukan analisis sentimen untuk menyaring opini dan mengklasifikasinya sehingga hasil dari klasifikasi nanti bisa digunakan untuk melakukan analisis pengguna aplikasi *e-learning* berdasarkan aplikasi yang digunakan.

Analisis sentimen adalah proses pemahaman, ekstraksi, dan pemrosesan data teks yang diperoleh dari masyarakat mengenai suatu topik, pendapat, penilaian, atau objek secara otomatis berdasarkan pada media sosial atau situs web yang tersedia untuk mendapatkan informasi yang terkandung di dalamnya (Buntoro, 2017). Analisis sentimen merupakan solusi guna menyaring pendapat publik dan membaginya ke dalam kategori positif dan negatif. (Muttaqin & Kharisudin, 2021), sehingga hasil klasifikasi dapat dimanfaatkan untuk membantu publik guna

mengetahui respon pengguna terhadap aplikasi *e-learning* berdasarkan aplikasi yang digunakan.

Dalam analisis sentimen, teknik *Naïve Bayes* digunakan untuk mengklasifikasikan data sebagai teks. Selanjutnya, data teks yang diklasifikasikan tersebut akan digunakan untuk mengklasifikasikan kata berdasarkan kelas berdasarkan pendapat yang sudah diambil. *Naïve Bayes* adalah teknologi preprocessing untuk klasifikasi fitur yang meningkatkan efisiensi, skalabilitas, dan akurasi proses klasifikasi teks. (Widagdo, W. A., & Nasiri, 2020).

Hasil dari penelitian ini adalah berupa ulasan pengguna yang dapat digunakan untuk memahami pendapat publik tentang tiga aplikasi *e-learning* rekomendasi terbaik saat ini yang digunakan peneliti untuk melakukan penelitian, apakah ada lebih banyak tanggapan positif atau negatif sehingga hasil penelitian dapat digunakan untuk evaluasi dan rencana strategis untuk meningkatkan kualitas layanan atau performa aplikasi sehingga pengguna dapat menikmati aplikasi *e-learning*. Berdasarkan penjelasan di atas, penelitian ini akan melakukan penerapan algoritma *naïve bayes* untuk mendeteksi sentimen pengguna aplikasi *e-learning* (studi kasus: ruangguru, pahamify, merdeka mengajar).

1.2 Rumusan Masalah

Dengan mempertimbangkan konteks yang telah dijelaskan di atas, rumusan masalah dalam penelitian ini adalah bagaimana melakukan penerapan algoritma *naïve bayes* untuk mendeteksi sentimen pengguna aplikasi *e-learning* (studi kasus: Ruangguru, Pahamify, Merdeka Mengajar).

1.3 Batasan Masalah

Agar penelitian ini bisa memudahkan dalam pembahasan dan lebih terarah, terdapat Batasan masalah pada penelitian ini sebagai berikut:

1. Pembahasan analisis dalam penelitian ini mengulas tiga aplikasi *e-learning* yang beroperasi di Indonesia yaitu Ruangguru, Pahamify, Merdeka Mengajar yang bersumber dari *play.google.com*

2. Dataset yang digunakan dalam penelitian ini berasal dari ulasan pengguna aplikasi Ruangguru, Pahamify, Merdeka Mengajar pada kolom komentar *playstore* masing-masing aplikasi
3. Kelas untuk klasifikasi terdiri atas kelas positif dan negatif.
4. Bahasa Indonesia digunakan dalam ulasan pengguna.
5. Pelabelan dataset dilakukan secara otomatis.
6. *Output* dari penelitian ini berupa hasil klasifikasi positif dan negatif, serta *wordcloud* yang menampilkan kata yang sering muncul pada data ulasan pengguna positif maupun negatif di masing-masing aplikasi.

1.4 Tujuan

Tujuan dari penelitian yang dilakukan penulis adalah untuk menghasilkan klasifikasi opini ulasan pengguna terhadap aplikasi *e-learning* (studi kasus: Ruangguru, Pahamify, Merdeka Mengajar) kedalam 2 kelas yaitu positif dan negatif sebagai bahan pertimbangan *e-learning* untuk melihat ulasan dari pengguna apakah baik atau buruk.

1.5 Manfaat

Berdasarkan analisis sentimen yang dilakukan dari penelitian tugas akhir ini yaitu diharapkan dapat digunakan sebagai tolak ukur *e-learning* untuk melakukan penilaian dari *e-learning* itu sendiri.

BAB II

LANDASAN TEORI

Penelitian terdahulu digunakan sebagai acuan dasar untuk menambah perspektif penulis. Penulis mencari penelitian terdahulu yang memiliki jenis penelitian yang kurang lebih sama dan kemudian mencari perbedaan dari penelitian tersebut, dapat dilihat di Tabel 2.1 berikut.

Tabel 2. 1 Penelitian Terdahulu dengan metode yang sama

| No | Nama Peneliti | Tahun Terbit | Judul Penelitian | Hasil |
|---|---|--------------|--|---|
| 1. | Muljono, Dian Putri Artanti, Abdul Syukur, Adi Prihandono (Muljono, Artanti, Prihandono, & Setiadi, 2018) | 2018 | Analisa Sentimen Untuk Penilaian Pelayanan Situs Belanja <i>Online</i> Menggunakan Algoritma <i>Naïve Bayes</i> | Studi kasus: Analisa Sentimen Untuk Penilaian Pelayanan Situs Belanja <i>Online</i> Menggunakan Algoritma <i>Naïve Bayes</i> Tools: <i>Confusion Matrix</i> Sumber data: Twitter <i>E-Commerce</i> : Lazada, Bukalapak, Blibli, dan Tokopedia Akurasi :93.33% |
| Perbedaan: Pada penelitian milik Muljono melakukan analisis sentimen untuk penilaian pelayanan situs belanja <i>online</i> . | | | | |
| 2. | Adika Widagdo, Bambang Soedijono W. A, Asro Nasiri (Widagdo, W. A, & Nasiri, 2020) | Sri 2020 | Analisa Tingkat Kepopuleran <i>E-Commerce</i> Di Indonesia Berdasarkan Sentimen Sosial Media Menggunakan Metode <i>Naïve Bayes</i> | Studi kasus: Analisa Tingkat Kepopuleran <i>E-Commerce</i> Di Indonesia Berdasarkan Sentimen Sosial Media Menggunakan Metode <i>Naïve Bayes</i> Tools: RapidMiner Testing: <i>Confusion Matrix</i> Sumber data: Twitter <i>E-Commerce</i> : Tidak Disebutkan Akurasi :97.14% |
| Perbedaan: Pada penelitian milik Adika & Asro objek yang dijadikan penelitian adalah aplikasi <i>E-Commerce</i> di Indonesia sedangkan pada penelitian ini menggunakan aplikasi <i>e-learning</i> / pembelajaran secara daring | | | | |
| 3. | Asep Hendra dan Fitriyani (Hendra & Fitriyani, 2021) | 2021 | Analisis Sentimen Review Halodoc Menggunakan <i>Naïve Bayes Classifier</i> | Studi kasus: Analisis Sentimen Review Halodoc Menggunakan <i>Naïve Bayes Classifier</i> Tools: <i>Confusion Matrix</i> Sumber data: <i>Google Playstore</i> Akurasi :81.68% |
| Pembeda: Pada penelitian milik Asep & Fitriyani melakukan analisis sentimen <i>review</i> pada aplikasi Halodoc | | | | |

2.1 Analisis Sentimen

Analisis sentimen (analisis sentimen) atau penambangan opini (penambangan opini) adalah studi komputasi untuk mengenali dan mengekspresikan opini, sentimen, evaluasi, sikap, emosi, subjektivitas, penilaian, atau pandangan yang terdapat dalam teks biasa. (Wilma, Lawalata, Pailin, & Ohoiren, 2021).

Analisis sentimen digunakan untuk memahami dan mengelola data teks dengan tujuan mendapatkan informasi dari kalimat-kalimat opini. Fokus analisis sentimen adalah mengidentifikasi kecenderungan pendapat atau opini seseorang terhadap suatu masalah atau objek, baik itu bersifat positif, negatif, maupun netral. (Zalyhaty, Nurcahyawati, & Sutomo, 2020).

Text mining merupakan proses eksplorasi dan analisis data teks yang tidak terstruktur dengan menggunakan bantuan perangkat lunak untuk mengidentifikasi pola, topik, dan kata kunci dalam informasi data yang diberikan. Proses ini mencakup pemrosesan dan analisis jumlah data yang cukup besar, struktur teks yang kompleks, dimensi yang besar, dan *noise*. (Pravina, Cholissodin, & Adikara, 2019).

2.2 Python

Python merupakan sebuah bahasa pemrograman yang dapat digunakan dalam data mining. *Python* memiliki banyak *library* yang membantu dalam proses data mining. Beberapa kelebihan dari bahasa pemrograman *python* antara lain seperti mendukung *multiplatform*, memiliki sistem pengelolaan *memory* yang otomatis, pengembangan program dilakukan dengan cepat dan *coding* yang lebih sedikit (Ginting, Kusri, & Taufiq, 2020).

Open source library python

Sejumlah *library* digunakan untuk menyederhanakan dan memudahkan dalam pengembangan sistem, pada penelitian ini menggunakan beberapa *library* yaitu sebagai berikut:

- a. *Pandas* yaitu *library python* yang mempunyai kegunaan untuk analisis data dan untuk manipulasi data.
- b. *Numpy* yaitu *library python* yang berfokus dan berguna untuk pemrosesan dan analisis data pada perhitungan numerik.

- c. *Google play scraper* yaitu *library python* yang berfungsi untuk mengekstrak data dari *google play store* yang memungkinkan untuk bisa mengambil informasi data seputar detail aplikasi, ulasan pengguna, rating, dan lainnya.
- d. *Re (regular expressions)* yaitu *library python* yang berfungsi untuk melakukan pencocokan data dalam operasi pencarian dan penggantian teks berdasarkan pola.
- e. *NLTK (natural language toolkit)* yaitu *library python* yang berfungsi untuk mengelola bahasa alami secara umum, menghilangkan akhiran kata, pembagian data menjadi kata, mengubah kata ke bentuk dasarnya. Adapun fungsi lain seperti menyediakan kumpulan teks dalam berbagai bahasa.
- f. *Sastrawi* yaitu *library python* yang digunakan untuk pemrosesan bahasa alami dalam Bahasa Indonesia yang berfokus pada pemrosesan teks Bahasa Indonesia
- g. *Sklearn (scikit-learn)* yaitu *library python* yang digunakan untuk pembelajaran pada *machine learning* dan analisis data, juga berfungsi untuk mengevaluasi performa model dan meningkatkan kinerja model.
- h. *Wordcloud* yaitu *library python* yang digunakan untuk membuat visualisasi dari data teks yang nantinya akan di tunjukkan dalam bentuk *word cloud*, dan ukuran kata mengartikan bahwa kata tersebut yang sering muncul dari teks.
- i. *Matplotlib* yaitu *library python* yang berfungsi untuk membuat visualisasi data.

2.3 Text Preprocessing

Text preprocessing merupakan proses untuk membersihkan data agar mengurangi *noise* data dan juga mengatasi hilangnya informasi. Proses text mining memerlukan beberapa tahap awal karena struktur data teks tidak teratur. Persiapan adalah kunci untuk mengubah teks menjadi lebih terstruktur. Dalam analisis dataset penelitian ini menghasilkan penulisan ulasan pengguna yang unik dan mempunyai perbedaan di tiap akun pengguna, adapun karakter khusus pada dataset seperti

emoticon, *number*, dan simbol tanda baca yang juga berada pada dataset di tiap objek penelitian. Sehingga muncul beberapa teknik yang digunakan untuk mengelolanya, salah satu implementasi dari *text mining* adalah tahap *text preprocessing*.

