PERBANDINGAN NAÏVE BAYES DAN LOGISTIC REGRESSION DALAM SENTIMENT ANALYSIS PADA REVIEW MARKETPLACE MENGGUNAKAN RATING-BASED LABELING

Satya Abdul Halim Bahtiar

19917014

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer

Konsentrasi Sains Data

Program Studi Informatika Program Magister

Fakultas Teknologi Industri

Universitas Islam Indonesia

2023

Lembar Pengesahan Pembimbing

PERBANDINGAN NAÏVE BAYES DAN LOGISTIC REGRESSION DALAM SENTIMENT ANALYSIS PADA REVIEW MARKETPLACE MENGGUNAKAN RATING-BASED LABELING

Satya Abdul Halim Bahtiar
19917014

Yogyakarta, Agustus 2023

Pembimbing 1	Pembimbing 2
Chandra Kusuma Dewa, S.Kom., M.Cs., Ph.D.	Dr. Ahmad Luthfi, S.Kom., M.Kom.

Lembar Pengesahan Penguji

PERBANDINGAN NAÏVE BAYES DAN LOGISTIC REGRESSION DALAM SENTIMENT ANALYSIS PADA REVIEW MARKETPLACE MENGGUNAKAN RATING-BASED LABELING

Satya Abdul Halim Bahtiar
19917014

Yogyakarta, Agustus 2023

Tim Penguji,

Chandra Kusuma Dewa, S.Kom., M.Cs., Ph.D.
Ketua

Dr. Ahmad Luthfi, S.Kom., M.Kom.
Anggota I

Irving Vitra Paputungan, S.T., M.Sc., Ph.D.
Anggota II

Mengetahui,
Ketua Program Studi Informatika Program Magister
Universitas Islam Indonesia

Irving Vitra Paputungan, S.T., M.Sc., Ph.D.

Abstrak

PERBANDINGAN NAÏVE BAYES DAN LOGISTIC REGRESSION DALAM SENTIMENT ANALYSIS PADA REVIEW MARKETPLACE MENGGUNAKAN RATING-BASED LABELING

Penelitian ini berfokus pada analisis sentimen di Google Play Store, platform untuk mengunduh aplikasi Android dan memberikan ulasan. Analisis sentimen penting untuk memahami respons pengguna terhadap aplikasi, terutama di *marketplace*. Dalam penelitian ini, dua algoritma machine learning, yaitu Naïve Bayes dan Logistic Regression, digunakan untuk mengklasifikasikan ulasan pengguna. Naïve Bayes dan Logistic Regression sering digunakan untuk klasifikasi sentimen karena keunggulan simplicitas, efisiensi komputasi, dan kemampuan memberikan interpretasi dan estimasi probabilitas prediksi yang baik. Penilaian rating aplikasi digunakan sebagai referensi untuk menentukan sentimen dari setiap komentar. Dataset dibagi menjadi dua kondisi: menggunakan 2 label (positif & negatif) dan 3 label (positif, netral, & negatif). Hasil pengujian menunjukkan bahwa performa tertinggi diperoleh dengan menggunakan Logistic Regression pada dataset Shopee dengan 2 label. Akurasinya mencapai 84,58%, presisinya 84,66%, dan recallnya 84,63%. Selain itu, waktu proses tercepat terjadi saat menguji dataset Lazada 2 label dengan Naïve Bayes, hanya memerlukan 0,038 detik. Secara keseluruhan, penelitian ini menunjukkan bahwa dataset dengan 2 label cenderung menghasilkan tingkat akurasi yang lebih tinggi dibandingkan dengan dataset 3 label.

Kata kunci

Naïve Bayes, Logistic Regression, Marketplace, Google Play Store, Rating-based Labeling

Abstract

Comparison of Naïve Bayes and Logistic Regression in Sentiment Analysis on Marketplace Reviews Using Rating-Based Labeling

This research focuses on sentiment analysis in the Google Play Store, a platform for downloading Android applications and providing reviews. Sentiment analysis is essential for understanding user responses to applications, particularly in the app marketplace. In this study, two machine learning algorithms, Naïve Bayes and Logistic Regression, are employed to classify user reviews. Naïve Bayes and Logistic Regression are commonly used for sentiment classification due to their advantages of simplicity, computational efficiency, and ability to provide meaningful interpretation and probability estimation for predictions. The application rating is used as a reference to determine the sentiment of each comment. The dataset is divided into two conditions: using 2 labels (positive & negative) and 3 labels (positive, neutral, & negative). The test results indicate that the highest performance is achieved by classifying with Logistic Regression on the Shopee dataset with 2 labels. The accuracy reaches 84.58%, precision reaches 84.66%, and recall reaches 84.63%. Additionally, the fastest processing time occurs when testing the Lazada 2-label dataset with Naïve Bayes, taking only 0.038 seconds. Overall, the research suggests that datasets with 2 labels tend to yield higher accuracy compared to datasets with 3 labels.

Keywords

Naïve Bayes, Logistic Regression, Marketplace, Google Play Store, Rating-based Labeling

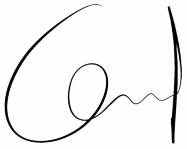
Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Agustus 2023



Satya Abdul Halim Bahtiar, S.Kom

Daftar Publikasi

Publikasi yang menjadi bagian dari tesis

Bahtiar, S. A. H., Dewa, C. K., Luthfi, A. (2023). Comparison of Naïve Bayes and Logistic Regression in Sentiment Analysis on Marketplace Reviews Using Rating-Based Labeling. Journal of Information Systems and Informatics,4(3).

Sitasi publikasi 1

Kontributor	Jenis Kontribusi
Satya Abdul Halim Bahtiar	Melakukan komputasi, analisis model dan menulis *paper*
Chandra Kusuma Dewa	Melakukan analisis model dan mengedit *paper*
Ahmad Luthfi	Melakukan analisis model dan mengedit *paper*

Halaman Kontribusi

Dalam penulisan tesis ini pembimbing I memberikan masukan terkait judul penelitian dan bersama pembimbing II memberikan beberapa masukkan sebagai perbaikan dari cara penulisan tesis serta analisis dan pengolahan dataset.

Halaman Persembahan

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيْم

إِنَّ صَلاتِي وَنُسُكِي وَمَحْيَايَ وَمَمَاتِي لِلَّهِ رَبِّ الْعَالَمِينَ

"Inna Sholati Wanusuki Wamahyaya Wamamati Lillahirabbil Alamin "

"Sesungguhnya shalatku, ibadahku, hidupku dan matiku hanyalah untuk Allah, Tuhan semesta alam."

Kata Pengantar

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيْم

Alhamdulillahirabbil'alamin, segala puji syukur senantiasa kita limpahkan kepada Allah SWT, karena atas Ijin dan nikmat-Nya, tesis yang berjudul "Perbandingan Naïve Bayes Dan Logistic Regression Dalam Sentiment Analysis Pada Review Marketplace Di Google Play Store Menggunakan Rating-Based Labeling" dapat berjalan dengan lancar dalam penyelesaiannya. Tesis ini diajukan sebagai bagian dalam menyelesaikan studi dan sebagai salah satu syarat untuk memperoleh gelar Magister Komputer pada Program Studi Magister Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Dalam penyelesaian Tesis ini, penulis banyak mendapatkan bantuan dari berbagai pihak, untuk itu penulis menyampaikan ucapan terima kasih setulusnya kepada:

1. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D., Rektor Universitas Islam Indonesia.
2. Bapak Irving Vitra Paputungan, S.T., M.Sc., Ph.D., Ketua Program Studi Magister Teknik Informatika Universitas Islam Indonesia.
3. Bapak Chandra Kusuma Dewa, M.Kom., M.Cs., Ph.D., selaku dosen pembimbing satu, yang telah banyak membantu penulis dalam memberikan ide, saran dan perbaikan dalam tesis ini.
4. Bapak Dr. Ahmad Luthfi, S.Kom., M.Kom., selaku dosen pembimbing dua yang banyak memberikan kemudahan baik pengarahan maupun bimbingan selama pengajuan dan pengerjaan Tesis.
5. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., bapak Dr. Ing. Ridho Rahmadi, S.Kom., M.Sc. dan bapak Ahmad Fathan Hidayatullah S.T., M.Sc., yang telah menginspirasi saya dan mengnalkan saya dengan Magister Informatika UII.
6. Dosen Program Studi Magister Teknik Informatika yang telah memberikan bekal ilmu pengetahuan kepada penulis, semoga ilmunya menjadi amal jariyah di dunia maupun akhirat.
7. Staff Akademik Program Pascasarjana Fakultas Teknologi Universitas Islam Indonesia, yang telah membantu dalam segala urusan administrasi di kampus.

8. Sahabat seperjuangan Sains Data (Yurio, Malik, Fahmi, Yopi, Eko, Windi, Rifai dan Atin)
9. Teman-teman Magister Teknik Informatika Universitas Islam Indonesia Yogyakarta seangkatan.

Saya menyadari bahwa dalam penulisan tesis ini masih banyak kelemahan dan kekurangan, untuk itu kritik dan saran yang sifatnya membangun sangat penulis harapkan agar tesis ini dapat menjadi lebih baik.

Daftar Isi

Daftar Tabel

Daftar Gambar

Glosarium

Marketplace	- Lokapasar atau supermarket online
Kemkominfo	- Kementerian Komunikasi dan Informatika
NB	- Naïve Bayes
LR	- Logistic Regression
SVM	- Support Vector Machine
FS	- Feature Selection
PSO	- Particle Swarm Optimization
K-NN	- K-Nearest Neighbour
SLR	- Systematic Literature Review

BAB 1
Pendahuluan

1.1 Latar Belakang

Marketplace di Indonesia telah mengalami pertumbuhan yang pesat dalam beberapa tahun terakhir. Peningkatan pengguna internet dan adopsi perangkat mobile, serta perubahan pola belanja masyarakat Indonesia, telah mendorong perkembangan Marketplace yang sangat dinamis. Saat ini, banyak pengguna di Indonesia mengandalkan aplikasi marketplace untuk melakukan pembelian online, membandingkan harga, mencari produk, dan memberikan ulasan atau review tentang produk yang mereka beli.

Dalam konteks pasar aplikasi di Indonesia, Google Play Store juga merupakan salah satu platform utama untuk mengunduh aplikasi di perangkat Android. Indonesia merupakan pasar yang potensial untuk aplikasi mobile, dengan jumlah pengguna smartphone yang terus berkembang pesat. Pengguna dapat memberikan ulasan atau review terhadap aplikasi yang mereka gunakan, baik itu pengalaman positif, negatif, atau netral. Oleh karena itu, analisis sentimen pada review Google Play Store menjadi penting bagi pengembang aplikasi dan pemilik bisnis untuk memahami respons pengguna terhadap aplikasi mereka dan mengambil tindakan yang sesuai.

Dalam sentiment analisis marketplace di indonesia pada review Google Play Store, algoritma machine learning digunakan untuk mengklasifikasikan ulasan pengguna menjadi positif, negatif, atau netral. Dua algoritma machine learning yang digunakan untuk tugas ini adalah Naïve Bayes dan Logistic Regression.

Naïve Bayes adalah salah satu metode klasifikasi yang sederhana dan efisien yang berdasarkan pada teorema Bayes. Metode ini mengasumsikan independensi antara fitur atau atribut dalam data yang diberikan, sehingga dikenal sebagai "Naïve". Naïve Bayes telah banyak digunakan dalam berbagai aplikasi, termasuk analisis sentimen, karena kemampuannya dalam menangani data dengan cepat dan mudah diimplementasikan.

Di sisi lain, Logistic Regression adalah metode klasifikasi yang menggunakan fungsi logistik untuk memodelkan probabilitas kelas yang berbeda dalam data yang diberikan. Metode ini menghasilkan prediksi dalam bentuk probabilitas, dan kemudian menggunakan ambang batas tertentu untuk mengklasifikasikan data menjadi kelas positif, negatif, atau netral. Logistic Regression juga merupakan metode klasifikasi yang umum

digunakan dalam analisis sentimen, karena kemampuannya untuk menghasilkan probabilitas prediksi yang berguna untuk menentukan sentimen dari teks.

Dalam konteks sentiment analisis, perbandingan antara Naïve Bayes dan Logistic Regression menjadi topik penelitian yang menarik. Tujuan dari tesis ini adalah untuk membandingkan kinerja kedua algoritma ini dalam konteks sentiment analisis marketplace di indonesia pada review Google Play Store. Penelitian ini diharapkan dapat memberikan pemahaman yang lebih baik tentang Perbandingan algoritma naïve bayes dan logistic regression dalam sentiment analisis marketplace di indonesia pada review Google Play Store, dan menjadi kontribusi yang berharga dalam pengembangan metode analisis sentimen yang efisien dan akurat dalam konteks marketplace di Indonesia.

1.2 Rumusan Masalah

Berdasarkan pembahasan pada latar belakang maka rumusan masalah pada penelitian ini adalah bagaimana perbandingan kinerja antara algoritma Naïve Bayes dan Logistic Regression dalam melakukan sentiment analisis pada review aplikasi marketplace di Indonesia yang terdapat pada Google Play Store menggunakan *rating-based labeling*.

1.3 Tujuan Penelitian

Adapun tujuan penelitian ini adalah:

1. Menerapkan algoritma Naïve Bayes dan Logistic Regression dalam melakukan sentiment analisis pada review aplikasi marketplace di Indonesia yang terdapat pada Google Play Store menggunakan *rating-based labeling*.
2. Membandingkan kinerja algoritma Naïve Bayes dan Logistic Regression pada dataset yang sudah tersedia.
3. Menerapkan *rating-based labeling* pada dataset yang tersedia.

1.4 Batasan Masalah

Adapun batasan masalah pada penelitian ini mencakup:

1. Algoritma yang dibandingkan adalah Naïve Bayes dan Logistic Regression.
2. Data yang digunakan diambil dari review aplikasi marketplace Tokopedia, Shopee, Lazada dan Blibli pada Google Play Store.
3. Review yang digunakan adalah review Bahasa Indonesia.

1.5 Manfaat Penelitian

1. Manfaat Akademis
    a. Dapat memberikan informasi atau pengetahuan bagi pengembang aplikasi marketplace.
    b. Dapat menambah wawasan tentang implementasi Naïve Bayes dan Logistic Regression pada proses klasifikasi sentimen pada dataset review aplikasi di *Google Play Store*.
    c. Sebagai acuan atau rujukan bagi penelitian selanjutnya yang ingin meneliti tentang analisis sentimen pada dataset review aplikasi di *Google Play Store*.
    d. Sebagai acuan bagi penelitian tentang implementasi Rating-Based Labeling pada dataset review aplikasi di *Google Play Store*.
    e. Dapat memberikan informasi yang berguna khususnya pada ruang lingkup pengembang aplikasi.
2. Manfaat Praktis

    Dapat menjadi sebuah referensi ilmiah yang berguna bagi pengembang aplikasi marketplace dalam mengambil keputusan.

1.6 Sistematika Penulisan

1. Abstrak berisi rangkuman laporan isi laporan secara umum.
2. Pernyataan Keaslian Tulisan.
3. Daftar Publikasi berisi jurnal internasional.
4. Halaman Kontribusi berisi dosen yang terlibat dalam penelitian.
5. Halaman Persembahan berisi ucapan terhadap keluarga.
6. Kata Pengantar berisi ucapan terimakasih kepada berbagai pihak yang terlibat dalam penelitian.
7. Daftar Isi berisi daftar judul bab, dan sub bab laporan.
8. Daftar Tabel berisi daftar nama tabel.
9. Daftar Gambar berisi daftar nama Gambar.
10. BAB 1 Pendahuluan
    a. 1.1 Latar Belakang berisi alasan penulis memilih tema
    b. 1.2 Rumusan Masalah berisi pertanyaan penelitian.
    c. 1.3 Tujuan Penelitian berisi tujuan penelitian.
    d. 1.4 Batasan Masalah berisi batasan apa saja yang akan dibahas dalam penelitian.

e. 1.5 Manfaat Penelitian berisi manfaat yang didapatkan sesudah melakukan penelitian.
f. 1.6 Sistematika Penulisan berisi daftar penulisan dari BAB awal sampai akhir

11. BAB 2 Tinjauan Pustaka
    a. 2.1 Pendahuluan.
    b. 2.2 Konsep Pengetahuan Model berisi pengetahuan apa saja yang digunakan dalam penelitian.
    c. 2.3 Analisa Kebutuhan berisi tahapan analisis kebutuhan yang diperlukan untuk melakukan penelitian.
12. BAB 3 Metodologi
    a. 3.1 Tahapan Penelitian berisi tahapan yang dilakukan dalam penelitian secara urut.
    b. 3.2 Literature review berisi tahapan yang dilakukan peneliti dalam mengumpulkan informasi yang mendukung penelitian.
    c. 3.2 Pengumpulan data berisi tahapan pengumpulan data dari review aplikasi di Google Play Store
    d. 3.3 Pelebelan menggunakan rating-based labeling.
    e. 3.4 Preprocessing data berisi tahapan penerapan *text preprocessing* pada data yang sudah di dapat.
    f. 3.5 Modeling menggunakan Naïve bayes dan Logistic Regression adalah tahapan penerapan testing dan training model pada dataset yang tersedia.
    g. 3.6 Evaluasi Model berisikan tahapan evaluasi model yang sudah dilatih dan diuji.
13. BAB 4 Hasil Dan Pembahasan
    a. Hasil pengujian Naïve Bayes dengan dataset 2-label
    b. Hasil pengujian Logistic Regression dengan dataset 2-label.
    c. Hasil pengujian Naïve Bayes dengan dataset 3-label
    d. Hasil pengujian Logistic Regression dengan dataset 3-label.
    e. Pembahasan berisi pembahasan hasil penelitian.
14. Kesimpulan dan Saran
    a. Kesimpulan berisi kesimpulan yang didapatkan setelah melakukan penelitian.

b. Saran berisi saran untuk penelitian selanjutnya yang terkait dengan penelitian ini.