Sebelum pemrosesan teks lainnya, data harus dibersihkan melalui proses yang disebut "*Text Pre-Processing*". Proses ini melibatkan pengolahan data untuk menghilangkan *noise* atau informasi yang hilang atau tidak lengkap. (Pravina, Cholissodin, & Adikara, 2019). Terdapat 5 tahap dalam proses *text pre-processing*, yaitu:

1. *Cleansing*

Merupakan tahapan yang bertujuan untuk menghapus karakter-karakter selain alfabet dari huruf a-z (termasuk tanda baca, *emoticon*, *number*) (Pravina, Cholissodin, & Adikara, 2019). Contoh data sebelum dan sesudah melalui tahap *cleansing* ditampilkan pada Tabel 2.2

Tabel 2. 2 Tahap *Cleansing*

| Sebelum <i>Cleansing</i> | Setelah <i>Cleansing</i> |
|---|---|
| Minta tolong diperbaiki segera di HP saya aplikasinya force close. Mohon bantuannya ya pihak developer 🙏🙏🙏» | Minta tolong diperbaiki segera di HP saya aplikasinya force close Mohon bantuannya ya pihak developer |

2. *Case Folding*

Proses ini bertujuan untuk mengonversi huruf kapital menjadi huruf kecil dalam data atau untuk menormalkan semua kata menjadi huruf kecil. (Izzah & Girsang, 2021). Contoh data sebelum dan sesudah melalui tahap *case folding* ditampilkan pada Tabel 2.3

Tabel 2. 3 *Case Folding*

| Sebelum <i>Case Folding</i> | Setelah <i>Case Folding</i> |
|---|---|
| Minta tolong diperbaiki segera di HP saya aplikasinya force close Mohon bantuannya ya pihak developer | minta tolong diperbaiki segera di hp saya aplikasinya force close mohon bantuannya ya pihak developer |

3. *Stopword Removal*

Kumpulan kata-kata umum yang tidak memiliki makna penting dalam sebagian besar kalimat atau dokumen. Menghapus kata-kata umum ini bertujuan untuk mengurangi jumlah kata yang disimpan oleh korpus atau sistem. (Mukminin,

Sunarto, & Nurcahyawati, 2021). Contoh data sebelum dan sesudah melalui tahap *stopword removal* ditampilkan pada Tabel 2.4

Tabel 2. 4 Stopword Removal

| Sebelum <i>Stopword Removal</i> | Setelah <i>Stopword Removal</i> |
|---|--|
| minta tolong diperbaiki segera di hp saya aplikasinya force close mohon bantuannya ya pihak developer | tolong diperbaiki hp aplikasinya force close mohon bantuannya ya developer |

4. *Tokenizing*

Merupakan tahapan proses pemisahan dokumen data menjadi token atau kata, pada langkah ini, dilakukan pemisahan setiap kata dalam kalimat yang terpisah oleh tanda koma (.). Proses ini melibatkan pemecahan kalimat dan kata. (Izzah & Girsang, 2021). Contoh data sebelum dan sesudah melalui tahap *tokenizing* ditampilkan pada Tabel 2.5

Tabel 2. 5 Tokenizing

| Sebelum <i>Tokenizing</i> | Setelah <i>Tokenizing</i> |
|--|---|
| tolong diperbaiki hp aplikasinya force close mohon bantuannya ya developer | 'tolong', 'diperbaiki', 'hp', 'aplikasinya', 'force', 'close', 'mohon', 'bantuannya', 'ya', 'developer' |

5. *Stemming*

Stemming adalah proses mengubah kata menjadi bentuk dasarnya. Proses *stemming* di Bahasa Indonesia cukup kompleks karena hanya membuat kata dasar di setiap komentar dan menghapus semua imbuhan pada kata-kata yang ada di dokumen. (Pravina, Cholissodin, & Adikara, 2019). Contoh data sebelum dan sesudah melalui tahap *stemming* ditampilkan pada Tabel 2.6

Tabel 2. 6 Stemming

| Sebelum <i>Stemming</i> | Setelah <i>Stemming</i> |
|---|--|
| 'tolong', 'diperbaiki', 'hp', 'aplikasinya', 'force', 'close', 'mohon', 'bantuannya', 'ya', 'developer' | tolong baik hp aplikasi force close mohon bantu ya developer |

2.4 *Naïve Bayes*

Menurut (Darwis, Siskawati, & Abidin, 2021) *Naive Bayes* adalah teknik *machine learning* yang menggunakan perhitungan probabilitas dan statistik yang diciptakan oleh ilmuwan Inggris Thomas Bayes. Cara kerja *Naive Bayes* adalah memprediksi probabilitas di masa depan berdasarkan pengalaman di masa lalu.

Dalam metode ini semua atribut atau variabel mempunyai peran masing-masing pada pengambilan keputusan, tidak ada yang bergantung pada variabel kelas.

Dalam penelitian ini, algoritma *naïve bayes* disesuaikan untuk memenuhi kelas data yang akan diklasifikasikan. Rumus *naïve bayes* yang disesuaikan adalah sebagai berikut:

$$P(W_k|C_i) = \frac{nk + 1}{n + |\text{Vocabulary}|} \quad (1)$$

Dimana:

$P(W_k)$: Probabilitas (peluang) kemunculan kata

C_i : Kategori kelas

$P(W_k|C_i)$: Probabilitas kemunculan kata (W_k) pada suatu dokumen dengan kategori kelas (C_i)

nk : Nilai kemunculan kata pada kategori C_i

n : Jumlah keseluruhan kata pada kategori C_i

Vocabulary: Jumlah keseluruhan kata

2.5 Confusion Matrix

Menurut (Pravina, Cholissodin, & Adikara, 2019) metode yang dikenal sebagai *Confusion Matrix* digunakan untuk mengevaluasi klasifikasi model untuk menentukan objek mana yang benar dan mana yang salah. Matriks prediksi yang akan dibandingkan dengan kelas asli terdiri dari informasi aktual dan perkiraan nilai klasifikasi. Setelah sistem melakukan pengklasifikasian komentar dengan sukses, ada ukuran yang diperlukan untuk menentukan kevalidan atau keakuratan klasifikasi yang dibuat oleh sistem. Pada Tabel 2.7 di bawah ini menunjukkan *Confusion Matrix* yang digunakan untuk membantu dalam perhitungan sistem evaluasi.

Tabel 2. 7 Confusion Matrix

| Classification | Predicted Positive | Predicted Negative |
|------------------------------|--|--|
| <i>Actual Positive Cases</i> | <i>Number of True Positive Cases (TP)</i> | <i>Number of False Negative Cases (FN)</i> |
| <i>Actual Negative Cases</i> | <i>Number of False Positive Cases (FP)</i> | <i>Number of True Negative Cases (TN)</i> |

Pengujian akurasi ini menggunakan *confusion matrix* empat kondisi berikut: *True Positive* (TP), *True Negatif* (TN), *False Positive* (FP), dan *False Negative* (FN). Nilai *accuracy*, *precision*, dan *recall* dapat dihitung dengan menggunakan *confusion matrix*. Rumus perhitungan detailnya dapat dilihat di bawah ini:

Perhitungan berikut menunjukkan akurasi model dalam mengklasifikasikan dengan benar:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

Akurasi antara data permintaan dan hasil prediksi model digambarkan dengan *precision* seperti perhitungan berikut:

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

Recall menggambarkan kesuksesan model dalam menemukan kembali sebuah informasi dengan perhitungan di bawah ini:

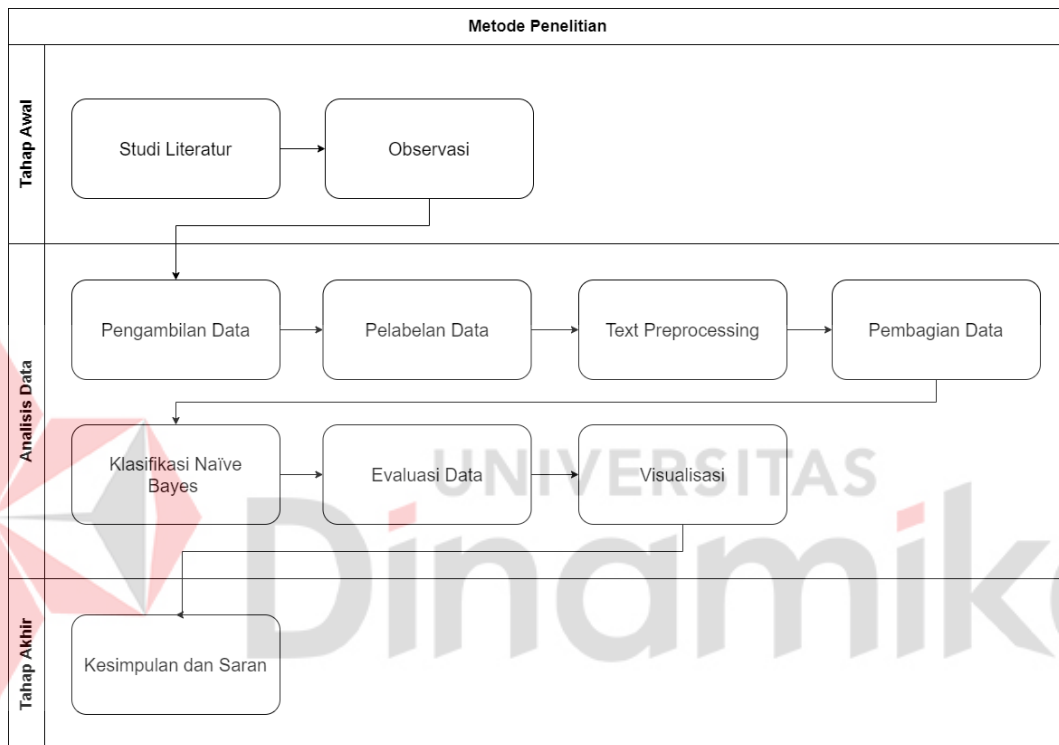
$$Recall = \frac{TP}{TP + FN} \quad (4)$$

2.6 Visualisasi

Menurut Hayadi (2017), Visualisasi juga merupakan upaya untuk mendeskripsikan maksud tertentu menjadi bentuk informasi yang mudah dipahami. Ini digunakan untuk mempermudah pengguna untuk melakukan analisis dan penalaran terkait informasi yang dibutuhkan serta menjadi bukti. Visualisasi juga merupakan metode penyampaian informasi yang berguna untuk menjelaskan konsep menggunakan gambar, diagram, atau animasi yang dapat diinterpretasikan, dieksplorasi, dan dianalisis dengan data yang terkait.

BAB III METODOLOGI PENELITIAN

Pada tahap ini menjelaskan tentang tahap-tahap yang akan dilakukan untuk menyelesaikan penelitian. Pada penelitian ini menggunakan tiga tahapan yaitu, tahap awal, tahap analisis data, dan tahap akhir. Gambaran dari metode penelitian terdapat pada Gambar 3.1



Gambar 3. 1 Diagram Alir Metode Penelitian

3.1 Tahap Awal

Tahap awal merupakan tahapan pertama yang dilakukan dalam penelitian ini sebelum melakukan analisis data. Diagram alir pada tahap awal berikut ini dapat dilihat pada Gambar 3.2



Gambar 3. 2 Diagram Alir Tahap Awal

3.1.1 Studi Literatur

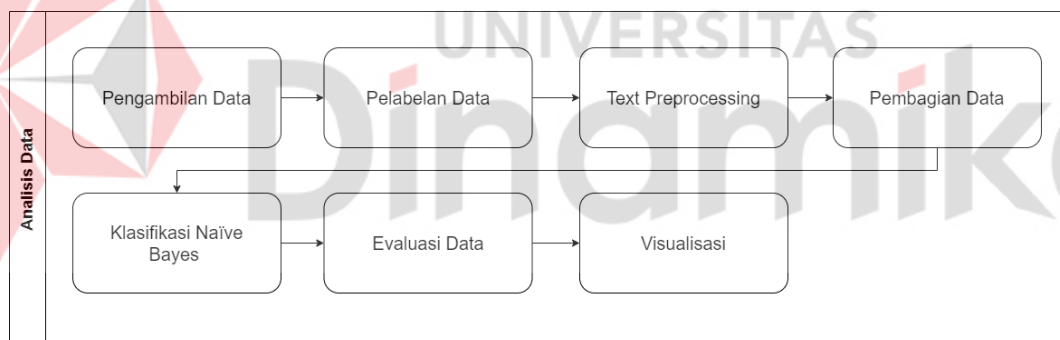
Studi literatur dilakukan dan merupakan tahapan pertama dalam penelitian ini untuk membantu dalam penulisan terhadap topik yang diambil. Tahap ini menganalisis terhadap penelitian-penelitian sebelumnya dan referensi jurnal lainnya yang dapat membantu dalam menyelesaikan penelitian ini.

3.1.2 Observasi

Observasi yang dilakukan pada tahap ini yaitu melakukan pengamatan, mempelajari, dan memahami objek atau topik yang diteliti. Pada penelitian ini dilakukan dengan mengamati ulasan pengguna pada kolom komentar *Google Playstore* di tiga aplikasi objek penelitian.

3.2 Analisis Data

Tahap ini menjelaskan tentang analisis data untuk memahami kebutuhan data yang akan digunakan dalam menyelesaikan tugas akhir ini, dan selanjutnya dilakukan pemrosesan data. Diagram alir analisis data dapat dilihat pada Gambar 3.3



Gambar 3. 3 Diagram Alir Analisis Data

3.2.1 Pengambilan Data

Pada tahap ini dilakukan pencarian data yang relevan dan memahami data tersebut dengan melakukan penambangan atau pengambilan data menggunakan bahasa pemrograman *Python* dan dengan bantuan *tools Google Colab*. Kemudian melakukan *import library pandas*. *Pandas*, *library python* yang berfungsi untuk memanipulasi data, selanjutnya menggunakan *library Google play scraper* untuk *crawling* data dari *playstore*, lalu mengubah data *crawling* tersebut menjadi format *.csv*. Hasilnya disimpan dalam bentuk *.csv (comma-separated values)*. Data hasil *crawling* yang diperoleh sebanyak 2000 data disetiap objek studi kasusnya. Lalu

data ulasan diamati dan diklasifikasikan berdasarkan *reviewId*, *userName*, *userImage*, *content*, *score*, *thumbsUp*, *reviewCreated*, *at*, *replyContent*, *replyAt*, *sortOrder*, *appId*.

3.2.2 Pelabelan Data

Pelabelan data atau penentuan kelas atribut yang dilakukan sesuai dengan subjektivitas peneliti. Tahapan ini dilakukan dengan cara melakukan pengambilan data ulasan pengguna yang telah diambil pada saat pengambilan data di kolom komentar *google playstore* tiap objek dan pembagian kelas yaitu sentimen positif untuk *review* aplikasi *e-learning* yang bersifat positif dan sentimen negatif untuk *review* aplikasi *e-learning* yang bersifat negatif secara otomatis dengan bantuan *tools Google Colab*. Contoh dari tahap pelabelan data terdapat pada tabel 3.1 di bawah ini.