15. Daftar Pustaka berisi pustaka rujukan yang digunakan di penelitian ini.
16. Lampiran berisi lampiran *source code* yang digunakan pada penelitian ini.

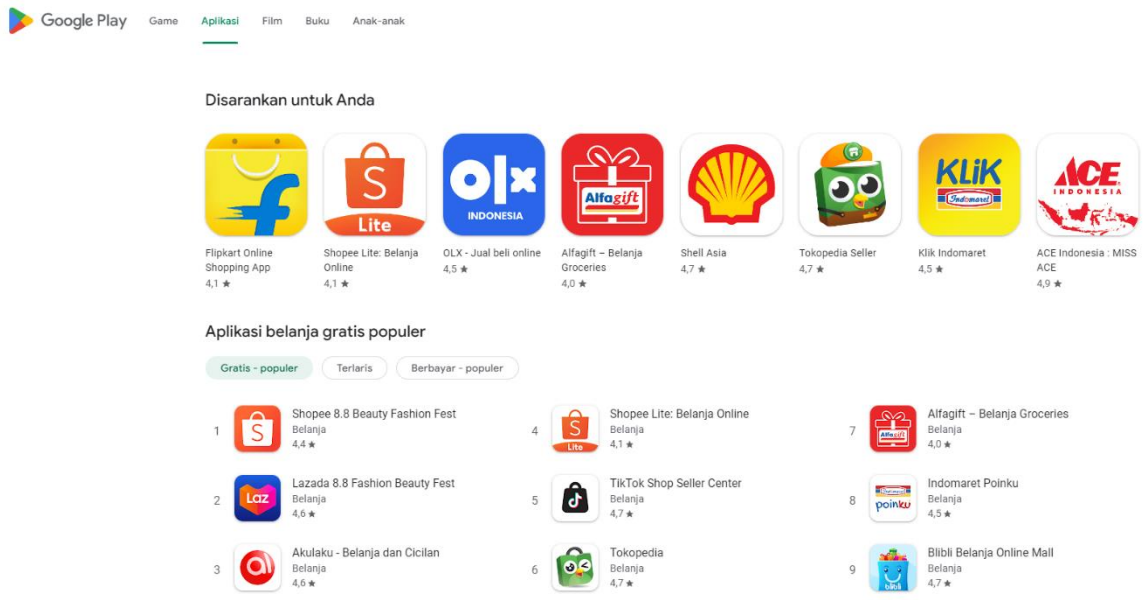
BAB 2

Tinjauan Pustaka

2.1 Pendahuluan

Pada era digital yang terus berkembang, Indonesia menjadi salah satu negara dengan pertumbuhan pengguna internet yang pesat. Pengguna internet di Indonesia terus meningkat dari tahun ke tahun. Kementerian Komunikasi dan Informatika (Kemenkominfo) mengungkapkan pengguna internet di Indonesia saat ini mencapai 63 juta orang. Dari angka tersebut, 95 persennya menggunakan internet untuk mengakses jejaring sosial (KOMINFO, n.d.-b). Didorong oleh adopsi teknologi digital yang semakin meluas di berbagai segmen masyarakat, perdagangan daring juga mengalami pertumbuhan yang signifikan, di mana konsumen semakin mengadopsi pembelian secara online sebagai cara yang lebih praktis dan efisien untuk memenuhi kebutuhan mereka.

Tren pertumbuhan pengguna internet dan Marketplace di Indonesia tidak hanya tercermin dalam jumlah pengguna dan transaksi online, tetapi juga dalam kontribusi terhadap perekonomian nasional. Marketplace di Indonesia telah menjadi sektor ekonomi yang strategis, dengan kontribusi yang signifikan terhadap Produk Domestik Bruto (PDB) negara ini. Fenomena ini mencerminkan perubahan perilaku konsumen yang semakin beralih ke platform online untuk melakukan transaksi, termasuk pembelian produk dan layanan.



Gambar 2.1 Ragam aplikasi Marketplace di Google Play Store

Direktur Pemberdayaan Informatika, Direktorat Jenderal Aplikasi Informatika Kemkominfo, Septriana Tangkary menyatakan pertumbuhan nilai perdagangan elektronik di Indonesia mencapai 78 persen, tertinggi di dunia.

"Indonesia merupakan negara 10 terbesar pertumbuhan 'e-commerce' dengan pertumbuhan 78 persen dan berada di peringkat ke-1. Sementara Meksiko berada di peringkat kedua, dengan nilai pertumbuhan 59 persen," kata Septriana Tangkary di Pamekasan, Jawa Timur, Rabu (27/2) (KOMINFO, n.d.-a). Dalam melakukan perdagangan elektronik, masyarakat Indonesia banyak menggunakan aplikasi *marketplace* android yang ada di Google Play Store, beberapa aplikasi yang sering digunakan yaitu Shopee, Tokopedia, Lazada dan Blibli

Dalam penelitiannya Indra Kurniawan menggunakan dan membandingkan algoritma NB dan SVM dalam mengklasifikasikan sentiment positif dan negatif pada opini pelanggan Tokopedia, Shopee, Lazada di Twitter. Tahapan yang dilakukan dalam sentiment analysis di penelitian ini adalah pengumpulan data, pelebelan, preprocessing, pemodelan dan evaluasi. Peneliti menggunakan metode SMOTE untuk mengatasi imbalance class dan juga menggunakan Cross Validation k-folds 10 dalam evaluasinya. Dari hasil pengujian perbandingan ditemukan bahwa SVM lebih akurat dalam memprediksi daripada NB, namun keduanya memiliki tingkat akurasi diatas 80% (Kurniawan et al., 2023).

Dengan menggunakan data dari sumber yang lain yaitu review aplikasi klikindomaret dari Google Play Store Muhammad Azhar (Azhar et al., 2020) juga membandingkan NB dan SVM untuk menganalisis sentimen, namun dalam penelitian ini algoritma NB hanya dapat memprediksi sentimen dengan akurasi 69.74%. Kemudian peneliti menerapkan optimasi feature selection (FS) menggunakan Particle Swarm Optimization (PSO) dan didapatkan peningkatan akurasi NB menjadi 75,21%. Berbeda dengan algoritma NB, ketika peneliti menerapkan SVM didapatkan akurasi sebesar 81,21% tanpa optimasi dan akurasi sebesar 81,84% setelah menerapkan optimasi PSO.

(Pratmanto et al., 2020) berhasil mendapatkan 96.667% akurasi dalam penelitiannya menggunakan NB untuk analisis sentiment pada review aplikasi shopee di Google Play Store. Pranoto menuliskan bahwa akurasi itu didapatkan dengan menggunakan teknik partisi pada data aplikasi shopee di Google Play Store, dari 200 data review yang diambil terdiri dari 100 review positif dan 100 review negatif. Dengan data tersebut peneliti membagi menjadi dua partisi. Partisi data pertama yaitu training terdiri dari 140 data, partisi data kedua yaitu testing, terdiri dari 60 data.

Dengan menggunakan K-Fold, juga mengaplikasikan unigram dan bigram Arif Nur Rohman (Rohman et al., 2020) membandingkan NB dengan Algoritma K-NN untuk menganalisis sentiment pada review produk fashion di marketplace shopee. Hasil optimal didapatkan dengan menerapkan algoritma K-NN pada Bigram dan pengujian validasi K-Fold ke 9.

Metode SVM digunakan oleh Muhammadi Iqbal Ahmadi (Ahmadi et al., 2020) untuk mengklasifikasi review positif dan negatif pengguna pada aplikasi Marketplace Google play store. Dalam penelitian ini penulis menggunakan 10 fold cross validation sebagai model evaluasi dengan data berjumlah 3,000 ulasan, sebagai data training sebanyak 1500. yang diambil dari masing-masing aplikasi sebanyak 300 ulasan sebagai data testing. Dari review lima aplikasi Marketplace yang digunakan pada penelitian ini, dihasilkan tingkat akurasi paling tinggi adalah pada klasifikasi sentiment pada Tokopedia dengan nilai akurasi yaitu Tokopedia 90,67%, JD.ID 75,33%, Blibli 74,00%, Shopee 70,00%, Lazada 69,00%.

Dari berberapa penelitian sebelumnya maka pada penelitian kali ini akan berfokus pada perbandingan dua algoritma, yaitu NB dan Logistic Regression dan menggunakan data review empat aplikasi Marketplace Indonesia di Google Play Store. Diharapkan dengan penelitian ini dapat menemukan perbandingan tingkat akurasi kedua algoritma dalam mengklasifikasi sentiment pada data yang ada.

Tabel 2.1 Cluster Pengetahuan

No	Sub Tema	Keyword	Ulasan Kritis	Pustaka
1.	Perbandingan Algoritma Naïve Bayes Dan SVM Dalam Sentimen Analisis Marketplace Pada Twitter	Sentimen Analisis, Marketplace, Naïve Bayes, Support Vector Machine, SMOTE	Penelitian ini menggunakan SMOTE untuk mengatasi imbalance class dan menerapkan k-folds untuk evaluasi model	(Kurniawan et al., 2023)
2.	Marketplace Sentiment Analysis Using Naïve Bayes	Naïve Bayes, Particle Swarm Optimization, Support Vector	Penerapan Particle Swarm Optimization meningkatkan tingkat akurasi pada kedua algoritma yang dipakai.	(Azhar et al., 2020)

	And Support Vector Machine	Machine, Feature Selection, Consumer Review.		
3.	App Review Sentiment Analysis Shopee Application In Google Play Store Using Naïve Bayes Algorithm	Sentimen Analisis, Marketplace, Naïve Bayes	Menggunakan NB dengan dataset imbang atara sentiment positif dan negatif dan pembagian 70:30 untuk data training dan testing	(Pratmanto et al., 2020)
4.	Natural Language Processing on Marketplace Product Review Sentiment Analysis	Sentiment analysis, NLP, Text Mining, Naïve Bayes, K-NN	Review produk di shopee menggunakan NB & K-NN, menerapkan n-gram dan word normalization	(Rohman et al., 2020)
5.	Sentiment Analysis Online Shop On The Play Store Using Method Support Vector Machine (Svm)	Google Play Store, Support Vector Machine, Sentimen Analysis	Data yang digunakan cukup beragam dari 5 review aplikasi *Marketplace* di google play. Menggunakan SVM untuk mengklasifikasikan sentiment	(Ahmadi et al., 2020)
6	Application of Naïve Bayes Classification to Analyze Performance Using Stopwords	Naïve Bayes, classification, preprocessing, processing time	Penerapan teks preprocessing membuat f1-score lebih baik dan waktu proses lebih cepat	(Jefriyanto et al., 2023)

2.2 Konsep Pengetahuan

Konsep Pengetahuan merupakan konsep dan istilah yang dipakai dalam penelitian "perbandingan naïve bayes dan logistic regression dalam sentiment analysis pada review marketplace di google play store menggunakan rating-based labeling"

2.2.1 Systematic Literature Review

Metode SLR dilakukan dengan langkah-langkah yang terstruktur dan mengikuti protokol tertentu, sehingga mampu menghindari bias dan pemahaman subyektif dari peneliti (Wahono, n.d.). Dalam pelaksanaan *Systematic Literature Review* (SLR) pertama dilakukan perencanaan terkait topik apa yang akan dibahas, kemudian proses identifikasi literatur yang berkaitan dengan topik hingga akhirnya merangkum hasil literasi dan memilih jurnal yang tepat. SLR dilakukan dalam 3 bagian *planning, conducting, dan reporting.*

Berikut adalah langkah-langkah umum dalam *Systematic Literature Review*:

1. Merumuskan Pertanyaan: Tentukan pertanyaan penelitian atau topik yang ingin Anda teliti. Pertanyaan ini akan menjadi panduan untuk mempersempit cakupan literatur yang akan Anda cari.
2. Mengembangkan protocol review: Buat tata cara dan urutan review yang akan dilakukan.
3. Mengidentifikasi literature yang relevan: Proses mengumpulkan studi yang relevan dengan penelitian yang akan dilakukan.
4. Melakukan seleksi studi utama: Melakukan pemilihan studi utama yang akan dijadikan rujukan dalam penelitian.
5. Menilai kualitas studi: Memerika kualitas rujukan studi yang akan digunakan.
6. Melakukan sintesis bukti: Melakukan penggabungan hasil yang didapat dari studi rujukan dan menghasilkan kerangka yang akan menjadi acuan dalam penelitian.
7. Menulis paper: Menyusun kerangka laporan penelitian yang akan dipublikasikan
8. Memilih jurnal yang tepat: Memilih jurnal yang tepat untuk dicantumkan dalam jurnal

Gambar 2.2 Tahapan SLR

2.2.2 Google-Play-Scraper

Scraping, atau web scraping, adalah proses ekstraksi informasi dari halaman web dengan menggunakan perangkat lunak atau alat otomatis. Tujuannya adalah untuk mengumpulkan data dari berbagai sumber online dengan cara yang lebih efisien dan otomatis. Teknik scraping melibatkan pengambilan konten teks, gambar, data tabel, atau informasi lainnya dari situs web dan menyimpannya dalam format yang dapat diolah, seperti file spreadsheet atau database.

Scraping dapat digunakan untuk berbagai tujuan, termasuk:

1. Pengumpulan Data: Mengumpulkan data seperti harga produk, ulasan pelanggan, informasi kontak, atau data lainnya dari situs web.
2. **Penelitian dan Analisis:** Mengumpulkan data untuk analisis penelitian atau bisnis, seperti analisis pasar, tren industri, atau perbandingan produk.
3. **Monitor Informasi:** Melacak perubahan pada halaman web tertentu secara berkala, misalnya untuk memantau harga produk atau berita terbaru.

4. **Pembuatan Konten:** Mengumpulkan data untuk membuat konten, seperti mengambil kutipan dari berita atau artikel untuk dibagikan atau direferensikan.

Namun, perlu diingat bahwa tidak semua situs web mengizinkan atau mendukung praktik scraping, dan beberapa mungkin memiliki aturan atau kebijakan yang melarang pengambilan data mereka. Oleh karena itu, penting untuk mematuhi etika dan hukum serta memperhatikan pedoman penggunaan situs web saat melakukan scraping.

Scraping dapat dilakukan menggunakan berbagai teknik dan alat. Berikut adalah langkah-langkah umum untuk melakukan web scraping:

1. **Pilih Situs Web:** Tentukan situs web yang ingin Anda scrap. Pastikan untuk memahami aturan dan kebijakan situs terkait scraping sebelum memulai.
2. **Pilih Teknik Scraping:** Ada beberapa cara untuk melakukan scraping, termasuk menggunakan perangkat lunak atau bahasa pemrograman seperti Python. Pilih teknik yang sesuai dengan kebutuhan Anda.
3. **Gunakan Library atau Perangkat Lunak Scraping:** Jika Anda memilih menggunakan bahasa pemrograman seperti Python, Anda dapat menggunakan library atau modul khusus seperti BeautifulSoup atau Scrapy. Ini akan membantu Anda mengambil dan mengolah data dari situs web dengan lebih mudah.
4. **Analisis Struktur Halaman:** Periksa struktur halaman web yang ingin Anda scrap. Identifikasi elemen HTML yang berisi data yang ingin Anda ambil, seperti teks, gambar, tabel, atau atribut lainnya.
5. **Buat Skrip Scraping:** Gunakan bahasa pemrograman atau perangkat lunak scraping untuk membuat skrip yang akan mengekstrak data dari situs web. Ini mungkin melibatkan pencarian elemen HTML berdasarkan tag, kelas, ID, atau struktur lainnya.
6. **Ekstrak Data:** Jalankan skrip Anda untuk melakukan scraping. Data yang ditarik dapat disimpan dalam berbagai format, seperti file CSV, Excel, atau database.

Google-Play-Scraper adalah library Python yang digunakan untuk mengambil atau mengekstrak informasi dari situs web *Google Play Store* secara otomatis (JoMingyu, n.d.). Penggunaan Google-Play-Scraper melibatkan penggunaan skrip komputer untuk mengambil data yang diperlukan dari halaman web *Google Play Store*, seperti detail aplikasi maupun rating dan review. Dengan menggunakan Google-Play-Scraper, dapat digunakan untuk mengumpulkan data dalam jumlah besar dari *Google Play Store* dan kemudian menganalisis atau menggunakannya untuk berbagai tujuan, seperti penelitian, analisis, atau pengembangan aplikasi. Namun, penting untuk menghormati aturan dan etika

penggunaan data yang berlaku dalam proses pengambilan data. Dokumentasi Google-Play-Scraper dapat dilihat pada https://github.com/JoMingyu/google-play-scraper (PlanB, 2019/2023).

App Detail

```
from google_play_scraper import app

result = app(
    'com.nianticlabs.pokemongo',
    lang='en', # defaults to 'en'
    country='us' # defaults to 'us'
)
```

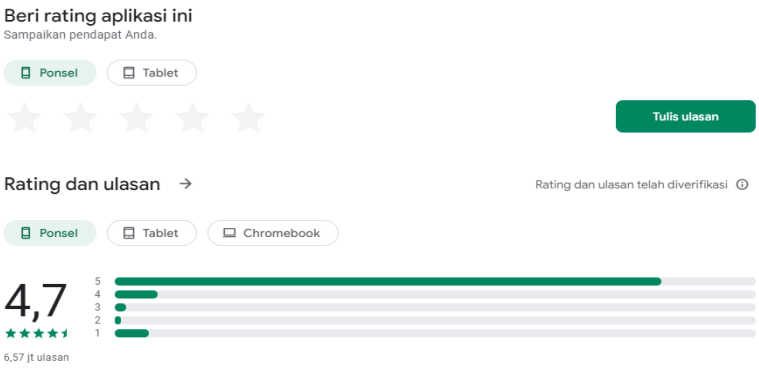
Gambar 2.3 Contoh kode perintah Google-Play-Scraper.