Tabel 3. 1 Identifikasi Masalah Saat Ini

| <i>Content</i> | Kategori | Data |
|---|----------|----------|
| Minta tolong diperbaiki segera di HP saya aplikasinya force close. Mohon bantuannya ya pihak developer ðŸ™• ðŸ• » | Negatif | Training |
| Aplikasinya aneh, masa setelah berlangganan besoknya gabisa di buka aplikasinya, mohon untuk di perbaiki. | Negatif | Training |
| Aplikasinya sudah bagus Saran aja min kalau bisa dibuat full screen di laptop/komputer supaya lebih jelas dan kalau boleh sertakan dark mode | Positif | Training |
| Saya suka banget dengan aplikasi pahamify karena fitur-fiturnya mudah dimengerti ketika mempelajari suatu materi... | Positif | Training |
| Aplikasinya sangat membantu untuk lebih memahami tentang materi pelajaran lebih mudah dibandingkan dengan disekolah | Positif | Training |
| Ketika saya buka aplikasi ini, tiba2 keluar sendiri. Padahal penyimpanan saya masih 14 GB. Cache juga sudah saya bersihkan. Saya sangat butuh aplikasi ini untuk belajar. Tolong bantuannya | Negatif | Testing |

3.2.3 Text Preprocessing

Tahap *Text Preprocessing* merupakan proses ekstraksi, membersihkan, memilih, dan memproses data untuk analisis lanjut dengan mengubah bentuk data yang tidak terstruktur menjadi bentuk data yang terstruktur dari kata dasar. Dilakukan proses seleksi data yang akan digunakan dalam penelitian ini yaitu data

content pada *dataset*. Proses ini dilakukan melalui *library* yang digunakan pada *google colab* menggunakan bahasa pemrograman *python*, proses tersebut terdiri dari *cleansing*, *case folding*, *stopword removal*, *tokenizing*, dan *stemming*.

Cleansing

Pada proses ini data ulasan dibersihkan menjadi format teks yang diinginkan, pada prosesnya ini meliputi penghapusan angka atau nomor, tanda baca, *emoticon*. Contoh hasil *cleansing* dapat dilihat pada Tabel 3.2

Tabel 3. 2 Hasil *Cleansing*

| Sebelum <i>cleansing</i> | Setelah <i>cleansing</i> |
|--|---|
| Minta tolong diperbaiki segera di HP saya aplikasinya force close. Mohon bantuannya ya pihak developer » | Minta tolong diperbaiki segera di HP saya aplikasinya force close Mohon bantuannya ya pihak developer |
| Aplikasinya aneh, masa setelah berlangganan besoknya gabisa di buka aplikasinya, mohon untuk di perbaiki. | Aplikasinya aneh masa setelah berlangganan besoknya gabisa di buka aplikasinya mohon untuk di perbaiki |
| Aplikasinya sudah bagus Saran aja min kalau bisa dibuat full screen di laptop/komputer supaya lebih jelas dan kalau boleh sertakan dark mode | Aplikasinya sudah bagus Saran aja min kalau bisa dibuat full screen di laptopkomputer supaya lebih jelas dan kalau boleh sertakan dark mode |
| Saya suka banget dengan aplikasi pahamify karena fitur-fitur nya mudah dimengerti ketika mempelajari suatu materi... | Saya suka banget dengan aplikasi pahamify karena fiturfitur nya mudah dimengerti ketika mempelajari suatu materi |
| Aplikasinya sangat membantu untuk lebih memahami tentang materi pelajaran lebih mudah dibandingkan dengan disekolah | Aplikasinya sangat membantu untuk lebih memahami tentang materi pelajaran lebih mudah dibandingkan dengan disekolah |

Case Folding

Proses ini mengubah huruf kapital atau huruf besar yang ada pada data menjadi huruf kecil atau menyamakan semua bentuk kata menjadi huruf kecil semua. Hasil *Case folding* ditampilkan pada Tabel 3.3

Tabel 3. 3 Hasil *Case Folding*

| Sebelum <i>Case Folding</i> | Setelah <i>Case Folding</i> |
|---|---|
| Minta tolong diperbaiki segera dizHP saya aplikasinya force close Mohon bantuannya ya pihak developer | minta tolong diperbaiki segera di hp saya aplikasinya force close mohon bantuannya ya pihak developer |
| Aplikasinya aneh masa setelah berlangganan besoknya gabisa di buka aplikasinya mohon untuk di perbaiki | aplikasinya aneh masa setelah berlangganan besoknya gabisa di buka aplikasinya mohon untuk di perbaiki |
| Aplikasinya sudah bagus Saran aja min kalau bisa dibuat full screen di laptopkomputer supaya lebih jelas dan kalau boleh sertakan dark mode | aplikasinya sudah bagus saran aja min kalau bisa dibuat full screen di laptopkomputer supaya lebih jelas dan kalau boleh sertakan dark mode |

| Sebelum <i>Case Folding</i> | Setelah <i>Case Folding</i> |
|---|---|
| Saya suka banget dengan aplikasi pahamify karena fiturfitur nya mudah dimengerti ketika mempelajari suatu materi | saya suka banget dengan aplikasi pahamify karena fiturfitur nya mudah dimengerti ketika mempelajari suatu materi |
| Aplikasinya sangat membantu untuk lebih memahami tentang materi pelajaran lebih mudah dibandingkan dengan disekolah | aplikasinya sangat membantu untuk lebih memahami tentang materi pelajaran lebih mudah dibandingkan dengan disekolah |

Stopword Removal

Dalam proses ini terdapat daftar kata umum yang tidak memiliki makna penting dalam sebagian kalimat atau dokumen, sehingga kata umum akan dihapus yang berfungsi untuk mengurangi jumlah kata yang disimpan oleh *corpus* atau sistem. Hasil *Stopword Removal* ditampilkan pada Tabel 3.4

Tabel 3. 4 Hasil *Stopword Removal*

| Sebelum <i>Stopword Removal</i> | Setelah <i>Stopword Removal</i> |
|---|--|
| minta tolong diperbaiki segera di hp saya aplikasinya force close mohon bantuannya ya pihak developer | tolong diperbaiki hp aplikasinya force close mohon bantuannya ya developer |
| aplikasinya aneh masa setelah berlangganan besoknya gabisa di buka aplikasinya mohon untuk di perbaiki | aplikasinya aneh berlangganan besoknya gabisa buka aplikasinya mohon perbaiki |
| aplikasinya sudah bagus saran aja min kalau bisa dibuat full screen di laptopkomputer supaya lebih jelas dan kalau boleh sertakan dark mode | aplikasinya bagus saran aja min full screen laptopkomputer sertakan dark mode |
| saya suka banget dengan aplikasi pahamify karena fiturfitur nya mudah dimengerti ketika mempelajari suatu materi | suka banget aplikasi pahamify fiturfitur nya mudah dimengerti mempelajari materi |
| aplikasinya sangat membantu untuk lebih memahami tentang materi pelajaran lebih mudah dibandingkan dengan disekolah | aplikasinya membantu memahami materi pelajaran mudah dibandingkan disekolah |

Tokenizing

Proses ini melakukan pemisahan setiap kata dalam suatu kalimat yang dipisahkan dengan tanda koma (,). Pemecahan kalimat dan kata. Hasil *Tokenizing* ditampilkan pada Tabel 3.5

Tabel 3. 5 Hasil *Tokenizing*

| Sebelum <i>Tokenizing</i> | Setelah <i>Tokenizing</i> |
|---|---|
| tolong diperbaiki hp aplikasinya force close mohon bantuannya ya developer | 'tolong', 'diperbaiki', 'hp', 'aplikasinya', 'force', 'close', 'mohon', 'bantuannya', 'ya', 'developer' |
| aplikasinya aneh berlangganan besoknya gabisa buka aplikasinya mohon perbaiki | 'aplikasinya', 'aneh', 'berlangganan', 'besoknya', 'gabisa', 'buka', 'aplikasinya', 'mohon', 'perbaiki' |
| aplikasinya bagus saran aja min full screen laptopkomputer sertakan dark mode | 'aplikasinya', 'bagus', 'saran', 'aja', 'min', 'full', 'screen', 'laptopkomputer', 'sertakan', 'dark', 'mode' |

| Sebelum <i>Tokenizing</i> | Setelah <i>Tokenizing</i> |
|--|---|
| suka banget aplikasi pahamify fiturfitur nya mudah dimengerti mempelajari materi | 'suka', 'banget', 'aplikasi', 'pahamify', 'fiturfitur', 'nya', 'mudah', 'dimengerti', 'mempelajari', 'materi' |
| aplikasinya membantu memahami materi pelajaran mudah dibandingkan disekolah | 'aplikasinya', 'membantu', 'memahami', 'materi', 'pelajaran', 'mudah', 'dibandingkan', 'disekolah' |

Stemming

Proses ini merubah kata-kata menjadi kata dasarnya (*stem word*) atau mengembalikan kata ke bentuk dasarnya dengan pedoman penggunaan kata Bahasa Indonesia. Hasil *Stemming* ditampilkan pada Tabel 3.6

Tabel 3. 6 Hasil *Stemming*

| Sebelum <i>Stemming</i> | Setelah <i>Stemming</i> |
|---|--|
| 'tolong', 'diperbaiki', 'hp', 'aplikasinya', 'force', 'close', 'mohon', 'bantuannya', 'ya', 'developer', 'aplikasinya', 'aneh', 'berlangganan', 'besoknya', 'gabisa', 'buka', 'aplikasinya', 'mohon', 'perbaiki', 'aplikasinya', 'bagus', 'saran', 'aja', 'min', 'full', 'screen', 'laptopkomputer', 'sertakan', 'dark', 'mode' | tolong baik hp aplikasi force close mohon bantu ya developer aplikasi aneh langgan besok gabisa buka aplikasi mohon baik aplikasi bagus saran aja min full screen laptopkomputer serta dark mode |
| 'suka', 'banget', 'aplikasi', 'pahamify', 'fiturfitur', 'nya', 'mudah', 'dimengerti', 'mempelajari', 'materi', 'aplikasinya', 'membantu', 'memahami', 'materi', 'pelajaran', 'mudah', 'dibandingkan', 'disekolah' | suka banget aplikasi pahamify fiturfitur nya mudah erti ajar materi aplikasi bantu paham materi ajar mudah banding sekolah |

3.2.4 Pembagian Data

Pada tahap ini, data dibagi menjadi data latih dan uji. Data latih digunakan untuk melatih algoritma *Naïve Bayes*, sedangkan data uji digunakan untuk melakukan uji penilaian pada *Naïve Bayes* setelah data dilatih. Untuk menyelesaikan proses ini, *google colab* menggunakan *library python Sklearn* dengan *train_test_split*. Untuk digunakan sebagai *input* dengan metode *Naive Bayes*, data harus dilabeli dan melalui proses *preprocessing*.

3.2.5 Klasifikasi *Naïve Bayes*

Tahapan ini dibuatkan sebuah model guna klasifikasi *dataset* komentar yang bertujuan untuk mengklasifikasikan data dengan menghitung apakah setiap kata dalam kalimat dianggap sebagai kalimat positif atau negatif. Data yang

digunakan harus sudah diproses di tahap *preprocessing* sebelumnya. Klasifikasi data pada penelitian ini memiliki fungsi untuk menentukan kelas pada data yaitu positif dan negatif. Contoh dari tahap klasifikasi *Naïve Bayes* terdapat pada tabel 3.7 di bawah ini.

Tabel 3. 7 Data Hasil *Text Preprocessing*

| Kategori | Content | Jumlah Kata |
|----------|---|-------------|
| Negatif | tolong baik hp aplikasi force close mohon bantu ya developer | 10 |
| Negatif | aplikasi aneh langgan besok gabisa buka aplikasi mohon baik | 9 |
| Positif | aplikasi bagus saran aja min full screen laptopkomputer serta dark mode | 11 |
| Positif | suka banget aplikasi pahamify fiturfitur nya mudah erti ajar materi | 10 |
| Positif | aplikasi bantu paham materi ajar mudah banding sekolah | 8 |

Sesudah proses *text preprocessing* selesai maka selanjutnya dilakukan perhitungan probabilitas ulasan positif dan negatif. Adapun banyaknya ulasan yang dipakai sebanyak 5, maka probabilitas ulasan positif sebesar $P(\text{positif}) = \frac{3}{5} = 0.6$ dan probabilitas ulasan negatif $P(\text{negatif}) = \frac{2}{5} = 0.4$ selanjutnya dilakukan pembagian setiap kata yang ada pada ulasan. Dari data hasil diatas diketahui bahwa jumlah kata yang ada pada data tersebut berjumlah 48 kata, dan masing-masing kata dari ulasan positif berjumlah 29 kata sedangkan kata dari ulasan negatif berjumlah 19 kata. Kemudian dihitung setiap kata dengan rumus *Naïve Bayes* sesuai dengan frekuensi kata pada data *training*.