2.2.3 Rating-Based Labeling

Google Play Store adalah platform penyedia aplikasi android yang memungkinkan pengguna untuk mengunduh, membeli, dan mencari perangkat lunak. Platform ini memungkinkan pengguna untuk berbagi peringkat dan ulasan mereka tentang aplikasi dalam bentuk komentar teks (Aldabbas et al., 2020). Sebagai contoh, pengguna dapat menyatakan kepuasan, keluhan terhadap aplikasi, atau mengajukan permintaan fitur baru untuk aplikasi tertentu. Ulasan ini menyediakan banyak jenis informasi tentang aplikasi, seperti kepuasan pengguna, dokumentasi, laporan fitur, dan laporan bug dari pengalaman pengguna terhadap fitur-fitur khusus dalam aplikasi. Ulasan pada aplikasi ini dapat dianggap sebagai "Suara Pengguna" yang sangat membantu bagi upaya pengembangan dan peningkatan aplikasi di masa depan. Contoh ulasan pengguna dapat dilihat pada gambar 2.2

Gambar 2.4 Ulasan pengguna di Google Play Store.

Fitur *Rating* dan *Review* di Google Play Store dapat menunjukkan tingkat kinerja suatu aplikasi. Setiap *rating* bisa disertai *review* atau hanya *rating* saja. *Rating* berbintang dengan skor 5 menunjukkan sangat bagus, skor 4 menunjukkan bagus, skor 3 menunjukkan cukup, skor 2 menunjukkan buruk, dan skor 1 menunjukkan sangat buruk (Raksaka Indra Alhaqq et al., 2022).



Gambar 2.5 Rating pengguna di Google Play Store.

Dari sistem *rating* dan *review* yang ada, maka nilai *rating* yang diberikan oleh pengguna dapat diubah kedalam label sentimen untuk masing-masing rating dan review, yang nantinya bisa digunakan dalam penelitian analisis sentimen pada data tersebut.

2.2.4 Natural Language Toolkit (NLTK)

Natural Language Toolkit (NLTK) merupakan sebuah pustaka Python yang menyediakan sejumlah algoritma pemrosesan bahasa alami (Natural Language Processing atau NLP).

Pustaka ini bersifat sumber terbuka, mudah digunakan, didukung oleh komunitas yang luas, dan memiliki dokumentasi yang lengkap. NLTK mencakup berbagai algoritma, termasuk tokenisasi, part- of-speech tagging, stemming, dan analisis sentiment (Khemani & Adgaonkar, 2021).

Berikut adalah beberapa fitur dan fungsionalitas utama dari NLTK:

1. **Tokenization**: NLTK dapat digunakan untuk memisahkan teks menjadi unit-unit yang lebih kecil, seperti kata-kata atau kalimat. Proses ini disebut tokenisasi, dan NLTK menyediakan alat yang efisien untuk melakukan tugas ini.
2. **Stop Words Removal**: NLTK memiliki daftar kata-kata umum yang tidak memberikan banyak informasi (seperti kata penghubung dan kata depan) yang sering dihapus dalam analisis teks untuk meningkatkan efisiensi dan akurasi.
3. **Stemming dan Lemmatization**: NLTK mendukung teknik stemming (memotong akhiran kata) dan lemmatization (mengubah kata menjadi bentuk dasarnya) untuk mengurangi variasi kata dan menghasilkan bentuk kata yang lebih umum.
4. **Sentiment Analysis**: NLTK dapat digunakan untuk melakukan analisis sentimen, yaitu mengevaluasi dan mengidentifikasi sentimen atau emosi dalam teks, seperti apakah teks tersebut bersifat positif, negatif, atau netral.
5. **Part-of-Speech Tagging**: NLTK memungkinkan pengguna untuk melabeli kata-kata dalam teks dengan bagian dari pidato yang sesuai, seperti kata benda, kata kerja, kata sifat, dan lain-lain.
6. **Structured Text Processing**: NLTK mendukung pengolahan teks yang lebih kompleks, seperti pemisahan paragraf, pengenalan entitas (misalnya, pengenalan nama-nama orang atau tempat), dan analisis sintaksis.
7. **Text Corpora**: NLTK menyediakan kumpulan data teks (korpora) dalam berbagai bahasa dan domain yang dapat digunakan untuk melatih dan menguji model NLP.
8. **Advanced Natural Language Processing**: NLTK juga mendukung tugas-tugas NLP lanjutan seperti pembangunan model bahasa, pemodelan tema, pembuatan model klasifikasi teks, dan banyak lagi.

NLTK sangat berguna bagi para peneliti, pengembang, dan praktisi NLP untuk melakukan berbagai tugas pemrosesan bahasa alami dengan cepat dan efisien. Pustaka ini adalah salah satu alat utama dalam komunitas NLP di Python dan telah menjadi fondasi untuk banyak proyek dan riset di bidang ini. Pustaka NLTK dan dokumentasinya dapat dilihat pada tautan https://www.nltk.org/ (*NLTK :: Natural Language Toolkit*, n.d.).

2.2.5 Sastrawi

Sastrawi adalah sebuah pustaka Python yang memiliki fungsi untuk mengonversi kata-kata dalam Bahasa Indonesia yang bervariasi bentuknya menjadi bentuk dasar (stem). Penggunaan stemming sastrawi dilakukan untuk mengatasi kekurangan NLTK dalam melakukan stemming kata yang berbahsa Indonesia. Pustaka Sastrawi menggunakan berbagai algoritma sebagai inti fungsionalitasnya, termasuk Algoritma Nazief dan Adriani, Enhanced Confix Stripping, dan Modified Enhanced Confix Stripping. Ini adalah sebuah pustaka stemmer yang dirancang untuk mengatasi masalah penggantian kata-kata dengan kata-kata dasar dalam bahasa Indonesia. Sastrawi menerapkan algoritma berdasarkan Nazief dan Adriani, yang kemudian ditingkatkan dengan algoritma CS (Confix Stripping), algoritma ECS (Enhanced Confix Stripping), dan terus ditingkatkan dengan Modified ECS (Rosid et al., 2020).

Berikut ini adalah beberapa fitur dan komponen penting dari Pustaka Sastrawi:

1. **Stemming Bahasa Indonesia**: Salah satu fitur utama dari Pustaka Sastrawi adalah algoritma stemming Bahasa Indonesia yang disediakannya. Stemming adalah proses menghapus infleksi kata untuk mengembalikan kata ke bentuk dasarnya. Pustaka Sastrawi menyediakan implementasi algoritma stemming yang membantu menghasilkan kata dasar dari kata-kata dalam teks.
2. **Kamus Kata Dasar**: Pustaka Sastrawi juga menyediakan kamus kata dasar Bahasa Indonesia yang digunakan dalam proses stemming. Kamus ini berisi kumpulan kata-kata dasar dan kata-kata turunan yang digunakan sebagai referensi dalam proses pemangkasan kata-kata.
3. **Pemangkasan Kata (Stemming) Kontekstual**: Selain algoritma stemming dasar, Pustaka Sastrawi juga menyediakan pemangkasan kata yang lebih kontekstual dan lebih cerdas. Hal ini memungkinkan pemrosesan kata-kata berdasarkan konteks kalimat dan kalimat sekitarnya.
4. **Pemrosesan Dokumen**: Pustaka Sastrawi memiliki fungsi-fungsi untuk memproses teks dalam bentuk dokumen, termasuk pemecahan teks menjadi kalimat-kalimat, dan menghapus karakter-karakter yang tidak diinginkan.
5. **Penyaringan Kata-Kata Tidak Penting**: Pustaka ini juga memiliki fitur penyaringan kata-kata tidak penting (stop words) dalam bahasa Indonesia. Kata-kata tidak penting ini dapat dihilangkan untuk mempermudah analisis dan pemrosesan teks.

Pustaka Sastrawi adalah salah satu pustaka yang sangat penting bagi komunitas peneliti dan pengembang di bidang pemrosesan bahasa alami dalam bahasa Indonesia. Dengan menyediakan algoritma stemming dan alat-alat pemrosesan teks lainnya, pustaka ini membantu mendorong perkembangan teknologi NLP di Indonesia dan mendukung berbagai aplikasi seperti analisis sentimen, pengelompokan dokumen, dan banyak lagi.

2.2.6 Naïve Bayes

Naïve Bayes adalah sebuah metode klasifikasi dalam pembelajaran mesin yang berdasarkan pada teorema Bayes dengan asumsi bahwa setiap fitur atau atribut dalam data adalah independen secara bersyarat terhadap kelas atau labelnya. Meskipun asumsi tersebut sering kali tidak memenuhi kondisi di dunia nyata, metode Naïve Bayes tetap efektif dan sering digunakan dalam berbagai aplikasi, termasuk analisis sentimen, klasifikasi teks, dan kategorisasi dokumen. Algoritma ini memanfaatkan probabilitas dan statistik untuk memprediksi kelas atau label yang paling mungkin dari suatu data, berdasarkan pada data latih yang telah diberi label sebelumnya. Meskipun sederhana dan memiliki asumsi yang kuat, Naïve Bayes sering menghasilkan kinerja yang baik dalam banyak kasus, terutama ketika dataset cukup besar.

Rumus dasar untuk metode klasifikasi Naïve Bayes adalah berdasarkan pada Teorema Bayes. Dalam konteks klasifikasi, teorema ini digunakan untuk menghitung probabilitas suatu kelas (target) berdasarkan atribut atau fitur yang ada dalam data. Rumus dasar untuk Naïve Bayes dituliskan pada rumus (2.1)

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \tag{2.1}$$

Di mana:

$P(C|X)$ adalah probabilitas kelas C diberikan atribut X, yang merupakan nilai yang ingin diprediksi (posterior prob ability).

$P(X|C)$ adalah probabilitas atribut X diberikan kelas C (likelihood).

$P(C)$ adalah probabilitas kelas C (prior probability).

$P(X)$ adalah probabilitas atribut X secara keseluruhan (evidence).

Dalam Naïve Bayes, asumsi dasar yang dibuat adalah bahwa atribut-atribut atau fitur-fitur dalam data saling independen secara kondisional terhadap kelas tertentu. Ini mengarah pada penyederhanaan rumus menjadi rumus (2.2)

$$P(C|X) = \frac{P(X_1|C) \cdot P(X_2|C) \cdot \ldots \cdot P(X_n|C) \cdot P(C)}{P(X)} \quad (2.2)$$

Di mana $X_1, X_2, \ldots, X_n$ adalah atribut-atribut dalam data.

Dalam praktiknya, estimasi probabilitas $P(X_i|C)$ dan $P(C)$ dilakukan berdasarkan data pelatihan. Selanjutnya, probabilitas posterior $P(C|X)$ dapat digunakan untuk memprediksi kelas yang paling mungkin untuk atribut X.

Ada dua variasi dari metode Naïve Bayes yang digunakan untuk klasifikasi dalam pembelajaran mesin, yaitu Gaussian Naïve Bayes dan Multinomial Naïve Bayes. Perbedaan utama di antara keduanya terletak pada asumsi yang dibuat mengenai distribusi atribut atau fitur dalam data.

Gaussian Naïve Bayes digunakan ketika atribut dalam data dianggap memiliki distribusi Gaussian atau distribusi normal. Metode ini cocok untuk data numerik kontinu, seperti data yang memiliki atribut berupa bilangan riil (misalnya, tinggi, berat badan, dll.). Asumsi utama dari Gaussian Naïve Bayes adalah bahwa atribut-atribut ini terdistribusi secara normal dalam setiap kelas. Oleh karena itu, metode ini menghitung probabilitas kelas berdasarkan distribusi Gaussian dari setiap atribut numerik.

Rumus posterior probability untuk Gaussian Naïve Bayes dituliskan pada (2.3)

$$P(C|X) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \cdot P(C) \quad (2.3)$$

Di mana:

$P(C|X)$ adalah probabilitas kelas C diberikan atribut X.

x adalah nilai atribut dalam data yang ingin diprediksi.

μ adalah rata-rata (mean) atribut X untuk kelas C.

σ^2 adalah varians atribut X untuk kelas C.

$P(C)$ adalah probabilitas kelas C.

Multinomial Naïve Bayes cocok digunakan pada data yang memiliki atribut diskrit atau berhitung, terutama data yang dapat dihitung sebagai frekuensi (misalnya, jumlah kata dalam dokumen teks). Metode ini umumnya digunakan dalam analisis teks dan klasifikasi dokumen, di mana setiap atribut mewakili jumlah kemunculan suatu kata atau token.

Asumsi utama dari Multinomial Naïve Bayes adalah bahwa atribut-atribut ini terdistribusi multinomial dalam setiap kelas, dan metode ini menghitung probabilitas kelas berdasarkan frekuensi kemunculan atribut-atribut ini dalam setiap kelas. Rumus posterior probability untuk Multinomial Naïve Bayes dituliskan dalam (2.4)

$$P(C|X) = P(C) \cdot \prod_{i=1}^{n} P(X_i|C)^{X_i} \quad (2.4)$$

Di mana:

$P(C|X)$ adalah probabilitas kelas C diberikan atribut X.

$P(C)$ adalah probabilitas kelas C.

$P(X_i|C)$ adalah probabilitas atribut X_i diberikan kelas C.

X_i adalah jumlah kemunculan atribut X_i dalam data.

Dengan demikian, pilihan antara Gaussian Naïve Bayes dan Multinomial Naïve Bayes tergantung pada jenis data yang akan digunakan dan asumsi distribusi yang paling sesuai dengan atribut-atribut dalam data. Jika atribut adalah data numerik kontinu, Gaussian Naïve Bayes mungkin lebih sesuai. Namun, jika atribut adalah data diskrit atau berhitung, Multinomial Naïve Bayes mungkin menjadi pilihan yang lebih baik.

2.2.7 Logistic Regression

Logistic Regression adalah salah satu teknik analisis regresi yang digunakan untuk memodelkan hubungan antara variabel dependen biner (dichotomous) dengan satu atau lebih variabel independen. Tujuan utama dari Logistic Regression adalah untuk memprediksi probabilitas kejadian dari suatu peristiwa tertentu berdasarkan nilai-nilai dari variabel independen yang ada. Meskipun memiliki kata "regression" dalam namanya, Logistic Regression sebenarnya digunakan untuk masalah klasifikasi, bukan regresi. Tujuannya adalah memprediksi probabilitas bahwa suatu instance data akan termasuk dalam satu kelas atau kategori tertentu. Hasil prediksi ini kemudian dapat diubah menjadi keputusan klasifikasi berdasarkan threshold tertentu.

Model ini menggunakan fungsi logistik (atau sigmoid) untuk menghasilkan output dalam bentuk probabilitas antara 0 dan 1, yang kemudian dapat digunakan untuk mengklasifikasikan data ke dalam kategori yang sesuai. Rumus logistic regression ditulis dalam (2.5)

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p)}} \quad (2.5)$$

Di mana:

$P(Y = 1|X)$ adalah adalah probabilitas bahwa variabel dependen Y adalah 1 (klasifikasi positif) pada kondisi tertentu dari variabel independen X.

e adalah adalah bilangan Euler (2.71828...).

$\beta_0 + \beta_1 + \beta_2 + \cdots + \beta_p$ adalah koefisien yang harus diestimasi dari data pelatihan.

$X_1 + X_2 + \cdots + X_p$ adalah nilai-nilai variabel independen.

Salah satu inti dari Logistic Regression adalah menghitung odds ratio (rasio peluang) dari suatu kejadian dalam kelas yang diinginkan dibandingkan dengan kelas yang tidak diinginkan. Ini memungkinkan kita untuk memahami bagaimana variabel prediktor mempengaruhi probabilitas kejadian.

Pemodelan dalam Logistic Regression didasarkan pada fungsi likelihood, yaitu kemungkinan observasi data diberikan parameter model tertentu. Tujuan utamanya adalah mencari parameter yang memaksimalkan likelihood ini, yang pada akhirnya menghasilkan model yang sesuai dengan data yang diberikan.

Salah satu bentuk umum dari Logistic Regression adalah Binary Logistic Regression atau Logistic Regression Bernoulli. Model ini cocok untuk kasus di mana variabel dependen adalah biner, seperti ya/tidak atau 1/0.

Selain Binary Logistic Regression, ada juga Multinomial dan Ordinal Logistic Regression yang digunakan ketika variabel dependen memiliki lebih dari dua kategori. Multinomial digunakan untuk kasus klasifikasi dengan lebih dari dua kelas, sedangkan Ordinal digunakan untuk data dengan skala ordinal.

Logistic Regression dapat diimplementasikan dengan menggunakan berbagai library dan framework machine learning, seperti scikit-learn untuk Python. Pada dasarnya, Logistic Regression adalah pendekatan yang relatif sederhana namun kuat dalam analisis dan prediksi data kategorikal.

2.2.8 Evaluasi Model

Langkah terakhir dalam penelitian ini adalah melakukan pengukuran evaluasi terhadap kinerja dari pemodelan klasifikasi machine learning yang telah dilakukan pada tahapan sebelumnya. Tujuan dari pengukuran evaluasi ini adalah untuk membandingkan kinerja dan efektivitas dari kedua pemodelan klasifikasi machine learning yang digunakan.

Evaluasi model merupakan tahap penting dalam proses pembuatan model untuk memastikan kinerja model yang baik pada data yang belum pernah dilihat sebelumnya.