Tabel 3. 8 Contoh Frekuensi Kata Uji pada Data *Training*

| No | Kata | Frekuensi <i>Training</i> Negatif | Frekuensi <i>Training</i> Positif |
|----|-----------|--------------------------------------|--------------------------------------|
| 1 | Tolong | 1 | 0 |
| 2 | Baik | 2 | 0 |
| 3 | Hp | 1 | 0 |
| 4 | Aplikasi | 3 | 3 |
| 5 | Force | 1 | 0 |
| 6 | Close | 1 | 0 |
| 7 | Mohon | 2 | 0 |
| 8 | Bantu | 1 | 1 |
| 9 | Ya | 1 | 0 |
| 10 | Developer | 1 | 0 |
| 11 | Aneh | 1 | 0 |
| 12 | Langgan | 1 | 0 |
| 13 | Besok | 1 | 0 |
| 14 | Gabisa | 1 | 0 |

| No | Kata | Frekuensi <i>Training Negatif</i> | Frekuensi <i>Training Positif</i> |
|----|----------------|--------------------------------------|--------------------------------------|
| 15 | Buka | 1 | 0 |
| 16 | Bagus | 0 | 1 |
| 17 | Saran | 0 | 1 |
| 18 | Aja | 0 | 1 |
| 19 | Min | 0 | 1 |
| 20 | Full | 0 | 1 |
| 21 | Screen | 0 | 1 |
| 22 | Laptopkomputer | 0 | 1 |
| 23 | Serta | 0 | 1 |
| 24 | Dark | 0 | 1 |
| 25 | Mode | 0 | 1 |
| 26 | Suka | 0 | 1 |
| 27 | Banget | 0 | 1 |
| 28 | Pahamify | 0 | 1 |
| 29 | Fiturfitur | 0 | 1 |
| 30 | Nya | 0 | 1 |
| 31 | Mudah | 0 | 2 |
| 32 | Erti | 0 | 1 |
| 33 | Ajar | 0 | 2 |
| 34 | Materi | 0 | 2 |
| 35 | Paham | 0 | 1 |
| 36 | Banding | 0 | 1 |
| 37 | Sekolah | 0 | 1 |

Berdasarkan dari data tabel 3.8 diatas mempunyai nilai yang dapat dicari nilai probabilitasnya dengan menggunakan rumus $P(W_k/C_i)$ yaitu sebagai berikut:

$$P(\text{Tolong}|\text{negatif}) = (1+1) / (19+48) = 0.02985$$

$$P(\text{Tolong}|\text{positif}) = (0+1) / (29+48) = 0.01298$$

$$P(\text{Bagus}|\text{negatif}) = (0+1) / (19+48) = 0.01492$$

$$P(\text{Bagus}|\text{positif}) = (1+1) / (29+48) = 0.02597$$

$$P(\text{Aplikasi}|\text{negatif}) = (3+1) / (19+48) = 0.05970$$

$$P(\text{Aplikasi}|\text{positif}) = (3+1) / (29+48) = 0.05194$$

Berdasarkan hasil dari perhitungan dengan menggunakan rumus $P(W_k/C_i)$ dapat dilihat sebagai berikut:

Tabel 3. 9 Contoh Hasil Probabilitas Kata

| No | Kata | $N_k \text{Negatif}$ | $N_k \text{Positif}$ | $P(W_k/C_i)$ <i>negatif</i> | $P(W_k/C_i)$ <i>positif</i> |
|----|--------|----------------------|----------------------|--------------------------------|--------------------------------|
| 1 | Tolong | 1 | 0 | 0.02985 | 0.01298 |
| 2 | Baik | 2 | 0 | 0.04477 | 0.01298 |

| No | Kata | $N_k Negatif$ | $N_k Positif$ | $P(W_k C_i)$ <i>negatif</i> | $P(W_k C_i)$ <i>positif</i> |
|----|----------------|---------------|---------------|--------------------------------|--------------------------------|
| 3 | Hp | 1 | 0 | 0.02985 | 0.01298 |
| 4 | Aplikasi | 3 | 3 | 0.05970 | 0.05194 |
| 5 | Force | 1 | 0 | 0.02985 | 0.01298 |
| 6 | Close | 1 | 0 | 0.02985 | 0.01298 |
| 7 | Mohon | 2 | 0 | 0.04477 | 0.01298 |
| 8 | Bantu | 1 | 1 | 0.02985 | 0.02597 |
| 9 | Ya | 1 | 0 | 0.02985 | 0.01298 |
| 10 | Developer | 1 | 0 | 0.02985 | 0.01298 |
| 11 | Aneh | 1 | 0 | 0.02985 | 0.01298 |
| 12 | Langgan | 1 | 0 | 0.02985 | 0.01298 |
| 13 | Besok | 1 | 0 | 0.02985 | 0.01298 |
| 14 | Gabisa | 1 | 0 | 0.02985 | 0.01298 |
| 15 | Buka | 1 | 0 | 0.02985 | 0.01298 |
| 16 | Bagus | 0 | 1 | 0.01492 | 0.02597 |
| 17 | Saran | 0 | 1 | 0.01492 | 0.02597 |
| 18 | Aja | 0 | 1 | 0.01492 | 0.02597 |
| 19 | Min | 0 | 1 | 0.01492 | 0.02597 |
| 20 | Full | 0 | 1 | 0.01492 | 0.02597 |
| 21 | Screen | 0 | 1 | 0.01492 | 0.02597 |
| 22 | Laptopkomputer | 0 | 1 | 0.01492 | 0.02597 |
| 23 | Serta | 0 | 1 | 0.01492 | 0.02597 |
| 24 | Dark | 0 | 1 | 0.01492 | 0.02597 |
| 25 | Mode | 0 | 1 | 0.01492 | 0.02597 |
| 26 | Suka | 0 | 1 | 0.01492 | 0.02597 |
| 27 | Banget | 0 | 1 | 0.01492 | 0.02597 |
| 28 | Pahamify | 0 | 1 | 0.01492 | 0.02597 |
| 29 | Fiturfitur | 0 | 1 | 0.01492 | 0.02597 |
| 30 | Nya | 0 | 1 | 0.01492 | 0.02597 |
| 31 | Mudah | 0 | 2 | 0.01492 | 0.03896 |
| 32 | Erti | 0 | 1 | 0.01492 | 0.02597 |
| 33 | Ajar | 0 | 2 | 0.01492 | 0.03896 |
| 34 | Materi | 0 | 2 | 0.01492 | 0.03896 |
| 35 | Paham | 0 | 1 | 0.01492 | 0.02597 |
| 36 | Banding | 0 | 1 | 0.01492 | 0.02597 |
| 37 | Sekolah | 0 | 1 | 0.01492 | 0.02597 |

Dari hasil probabilitas kata berdasarkan tabel 3.9 diatas maka bisa diketahui probabilitas perkata. Data *training* sebelumnya bisa digunakan untuk melihat hasil *test* model dengan mengalihkan semua nilai probabilitas. Berikut data *testing* yang digunakan "Ketika saya buka aplikasi ini, tiba2 keluar sendiri. Padahal penyimpanan saya masih 14 GB. Cache juga sudah saya bersihkan. Saya sangat butuh aplikasi ini untuk belajar. Tolong bantuannya" dan hasil perubahan setelah melalui *text preprocessing* yaitu "buka aplikasi simpan gb cache sih butuh aplikasi ajar tolong bantu".

Positif:

$$\begin{aligned}
&= P(\text{positif}) \times P(\text{buka}|\text{positif}) \times P(\text{aplikasi}|\text{positif}) \times P(\text{ajar}|\text{positif}) \times P(\text{tolong}|\text{positif}) \\
&\times P(\text{bantu}|\text{positif}). \\
&= 0.6 \times 0.01298 \times 0.05194 \times 0.03896 \times 0.01298 \times 0.02597 \\
&= 0.00000000531
\end{aligned}$$

Negatif:

$$\begin{aligned}
&= P(\text{negatif}) \times P(\text{buka}|\text{negatif}) \times P(\text{aplikasi}|\text{negatif}) \times P(\text{ajar}|\text{negatif}) \times \\
&P(\text{tolong}|\text{negatif}) \times P(\text{bantu}|\text{negatif}). \\
&= 0.4 \times 0.02985 \times 0.05970 \times 0.01492 \times 0.02985 \times 0.02985 \\
&= 0.00000000947
\end{aligned}$$

Perhitungan diatas merupakan proses klasifikasi sentimen berdasarkan hasil perhitungan probabilitas yang mempunyai nilai hasil tertinggi yaitu sebesar 0.00000000947 sehingga ulasan tersebut diklasifikasikan kedalam kelas ulasan “Negatif”.

3.2.6 Evaluasi Data

Evaluasi data dapat dilakukan untuk mengetahui keakuratan algoritma data, akurasi akan diukur dengan menggunakan *Confusion Matrix* seperti pada tabel 3.10 berikut.

Tabel 3. 10 Contoh Confusion Matrix

| Classification | Predicted Positive | Predicted Negative |
|------------------------------|--------------------|--------------------|
| <i>Actual Positive Cases</i> | 3 (TP) | 0 (FN) |
| <i>Actual Negative Cases</i> | 0 (FP) | 3 (TN) |

Accuracy

$$= 3 + 3 / 3 + 3 + 0 + 0$$

$$= 6 / 6$$

$$= 100\%$$

Precision

$$= 3 / 3 + 0$$

$$= 3 / 3$$

$$= 100\%$$

Recall

$$= 3 / 3 + 0$$

$$= 3 / 3$$

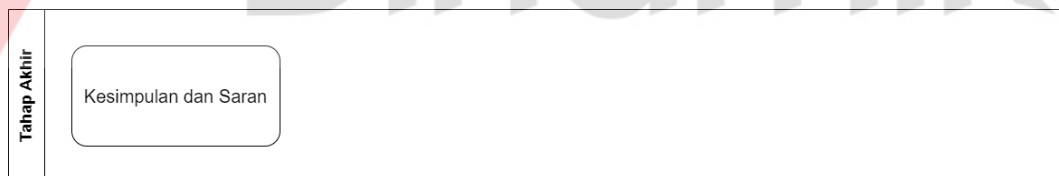
$$= 100\%$$

3.2.7 Visualisasi

Dilakukannya tahap visualisasi data adalah untuk menampilkan data hasil akhir dalam bentuk *wordcloud*, yang bertujuan untuk membuat hasil lebih mudah dipahami oleh peneliti dan pembaca. Visualisasi data menggunakan *wordcloud* menghasilkan representasi *visual* dari kata-kata yang paling sering muncul dalam teks yang telah dianalisis sebelumnya. Semakin besar suatu kata pada *wordcloud*, semakin sering kata tersebut muncul dalam kalimat atau teks tersebut. Visualisasi *wordcloud* menggunakan *library wordcloud* di *python*.

3.3 Tahap Akhir

Bagian ini menjelaskan tentang tahap akhir untuk menyelesaikan penelitian ini. Diagram alir pada tahap akhir berikut dapat dilihat pada Gambar 3.4



Gambar 3. 4 Diagram Alir Tahap Akhir

3.3.1 Kesimpulan dan Saran

Tahap kesimpulan bertujuan untuk mengetahui perbandingan hasil analisis sentimen dan hasil evaluasi pada 3 objek aplikasi *e-learning* yaitu Ruangguru, Pahamify, Merdeka Mengajar. Pada tahap saran, tujuannya adalah memberikan rekomendasi kepada pembaca agar dapat melakukan penelitian yang lebih baik atau menggunakan penelitian tersebut sebagai referensi untuk penelitian lain dengan topik yang serupa.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini akan mengulas hasil analisis berdasarkan data yang telah diproses dari situs *web play.google.com* yang berisi komentar tentang tiga *E-Learning* dan pembahasan tentang penelitian yang diterapkan mengikuti langkah-langkah yang telah ditetapkan sesuai dengan metode yang digunakan.

4.1 Analisis

Analisis dimulai dengan identifikasi masalah yang digunakan sebagai dasar dalam proses pengerjaan yang akan dimulai, pada identifikasi masalah diketahui bahwa adanya peningkatan kegiatan belajar mandiri secara *online* dengan menggunakan aplikasi *e-learning* dan terdapat tiga *e-learning* di Indonesia yang sering digunakan aplikasinya yaitu Ruangguru, Pahamify, dan Merdeka Mengajar. Ulasan pengguna juga merupakan salah satu komponen yang penting dalam keberlangsungan keperluan proses bisnis yang ada untuk mempertahankan kualitas suatu bisnis. Oleh karena itu dalam penelitian ini akan menganalisis apakah layanan *e-learning* yang diberikan sudah memenuhi kebutuhan dari pengguna atau belum dengan mengetahui ulasan positif dan negatif di tiap *e-learning*.

4.2 Pengambilan Data

Tahap pertama adalah pengambilan data. Pengambilan data itu sendiri digunakan untuk keperluan kebutuhan data yang dipakai dalam penelitian, dan pengambilan data dilakukan secara manual menggunakan bahasa pemrograman *python* dan dengan bantuan *google colab*.

Tabel 4. 1 Instalasi dan *Import Library*

| Code |
|---|
| <pre>!pip install google-play- scraper from google_play_scraper import app import pandas as pd import numpy as np</pre> |

Dalam *Python* “!pip” adalah perintah khusus yang digunakan untuk menginstal paket atau modul *python* melalui *pip*, memberikan instruksi kepada *python* untuk menjalankan perintah tersebut di terminal dan kemudian modul akan diinstal secara otomatis. Menginstal *library python* “*google-play-scraper*”. *Library* ini digunakan untuk mengekstrak data dari *google play store* seperti informasi aplikasi, ulasan, peringkat, dan lainnya. Kemudian melakukan *import* modul “*app*” dari *library* “*google_play_scraper*”. Dan mengimport *library pandas* untuk memanipulasi data dan *numpy* untuk melakukan perhitungan numerik pada Tabel 4.1.

Tabel 4. 2 Pengambilan Data

| Code |
|--|
| <pre>from google play scraper import Sort, reviews result, continuation token = reviews('com.ruangguru.livestudents', lang='id', country='id', sort=Sort.MOST RELEVANT, count=2000, filter score with=None)</pre> |

Selanjutnya melakukan proses *scrape* data dari *google play store* dengan mengurutkan data ulasan. Pertama, mengambil data ulasan berdasarkan alamat website aplikasi yang ada pada *google play store*. Kemudian “*lang*” untuk menentukan data yang berbahasa Indonesia, dan “*country*” untuk menentukan data yang berasal dari pengguna di Indonesia, “*sort*” untuk mengurutkan dari data yang paling relevan, dan “*count*” mengambil banyaknya data berjumlah 2000, “*filter score*” menunjukkan rating ulasan yang diambil seperti pada Tabel 4.2.