Salah satu teknik yang digunakan untuk mengevaluasi dan merangkum kinerja pemodelan klasifikasi machine learning adalah confusion matrix. Confusion matrix adalah matriks yang merangkum total hasil klasifikasi yang benar dan salah. Dengan memperhitungkan nilai true positive (TP), false positive (FP), false negative (FN), dan true negative (TN), kita dapat menganalisis kinerja dari model klasifikasi yang telah dibangun. Evaluasi model dilakukan dengan confusion matrix untuk mengetahui akurasi, precission, recall (Vujovic, 2021).

Confusion matrix adalah alat penting dalam klasifikasi dan pembelajaran mesin yang memungkinkan kita untuk memvisualisasikan kinerja model prediktif dengan membandingkan hasil prediksi dengan kebenaran sebenarnya. Ini sangat berguna saat bekerja dengan masalah klasifikasi biner (dua kelas), tetapi juga dapat diperluas untuk klasifikasi multi-kelas.

Matriks itu sendiri adalah tabel dengan baris dan kolom yang mewakili kelas yang diprediksi dan kelas sebenarnya, secara berurutan. Contoh *confusion matrix* untuk klasifikasi biner atau memiliki dua kelas dapat dilihat di gambar 2.6

		NILAI YANG SEBENARNYA	
		POSITIF (1)	**NEGATIF (0)**
NILAI YANG DIPREDIKSI	**POSITIF (1)**	**TRUE POSITIF (TP)**	**FALSE POSITIF (FP)**
	NEGATIF (0)	**FALSE NEGATIVE (FN)**	**TRUE NEGATIVE (TN)**

Gambar 2.6 Confusion matrix 2 variabel.

Berikut adalah rincian komponen dari confusion matrix:

1. **True Positive (TP)**: Nilai yang sebenarnya positif dan diprediksi dengan benar sebagai positif oleh model.
2. **True Negative (TN)**: Nilai yang sebenarnya negatif dan diprediksi dengan benar sebagai negatif oleh model.

3. **False Positive (FP)**: Nilai yang sebenarnya negatif tetapi diprediksi dengan salah sebagai positif oleh model.
4. **False Negative (FN)**: Nilai yang sebenarnya positif tetapi diprediksi dengan salah sebagai negatif oleh model.

Tidak hanya untuk 2 variabel, confusion matrix juga bisa menggunakan *multi class* lebih dari dua variabel, dengan tetap mnerapkan *true value* / nilai yang sesungguhnya dan *false value* / nilai yang salah. Contoh *confusion matrix* untuk klasifikasi *multi class* atau memiliki 3 kelas dapat dilihat di gambar 2.7

		NILAI YANG SEBENARNYA		
		POSITIF (1)	**NETRAL (0)**	**NEGATIF (-1)**
NILAI YANG DIPREDIKSI	**POSITIF (1)**	**TRUE POSITIF (TP)**	**FALSE POSITIF (FP)**	**FALSE POSITIF (FP)**
	NETRAL (0)	**FALSE NETRAL (FNT)**	**TRUE NETRAL (TNT)**	**FALSE NETRAL (FNT)**
	NEGATIF (-1)	**FALSE NEGATIVE (FN)**	**FALSE NEGATIVE (FN)**	**TRUE NEGATIVE (TN)**

Gambar 2.7 Confusion matrix 3 variabel.

Berikut adalah rincian komponen dari confusion matrix:

1. **True Positive (TP)**: Nilai yang sebenarnya positif dan diprediksi dengan benar sebagai positif oleh model.
2. **True Netral (TNT)**: Nilai yang sebenarnya netral dan diprediksi dengan benar sebagai netral oleh model.
3. **True Negative (TN)**: Nilai yang sebenarnya negatif dan diprediksi dengan benar sebagai negatif oleh model.
4. **False Positive (FP)**: Nilai yang sebenarnya netral atau negatif tetapi diprediksi dengan salah sebagai positif oleh model.
5. **False Netral (FNT)**: Nilai yang sebenarnya positif atau negatif tetapi diprediksi dengan salah sebagai netral oleh model.

6. **False Negative (FN)**: Nilai yang sebenarnya positif atau netral tetapi diprediksi dengan salah sebagai negatif oleh model.

Dengan menggunakan *confusion matrix* kita bisa mengetahui *accuracy*, *precision*, dan *recall*. Akurasi adalah suatu ukuran evaluasi yang memperlihatkan sejauh mana model mampu melakukan prediksi yang tepat secara keseluruhan terhadap seluruh data yang diujikan. Akurasi dihitung menggunakan (2.6)

$$ACC = \frac{TP + TN}{TP + TN + FN + FP} = \frac{TP + TN}{P + N} \tag{2.6}$$

Precission adalah evaluasi yang mengukur seberapa baik model dapat mengenali data positif secara akurat dari total prediksi yang dinyatakan positif. Precison diukur dengan (2.7)

$$PREC = \frac{TP}{TP + FP} \tag{2.7}$$

Recall merupakan evaluasi yang menilai kemampuan model dalam mengenali atau menemukan data positif dari keseluruhan data positif yang tersedia. Metrik ini mengukur persentase data positif yang berhasil diidentifikasi atau ditemukan oleh model dibandingkan dengan total data positif yang ada sebenarnya. Recall dihitung dengan(2.8)

$$REC = \frac{TP+TN}{TP+FN} = \frac{TP}{P} \tag{2.8}$$

Sedangkan dalam menghitung waktu proses yang dihitung dalam penelitian ini adalah proses ketika model dilatih menggunakan data latih hingga model diujikan menggunakan data training dan menghasilkan prediksi. Untuk menghitung waktu proses model, dapat menggunakan beberapa metode atau fungsi yang disediakan oleh bahasa pemrograman atau *library*. Dalam penelitian ini, waktu proses diukur dalam detik (s).

Secara umum, langkah-langkah umum untuk menghitung waktu proses model adalah:

1. **Catat Waktu Awal (Start Time)**: Pada awal proses yang ingin dihitung waktunya, catat waktu mulai eksekusi menggunakan fungsi atau metode yang tersedia dalam

bahasa pemrograman. Misalnya, dalam Python dapat menggunakan modul **time** atau **datetime** untuk ini.

2. **Lakukan Proses Model**: Jalankan proses yang ingin diukur waktu eksekusinya, seperti melatih model, melakukan prediksi, atau evaluasi model.
3. **Catat Waktu Akhir (End Time)**: Setelah proses selesai, catat waktu akhir eksekusi menggunakan fungsi atau metode yang sama seperti langkah pertama.
4. **Hitung Waktu**: Hitung selisih waktu antara waktu akhir dan waktu awal untuk mendapatkan waktu total eksekusi proses. Hasil didapatkan dalam satuan detik (s).

2.3 Analisa Kebutuhan

Analisis kebutuhan adalah proses identifikasi, pengumpulan, dan pemahaman terhadap kebutuhan yang harus dipenuhi dalam sebuah penelitian. Dalam penelitian ini analisis kebutuhan yang akan dilakukan adalah terkait kebutuhan perangkat keras dan perangkat lunak.

2.3.1 Perangkat Keras

Dalam penelitian ini penulis menggunakan Google Colaboratory atau Google Colab untuk melakukan komputasi, maka kebutuhan perangkat keras minimal untuk menjalankan Google Colab termasuk:

1. **Koneksi Internet**: Karena Google Colab adalah platform cloud, maka diperlukan perangkat keras yang dapat digunakan untuk mengakses dan menggunakan layanan internet.
2. **Web Browser**: Google Colab dapat diakses melalui browser web, dan yang digunakan adalah Google Chrome, maka perangkat keras harus dapat menjalankan Google Chrome dengan baik.
3. **Perangkat Input**: Dengan adanya rencana untuk mengunggah atau memanipulasi data, pastikan perangkat input seperti mouse dan keyboard harus dapat trsambung ke perangkat keras.
4. **RAM dan Prosesor**: Perangkat keras membutuhkan RAM dan prosesor yang baik untuk dapat mendukung Google Colab mengelola tugas, dari tugas ringan hingga beban tugas yang dibutuhkan dengan baik.

5. **Kapasitas Penyimpanan**: Perangkat keras diharuskan dapat menyimpan data maupun file yang diperlukan. Pastikan masih cukup ruang penyimpanan untuk mengunduh, mengunggah dan memproses.

perangkat Keras yang digunakan memiliki spesifikasi seperti yang tertera pada tabel 2.2

Tabel 2.2 Spesifikasi perangkat keras

Processor	Intel(R) Core(TM) i3-10100F CPU @ 3.60GHz 3.60 GHz
RAM	Visipro DDR4 8GB
Graphic Card	NVIDIA GeForce GT 730 2 GB
Storage	SSD 240GB
Network	Wireless adapter

2.3.2 Google Colaboratory

Google Colaboratory, yang umumnya disebut sebagai Google Colab, adalah platform pengembangan berbasis cloud yang disediakan oleh Google secara gratis. Ini memungkinkan pengguna untuk menulis dan menjalankan kode Python dalam lingkungan Jupyter Notebook tanpa memerlukan konfigurasi atau instalasi lokal. Google Colab sangat populer di kalangan ilmuwan data, peneliti, dan pengembang karena memberikan akses mudah ke sumber daya komputasi yang kuat tanpa perlu mengeluarkan biaya untuk perangkat keras atau infrastruktur.

Beberapa fitur dari Google Colab adalah:

1. **Lingkungan Jupyter Notebook**: Google Colab menyediakan lingkungan Jupyter Notebook yang interaktif dan user-friendly. Pengguna dapat dengan mudah membuat, menjalankan, dan mengelola kode serta catatan di dalamnya.
2. **Penggunaan GPU dan TPU**: Google Colab memungkinkan pengguna untuk mengakses unit pemrosesan grafis (GPU) dan unit pemrosesan tensor (TPU) untuk mempercepat komputasi pada tugas-tugas yang membutuhkan pemrosesan intensif seperti pelatihan model machine learning.
3. **Pustaka Populer Terinstal**: Banyak pustaka dan alat populer seperti NumPy, pandas, TensorFlow, PyTorch, dan lainnya sudah terinstal di Google Colab. Pengguna tidak perlu menginstalnya secara manual.

4. **Kolaborasi**: Pengguna dapat berkolaborasi pada proyek-proyek yang sama dengan berbagi notebook dengan orang lain. Ini memungkinkan kerja tim yang efisien.
5. **Penyimpanan dan Integrasi Google Drive**: Google Colab terintegrasi dengan Google Drive, memungkinkan pengguna untuk menyimpan, mengelola, dan berbagi notebook serta data secara mudah.
6. **Akses ke Data dan Sumber Daya Google Cloud**: Pengguna dapat dengan mudah mengakses data dari Google Cloud Storage dan BigQuery, serta menggunakan layanan AI dan machine learning dari Google Cloud.
7. **Fasilitas Markdown**: Selain kode, pengguna dapat menambahkan teks, gambar, dan penjelasan dengan menggunakan sintaks Markdown di dalam notebook.

Google Colab adalah alat yang sangat berguna untuk pembelajaran, eksperimen cepat, dan pengembangan proyek-proyek data science dan machine learning tanpa perlu khawatir tentang konfigurasi perangkat keras atau infrastruktur.

BAB 3
Metodologi

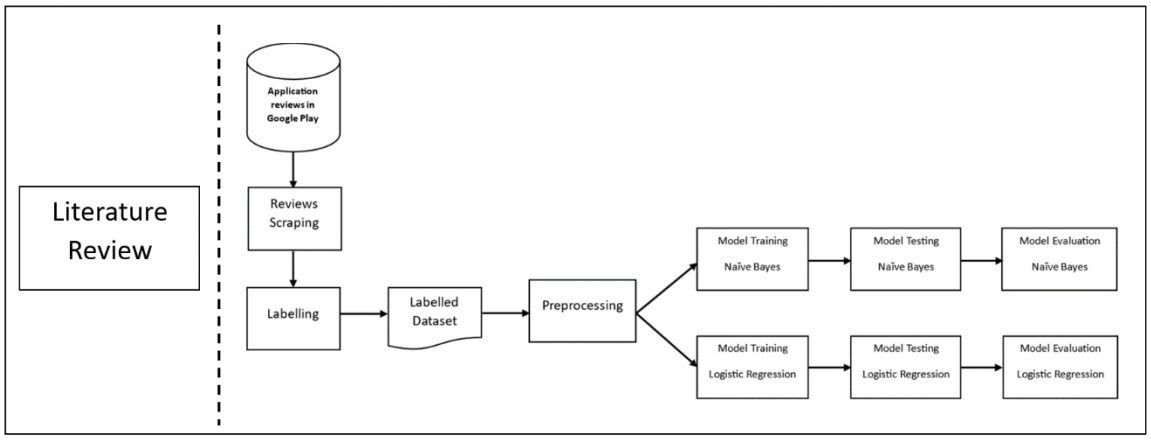
3.1 Tahapan Penilitian

Langkah-langkah atau alur dari metode penelitian yang dilakukan oleh penulis untuk membandingkan metode naïve bayes dan logistic regression dalam sentiment analysis pada review marketplace di google play store menggunakan rating-based labeling adalah:

1. **Studi Literatur**: Penulis melakukan tinjauan literatur untuk mengumpulkan informasi dan pengetahuan yang sudah ada tentang analisis sentimen, metode Naïve Bayes, Logistic Regression, serta aplikasi mereka dalam konteks analisis review di platform Google Play Store. Studi literatur membantu penulis memahami dasar-dasar teori dan konsep yang relevan sebelum memulai penelitian.
2. **Pengumpulan Data**: Pada langkah ini, penulis mengumpulkan data berupa review dari aplikasi marketplace di Google Play Store. Data ini dapat mencakup berbagai aspek, seperti teks ulasan, peringkat aplikasi, dan informasi terkait lainnya. Pengumpulan data yang representatif dan relevan penting untuk memastikan hasil penelitian memiliki validitas.
3. **Pelabelan Data**: Setelah data terkumpul, penulis melabeli setiap ulasan dengan sentimen yang sesuai, yaitu positif, negatif, atau netral. Proses pelabelan ini merupakan tahap kritis karena menjadi dasar bagi model machine learning untuk belajar dan melakukan klasifikasi sentimen.
4. **Preprocessing Data**: Data ulasan sering kali memerlukan preprocessing sebelum dimasukkan ke dalam model. Ini bisa mencakup langkah-langkah seperti mengubah teks menjadi huruf kecil, menghapus tanda baca, menghilangkan kata-kata umum (stop words), serta stemming atau lemmatisasi untuk mengubah kata-kata menjadi bentuk dasar.
5. **Pemodelan**: Pada langkah ini, penulis mengimplementasikan metode Naïve Bayes dan Logistic Regression untuk membangun model klasifikasi sentimen. Data yang sudah diolah dan dilabeli digunakan untuk melatih kedua model ini. Naïve Bayes akan menggunakan probabilitas kondisional untuk mengklasifikasikan ulasan ke dalam sentimen tertentu, sementara Logistic Regression akan memodelkan hubungan antara variabel independen (fitur) dan variabel dependen (sentimen).

6. **Evaluasi**: Setelah model terlatih, langkah terakhir adalah melakukan evaluasi. Dalam konteks ini, penulis mengukur performa kedua model dalam hal akurasi, presisi, dan recall. Akurasi mengukur sejauh mana model benar dalam mengklasifikasikan sentimen secara keseluruhan. Presisi mengukur seberapa tepat model dalam mengklasifikasikan sentimen positif atau negatif, sedangkan recall mengukur seberapa baik model dalam mengidentifikasi sentimen yang sebenarnya positif atau negatif.

   Alur tahapan yang dilakukan oleh peneliti dapat dilihat pada gambar 3.1



Gambar 3.1 Tahapan Penelitian.

3.2 Literature Review

Dalam literatur review, penulis melakukan penyelidikan mendalam terhadap berbagai sumber informasi yang relevan untuk memahami dan mengumpulkan pengetahuan yang telah ada seputar topik tertentu. Berikut adalah tahapan yang dilakukan dalam literatur review:

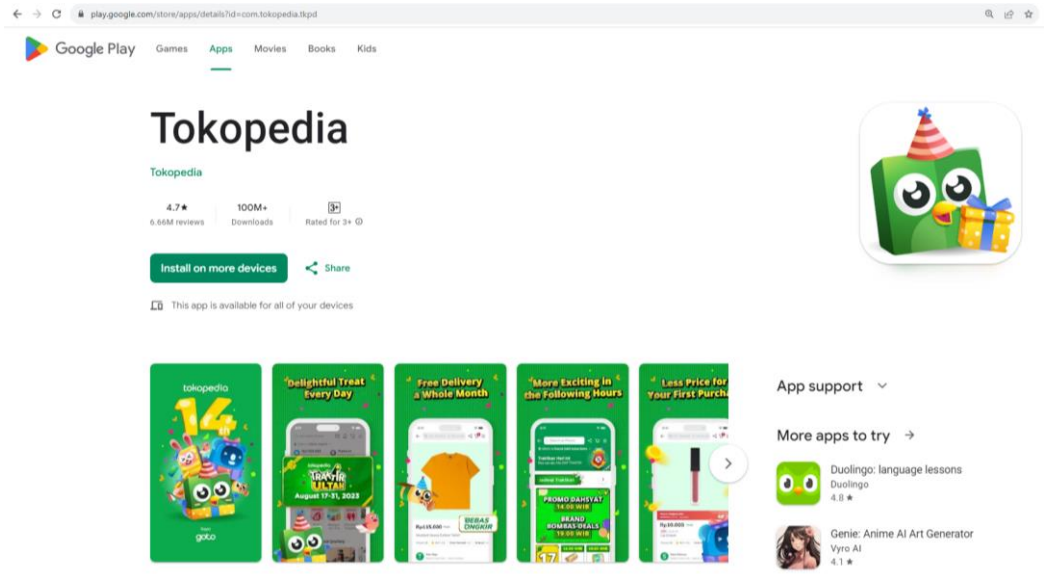
1. **Identifikasi Topik dan Tujuan**: Penulis mengidentifikasi topik penelitian yang akan didalami yaitu penelitian yang berhubungan dengan perbandingan model, algoritma naïve bayes, algoritma logistic regression, sentiment analysis, pemelitian dengan data review google play store, penelitian dengan data ulasan *marketplace* dan teknik pelebelan.
2. **Pengumpulan Sumber Informasi**: Penulis mengumpulkan sumber-sumber informasi yang relevan, seperti jurnal ilmiah, artikel, buku, laporan penelitian, dan sumber-sumber lainnya yang terkait dengan topik penelitian. Pengumpulan sumber-sumber ini dapat dilakukan melalui basis data akademis, perpustakaan, situs web

resmi, dan sumber-sumber lainnya. Sumber-sumber yang dikumpulkan berjumlah 27 sumber.

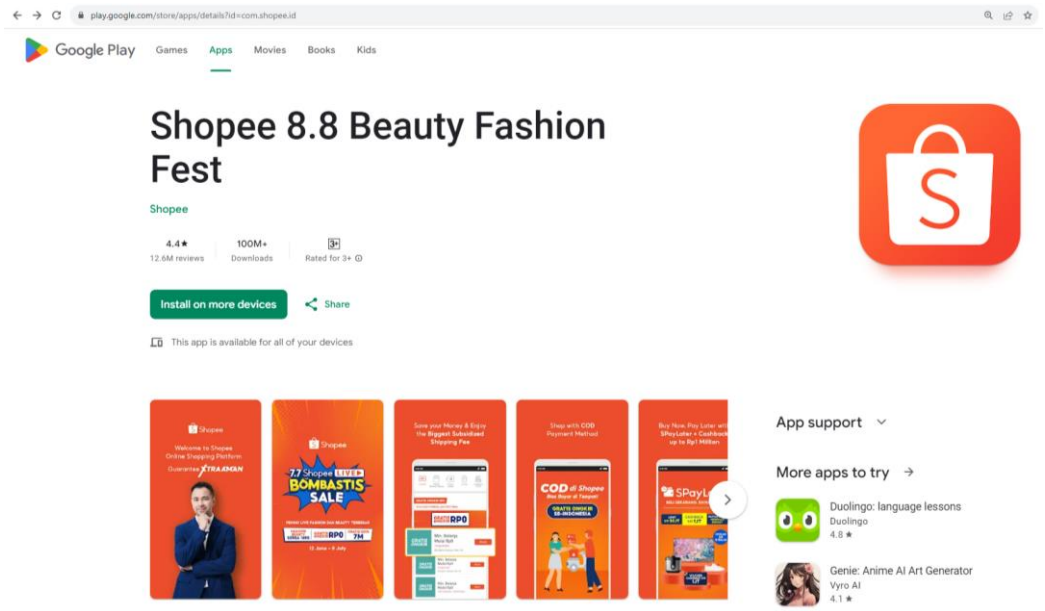
3. **Seleksi dan Penilaian Sumber**: Penulis menilai kualitas dan kredibilitas setiap sumber informasi yang dikumpulkan. Sumber-sumber yang relevan dan berkualitas tinggi dipilih untuk disertakan dalam literatur review, sementara sumber yang kurang relevan atau tidak kredibel dihindari. Sumber -sumber yang dipakai adalah jurnal-jurnal yang dianggap berkualitas dinilai berdasarkan jurnal tersebut diterbitkan oleh sumber yang terpercaya. Sebanyak 12 jurnal dipilih sebagai sumber yang berkualitas dan dijadikan rujukan dalam penelitian ini.
4. **Analisis dan Sinthesis**: Penulis menganalisis isi dari setiap sumber informasi yang terpilih. Mereka mengidentifikasi tema, argumen, temuan, dan konsep-konsep kunci yang muncul dalam literatur terkait. Selanjutnya, penulis mensintesis informasi dari berbagai sumber untuk mengembangkan pemahaman yang lebih komprehensif tentang topik tersebut.
5. **Pelaporan Hasil**: setelah melakukan analisis dan sinthesis informasi yang ada, penulis merancang metode penelitian mulai dari pengumpulan data hingga melakukan analisis data dan kemudian melakukan pelaporan dalam bentuk jurnal dan diterbitkan secara resmi dalam jurnal bereputasi.

3.3 Pengumpulan Data

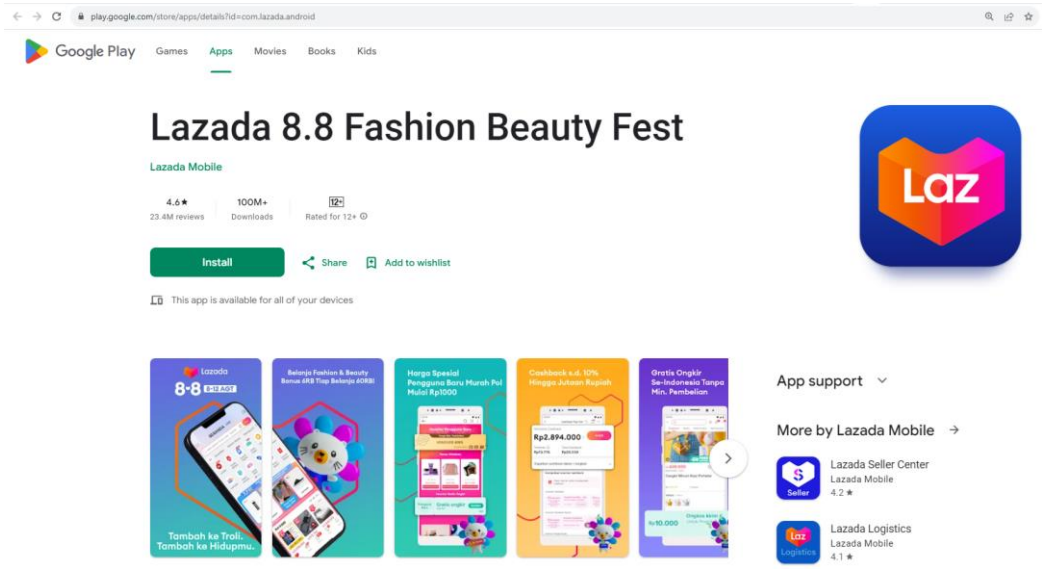
Pada saat pengumpulan data penelitian ini menggunakan library python “Google-play-scraper” untuk mengekstraksi informasi dari halaman website. Langkah yang pertama adalah menentukan target scraping atau alamat website yang akan diekstrak. Google-play-scraper dikhususkan untuk mengambil data aplikasi dari website Google Play Store, maka target yang dimaksud adalah aplikasi tujuan, dalam penelitian ini aplikasi yang akan menjadi target adalah empat aplikasi *marketplace* yang masuk dalam top aplikasi dalam kategori belanja, aplikasi tersebut adalahTokopedia, Shopee, Lazada dan Blibli. Halaman website dari aplikasi - apliaksi tersebut dapat dilihat pada gambar dibawah ini.



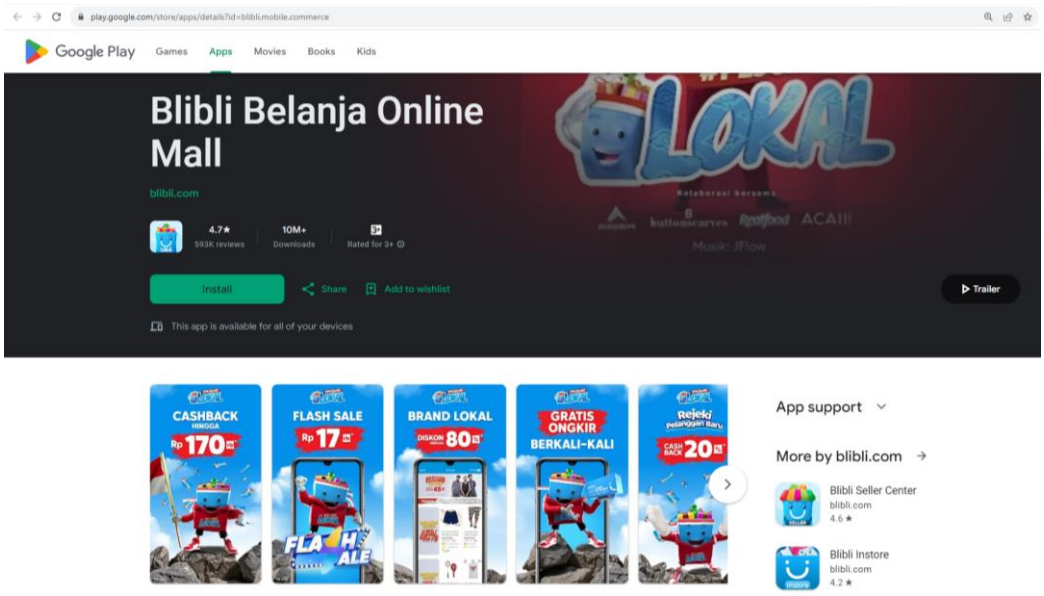
Gambar 3.2 Halaman aplikasi Tokopedia pada Google Play Store.



Gambar 3.3 Halaman aplikasi Shopee pada Google Play Store.

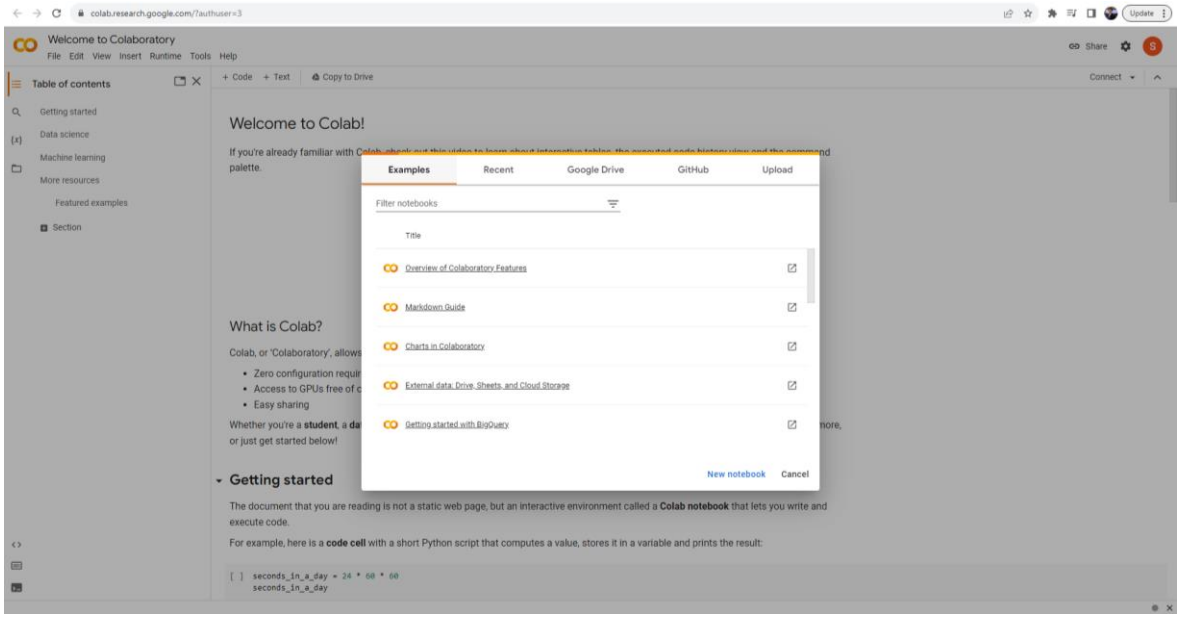


Gambar 3.4 Halaman aplikasi Lazada pada Google Play Store.



Gambar 3.5 Halaman aplikasi Tokopedia pada Google Play Store.

Langkah selanjutnya adalah membuka layanan Google Colab pada Google Chrome atau *browser* yang ada pada PC dengan mengetikan tautan https://colab.research.google.com/ hingga muncul tampilan seperti gambar 3.6. Setelahnya adalah memulai lembar kerja baru dengan mengklik ”New Notebook”.



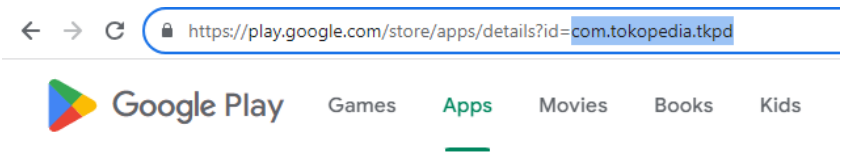
Gambar 3.6 Halaman awal Google Colab.

Setelah memulai lembar kerja baru pada Google Colab yang dilakukan selanjutnya adalah menginstal library Google-play-scraper pada lembar kerja dengan mengetikan perintah *"!pip install google-play-scraper*"

```
[ ]  #install library google_play_scraper
     !pip install google-play-scraper
```

Gambar 3.7 Menginstall library Google-play-scraper.

Sebelum mengekstrak review dari aplikasi marketpalce, harus diketahui dahulu "ID" atau kode identitas dari aplikasi tersebut. Kode identitas tersebut dapat dilihat dari url yang ada pada address bar, keterangan kode identitas apliaksi terletak setelah "*https://play.google.com/store/apps/details?id=*". sebagai contoh pada gambar 3.8 adalah adress bar dari aplikasi tokopedia yang memiliki url penuh "*https://play.google.com/store/apps/details?id=com.tokopedia.tkpd*" maka kode identitas dari Tokopedia adalah " *com.tokopedia.tkpd"*. Kode ini yang nantinya dimasukan pada *source code* Google-play-scraper.



Gambar 3.8 ID Aplikasi Tokopedia.

Setelah mendapatkan ID aplikasi langkah selanjutnya adalah impor modul yang diperlukan dari pustaka google-play-scraper ke dalam skrip dengan menuliskan "from google_play_scraper import Sort, reviews_all" dan dilanjut menuliskan skrip *scraping* google-play-scraper, tahapan ini dapat dilihat pada gambar 3.9.

```
[ ]  from google_play_scraper import Sort, reviews_all

     #scrape semua review tokopedia
     result_tp = reviews_all(
         'com.tokopedia.tkpd',
         sleep_milliseconds=0, # defaults to 0
         lang='id', # defaults to 'en'
         country='id', # defaults to 'us'
         sort=Sort.NEWEST, # defaults to Sort.MOST_RELEVANT
         # filter_score_with=5 # defaults to None(means all score)
     )
```

Gambar 3.9 Kode *scraping* aplikasi Tokopedia menggunakan Google-play-scraper.

Dari gambar diatas terdapat *result_tp* yang menjadi variabel untuk menampung hasil *scraping* dari *review* aplikasi Tokopedia. *sleep_milliseconds=0* adalah pengaturan untuk jeda eksekusi dari kode, angka 0 menunjukan bahwa skrip tidak menggunakan waktu jeda, jika nilainya diubah maka akan menunjukan ukuran jeda dalam milidetik. *lang='id'* adalah filter untuk bahasa dari review, 'id' menunjukan Bahasa yang diambil adalah Bahasa Indonesia, kode bahasa 'id' bisa diubah dengan Bahasa lain sesuai kode Bahasa yang kita inginkan. *country='id'* adalah filter dari dimana aplikasi tersebut diunduh, 'id' menunjukan *review* yang diambil hanyalah *review* yang berasal dari pengguna di negara Indonesia. *sort=Sort.NEWEST* adalah perintah untuk mensortir *review* dari yang paling baru. Sedangkan untuk perintah *filter_score_with=* adalah perintah untuk menyaring *review* berdasarkan rating tertentu, karena yang diambil adalah semua rating maka perintah ini tidak diaktifkan.

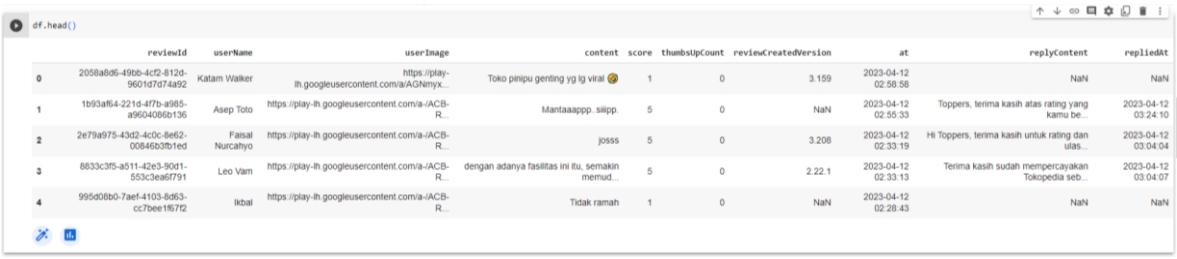
Setelah data ditampung pada variabel *result_tp* maka langkah selanjutnya menampilkan data hasil scraping menggunakan google-play-scraper dengan cara memasukannya terlebih dahulu pada *dataframe* menggunakan *library pandas.* Untuk menggunakan *library pandas* sebelumnya perlu memanggil *library* tersebut dengan perintah *import pandas as pd* dan untuk memasukan dataset ke dataframe pandas menggunakan perintah *df = pd.DataFrame(result_tp)*.

```
[ ]  import pandas as pd

     df = pd.DataFrame(result_tp)
```

Gambar 3.10 Kode untuk memasukan data ke dataframe pandas.

Untuk melihat isi *dataframe* menggunakan perintah *head()*, jika kita tidak mengatur jumlah data yang akan ditampilkkan dengan mengisi angka pada dalam kurung maka secara *default* data yang akan tampil berjumlah 5.