Tabel 4. 3 Pengambilan Kolom Review

```

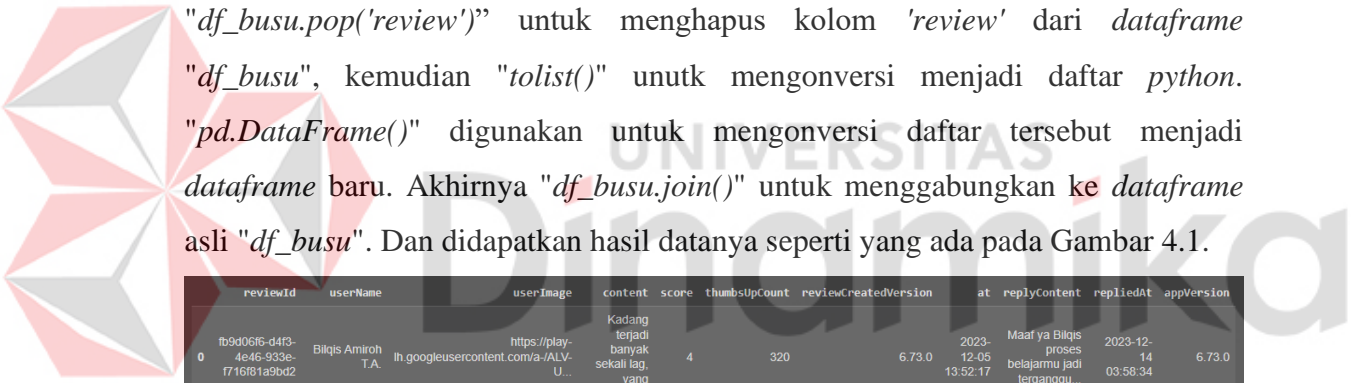
Code
df_busu = pd.DataFrame(np.array
(result), columns=['review'])

df_busu =
df_busu.join(pd.DataFrame
(df_busu.pop('review').tolist()))

df_busu.head()

```

Proses pada Tabel 4.3 bertujuan untuk mengonversi hasil ulasan dari *google play store* menjadi *dataframe pandas* dengan kolom yang sesuai. Hasil ulasan dari *google play store* yang ada pada variable “result” akan diubah menjadi *array numpy* “*np.array(result)*” dan dikonversikan menjadi *dataframe* pada variabel “*df_busu*”. “*df_busu.pop('review')*” untuk menghapus kolom 'review' dari *dataframe* “*df_busu*”, kemudian “*tolist()*” untuk mengonversi menjadi daftar *python*. “*pd.DataFrame()*” digunakan untuk mengonversi daftar tersebut menjadi *dataframe* baru. Akhirnya “*df_busu.join()*” untuk menggabungkan ke *dataframe* asli “*df_busu*”. Dan didapatkan hasil datanya seperti yang ada pada Gambar 4.1.



| | reviewId | userName | userImage | content | score | thumbsUpCount | reviewCreatedVersion | at | replyContent | repliedAt | appVersion |
|---|--------------------------------------|--------------------|--------------------------------------|---|-------|---------------|----------------------|---------------------|---|---------------------|------------|
| 0 | fb9d06f6-d4f3-4e46-933e-f716R1a9bd2 | Bilqis Amirah T.A. | lh.googleusercontent.com/a-/ALV-U... | Kadang terjadi banyak sekali lag, yang disuruh... | 4 | 320 | 6.73.0 | 2023-12-05 13:52:17 | Maaf ya Bilqis belajarmu jadi terganggu... | 2023-12-14 03:58:34 | 6.73.0 |
| 1 | 625310e3-1695-4f6a-9c9f-49d9ed660654 | Shafinna AZ Zahra | lh.googleusercontent.com/a-/ALV-U... | Aplikasinya suka nge-lag atau nge-hang seperti... | 4 | 201 | 6.72.0 | 2023-12-02 13:13:43 | Maaf ya Shafinna proses belajarmu jadi terganggu... | 2023-12-03 04:00:07 | 6.72.0 |
| 2 | 6bac5ff-fba2-441f-8c0b-961ddfcec0cd | Ken Ahmad | lh.googleusercontent.com/a-/ALV-U... | Ralat ya, sebelumnya aku udah kasih 5 bintang ... | 2 | 113 | 6.75.0 | 2024-01-08 15:38:32 | Maaf ya Ken proses belajarmu jadi terganggu. C... | 2024-01-09 03:39:20 | 6.75.0 |
| 3 | 1320a6fb-9eaf-48dc-84f1-7d3b5538a41b | Aqila Azka | lh.googleusercontent.com/a-/ALV-U... | Aplikasinya sangat berguna bagi para pelajar u... | 5 | 34 | 6.75.0 | 2024-01-10 13:49:14 | Terimakasih Aqila atas review positif untuk Ap... | 2024-01-11 03:19:52 | 6.75.0 |

Gambar 4. 1 Hasil Konversi Ulasan ke *Dataframe*

Berdasarkan dari hasil *scrape* data dan konversi ulasan menghasilkan beberapa kolom seperti pada Gambar 4.1 kemudian beberapa kolom tersebut akan dipilih kembali mana kolom yang akan digunakan pada proses selanjutnya seperti pada Tabel 4.4 berikut.

Tabel 4. 4 Pemilihan kolom tertentu

```
Code
df_busu[['userName',
'score', 'at',
'content']].head()
```

Proses pada Tabel 4.4 bertujuan untuk pemilihan kolom tertentu dari *dataframe* "df_busu" yang kemudian disusun kembali berdasarkan kolom yang diinginkan, yaitu kolom *userName*, *score*, *at*, dan *content*. Dan hasilnya seperti pada Gambar 4.2 berikut.

| | userName | score | at | content |
|---|----------------------|-------|---------------------|---|
| 0 | Bilqis Amiroh T.A. | 4 | 2023-12-05 13:52:17 | Kadang terjadi banyak sekali lag, yang disuruh... |
| 1 | Shafinna Az Zahra | 4 | 2023-12-02 13:13:43 | Aplikasinya suka nge-lag atau nge-hang seperti... |
| 2 | Ken Ahmad | 2 | 2024-01-08 15:38:32 | Ralat ya, sebelumnya aku udah kasih 5 bintang ... |
| 3 | Aqila Azka | 5 | 2024-01-10 13:49:14 | Aplikasinya sangat berguna bagi para pelajar u... |
| 4 | Alfiyan Fadlurrahman | 3 | 2024-01-09 06:07:21 | tambahkan fitur tap tahan lama di layar untuk ... |

Gambar 4. 2 Hasil Pemilihan kolom

Selanjutnya data diurutkan kembali berdasarkan tanggal terbaru seperti pada Tabel 4.5 berikut.

Tabel 4. 5 Mengurutkan Data Terbaru

```
Code
new_df = df_busu[['userName',
'score', 'at', 'content']]
sorted_df =
new_df.sort_values(by='at',
ascending=False)
sorted_df.head()
```

Proses pada Tabel 4.5 bertujuan untuk mengurutkan *dataframe* baru berdasarkan nilai kolom 'at'. Dalam programnya sendiri "new_df" dibuat sebagai *dataframe* baru dan terdiri dari kolom yang dipilih dari *dataframe* sebelumnya "df_busu". Dibuat "sorted_df" dengan mengurutkan "new_df" berdasarkan nilai kolom 'at' secara menurun sehingga nilai yang lebih besar atau tanggal yang terbaru akan muncul terlebih dahulu. Dan hasilnya seperti pada Gambar 4.3 berikut.

| | userName | score | at | content |
|-----|---------------------|-------|---------------------|---|
| 105 | Muhammad Iqbal | 5 | 2024-01-28 13:29:12 | Saya berlangganan Brain Academy. Pelajaran mud... |
| 85 | Omah Yeti | 2 | 2024-01-27 07:10:21 | Karena gak bisa final battle gak bisa main zon... |
| 807 | Dimas Siswa prayoga | 5 | 2024-01-26 12:58:37 | Alhamdulillah bagus kak aplikasinya saran ku d... |
| 201 | Namira Kumiasari | 5 | 2024-01-26 04:30:20 | Suka bangettt, aku jadi bisa meningkatkan ilmu... |
| 806 | Prijayamuda | 5 | 2024-01-25 22:19:04 | Wah, seru banget, soalnya saya nonton sambil b... |

Gambar 4. 3 Hasil Pengurutan Data Terbaru

Dataframe baru digunakan dan dari kolom yang sudah dipilah diproses sebelumnya pada Tabel 4.4 kolom yang akan digunakan selanjutnya hanya kolom *content* dan *score* saja seperti pada Tabel 4.6 berikut.

Tabel 4. 6 Membuat dan Memilah 2 Kolom

Code

```
my_df = sorted_df[['userName',
'score', 'at', 'content']]
my_df=my_df[['content',
'score']]
```

Proses pada Tabel 4.6 bertujuan untuk membuat *dataframe* baru yang hanya berisi dua kolom dari *dataframe* yang telah diurutkan sebelumnya. Dibuatnya "*my_df*" sebagai *dataframe* baru ini terdiri dari kolom yang seperti ada pada "*sorted_df*" dan kemudian kali ini "*my_df*" hanya akan terdiri dari dua kolom yaitu 'content' dan 'score'. Hasil dari proses akan ditunjukkan pada Gambar 4.4 berikut.

| | content | score |
|-----|---|-------|
| 11 | Sudah berkali kali belanja disini, pesanan sel... | 1 |
| 387 | enakan games pet yang dulu dibandingkan sekarang. | 4 |
| 451 | Aplikasi tiba ² crash/tidak bisa masuk ke apk t... | 5 |
| 88 | Seneng makenya. Dan memudahkan. Tapi keamanann... | 1 |
| 553 | kenapa yah baru baru ini pengiriman j&t super ... | 2 |

Gambar 4. 4 Hasil Proses Penyisahan Kolom

4.3 Pelabelan Data

Pelabelan data dilakukan otomatis dengan bantuan *tools google colab*, dengan menggunakan *command if, elif* dengan nilai *score* yang sudah ditentukan. Implementasi penulisan code ditampilkan pada Tabel 4.7 berikut.

Tabel 4. 7 Pelabelan Data

```
Code
def pelabelan(score):
    if score < 3:
        return 'Negatif'
    elif score == 4:
        return 'Positif'
    elif score == 5:
        return 'Positif'
my_df['Label'] = my_df
['score'].apply(pelabelan)
my_df.head(50)
```

Proses pada Tabel 4.7 bertujuan untuk menambahkan kolom baru 'Label' ke dalam *dataframe* berdasarkan nilai dalam kolom "score". Terdapat "pelabelan(score)" untuk memberikan label terhadap nilai "score". Jika 'score' kurang dari 3 dinyatakan kedalam label negatif, sedangkan 'score' yang sama dengan 4 dan 5 berlabel positif, pada "apply(pelabelan)" digunakan untuk fungsi pelabelan pada kolom "score" dan hasilnya akan disimpan pada kolom "Label". Dan hasil dari proses pelabelan data akan ditunjukkan pada Gambar 4.5.

| | content | score | Label |
|-----|---|-------|---------|
| 11 | Sudah berkali kali belanja disini, pesanan sel... | 1 | Negatif |
| 387 | enakan games pet yang dulu dibandingkan sekarang. | 4 | Positif |
| 451 | Aplikasi tiba² crash/tidak bisa masuk ke apk t... | 5 | Positif |
| 88 | Seneng makanya. Dan memudahkan. Tapi keamanann... | 1 | Negatif |
| 553 | kenapa yah baru baru ini pengiriman j&t super ... | 2 | Negatif |
| 57 | kecewa. para pedagang nya kayak nya tipu tipu.... | 1 | Negatif |
| 166 | mudah cara co nya, banyak diskon nya terutama ... | 5 | Positif |
| 53 | Shopee memang sangat bagus Tapi SPX nya sangat... | 5 | Positif |
| 1 | Tidak banyak kata. Banyak iklan Bosque... dan ... | 1 | Negatif |
| 557 | Kqlau bisa expedisinya kita yg nentuin dong..... | 5 | Positif |
| 564 | Pengiriman shopee terlalu lama dan banyak tran... | 1 | Negatif |
| 95 | Susah login,, harus verifikasi wajah tapi gaga... | 2 | Negatif |
| 2 | Aplikasi nya bagus utuk beli barang yg kita ma... | 1 | Negatif |
| 89 | Semakin kesini semakin banyak kendala nya teru... | 3 | None |
| 34 | Masih terkendala di aplikasi yang lemot ini,se... | 2 | Negatif |

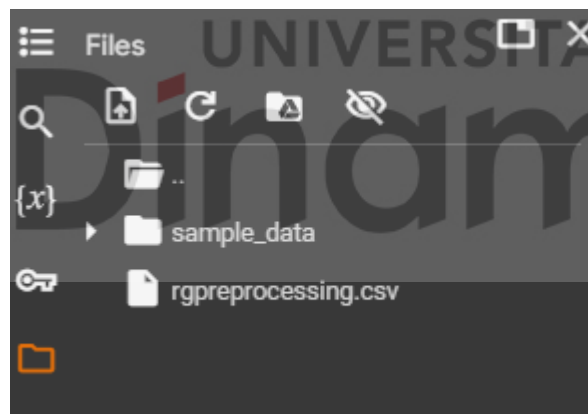
Gambar 4. 5 Hasil Pelabelan Data

Kemudian data *scrape* yang sudah melalui proses pemilahan kolom dan pelabelan data yang digunakan akan disimpan kedalam file bentuk format *.csv* dan tersimpan pada *cloud* di *google colab*. Hasil file *.csv* yang sudah tersimpan nantinya bisa diakses pada menu files yang terkoneksi dengan akun *google drive*. Implementasi penulisan code akan ditampilkan pada Tabel 4.8 berikut.

Tabel 4. 8 Simpan Data

| Code |
|--|
| <pre>my df.to csv("rgpreprocessing .csv", index = False)</pre> |

Proses pada Tabel 4.8 bertujuan untuk menyimpan *dataframe* ke dalam file *csv*. Nama file yang digunakan untuk menyimpan data yaitu "*rgpreprocessing.csv*" dan data akan disimpan dalam bentuk file *.csv* dan tanpa menyertakan indeks dalam file. Dan hasil dari proses menyimpan data akan ditunjukkan pada Gambar 4.6.



Gambar 4. 6 Hasil Menyimpan Data

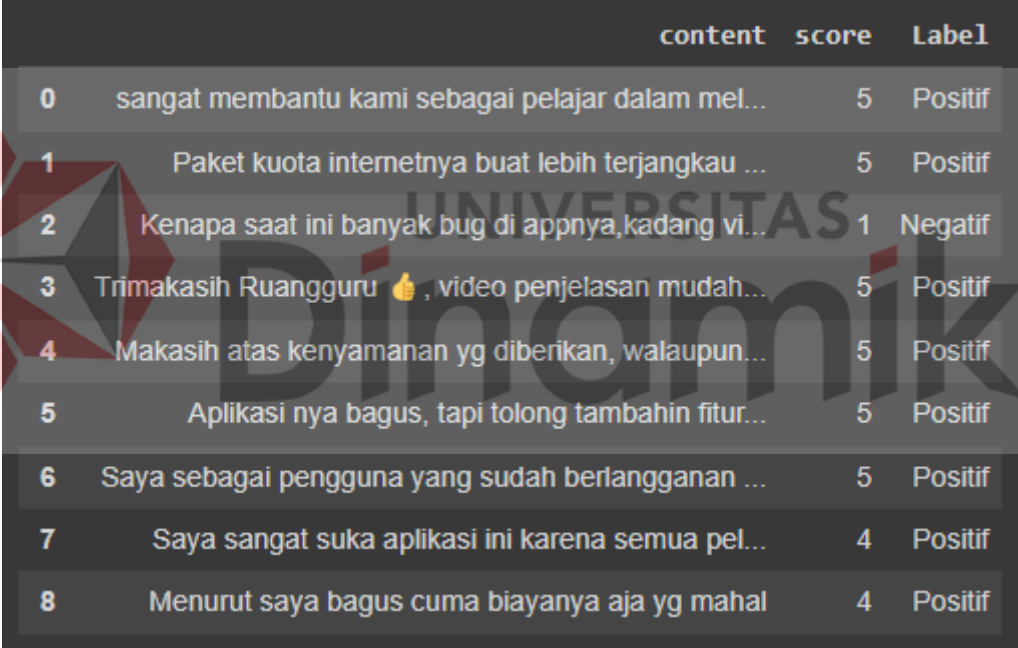
4.4 Text Pre-Processing

Tahap *Text Preprocessing* merupakan proses ekstraksi, membersihkan, memilih, dan memproses data yang akan dilakukan menjadi 5 tahap. Dalam langkah *Text Preprocessing*, dengan *Google Colab* digunakan sebagai bahasa pemrograman *Python*. *Text Preprocessing* bertujuan untuk menjamin bahwa data yang digunakan dalam analisis memiliki kualitas yang terbaik, relevan, dan siap untuk dipergunakan. Tahap atau langkah-langkah *Text Pre-processing* data dilakukan sebagai berikut:

Tabel 4. 9 Mengambil Data dari Penyimpanan *Cloud Colab*

```
Code
df = pd.read_csv('/content/
rgpreprocessing.csv')
df.head(50)
```

Setelah menyimpan data proses selanjutnya bertujuan untuk membaca file *csv* dengan nama "*rgpreprocessing.csv*" yang berada di dalam direktori */content/* pada *google colab*. "*read_csv*" digunakan untuk membaca file *csv* dan mengonversinya menjadi *dataframe* dan disimpan di variabel "*df*". Hasil dari proses pengambilan data akan ditunjukkan pada Gambar 4.7. Setelah selesai, kemudian dilanjutkan dengan implementasi beberapa tahapan yang ada di *text preprocessing* data.



| | content | score | Label |
|---|--|-------|---------|
| 0 | sangat membantu kami sebagai pelajar dalam mel... | 5 | Positif |
| 1 | Paket kuota internetnya buat lebih terjangkau ... | 5 | Positif |
| 2 | Kenapa saat ini banyak bug di appnya,kadang vi... | 1 | Negatif |
| 3 | Trimakasih Ruangguru 👍 , video penjelasan mudah... | 5 | Positif |
| 4 | Makasih atas kenyamanan yg diberikan, walaupun... | 5 | Positif |
| 5 | Aplikasi nya bagus, tapi tolong tambahin fitur... | 5 | Positif |
| 6 | Saya sebagai pengguna yang sudah berlangganan ... | 5 | Positif |
| 7 | Saya sangat suka aplikasi ini karena semua pel... | 4 | Positif |
| 8 | Menurut saya bagus cuma biayanya aja yg mahal | 4 | Positif |

Gambar 4. 7 Pengambilan dari *Cloud*

Cleansing

Cleansing dilakukan dengan menghapus angka, tanda baca, *url*. *Cleansing* menggunakan bantuan *Regular expression*. Implementasi penulisan code *cleansing* ditampilkan pada Tabel 4.10 berikut.

Tabel 4. 10 *Cleansing*

Code

```
import re
def clean_text(df,
text field,
new text field name):
    my df[new text field
name] = my df[new text field
name].apply(lambda elem:
re.sub(r"(@[A-Za-z0-9]+ |
([^\0-9A-Za-z
\t]) | (\w+:\//\S+)|^rt|http.+?"
, "", elem))
    # remove numbers
    my df[new text field name]
= my df[new text field name]
.apply(lambda elem:
re.sub(r"\d+", "", elem))
    return my df
```

Library re(regular expressions) di *import* terlebih dahulu, lalu fungsi “*clean_text*” didefinisikan dengan tiga parameter yang berisikan *dataframe*, kolom yang akan dicleansing, dan kolom baru sebagai hasil *cleansing*. Lalu “*apply()*” digunakan untuk fungsi *cleansing* pada kolom teks. Adapun “*re.sub()*” digunakan untuk mengganti pola tertentu sehingga membersihkan teks dari karakter-karakter yang tidak diinginkan, seperti tanda @, url, karakter *non-alphanumeric*, angka, dan sebagainya.

Case Folding

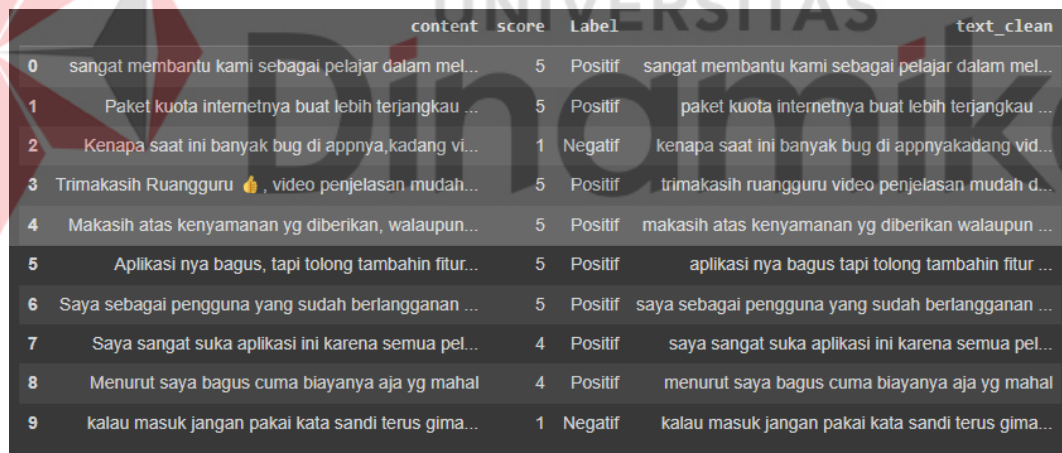
Tahap berikutnya merupakan *Case Folding*, pada tahap ini dilakukan perubahan setiap ulasan yang memiliki huruf besar atau *uppercase* diubah menjadi huruf kecil atau *lowercase* menggunakan fungsi *.str.lower()*. Implementasi penulisan code untuk langkah *case folding* ditampilkan pada Tabel 4.11 berikut.

Tabel 4. 11 *Case Folding*

```
Code
my df[new text field name] =
my df[text field].str.lower()

my df['text_clean'] =
my df['content'].str.lower()
my df['text_clean']
data_clean = clean_text(my_df,
'content', 'text_clean')
data_clean.head(10)
```

Berdasarkan dari penulisan code yang diterapkan kolom baru "text_clean" dibuat dan berisikan teks dari kolom "content" yang sudah diubah menjadi huruf kecil menggunakan fungsi "str.lower()". Kemudian fungsi "clean_text" digunakan untuk membersihkan teks dalam dataframe dan hasilnya akan disimpan dalam variabel "data_clean". Dan hasilkan data cleansing dan case folding seperti pada Gambar 4.8 berikut.



| | content | score | Label | text_clean |
|---|--|-------|---------|---|
| 0 | sangat membantu kami sebagai pelajar dalam mel... | 5 | Positif | sangat membantu kami sebagai pelajar dalam mel... |
| 1 | Paket kuota internetnya buat lebih terjangkau ... | 5 | Positif | paket kuota internetnya buat lebih terjangkau ... |
| 2 | Kenapa saat ini banyak bug di appnya,kadang vi... | 1 | Negatif | kenapa saat ini banyak bug di appnyakadang vid... |
| 3 | Trimakasih Ruangguru 🙌 , video penjelasan mudah... | 5 | Positif | trimakasih ruangguru video penjelasan mudah d... |
| 4 | Makasih atas kenyamanan yg diberikan, walaupun... | 5 | Positif | makasih atas kenyamanan yg diberikan walaupun ... |
| 5 | Aplikasi nya bagus, tapi tolong tambahin fitur... | 5 | Positif | aplikasi nya bagus tapi tolong tambahin fitur ... |
| 6 | Saya sebagai pengguna yang sudah berlangganan ... | 5 | Positif | saya sebagai pengguna yang sudah berlangganan ... |
| 7 | Saya sangat suka aplikasi ini karena semua pel... | 4 | Positif | saya sangat suka aplikasi ini karena semua pel... |
| 8 | Menurut saya bagus cuma biayanya aja yg mahal | 4 | Positif | menurut saya bagus cuma biayanya aja yg mahal |
| 9 | kalau masuk jangan pakai kata sandi terus gima... | 1 | Negatif | kalau masuk jangan pakai kata sandi terus gima... |

Gambar 4. 8 Hasil *Case Folding Text Pre-Processing*

Stopword Removal

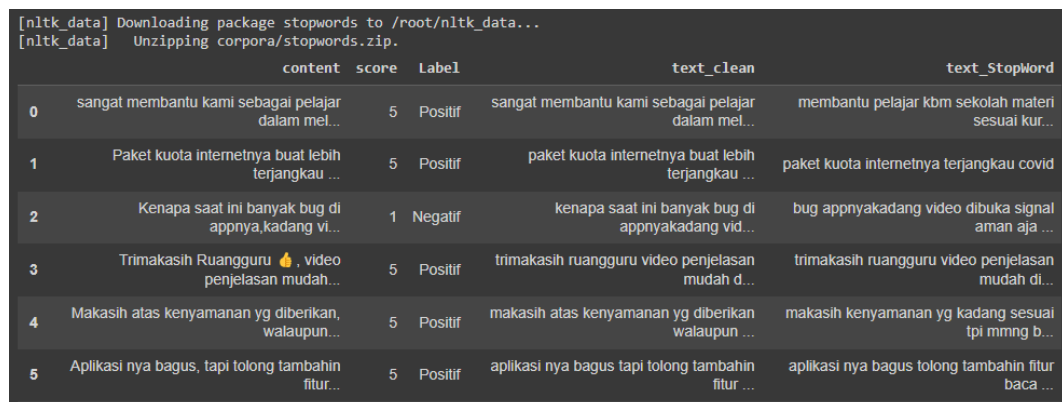
Tahap ini bertujuan untuk menghapus atau menghilangkan kata umum yang dianggap tidak memiliki makna penting dalam sebagian kalimat atau dokumen sehingga informasi lebih mudah kelola sistem. Setiap ulasan kata akan dibandingkan dengan kamus stopwords yang tersedia di *library corpus / stopwords*. *import library nltk* digunakan untuk membersihkan atau menyaring data sehingga fokus dapat diberikan pada kata-kata yang memiliki nilai informasi, menghapus

kata yang kurang memiliki makna penting, seperti kata penghubung, kata depan. Implementasi penulisan code untuk deklarasi variabel *stopwords* ditampilkan pada Tabel 4.12 berikut.