	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt
0	2058a8d6-49bb-4cf2-812d-9601d7d74a92	Katam Walker	https://play-lh.googleusercontent.com/a/AGNmyx...	Toko pinipu genting yg lg viral 😡	1	0	3.159	2023-04-12 02:58:58	NaN	NaN
1	1b93af64-221d-4f7b-a985-a9604086b136	Asep Toto	https://play-lh.googleusercontent.com/a-/ACB-R...	Mantaaappp..siiipp.	5	0	NaN	2023-04-12 02:55:33	Toppers, terima kasih atas rating yang kamu be...	2023-04-12 03:24:10
2	2e79a975-43d2-4c0c-8e62-00846b3fb1ed	Faisal Nurcahyo	https://play-lh.googleusercontent.com/a-/ACB-R...	josss	5	0	3.208	2023-04-12 02:33:19	Hi Toppers, terima kasih untuk rating dan ulas...	2023-04-12 03:04:04
3	8833c3f5-a511-42e3-90d1-553c3ea6f791	Leo Vam	https://play-lh.googleusercontent.com/a-/ACB-R...	dengan adanya fasilitas ini itu, semakin memud...	5	0	2.22.1	2023-04-12 02:33:13	Terima kasih sudah mempercayakan Tokopedia seb...	2023-04-12 03:04:07
4	995d08b0-7aef-4103-8d63-cc7bee1f67f2	Ikbal	https://play-lh.googleusercontent.com/a-/ACB-R...	Tidak ramah	1	0	NaN	2023-04-12 02:26:43	NaN	NaN

Gambar 3.11 Tampilan dataset saat ditampilkanalam *dataframe*.

Setelah mengetahui isi dataset sesuai dengan dataset yang dibutuhkan, maka selanjutnya adalah menyimpannya untuk digunakan pada proses selanjutnya dengan perintah *df.to_csv('Tokopedia_raw.csv', index=False, encoding='utf-8')*, yang berarti dataset disimpan dalam format csv dengan nama Tokopedia_raw.csv.

```
[ ]  df.to_csv('Tokopedia_raw.csv', index=False, encoding='utf-8')
```

Gambar 3.12 Perintah untuk menyimpan dataset dalam csv.

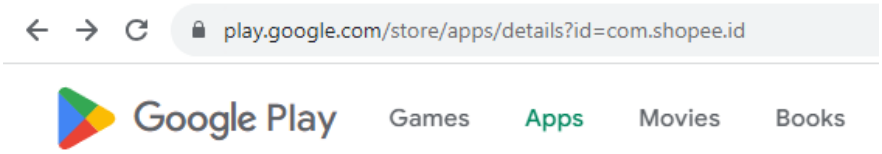
Jika ingin menyimpan data pada Google Drive kita bisa menghubungkan Google Colab dengan Google Drive dengan memanggil *drive* menggunakan perintah *from google.colab import drive* dan kemudaian mount drive dengan perintah *drive.mount('/content/drive').*

```
from google.colab import drive
drive.mount('/content/drive')
```

Gambar 3.13 Perintah untuk menghubungkan ke *Google Drive*.

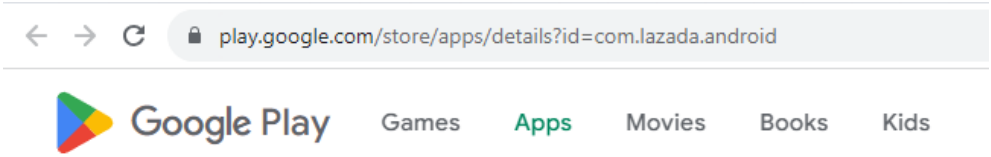
Langkah yang sudah dijelaskan sebelumnya diterapkan kepada target aplikasi yang lainnya dengan mengambil kode identitas dari aplikasi Shopee, Lazada dan Blibli.

Adress bar dari aplikasi Shopee yang memiliki url penuh ”*https://play.google.com/store/apps/details?id=com.shopee.id*” maka kode identitas dari Shopee adalah ”*com.shopee.id”*



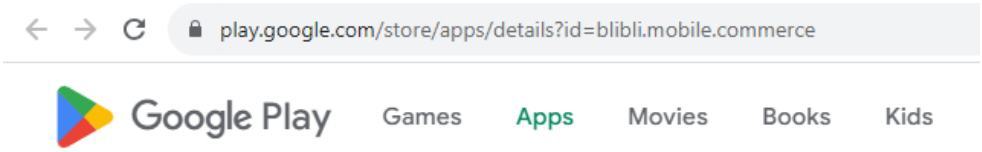
Gambar 3.14 ID Aplikasi Shopee.

Adress bar dari aplikasi Lazada yang memiliki url penuh ”*https://play.google.com/store/apps/details?id=com.lazada.android”* maka kode identitas dari Lazada adalah ” *com.lazada.android”*



Gambar 3.15 ID Aplikasi Lazada.

Adress bar dari aplikasi Blibli yang memiliki url penuh ”*https://play.google.com/store/apps/details?id=blibli.mobile.commerce*” maka kode identitas dari Shopee adalah ”*blibli.mobile.commerce”*



Gambar 3.16 ID Aplikasi Blibli.

Data yang didapatkan adalah data berupa teks dengan 10 variabel yaitu: 'reviewId', 'userName', 'userImage', 'content', 'score', 'thumbsUpCount', 'reviewCreatedVersion', 'at', 'replyContent', dan 'repliedAt'.

3.4 Pemberian Rating-Based Labeling

Pemberian label atau kategori tersebut berupa kelas atau kategori yang sudah ditentukan dan sesuai dengan tujuan klasifikasi sentimen. Dalam penelitian ini dilakukan rating based labeling dengan 2 macam kondisi pelebelan yaitu kondisi 2 label (positif & negatif). Label positif diambilkan dari komentar dengan rarting lima sedangkan lainnya akan dikategorikan menjadi negatif. Dan 3 label (positif, netral, & negatif), dengan

menggunakan rating 5 dan 4 menjadi positif, rating 3 menjadi netral, rating 2 dan 1 menjadi negatif.

Pelabelan dilakukan dengan cara mengubah nilai rating menjadi label yang sudah ditentukan sesuai dengan aturan yang akan ditetapkan. Pengubahan nilai rating menjadi 2 label dilakukan menggunakan Google Colab dengan script seperti gambar berikut.

```
[ ]  #ubah skor jadi label
     df.loc[df["score"] == 1, "score"] = "negatif"
     df.loc[df["score"] == 2, "score"] = "negatif"
     df.loc[df["score"] == 3, "score"] = "negatif"
     df.loc[df["score"] == 4, "score"] = "negatif"
     df.loc[df["score"] == 5, "score"] = "positif"
```

Gambar 3.17 *Script* untuk kondisi 2-label.

Dataset yang sudah disimpan dalam google drive terlebih dahulu dipanggil dengan menghubungkan Google Colab dengan Google Drive dan kemudian memasukan dataset tersebut ke dalam dataframe "*df*".

Untuk dataset dengan kondisi 3 label, pengubahan nilai rating dilakukan dengan menggunakan perintah seperti gambar berikut.

```
[ ]  #ubah skor jadi label
     df.loc[df["score"] == 1, "score"] = "negatif"
     df.loc[df["score"] == 2, "score"] = "negatif"
     df.loc[df["score"] == 3, "score"] = "netral"
     df.loc[df["score"] == 4, "score"] = "positif"
     df.loc[df["score"] == 5, "score"] = "positif"
```

Gambar 3.18 *Script* untuk kondisi 3-label.

Dari 10 variable data yang didapatkan yaitu: 'reviewId', 'userName', 'userImage', 'content', 'score', 'thumbsUpCount', 'reviewCreatedVersion', 'at', 'replyContent', dan 'repliedAt'. Selanjutnya yang akan digunakan adalah kolom 'content' yang berisi ulasan dan juga kolom 'score' yang berisi rating yang sudah diubah kedalam label.

```
label = df[['content','score']]
label.head()
```

Gambar 3.19 Pemilihan variable dataset.

Setelah proses pengubahan rating menjadi label baik 2-label maupun 3-label selanjutnya adalah proses pengambilan sample masing-masing label yang berjumlah 10.000 data, proses pengambilan data dilakukan untuk mempermudah dan mempersingkat waktu dalam proses selanjutnya yaitu *data preprocessing*. Perintah yang digunakan dapat dilihat pada gambar berikut.

```
from pandas.core.common import random_state
col = 'score'
sample = []

variants = list(df[col].dropna().unique())
print(variants)

for typ in variants:
  sample.append(label[label[col] == typ].sample(10000, replace=True, random_state=0))
datasample = pd.concat(sample)
datasample.head(50)
```

Gambar 3.20 *Script* untuk pengambilan 10.000 sample.

Selanjutnya masing – masing data sample dengan kondisi 2 label dan 3 label kembali disimpan dalam bentuk csv sebagai data yang sudah dilabeli untuk digunakan dalam proses selanjutnya.

3.5 Data Preprocessing

Sebelum dilakukan pemodelan menggunakan data review aplikasi, dataset tersebut melalui tahap *data preprocessing* atau dalam kasus data teks disebut juga *text preprocessing*. Tahapan ini dilakukan untuk mempersiapkan data agar dapat diproses secara efektif dan efisien oleh algoritma machine learning. Adapun tahapan yang dilakukan dalam proses ini adalah case folding, cleansing, tokenizing, remove stop words, stemming.

Tahap-tahap *text preprocessing* menggunakan beberapa library dan dilakukan dalam fungsi preprocessing. Berikut adalah penjelasan lebih lanjut tentang library yang digunakan:

- re: Library ini digunakan untuk bekerja dengan ekspresi reguler (regular expressions) dalam mengganti atau menghilangkan pola tertentu dalam teks.
- nltk: Natural Language Toolkit (NLTK) adalah library pemrosesan bahasa alami yang digunakan untuk tokenisasi kata, penghapusan stop words, dan pengambilan stopwords.
- nltk.tokenize: Submodul ini digunakan untuk melakukan tokenisasi kata.
- nltk.corpus: Submodul ini digunakan untuk mengakses corpus NLTK seperti stop words.

- Sastrawi.StopWordRemover.StopWordRemoverFactory: Digunakan untuk membuat objek penghapus stop words dari pustaka Sastrawi.
- Sastrawi.Stemmer.StemmerFactory: Digunakan untuk membuat objek stemmer dari pustaka Sastrawi.

Tahapan yang dilakukan dalam *text preprocessing* adalah:

1. Lowercasing: Teks awal diubah menjadi huruf kecil dengan menggunakan fungsi lower(). Ini membantu dalam menghindari perbedaan kata yang sama karena perbedaan huruf besar dan kecil.

```
# Lowercase the text
text = text.lower()
```

Gambar 3.21 Proses *lowercasing*.

2. Penghapusan Angka dan Karakter Khusus: Dalam langkah ini, ekspresi reguler re.sub(r'[^a-zA-Z\s]', '', text) digunakan untuk menghilangkan angka dan karakter khusus dari teks.

```
# Remove digits and special characters
text = re.sub(r'[^a-zA-Z\s]', '', text)
```

Gambar 3.22 Proses penghapusan angka dan karakter khusus.

3. Penghapusan Tautan (Links): Ekspresi reguler re.sub(r'http\S+', '', text) digunakan untuk menghapus tautan (URL) dari teks.

```
# Remove links
text = re.sub(r'http\S+', '', text)
```

Gambar 3.23 Proses penghapusan tautan (links).

4. Tokenisasi: Menggunakan nltk.word_tokenize(text), teks dipecah menjadi token atau kata-kata terpisah.

```
# Tokenizing
text_tokens = nltk.word_tokenize(text)
```

Gambar 3.24 Proses penghapusan tautan (links).

5. Penghapusan Stop Words dengan NLTK: Set stop words dari NLTK (stopwords.words('indonesian')) digunakan untuk menghapus kata-kata stop words (kata-kata umum yang tidak memiliki makna signifikan seperti "dan", "di", "ke") dari token.

```
# Remove stop words nltk
stop_words = set(stopwords.words('indonesian'))
filtered_text = [word for word in text_tokens if word not in stop_words]
text = ' '.join(filtered_text)
```

Gambar 3.25 Proses *Stop Words* dengan NLTK.

6. Penghapusan Stop Words dengan Sastrawi: Pustaka Sastrawi digunakan untuk menghapus stop words yang tersisa setelah tahap sebelumnya.

```
# Remove stop words sastrawi
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
text = stopword.remove(text)
```

Gambar 3.26 Proses *Stop Words* dengan Sastrawi.

7. Stemming: Pustaka Sastrawi juga digunakan untuk melakukan stemming, yaitu mengubah kata-kata menjadi bentuk dasarnya. Ini membantu dalam mengatasi variasi kata yang memiliki akar yang sama.

```
#steming
factory = StemmerFactory()
stemmer = factory.create_stemmer()
text = stemmer.stem(text)
```

Gambar 3.27 Proses Stemming Sastrawi.

Tahapan-tahapan ini dimasukan kedalam fungsi yang bernama preprocessing dan diterapkan pada kolom 'content' dari DataFrame df menggunakan fungsi apply(). Hasil dari tahap preprocessing disimpan dalam kolom baru bernama 'clean'.

```
[ ]  def preprocesing(text):

         # Lowercase the text
         text = text.lower()

         # Remove digits and special characters
         text = re.sub(r'[^a-zA-Z\s]', '', text)

         # Remove links
         text = re.sub(r'http\S+', '', text)

         # Tokenizing
         text_tokens = nltk.word_tokenize(text)

         # Remove stop words nltk
         stop_words = set(stopwords.words('indonesian'))
         filtered_text = [word for word in text_tokens if word not in stop_words]
         text = ' '.join(filtered_text)

         # Remove stop words sastrawi
         factory = StopWordRemoverFactory()
         stopword = factory.create_stop_word_remover()
         text = stopword.remove(text)

         #steming
         factory = StemmerFactory()
         stemmer = factory.create_stemmer()
         text = stemmer.stem(text)

         return text

     df['clean'] = df['content'].apply(lambda x: preprocesing(x))
```

Gambar 3.28 Fungsi preprocessing.

Setelah tahapan *text preprocessing* dilakukan didapatkan hasil teks menjadi lebih efesien untuk diproses dalam tahap modeling. Contoh teks hasil preprocessing dapat dilihat dalam table berikut.

Tabel 3.1 Hasil *text preprocessing*.

content	score	clean
Mantep lebih mudah tuk belanja saya suka	positif	mantep mudah tuk belanja suka
Mantap aplikasinya. Gampang dapet gratis ongkir. Semoga lebih banyak promonya.	positif	mantap aplikasi gampang dapet gratis ongkir moga promonya
Aplikasi belanja online yang mudah digunakan. Banyak free ongkir dan diskon lebaran.	positif	aplikasi belanja online mudah free ongkir diskon lebaran
Aplikasi yang bagus tidak sulit untuk menggunakanya. Tidak menyesal menginstal aplikasi ini	positif	aplikasi bagus sulit menggunakanya sesal menginstal aplikasi
Mudah penggunaan dan belanjanya	positif	mudah guna belanja

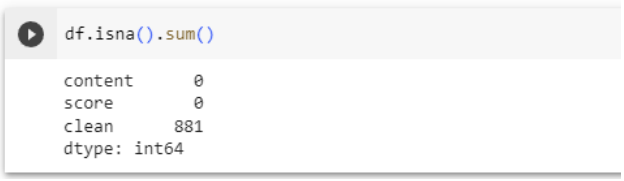
Aplikasi nya jelek banget, di iklan nya murah-murah pas di buka mahalnya minta ampun. Nyesal saya download aplikasi ini.....	negatif	aplikasi nya jelek banget iklan nya murahmurah pas buka mahal ampun nyesal download aplikasi
bangsat hei kalau punya apk toko jangan menampilkan iklan apalagi ketika saat online/main game online awas	negatif	bangsat hei apk toko tampil iklan onlinemain game online awas
Payah...berat amat di jaringan...terlalu sensitif...beda sama apps tetangga yg lebih ringan	negatif	payahberat jaringanterlalu sensitifbeda apps tetangga yg ringan
Tidak bisa batalkan pesanan padahal brg yang dipesan habis	negatif	batal pesan brg pes habis
Hasil refund jadi vocer, tidak bisa jadi uang kembali. Dan saat dipakai untuk belanja jadi tidak bisa gratis ongkir! Sangat mengecewakan!	negatif	hasil refund vocer uang pakai belanja gratis ongkir kecewa

Hasil dari text preprocessing ini disimpan dalam file csv dan ditandai sebagai ‘data clean’ untuk digunakan dalam proses selanjutnya.

3.6 Data Modeling dan Evaluasi

Dalam tahap ini model yang akan digunakan untuk memprediksi atau mengklasifikasikan adalah Multinominal Naïve Bayes dan Logistic Regression, seluruh data diuji dengan dua buah model yang dirancang. Dalam pembuatan model diperlukan data latih yang bertujuan untuk mengajarkan atau melatih model machine learning agar dapat mengenali pola dan mengidentifikasi hubungan antara variabel-variabel dalam data. Dalam fase pelatihan, model machine learning menggunakan data training untuk menyesuaikan parameter dan aturan yang diperlukan supaya dapat memahami karakteristik data dan membuat prediksi yang akurat. Setelah model dilatih menggunakan data latih selanjutnya model akan diuji kinerjanya dengan data testing yang belum pernah digunakan sebelumnya dalam pelatihan model. Di penelitian ini menggunakan 80% dari dataset sebagai data latih dan 20% sisanya sebagai data testing.

Langkah pertama adalah memuat data yang siap diproses, data tersebut adalah ‘data clean’ yang sebelumnya diproses dalam *text preprocessing*. Perlu dilakukan pemeriksaan terlebih dahulu apakah ada data yang menjadi kosong akibat proses sebelumnya. Fungsi yang digunakan adalah sebagai berikut



Gambar 3.29 Proses melihat data kosong.