Tabel 4. 12 *Stopword Removal*

| Code |
|---|
| <pre>import nltk.corpus nltk.download('stopwords') from nltk.corpus import stopwords stop = stopwords.words('indonesian') data_clean['text_StopWord'] = data_clean['text_clean'].apply(lambda x:' '.join([word for word in x.split() if word not in (stop)])) data_clean.head(50)</pre> |

Mengimport *corpus* dari *library nltk (natural language toolkit)* dan mengunduh *stopword*, serta mengimpor *stopword* bahasa Indonesia yang telah diunduh sebelumnya. Selanjutnya menghapus *stopword* dari teks yang ada pada kolom *text_clean* dalam *dataframe data_clean*, dengan adanya “*apply()*” untuk menerapkan fungsi pada kolom *text_clean*. Adapun fungsi “*split()*” untuk membagi teks menjadi kata-kata individual, kemudian setiap kata diproses dengan menghapus kata-kata yang ada dalam daftar *stopword* “*stop*”, serta fungsi “*join()*” untuk menggabungkan kembali dengan spasi sebagai pemisah. Hasil data *stopword removal* seperti pada Gambar 4.9 berikut.



| | content | score | Label | text_clean | text_Stopword |
|---|---|-------|---------|---|---|
| 0 | sangat membantu kami sebagai pelajar dalam mel... | 5 | Positif | sangat membantu kami sebagai pelajar dalam mel... | membantu pelajar kbm sekolah materi sesuai kur... |
| 1 | Paket kuota internetnya buat lebih terjangkau ... | 5 | Positif | paket kuota internetnya buat lebih terjangkau ... | paket kuota internetnya terjangkau covid |
| 2 | Kenapa saat ini banyak bug di appnya,kadang vi... | 1 | Negatif | kenapa saat ini banyak bug di appnyakadang vid... | bug appnyakadang video dibuka signal aman aja ... |
| 3 | Trimakasih Ruangguru 🙌, video penjelasan mudah... | 5 | Positif | trimakasih ruangguru video penjelasan mudah d... | trimakasih ruangguru video penjelasan mudah di... |
| 4 | Makasih atas kenyamanan yg diberikan, walaupun... | 5 | Positif | makasih atas kenyamanan yg diberikan walaupun ... | makasih kenyamanan yg kadang sesuai tpi mmng b... |
| 5 | Aplikasi nya bagus, tapi tolong tambahin fitur... | 5 | Positif | aplikasi nya bagus tapi tolong tambahin fitur ... | aplikasi nya bagus tolong tambahin fitur baca ... |

Gambar 4. 9 Hasil *Stopword Removal Text Pre-Processing*

Tokenizing

Tahap *Tokenizing* bertujuan untuk pemecahan kalimat menjadi kata-kata atau token berdasarkan spasi untuk mempermudah pemrosesan teks dan analisis. Penggunaan *function* dalam *library* dapat membantu untuk melakukan pemisahan kata-kata untuk dijadikan token. Implementasi penulisan code *tokenizing* ditampilkan pada Tabel 4.13 berikut.

Tabel 4. 13 *Tokenizing*

Code

```
import nltk
nltk.download('punkt')
from nltk.tokenize import sent_tokenize,
word_tokenize
data_clean['text_tokens'] =
data_clean['text_StopWord'].apply(lambda
x: word_tokenize(x))
data_clean.head()
```

Berdasarkan dari penulisan code, library *nltk* di *import* terlebih dahulu kemudian unduh modul yang diperlukan untuk tokenisasi '*punkt*' dan selanjutnya *import* fungsi "*sent_tokenize*" yang digunakan untuk memisahkan teks menjadi kalimat dan "*word_tokenize*" yang digunakan untuk memisahkan kalimat menjadi kata-kata. Dan akhirnya teks dalam kolom '*text_StopWord*' dipisahkan menjadi kata menggunakan fungsi "*word_tokenize*" dan akan disimpan hasilnya pada kolom '*text_tokens*'. Penerapan fungsi "*word_tokenize*" dilakukan menggunakan "*apply()*"

pada kolom 'text_StopWord'. Hasil data *tokenizing* seperti pada Gambar 4.10 berikut.

| | content | score | Label | text_clean | text_StopWord | text_tokens |
|---|--|-------|---------|---|---|---|
| 0 | sangat membantu kami sebagai pelajar dalam mel... | 5 | Positif | sangat membantu kami sebagai pelajar dalam mel... | membantu pelajar kbm sekolah materi sesuai kur... | [membantu, pelajar, kbm, sekolah, materi, sesu... |
| 1 | Paket kuota internetnya buat lebih terjangkau ... | 5 | Positif | paket kuota internetnya buat lebih terjangkau ... | paket kuota internetnya terjangkau covid | [paket, kuota, internetnya, terjangkau, covid] |
| 2 | Kenapa saat ini banyak bug di appnya,kadang vi... | 1 | Negatif | kenapa saat ini banyak bug di appnyakadang vid... | bug appnyakadang video dibuka signal aman aja ... | [bug, appnyakadang, video, dibuka, signal, ama... |
| 3 | Trimakasih Ruangguru 🙏 , video penjelasan mudah... | 5 | Positif | trimakasih ruangguru video penjelasan mudah d... | trimakasih ruangguru video penjelasan mudah di... | [trimakasih, ruangguru, video, penjelasan, mud... |
| 4 | Makasih atas kenyamanan yg diberikan, walaupun... | 5 | Positif | makasih atas kenyamanan yg diberikan walaupun ... | makasih kenyamanan yg kadang sesuai tpi mmng b... | [makasih, kenyamanan, yg, kadang, sesuai, tpi... |

Gambar 4. 10 Hasil *Tokenizing Text Pre-Processing*

Stemming

Tahap *Stemming* merupakan proses mengubah kata-kata menjadi bentuk dasarnya menggunakan *library Sastrawi*. *Sastrawi* adalah *library stemming Python* yang didesain khusus untuk bahasa Indonesia. Dalam proses ini, kata-kata dalam data akan disederhanakan kembali ke bentuk dasarnya dengan menghapus imbuhan dan awalan. Implementasi penulisan code instalasi *library python* ditampilkan pada Tabel 4.14 berikut.

Tabel 4. 14 Instalasi *Library*

Code

```
!pip install Sastrawi
from
Sastrawi.Stemmer.StemmerFactory
import StemmerFactory
factory = StemmerFactory()
stemmer =
factory.create_stemmer()
```

Menginstal *library Sastrawi*, lalu mengimpor *StemmerFactory* yang digunakan untuk melakukan *stemming* dari modul *Stemmer* dalam *library Sastrawi*, kemudian 'factory' akan dibuat sebagai objek dari 'StemmerFactory' dan akan digunakan untuk membuat objek *stemmer*. Kemudian fungsi "create_stemmer()" akan membuat objek 'Stemmer' dari objek 'factory' yang akan digunakan untuk melakukan proses *stemming* pada teks dalam bahasa Indonesia. Implementasi penulisan code *stemming* ditampilkan pada Tabel 4.15 berikut.

Tabel 4. 15 *Stemming*

Code

```
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}
hitung=0

for document in
data_clean['text_tokens']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

print(len(term_dict))
print("-----")
for term in term_dict:
    term_dict[term] =
stemmed_wrapper(term)
    hitung+=1
    print(hitung, ":", term, ":"
, term_dict[term])

print(term_dict)
print("-----")

def get_stemmed_term(document):
    return [term_dict[term] for term
in document]

data_clean['text_stemindo'] =
data_clean['text_tokens'].apply(lambda
x: ' '.join(get_stemmed_term(x)))
data_clean.head(20)
```

Stemming dilakukan dengan mendefinisikan fungsi *stemmed_wrapper()* yang berfungsi memanggil *stem()* dari *stemmer* untuk proses *stemming* pada kata. '*term_dict*' akan digunakan untuk menyimpan kata-kata unik dan kata yang sudah di-stem dan '*hitung*' untuk menghitung jumlah kata yang di-stem. Kemudian dari dokumen yang ada dalam kolom '*text_tokens*' pada *dataframe* '*data_clean*' setiap kata dalam dokumen akan dicek apakah sudah ada dalam *term_dict*, jika belum kata tersebut ditambahkan ke dalam *term_dict*. Kemudian menghitung panjang *term_dict* dan mencetaknya. Dari tiap kata yang ada dalam *term_dict* akan digunakan fungsi *stemmed_wrapper()* untuk di-stem dan kata yang sudah di-stem akan ditambah ke *term_dict* untuk dicetak bersama dengan kata aslinya dan hasil stemmingnya. Adapun fungsi *get_stemmed_term()* untuk menghasilkan daftar kata

yang telah di-*stem* dari setiap dokumen dalam kolom '*text_tokens*'. Menyimpan hasil *stemming* dalam kolom baru '*text_steamindo*' dalam *data_clean*. Berdasarkan dari penulisan *code* yang diterapkan menghasilkan data *stemming* seperti pada Gambar 4.11 dibawah

```
6269 : mudahtinggal : mudahtinggal
6270 : pelajaranselain : pelajaranselain
6271 : didapatsaya : didapatsaya
6272 : iniikalo : iniikalo
6273 : raishanyzoeb : raishanyzoeb
{'membantu': 'bantu', 'pelajar': 'ajan', 'kbm': 'kbm', 'sekolah': 'sekolah', 'materi': 'materi', 'sesuai': 'sesuai', 'kurikulum':
-----
```

| | content | score | Label | text_clean | text_StopWord | text_tokens | text_steamindo |
|---|---|-------|---------|---|---|---|---|
| 0 | sangat membantu kami sebagai pelajar dalam mel... | 5 | Positif | sangat membantu kami sebagai pelajar dalam mel... | membantu pelajar kbm sekolah materi sesuai kur... | [membantu, pelajar, kbm, sekolah, materi, sesu... | bantu ajar kbm sekolah materi sesuai kurikulum |
| 1 | Paket kuota internetnya buat lebih terjangkau ... | 5 | Positif | paket kuota internetnya buat lebih terjangkau ... | paket kuota internetnya terjangkau covid | [paket, kuota, internetnya, terjangkau, covid] | paket kuota internetnya terjangkau covid |
| 2 | Kenapa saat ini banyak bug di appnya,kadang vl... | 1 | Negatif | kenapa saat ini banyak bug di appnyakadang vid... | bug appnyakadang video dibuka signal aman aja ... | [bug, appnyakadang, video, dibuka, signal, ama... | bug appnyakadang video buka signal aman aja pe... |
| 3 | Trimakasih Ruangguru 🍌, video penjelasan mudah... | 5 | Positif | trimakasih ruangguru video penjelasan mudah d... | trimakasih ruangguru video penjelasan mudah di... | [trimakasih, ruangguru, video, penjelasan, mud... | trimakasih ruangguru video jelas mudah erti ba... |
| 4 | Makasih atas kenyamanan yg diberikan, walaupun... | 5 | Positif | makasih atas kenyamanan yg diberikan walaupun ... | makasih kenyamanan yg kadang sesuai tpi mmng b... | [makasih, kenyamanan, yg, kadang, sesuai, tpi,... | makasih nyaman yg kadang sesuai tpi mmng byak ... |

Gambar 4. 11 Hasil *Stemming Text Pre-Processing*

4.5 Pembagian Data

Setelah melakukan tahapan *text pre-processing* selanjutnya akan melakukan pembagian data latih dan data uji dilakukan dengan perbandingan 80:20 untuk total data. Data uji digunakan untuk menguji kinerja model atau algoritma yang digunakan, sedangkan data latih digunakan untuk melatih model atau algoritma. Pembagian data dilakukan *function* pada *library*. Implementasi penulisan *code* pembagian data ditampilkan pada Tabel 4.16 berikut.

Tabel 4. 16 Pembagian Data

Code

```
from sklearn.model_selection
import train_test_split
X_train, X_test,
y_train, y_test =
train_test_split(data_clean
['content'],
data_clean['Label'],

test_size = 0.20,

random_state = 0)
```

Pembagian data dilakukan dengan fungsi *train_test_split* untuk membagi dataset menjadi data latih dan data uji dari modul *model_selection* pada library *Sklearn (scikit-learn)*. Dari fungsi *train_test_split* terdapat empat set data: *X_train* (atribut data latih), *X_test* (atribut data uji), *y_train* (label data latih), dan *y_test* (label data uji). Data ini akan digunakan untuk melatih model dan menguji performanya. Pada “*data_clean['content']*” ini sebagai atribut(‘content’) dari dataset yang akan digunakan untuk melakukan prediksi, sedangkan pada “*data_clean['Label']*” ini sebagai target atau label(‘Label’) dari dataset yang akan diprediksi. Di penelitian ini ‘*test_size*’ menjadi 20% dari keseluruhan dataset, dan ‘*random_state*’ untuk menentukan nilai acak yang digunakan oleh algoritma untuk melakukan pembagian data, nilai acak yang digunakan adalah 0.

4.6 Implementasi Klasifikasi *Naïve Bayes*

Pada proses klasifikasi ini, yang dilakukan menggunakan algoritma *Naive Bayes*, dibagi menjadi dua proses: latih dan uji. Pertama algoritma dilatih untuk mengenali dan mempelajari data menggunakan proses pelatihan data. Setelah itu, dilakukan prediksi pada data pengujian, dan kemudian dilakukan pengujian dengan mempertimbangkan kemungkinan pelatihan dataset. Implementasi penulisan code *naïve bayes* ditampilkan pada Tabel 4.17 berikut.

Tabel 4. 17 *Naive Bayes*

Code

```
from sklearn.naive_bayes
import MultinomialNB

nb = MultinomialNB()
nb.fit(tfidf_train, y_train)
y_pred =
nb.predict(tfidf_test)
```

Klasifikasi *Naive Bayes* dilakukan dengan mengimpor *MultinomialNB* dari modul *naive_bayes* di *library Sklearn (scikit-learn)*. Objek '*nb*' yang menjadi model klasifikasi akan dilatih menggunakan data latih. Objek '*MultinomialNB()*' menjadi model yang akan digunakan untuk mempelajari pola dari data latih. Fungsi *fit()* digunakan untuk mempelajari pola dalam data latih. Atribut data latih (*tfidf_train*) dan label data latih (*y_train*) diberikan kepada fungsi *fit()*. Digunakannya *predict()* dari model '*nb*' untuk memprediksi label dari data uji *tfidf_test*. Hasil prediksi akan disimpan dalam variabel *y_pred* yang akan digunakan untuk melakukan evaluasi kinerja model pada langkah selanjutnya.