Dari proses diatas terlihat bahwa pada kolom *clean* terdapat 881 data kosong, hal itu dapat disebabkan oleh proses penyederhanaan teks sebelumnya, jika teks yang ada pada kolom *content* termasuk dalam kata tanpa makna yang dihilangkan pada proses penghapusan stopword maka hasil pada kolom *clean* bisa menjadi kosong.

Jika kondisi ini terjadi maka perlu dilakukan perataan data untuk mengatasi *imbalance* data. Proses yang dilakukan adalah dengan menghilangkan data yang kosong dengan perintah *dropna.*

```
[ ]  bersih = df.dropna(axis = 0, how ='any')
```

Gambar 3.30 Proses menghapus data kosong.

Selanjutnya menurunkan jumlah data dari 10.000 data menjadi 9.000 data tiap labelnya menggunakan proses sebagai berikut.

```
[ ]  from pandas.core.common import random_state
     col = 'score'
     sample = []

     variants = list(bersih[col].dropna().unique())
     print(variants)

     for typ in variants:
       sample.append(bersih[bersih1[col] == typ].sample(9000, replace=False, random_state=0))
     data = pd.concat(sample)
     data.head()
```

Gambar 3.31 Proses pengambilan data clean.

Setelah data seimbang dan tidak ada data kosong, langkah selanjutnya adalah memanggil berberapa *modul* scikit-learn (sklearn) yang akan digunakan. modul yang digunakan adalah:

1. time: Ini adalah modul yang digunakan untuk mengukur waktu eksekusi suatu kode. Modul ini sering digunakan untuk menghitung berapa lama waktu yang dibutuhkan untuk menjalankan suatu proses.

2. from MultinomialNB: Ini adalah implementasi dari algoritma Naive Bayes untuk data kategorikal (seperti data teks yang diubah menjadi representasi bag-of-words).
3. LogisticRegression: Ini adalah implementasi dari algoritma regresi logistik, yang digunakan dalam klasifikasi.
4. TfidfVectorizer: Ini adalah modul yang digunakan untuk mengubah teks menjadi representasi numerik berdasarkan metode TF-IDF (Term Frequency-Inverse Document Frequency).
5. train_test_split: Fungsi ini digunakan untuk membagi data menjadi data latih dan data uji.
6. accuracy_score, precision_score, recall_score: Ini adalah metrik evaluasi kinerja model seperti akurasi (accuracy_score), presisi (precision_score), dan recall (recall_score).

```
[ ]  import time
     from sklearn.naive_bayes import MultinomialNB
     from sklearn.linear_model import LogisticRegression
     from sklearn.feature_extraction.text import TfidfVectorizer
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import accuracy_score, precision_score, recall_score
```

Gambar 3.32 Modul yang digunakan dalam proses modeling.

Selanjutnya untuk data paka kolom score ditandai sebagai label dan data pada kolom clean ditandai dengan cleaned untuk selanjutnya data cleaned yang berupa teks dikonversi menjadi representasi vektor TF-IDF. Dan Menggunakan objek vectorizer untuk mengubah cleaned ke dalam bentuk matriks TF-IDF yang sesuai.

```
labels = data['score']
cleaned = data['clean']

# Convert the tokens into vectors
vectorizer = TfidfVectorizer()
features = vectorizer.fit_transform(cleaned)
```

Gambar 3.33 Proses konversi data teks ke vector.

Data kemudian dibagi menjadi data latih dan data training dengan pembagian 80:20, 80% adalah data latih dan 20% nya adalah data *training*. Proses pembagian data menggunakan modul train_test_split.

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2)
```

Gambar 3.34 Proses train test split.

Setelah data *test* dan *train* didapatkan selanjutnya dalah melakukan proses modeling dan evaluasi. Dalam modeling Naïve bayes menggunakan kode sebagai berikut.

```
start_time_nb = time.time()
nb = MultinomialNB()
nb.fit(X_train, y_train)
y_pred_nb = nb.predict(X_test)
end_time_nb = time.time()
```

Gambar 3.35 Proses modeling Multinominal Naïve Bayes.

Modul time diletakan diawal dan di akhir proses untuk menghitung lama proses dari modeling yang dilakukan. Selanjutnya dilakukan evaluasi dengan menghitung akurasi (accuracy_score), presisi (precision_score), dan recall (recall_score).

```
accuracy_nb = accuracy_score(y_test, y_pred_nb)
precision_nb = precision_score(y_test, y_pred_nb, average='macro')
recall_nb = recall_score(y_test, y_pred_nb, average='macro')

print("Naive Bayes")
print("Accuracy:", accuracy_nb)
print("Precision:", precision_nb)
print("Recall:", recall_nb)
print("Time:", end_time_nb - start_time_nb)
```

Gambar 3.36 Proses evaluasi Multinominal Naïve Bayes.

Sedangkan untuk modeling Logistic Regression menggunakan kode sebagai berikut.

```
start_time_lr = time.time()
lr = LogisticRegression()
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)
end_time_lr = time.time()
```

Gambar 3.37 Proses modeling Logistic Regression.

Selanjutnya dilakukan evaluasi dengan menghitung akurasi (accuracy_score), presisi (precision_score), dan recall (recall_score).

```
accuracy_lr = accuracy_score(y_test, y_pred_lr)
precision_lr = precision_score(y_test, y_pred_lr, average='macro')
recall_lr = recall_score(y_test, y_pred_lr, average='macro')

print("\nLogistic Regression")
print("Accuracy:", accuracy_lr)
print("Precision:", precision_lr)
print("Recall:", recall_lr)
print("Time:", end_time_lr - start_time_lr)
```

Gambar 3.38 Proses evaluasi Logistic Regression.

Seluruh proses baik pada Naïve Bayes maupun Logistic Regression diterapkan pada seluruh dataset dan hasilnya dicatat untuk dilakukan Analisa dan pembahasan.

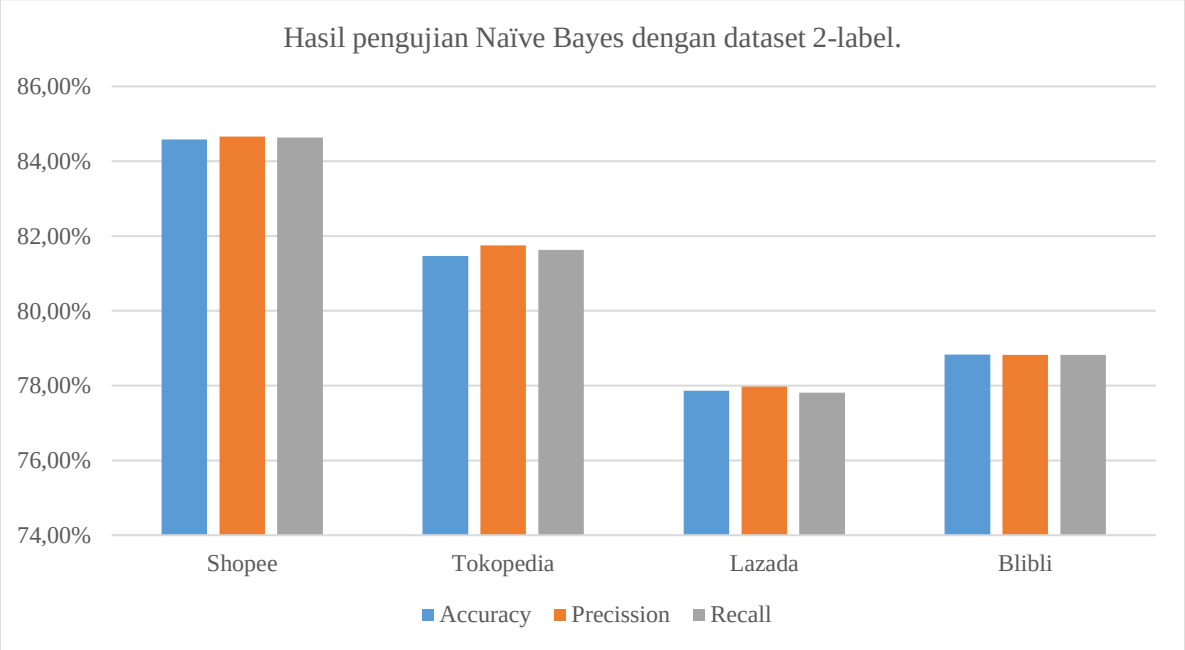
BAB 4
Hasil dan Pembahasan

4.1 Naïve Bayes dengan 2-label dataset

Hasil pengujian model Naïve Bayes menggunakan dataset marketplace 2-label diperoleh akurasi, presisi, recall, dan waktu pemrosesan sebagai berikut:

Tabel 4.1 Hasil pengujian Naïve Bayes dengan dataset 2-label.

Dataset	Accuracy	Precission	Recall	Time
Shopee	**84.33%**	**84.32%**	**84.34%**	0.067s
Tokopedia	80.88%	81.13%	81.03%	0.067s
Lazada	77.44%	78.03%	77.34%	**0.038s**
Blibli	79.11%	79.35%	79.02%	0.042s



Gambar 4.1 Hasil pengujian Naïve Bayes dengan dataset 2-label.

Pada tabel 4.1, diperoleh nilai akurasi, presisi, dan recall tertinggi untuk dataset Shopee dengan akurasi 84,33%, presisi 84,32%, dan recall 84,34%. Meskipun tidak signifikan, skor tertinggi 84,34% diperoleh dalam ingatan, menunjukkan bahwa model menunjukkan kemampuan tertinggi untuk mengidentifikasi semua objek di kelas target.

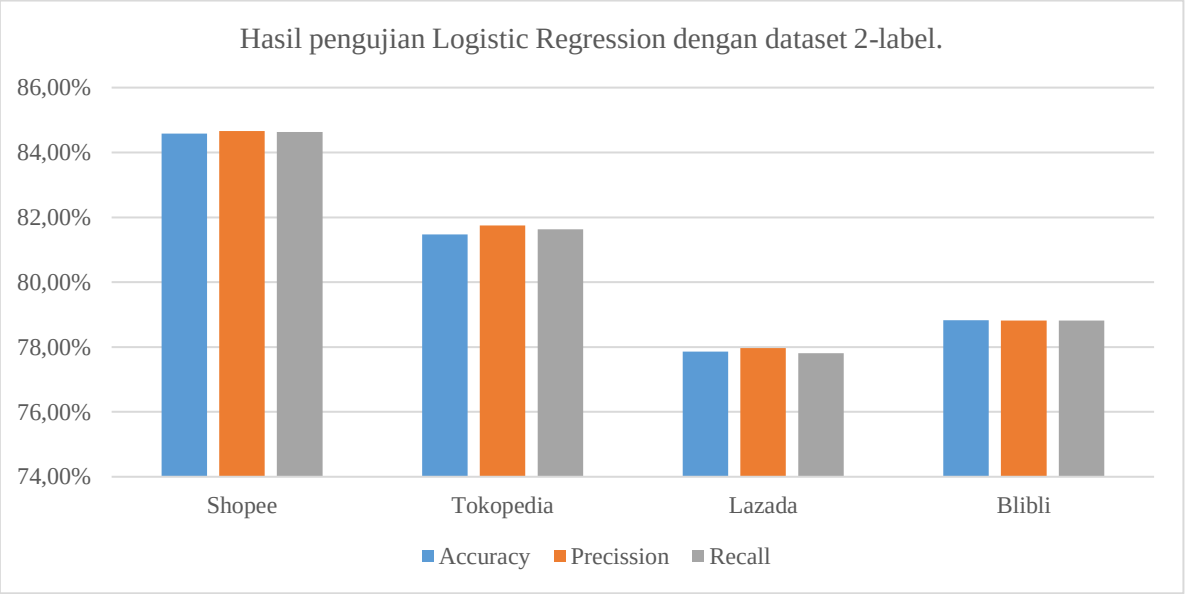
Sementara itu, waktu pemrosesan tercepat ditemukan dalam pengujian dataset Lazada, yang memakan waktu 0,038 detik.

4.2 Logistic Regression dengan 2-label dataset

Hasil pengujian model Logistic Regression menggunakan dataset marketplace 2-label diperoleh akurasi, presisi, recall, dan waktu pemrosesan sebagai berikut:

Tabel 4.2 Hasil pengujian Logistic Regression dengan dataset 2-label.

Dataset	Akurasi	Precission	Recall	Time
Shopee	**84.58%**	**84.66%**	**84.63%**	1.33s
Tokopedia	81.47%	81.75%	81.63%	1.27s
Lazada	77.86%	77.97%	77.81%	**0.47s**
Blibli	78.83%	78.82%	78.82%	0.65s



Gambar 4.2 Hasil pengujian Logistic Regression dengan dataset 2-label.

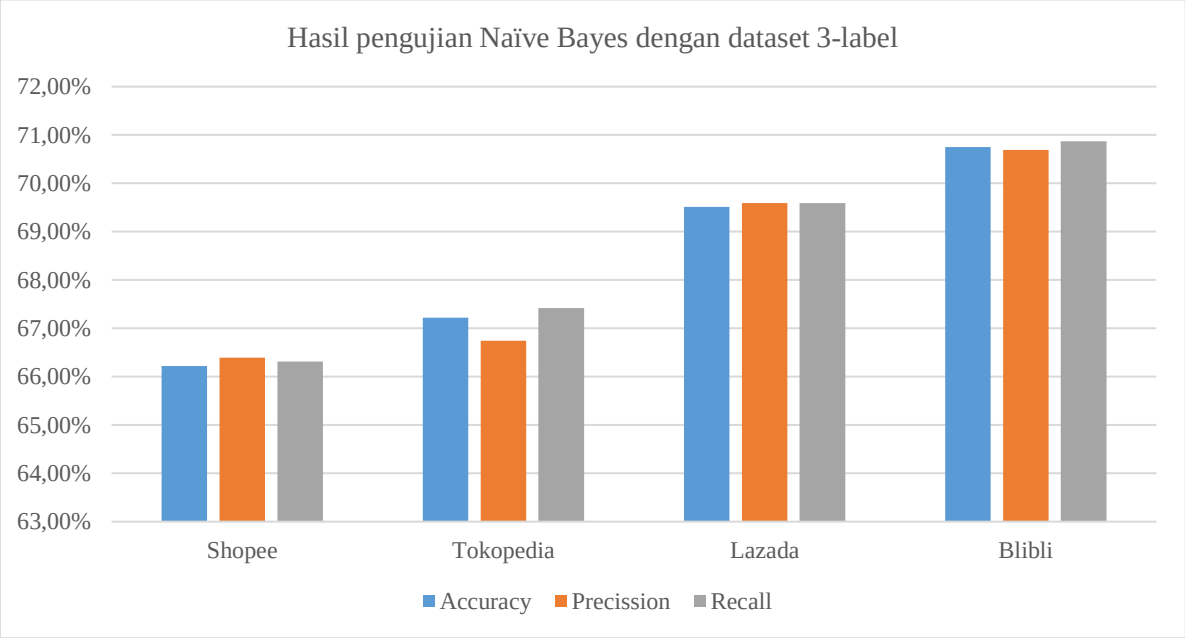
Pada tabel 4.2, masih diperoleh nilai akurasi, presisi, dan recall tertinggi untuk dataset Shopee, dengan nilai yang sedikit lebih tinggi: akurasi 84,58%, presisi 84,66%, dan recall 84,63%. Skor presisi tertinggi yang berarti model memiliki kemampuan tertinggi untuk memprediksi kelas target dengan benar. Meskipun waktu pemrosesan pada pengujian ini tidak lebih cepat dari pengujian sebelumnya, namun waktu pemrosesan tertinggi masih terdapat pada dataset Lazada, dengan nilai 0,47 detik.

4.3 Naïve Bayes dengan 3-label dataset

Hasil pengujian model Naïve Bayes menggunakan dataset marketplace 3-label diperoleh akurasi, presisi, recall, dan waktu pemrosesan sebagai berikut:

Tabel 4.3 Hasil pengujian Naïve Bayes dengan dataset 3-label.

Dataset	Akurasi	Precission	Recall	Time
Shopee	66.22%	66.39%	66.31%	**0.055s**
Tokopedia	67.22%	66.74%	67.42%	0.057s
Lazada	69.51%	69.59%	69.59%	0.058s
Blibli	**70.75%**	**70.69%**	**70.87%**	0.065s



Gambar 4.3 Hasil pengujian Naïve Bayes dengan dataset 3-label.

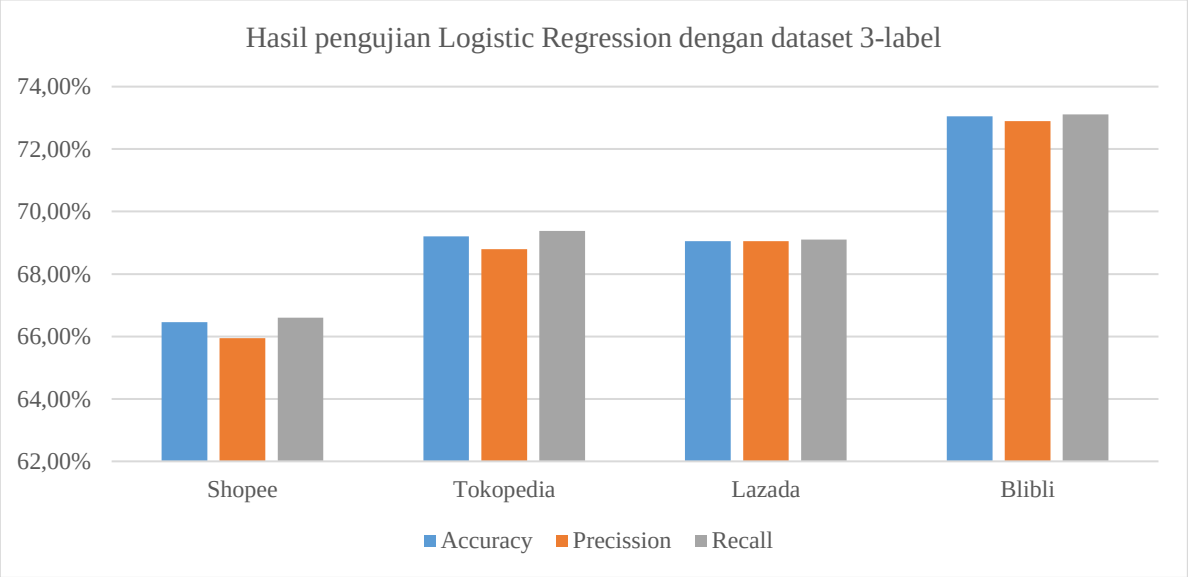
Pengujian himpunan data 3-label pada tabel 4.3 memperoleh nilai akurasi, presisi, dan recall yang lebih rendah dibandingkan dengan tabel 4 dan 5. Nilai tertinggi ditemukan pada dataset Blibli, dengan akurasi tertinggi yang dapat dicapai sebesar 70,75%, presisi 70,69%, dan recall 70,87%. Dalam dataset ini, kemampuan tertinggi model adalah dalam recall, yang berarti dapat menemukan objek di kelas target dengan skor 70,87%. Waktu pemrosesan tercepat ditemukan pada dataset Shopee, dengan nilai 0,055 detik, lebih cepat dari keseluruhan waktu pada tabel 4.2.

4.4 Logistic Regression dengan 3-label dataset

Hasil pengujian model Logistic Regression menggunakan dataset marketplace 3-label diperoleh akurasi, presisi, recall, dan waktu pemrosesan sebagai berikut:

Tabel 4.4 Hasil pengujian Logistic Regression dengan dataset 3-label.

Dataset	Akurasi	Precission	Recall	Time
Shopee	66.46%	65.95%	66.60%	3.60s
Tokopedia	69.20%	68.79%	69.38%	2.58s
Lazada	69.05%	69.05%	69.10%	2.35s
Blibli	73.05%	72.89%	73.11%	2.61s

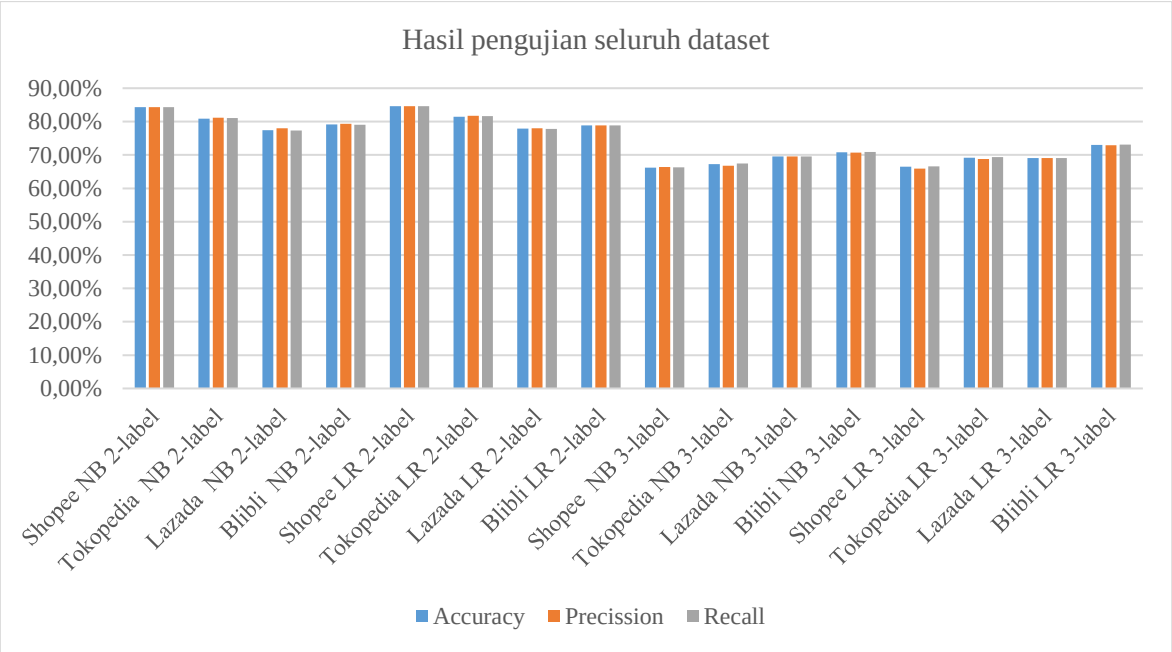


Gambar 4.4 Hasil pengujian Logistic Regression dengan dataset 3-label.

Pada tabel 7, nilai akurasi, presisi, dan recall lebih tinggi dibandingkan dengan tabel 6, dengan akurasi 73,05%, presisi 72,89%, dan recall 73,11%. Nilai-nilai ini diperoleh dalam pengujian dataset Blibli. Waktu pemrosesan tercepat ditemukan dalam dataset Lazada, dengan nilai 2,35 detik.

4.5 Pembahasan

Dari semua hasil didapatkan nilai akurasi, presisi, dan recall tertinggi dicapai saat pengujian dataset Shopee menggunakan *logistic regression* dengan 2 label. Nilai yang diperoleh adalah akurasi 84,58%, presisi 84,66%, dan recall 84,63%. Untuk hasil dari keseluruhan data dapat dilihat pada gambar 4.5.



Gambar 4.5 Hasil pengujian seluruh dataset.

Pada pengujian waktu proses ditemukan bahwa algoritma Naive Bayes mempunyai kemampuan proses lebih cepat dibandingkan Logistic Regression. Seluruh hasil pengujian algoritma Naive Bayes berada di bawah 0,067 detik, dengan waktu tercepat tercatat sebesar 0,038 detik, sedangkan Logistic Regression hanya meraih waktu proses tercepat sebesar 0,47 detik, dan waktu proses terlama sebesar 3,6 detik.

Tabel 4.5 Hasil lama waktu proses.

Dataset	**Time**
Shopee NB 2-label	0,06
Tokopedia NB 2-label	0,067
Lazada NB 2-label	**0,038**
Blibli NB 2-label	0,042
Shopee LR 2-label	1,33
Tokopedia LR 2-label	1,27
Lazada LR 2-label	0,47
Blibli LR 2-label	0,65
Shopee NB 3-label	0,055
Tokopedia NB 3-label	0,057
Lazada NB 3-label	0,058
Blibli NB 3-label	0,065
Shopee LR 3-label	3,6
Tokopedia LR 3-label	2,58
Lazada LR 3-label	2,35
Blibli LR 3-label	2,61

Berdasarkan pengujian pada kumpulan data pasar, Naïve Bayes dan Logistic Regression menunjukkan tingkat akurasi, presisi, dan perolehan kembali yang sebanding, berkisar antara 66% hingga 84%. Meskipun kedua algoritme menunjukkan kinerja yang serupa, Regresi Logistik memiliki sedikit keunggulan. Regresi logistik memperoleh hasil tertinggi pada akurasi dengan nilai 84,58%, presisi 84,66%, dan recall 84,63%.

Keunggulan Logistic Regression terkait Accuracy, Precission dan Recall dalam penelitian ini salah satunya diarenakan Logistic Regression memiliki metode Ridge Regression, Hal ini mengakibatkan koefisien menjadi lebih stabil dan kurang sensitif terhadap fluktuasi dalam data training. Dengan demikian, Ridge Regression membantu mencegah overfitting. Selain itu Logistic Regression juga memiliki metode Least Absolute Shrinkage and Selection Operator (LASSO) yang mengakibatkan beberapa koefisien menjadi persis nol, sehingga LASSO juga dapat digunakan untuk seleksi fitur otomatis. Dengan mengurangi beberapa koefisien menjadi nol, LASSO mampu menghasilkan model yang lebih sederhana dengan fitur-fitur yang lebih penting.

Keunggulan pada Logistic Regression ini juga sejalan dengan penelitian yang dilakukan oleh (Shah et al., 2020) dan (Itoo et al., 2021) tentang perbandingan Logistic Regression dengan berberapa algoritma lainnya yang menunjukan bahwa Logistic Regression dapat menghasilkan nilai accuracy dan precission yang tinggi.

Namun demikian, penting untuk ditekankan bahwa Naïve Bayes menunjukkan kinerja yang unggul dalam hal kecepatan pemrosesan, mencapai waktu terpendek selama evaluasi dataset Lazada dengan dua label, yaitu hanya 0,038 detik. Semua hasil pada pengujian waktu pemrosesan, Naïve Bayes memperoleh nilai dibawah 1 detik, dan hasil yang diperoleh lebih cepat dibandingkan waktu pemrosesan regresi logistik.

Algoritma Naïve Bayes didasarkan pada Teorema Bayes dan asumsi naif (naïve) bahwa semua fitur adalah independen satu sama lain. Karena asumsi ini, perhitungan probabilitas yang terlibat dalam Naïve Bayes menjadi lebih sederhana dan terpisah untuk setiap fitur, menghasilkan perhitungan yang lebih cepat.

Sama halnya dengan penelitian yang dilakukan (Jefriyanto et al., 2023) yang mengujikan performa Naïve Bayes pada klasifikasi teks Twitter yang dipadukan dengan implementasi preprocessing data menghasilkan waktu proses yang lebih cepat daripada tanpa preprocessing data, dalam penelitian tersebut waktu tercepat yang didapatkan adalah 0,141 detik. Dipeneliitian ini waktu yang didapatkan lebih cepat disebabkan karena adanya implementasi stopwords dari 2 library, yaitu NLTK dan sastrawi, dan juga

perbedaan karakteristik data yang dipakai. Data pada penelitian ini menggunakan data review dengan jumlah karakter lebih sedikit daripada data teks Twitter.

Di sisi lain, Logistic Regression melibatkan perhitungan yang lebih kompleks, termasuk perhitungan gradien untuk menemukan parameter yang optimal selama proses pelatihan. Ini melibatkan operasi matriks dan iterasi yang lebih rumit, yang dapat menghasilkan waktu pemrosesan yang lebih lama, namun itu juga yang menyebabkan Logistic Regression mampu memperoleh nilai accuracy, precission dan recall yang lebih tinggi.

Sedangkan untuk dataset berlabel, temuan penelitian menunjukkan bahwa dataset dengan 2 label cenderung menghasilkan accuracy, precision dan recall yang lebih tinggi dibandingkan dataset dengan 3 label. Dataset 2 label memiliki label yang lebih sederhana dan konsisten, sedangkan dataset dengan 3 label memungkinkan adanya ambiguitas atau kesalahan label terutama pada label netral dimana label netral berasal dari rating 3. Karena label netral berada ditengah rating memungkinkan data dalam label netral ada yang teridentifikasi masuk dalam label positif atau negatif.

Hasil ini memberikan wawasan berharga untuk mengoptimalkan analisis sentimen dalam aplikasi pasar dan menunjukkan bahwa memilih algoritma yang tepat dapat berdampak signifikan terhadap kinerja dan efisiensi pemrosesan.

Melihat hasil yang diperoleh, kombinasi pelabelan berbasis rating dengan algoritma Regresi Logistik dan Naive Bayes dapat mencapai tingkat akurasi, presisi, dan perolehan yang relatif tinggi, serupa dengan penelitian sebelumnya. Dalam penelitian masa depan, algoritme atau pengoptimalan lain dalam pelabelan berbasis peringkat juga dapat dieksplorasi.

BAB 5
Kesimpulan dan Saran

5.1 Kesimpulan

Pada penelitian ini Logisitc Regression mendapatkan nilai accuracy, precission dan recall lebih tinggi daripada Naive Bayes. Metode ridge regression dan LASO yang ada pada Logistic Regression menjadi berberapa hal yang mempengaruhi algoritma Logistic Regression mendapatkan nilai lebih tinggi daripada Naive Bayes. Namun metode perhitungan Naive Bayes yang lebih sederhana membuat algoritma ini dapat memproses data lebih cepat. Perbedaan tersebut menunjukan bahwa setiap algoritma mempunyai kelebihannya masing-masing, dan dapat digunakan seperti yang kita butuhkan.

Selain itu, menerapkan dataset dengan 2 label memberikan kinerja yang lebih baik dibandingkan dengan dataset dengan 3 label, membuat kategorisasi sentimen yang lebih efektif dalam *review* marketplace. Yang nantinya hasil dari penerapan algoritma ini dapat memberikan pandangan yang berguna bagi pelaku industri marketplace maupun bagi para peneliti dibidang ini.

5.2 Saran

Saran untuk penelitian selanjutnya adalah:

1. Dalam penelitian ini dibandingkan dua algoritma Naive Bayes dan Logistic Regression, untuk penelitian selanjutnya dapat ditambahkan algoritma pembanding lainnya yang juga dapat diterapkan dalam sentiment analisis.
2. Pada penelitian selanjutnya dapat dilakukan uji coba kepada data yang memiliki karakteristik rating yang mirip dengan data marketplace ini namun pada domain lain yang berbeda.
3. Pengimplementasiaan metode optimasi dapat dilakukan untuk meningkatkan hasil accuracy, precission dan recall.

Daftar Pustaka

Ahmadi, M. I., Apriani, F., Kurniasari, M., Handayani, S., & Gustian, D. (2020). *SENTIMENT ANALYSIS ONLINE SHOP ON THE PLAY STORE USING METHOD SUPPORT VECTOR MACHINE (SVM)*.

Aldabbas, H., Bajahzar, A., Alruily, M., Qureshi, A. A., Amir Latif, R. M., & Farhan, M. (2020). Google Play Content Scraping and Knowledge Engineering using Natural Language Processing Techniques with the Analysis of User Reviews. *Journal of Intelligent Systems*, *30*(1), 192–208. https://doi.org/10.1515/jisys-2019-0197

Azhar, M., Hafidz, N., Rudianto, B., & Gata, W. (2020). Marketplace Sentiment Analysis Using Naive Bayes And Support Vector Machine. *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, *8*(2), 91–100. https://doi.org/10.33558/piksel.v8i2.2272

Itoo, F., Meenakshi, & Singh, S. (2021). Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. *International Journal of Information Technology*, *13*(4), 1503–1511. https://doi.org/10.1007/s41870-020-00430-y

Jefriyanto, J., Ainun, N., & Ardha, M. A. A. (2023). Application of Naïve Bayes Classification to Analyze Performance Using Stopwords. *Journal of Information System, Technology and Engineering*, *1*(2), Article 2.

JoMingyu. (n.d.). *google-play-scraper: Google-Play-Scraper provides APIs to easily crawl the Google Play Store for Python without any external dependencies!* (1.2.4) [Python; MacOS, Microsoft]. Retrieved August 6, 2023, from https://github.com/JoMingyu/google-play-scraper

Khemani, B., & Adgaonkar, A. (2021). A Review on Reddit News Headlines with NLTK tool. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3834240

KOMINFO, P. (n.d.-a). *Kemkominfo: Pertumbuhan e-Commerce Indonesia Capai 78 Persen*. Website Resmi Kementerian Komunikasi Dan Informatika RI. Retrieved April 16, 2023, from https://play.google.com/store/apps

KOMINFO, P. (n.d.-b). *Kominfo: Pengguna Internet di Indonesia 63 Juta Orang*. Website Resmi Kementerian Komunikasi Dan Informatika RI. Retrieved April 16, 2023, from http:///index.php/content/detail/3415/Kominfo+%3A+Pengguna+Internet+di+Indonesia+63+Juta+orang/0/berita_satker

Kurniawan, I., Hananto, A. L., Hilabi, S. S., Hananto, A., & Rahman, A. Y. (2023). *Perbandingan Algoritma Naive Bayes Dan SVM Dalam Sentimen Analisis Marketplace Pada Twitter*. *10*(1).

NLTK :: Natural Language Toolkit. (n.d.). Retrieved July 15, 2023, from https://www.nltk.org/

PlanB. (2023). *Google-Play-Scraper* [Python]. https://github.com/JoMingyu/google-play-scraper (Original work published 2019)

Pratmanto, D., Rousyati, R., Wati, F. F., Widodo, A. E., Suleman, S., & Wijianto, R. (2020). App Review Sentiment Analysis Shopee Application In Google Play Store Using Naive Bayes Algorithm. *Journal of Physics: Conference Series*, *1641*(1), 012043. https://doi.org/10.1088/1742-6596/1641/1/012043

Raksaka Indra Alhaqq, I Made Kurniawan Putra, & Yova Ruldeviyani. (2022). Analisis Sentimen terhadap Penggunaan Aplikasi MySAPK BKN di Google Play Store. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, *11*(2), 105–113. https://doi.org/10.22146/jnteti.v11i2.3528

Rohman, A. N., Luviana Musyarofah, R., Utami, E., & Raharjo, S. (2020). Natural Language Processing on Marketplace Product Review Sentiment Analysis. *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, 1–5. https://doi.org/10.1109/ICORIS50180.2020.9320827

Rosid, M. A., Fitrani, A. S., Astutik, I. R. I., Mulloh, N. I., & Gozali, H. A. (2020). Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi. *IOP Conference Series: Materials Science and Engineering*, *874*(1), 012017. https://doi.org/10.1088/1757-899X/874/1/012017

Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. *Augmented Human Research*, *5*(1), 12. https://doi.org/10.1007/s41133-020-00032-0

Vujovic, Ž. Đ. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, *12*(6). https://doi.org/10.14569/IJACSA.2021.0120670

Wahono, R. S. (n.d.). *Systematic Literature Review: Pengantar, Tahapan dan Studi Kasus | RomiSatriaWahono.Net*. Retrieved June 6, 2023, from https://romisatriawahono.net/2016/05/15/systematic-literature-review-pengantar-tahapan-dan-studi-kasus/