4.7 Evaluasi Data

Tahap evaluasi adalah tahap yang bertujuan untuk melihat tolak ukur seberapa baik performa atau kinerja algoritma klasifikasi dengan membandingkan prediksi model dan nilai sebenarnya dari data *testing* yang digunakan dalam mengetahui probabilitas klasifikasi pada 3 aplikasi *e-learning* yang dijadikan objek penelitian. Hasil *Confusion Matrix* menjadi tolak ukur yang digunakan untuk mengukur performa atau kinerja tersebut adalah accuracy, precision, dan recall.

Tabel 4. 18 *Confusion Matrix*

Code

```

from sklearn.metrics
import accuracy_score,
precision_score,
recall_score, f1_score
from sklearn.metrics
import confusion matrix

clf = MultinomialNB()
clf.fit(X_train, y_train)
predicted = clf.predict(X_test)

print("MultinomialNB Accuracy:",
accuracy_score(y_test, predicted))
print("MultinomialNB Precision:",
precision_score(y_test, predicted,
average="binary",
pos_label="Negatif"))
print("MultinomialNB Recall:",
recall_score(y_test, predicted,
average="binary",
pos_label="Negatif"))
print("MultinomialNB f1_score:",
f1_score(y_test, predicted,
average="binary",
pos_label="Negatif"))

print(f'confusion matrix:\n
{confusion matrix(y_test,
predicted)}')

```

Pada Tabel 4.18 diatas menjelaskan bahwa dalam proses evaluasi dilakukan *import* beberapa metrik evaluasi kinerja model klasifikasi dari *library Sklearn*, yaitu *accuracy_score*, *precision_score*, *recall_score*, dan *f1_score* guna mengukur seberapa baik model klasifikasi yang telah dilatih dapat memprediksi label dengan benar. Dan *import confusion_matrix* dari *library Sklearn*. Kemudian '*clf*' sebagai objek model klasifikasi akan digunakan untuk evaluasi kinerja model, model klasifikasi '*clf.fit()*' dilatih menggunakan data latih, dan akan digunakan untuk memprediksi label '*predicted*' pada data uji (*X_test*). Lalu mencetak nilai *accuracy*, *precision*, *recall*, dan *f1* menggunakan metrik evaluasi yang telah diimpor sebelumnya dan mencetak *confusion_matrix*.

Confusion Matrix Ruangguru

```
MultinomialNB Accuracy: 0.8975069252077562
MultinomialNB Precision: 0.625
MultinomialNB Recall: 0.5319148936170213
MultinomialNB f1_score: 0.5747126436781609
confusion_matrix:
[[ 25  22]
 [ 15 299]]
```

Gambar 4. 12 Hasil Perhitungan *Confusion Matrix* Ruangguru

Pada Gambar 4.13 diatas menjelaskan bahwa nilai *Accuracy* yang dihasilkan merupakan 0.89, nilai *Precision* yang dihasilkan merupakan 0.62, nilai *Recall* yang dihasilkan merupakan 0.53, nilai *F1* yang dihasilkan merupakan 0.57. Nilai *True Negative* mendapatkan 15 yang menunjukkan bahwa label negatif yang diprediksi benar oleh *naïve bayes*. Nilai *False Negative* mendapatkan 25 yang menunjukkan bahwa label negatif yang diprediksi salah (*naïve bayes* melakukan prediksi positif yang seharusnya menjadi prediksi negatif). Nilai *True Positive* mendapatkan 299 label positif yang diprediksi benar oleh *naïve bayes*. Nilai *False Positive* mendapatkan 22 yang menunjukkan bahwa label positif yang diprediksi salah (*naïve bayes* melakukan prediksi negatif yang seharusnya menjadi prediksi positif).

Confusion Matrix Pahamify

```
MultinomialNB Accuracy: 0.8651685393258427
MultinomialNB Precision: 0.7083333333333334
MultinomialNB Recall: 0.6538461538461539
MultinomialNB f1_score: 0.68
confusion_matrix:
[[ 51  27]
 [ 21 257]]
```

Gambar 4. 13 Hasil Perhitungan *Confusion Matrix* Pahamify

Pada Gambar 4.14 diatas menjelaskan bahwa nilai *Accuracy* yang dihasilkan merupakan 0.86, nilai *Precision* yang dihasilkan merupakan 0.70, nilai *Recall* yang dihasilkan merupakan 0.65, nilai *F1* yang dihasilkan merupakan 0.68. Nilai *True Negative* mendapatkan 21 yang menunjukkan bahwa label negatif yang diprediksi benar oleh *naïve bayes*. Nilai *False Negative* mendapatkan 51 yang menunjukkan bahwa label negatif yang diprediksi salah (*naïve bayes* melakukan prediksi positif yang seharusnya menjadi prediksi negatif). Nilai *True Positive* mendapatkan 257 label positif yang diprediksi benar oleh *naïve bayes*. Nilai *False Positive* mendapatkan 27

yang menunjukkan bahwa label positif yang diprediksi salah (*naïve bayes* melakukan prediksi negatif yang seharusnya menjadi prediksi positif).

Confusion Matrix Merdeka Mengajar

```
MultinomialNB Accuracy: 0.9268929503916449
MultinomialNB Precision: 0.6666666666666666
MultinomialNB Recall: 0.4
MultinomialNB f1_score: 0.5
confusion_matrix:
[[ 14  21]
 [  7 341]]
```

Gambar 4. 14 Hasil Perhitungan *Confusion Matrix* Merdeka Mengajar

Pada Gambar 4.15 diatas menjelaskan bahwa nilai *Accuracy* yang dihasilkan merupakan 0.92, nilai *Precision* yang dihasilkan merupakan 0.66, nilai *Recall* yang dihasilkan merupakan 0.4, nilai *F1* yang dihasilkan merupakan 0.5. Nilai *True Negative* mendapatkan 7 yang menunjukkan bahwa label negatif yang diprediksi benar oleh *naïve bayes*. Nilai *False Negative* mendapatkan 14 yang menunjukkan bahwa label negatif yang diprediksi salah (*naïve bayes* melakukan prediksi positif yang seharusnya menjadi prediksi negatif). Nilai *True Positive* mendapatkan 341 label positif yang diprediksi benar oleh *naïve bayes*. Nilai *False Positive* mendapatkan 21 yang menunjukkan bahwa label positif yang diprediksi salah (*naïve bayes* melakukan prediksi negatif yang seharusnya menjadi prediksi positif).

4.8 Visualisasi

Pada tahap visualisasi ini hasil analisis akan ditampilkan dengan visualisasi melalui *wordcloud*, dengan tujuan untuk dapat menghasilkan atau menggambarkan kata-kata yang sering muncul pada data yang dianalisis ditiap kelasnya. Ukuran *font* yang berbeda-beda menggambarkan banyaknya kata yang sering muncul pada data yang dianalisis.

Visualisasi Merderka Mengajar



Gambar 4. 17 Wordcloud Merdeka Mengajar

Pada Gambar 4.18 menunjukkan bahwa kata yang paling sering muncul pada data yaitu kata "bantu", "aplikasi", "guru", "ajar" dari sekian banyak data yang ada.

4.9 Hasil dan Pembahasan

Hasil akurasi dari metode yang telah dikerjakan terdapat pada Tabel 4.19 sebagai berikut:

Tabel 4. 19 Hasil Akhir

| | Ruangguru | Pahamify | Merderka mengajar |
|----------|-----------|----------|-------------------|
| Accuracy | 89% | 86% | 92% |

Hasil dan pembahasan yang dapat disimpulkan yaitu:

1. Ruangguru mempunyai hasil akurasi dari model *naive bayes* sebesar 89%, kata-kata yang paling sering ditemukan dalam data yaitu kata "ruangguru", "ruang", "guru", "ajar".
2. Pahamify mempunyai hasil akurasi dari model *naive bayes* sebesar 86%, kata-kata yang paling sering ditemukan dalam data yaitu kata "ajar", "aplikasi", "bagus", "pahamify".

3. Merdeka Mengajar mempunyai hasil akurasi dari model *naive bayes* sebesar 92%, kata-kata yang paling sering ditemukan dalam data yaitu kata "bantu", "aplikasi", "guru", "ajar".



UNIVERSITAS
Dinamika

BAB V

KESIMPULAN DAN SARAN

Bab ini berisi tentang hasil analisis dari penelitian dan percobaan data tugas tugas akhir, pada kesimpulan mempunyai isi tentang kesimpulan hasil pengerjaan tugas akhir. Pada saran berisi tentang saran-saran untuk pengembangan penelitian serupa di masa mendatang.

5.1 Kesimpulan

Berdasarkan proses yang telah dilakukan di atas dapat ditarik kesimpulan bahwa:

1. Sesuai seperti pada subbab Hasil dan Pembahasan yang sudah diproses menggunakan Algoritma *Naive Bayes* mempunyai hasil akurasi di tiap aplikasi *e-learning* yaitu ruangguru sebesar 89%, pahamify sebesar 86%, merdeka mengajar sebesar 92%.
2. Penggunaan kamus bahasa gaul atau kata slang memungkinkan untuk dapat meningkatkan nilai akurasi.

5.2 Saran

Dalam pengerjaan tugas akhir tentunya memiliki beberapa kekurangan. Oleh sebab itu adapun beberapa saran yang diberikan untuk pengembangan lebih lanjut adalah sebagai berikut :

1. Menambahkan kamus yang terbaru, seputar bahasa gaul / *slang* dan singkatan, karena pada data ulasan masih banyak ditemukan bahasa yang kurang baku.
2. Menambahkan ulasan yang menggunakan bahasa inggris agar dapat melakukan perbandingan antara bahasa Indonesia dan bahasa inggris.

DAFTAR PUSTAKA

- Anindira, R. A., & Imran, A. I. (2021). STRATEGI DAN IMPLEMENTASI CUSTOMER RELATIONSHIP MANAGEMENT MELALUI APLIKASI KAI ACCESS DALAM MENINGKATKAN KEPUASAN PELANGGAN PT. KAI. *e-Proceeding of Management*, 4232-4241.
- Buntoro, G. A. (2017). Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter. *INTEGER: Journal of Information Technology*, 32-41.
- Darwis, D., Siskawati, N., & Abidin, Z. (2021). Penerapan Algoritma Naive Bayes untuk Analisis Sentimen Review Data Twitter BMKG Nasional. *Jurnal TEKNO KOMPAK*, 131-145.
- Ginting, V. S., Kusriani, & Taufiq, E. (2020). IMPLEMENTASI ALGORITMA C4.5 UNTUK MEMPREDIKSI KETERLAMBATAN PEMBAYARAN SUMBANGAN PEMBANGUNAN PENDIDIKAN SEKOLAH MENGGUNAKAN PYTHON. *Jurnal Teknologi Informasi dan Komunikasi*, 36 – 44.
- Hendra, A., & Fitriyani. (2021). Analisis Sentimen Review Halodoc Menggunakan Naïve Bayes Classifier. *JISKa*, 78-89.
- Hendriyanto, M. D., Ridha, A. A., & Enri, U. (2022). ANALISIS SENTIMEN ULASAN APLIKASI MOLA PADA GOOGLE PLAY STORE MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE. *Journal of Information Technology and Computer Science*, 1-7.
- Izzah, I. K., & Girsang, A. S. (2021). Modified TF-Assoc Term Weighting Method for Text Classification on News Dataset from Twitter. *IAENG International Journal of Computer Science*, 48.
- Mukminin, A., Sunarto, M. D., & Nurcahyawati, V. (2021). Analisis Sentimen Publik Terhadap Pelayanan Tes Swab-Pcr Covid-19 Di Indonesia Menggunakan Algoritma Support Vector Machine. *Jurnal Sistem Informasi dan Komputer Akuntansi*, 1-13.
- Muljono, Artanti, D. P., Prihandono, A., & Setiadi, D. I. (2018). Analisa Sentimen Untuk Penilaian Pelayanan Situs Belanja Online Menggunakan Algoritma Naïve Bayes. *Konferensi Nasional Sistem Informasi*, 165-170.

- Muttaqin, M. N., & Kharisudin, I. (2021). ANALISIS SENTIMEN APLIKASI GOJEK MENGGUNAKAN SUPPORT VECTOR MACHINE DAN K NEAREST NEIGHBOR. *UNNES Journal of Mathematics*, 22-27.
- Pravina, A. M., Cholissodin, I., & Adikara, P. P. (2019). Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2789-2797.
- Utami, I. S. (2016). Implementasi E-Learning untuk Meningkatkan Aktivitas Belajar Siswa. *Jurnal Komputer Terapan*, 169-178.
- Widagdo, A. S., W. A, B. S., & Nasiri, A. (2020). Analisis Tingkat Kepopuleran E-Commerce Di Indonesia Berdasarkan Sentimen Sosial Media Menggunakan Metode Naïve Bayes. *Jurnal INFORMA Politeknik Indonusa Surakarta*, 1-5.
- Wilma, L., Lawalata, V. O., Pailin, D. B., & Ohoirenan, R. (2021). Sentiment Analysis of Consumers for Determining the Packaging Features of Eucalyptus Oil Products. *JURNAL ILMIAH TEKNIK INDUSTRI*, 71-80.
- Zalyhaty, L. Q., Nurcahyawati, V., & Sutomo, E. (2020). Analisis Sentimen Tanggapan Masyarakat Terhadap Vaksin Covid-19 Menggunakan Algoritma Support Vector Machine (SVM). *Jurnal Sistem Informasi Universitas Dinamika*, 1-10.