PENERAPAN ALGORITMA *K-MEANS CLUSTERING* PADA ULASAN PENGGUNA MERDEKA MENGAJAR DI *PLAY STORE*

SKRIPSI

Oleh :

Nur Syamsu Wais Al Qorni

22.43.905

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA dan KOMPUTER
WIDYA CIPTA DHARMA

2024

PENERAPAN ALGORITMA *K-MEANS CLUSTERING* PADA ULASAN PENGGUNA MERDEKA MENGAJAR DI *PLAY STORE*

SKRIPSI

Skripsi Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Komputer

Oleh :

Nur Syamsu Wais Al Qorni

22.43.905

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA dan KOMPUTER
WIDYA CIPTA DHARMA

2024

LEMBAR PENGESAHAN

Skripsi Oleh : **Nur Syamsu Wais Al Qorni (22.43.905)**

Telah dipertahankan di depan dewan penguji

Pada tanggal : 29 JUL 2024

Dewan Penguji

Wahyuni, S.Kom., M.Kom.

Pitrasacha Adytia, S.T., M.T.

Siti Lailiyah, S.Kom., M.Kom.

Rizky Zakariyya Rasyad, S.E., M.M.

Mengetahui,
Ketua Program Studi Teknik Informatika

Wahyuni, S.Kom., M.Kom.

Mengesahkan,
Ketua STMIK Widya Cipta Dharma

H. Tommy Bustomi, S.Kom., M.Kom.

ii

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Nur Syamsu Wais Al Qorni

Nim : 22.43.905

Judul : Penerapan Algoritma *K-Means Clustering* Pada Ulasan Pengguna Merdeka Mengajar di *Play Store*

Menyatakan dan bertanggung jawab dengan sebenarnya bahwa Skripsi ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya. Jika waktu selanjutnya ada pihak lain yang mengklaim bahwa Skripsi ini sebagai karyanya, yang disertai dengan bukti-bukti yang cukup, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Samarinda, Senin 29 Juli 2024

Yang membuat pernyataan

Nur Syamsu Wais Al Qorni

iii

ABSTRAK

Nur Syamsu Wais Al Qorni, 2024, Penerapan Algoritma *K-Means Clustering* Pada Ulasan Pengguna Merdeka Mengajar di *Play Store*. Skripsi Jurusan Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer Widya Cipta Dharma, Pembimbing (I) Wahyuni, S.Kom., M.Kom., Pembimbing (II) Pitrasacha Adytia, S.T., M.T.

Kata Kunci : Algoritma *K-Means Clustering*, Analisis Klaster, Aplikasi Merdeka Mengajar, *CRISP-DM*, *Google Play Store*.

Dinas Pendidikan dan Kebudayaan atau Kemendikbudristek memberikan kemudahan bagi para tenaga pengajarnya melalui aplikasi Merdeka Mengajar. Aplikasi ini membantu para guru dalam mengajar secara efektif serta meningkatkan kompetensi mereka. Namun, banyaknya ulasan pengguna di *Google Play Store* mengenai aplikasi ini menunjukkan adanya kebutuhan untuk memahami fitur-fitur yang disukai dan dikeluhkan. Penelitian ini bertujuan untuk menerapkan Algoritma *K-Means Clustering* dalam menganalisis ulasan pengguna aplikasi Merdeka Mengajar.

Metode *CRISP-DM* digunakan dalam proses ini, yang mencakup tujuh tahapan: *Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, Deployment*, dan *Evaluation System*. Data ulasan dikumpulkan menggunakan teknik *scrapping* dan diproses menggunakan berbagai *library* seperti *NLP-ID*, *Sastrawi*, dan *NLTK*. Analisis klasterisasi menunjukkan bahwa model *Bag of Words (BoW)* dan *TF-IDF* optimal pada 3 klaster, sedangkan model *Word2Vec* menunjukkan hasil terbaik pada 7 klaster. Namun, penelitian ini menggunakan 3 klaster dengan *Word2Vec* untuk konsistensi kinerja.

Hasil klasterisasi mengidentifikasi tiga tema utama: "Aplikasi sangat membantu untuk Guru," "Memudahkan membuat Administrasi bagi Guru," dan "Pelatihan secara *online* maupun *offline*." Penelitian ini memberikan rekomendasi untuk pengembangan lebih lanjut, termasuk pengambilan data dari berbagai media sosial, pembangunan sistem berbasis *Android*, dan analisis sentimen yang lebih mendalam.

ABSTRACT

Nur Syamsu Wais Al Qorni, 2024, Application of the K-Means Clustering Algorithm in Merdeka Mengajar User Reviews on the Play Store. Thesis Department of Informatics Engineering, Widya Cipta Dharma College of Information and Computer Management, Supervisor (I) Wahyuni, S.Kom., M.Kom., Supervisor (II) Pitrasacha Adytia, S.T., M.T.

Keywords: K-Means Clustering Algorithm, Cluster Analysis, Merdeka Mengajar Application, CRISP-DM, Google Play Store.

The Department of Education and Culture or Kemendikbudristek provides convenience for its teaching staff through the Merdeka Mengajar application. This application helps teachers teach effectively and improve their competence. However, the many user reviews on the Google Play Store regarding this application indicate a need to understand the features that are liked and complained about. This study aims to apply the K-Means Clustering Algorithm in analyzing user reviews of the Merdeka Mengajar application.

The CRISP-DM method is used in this process, which includes seven stages: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment, and Evaluation System. Review data is collected using scraping techniques and processed using various libraries such as NLP-ID, Sastrawi, and NLTK. Clustering analysis shows that the Bag of Words (BoW) and TF-IDF models are optimal at 3 clusters, while the Word2Vec model shows the best results at 7 clusters. However, this study uses 3 clusters with Word2Vec for performance consistency.

The clustering results identified three main themes: "The application is very helpful for teachers," "Makes it easier to create administration for teachers," and "Training both online and offline." This study provides recommendations for further development, including data collection from various social media, building an Android-based system, and deeper sentiment analysis.

RIWAYAT HIDUP

Nur Syamsu Wais Al Qorni, lahir tanggal 12 September 2000 di Samarinda Kalimatan Timur. Merupakan anak pertama dari dua bersaudara pasangan Bapak Sunyoto S.Pd, MPd. dan Isnaniah SE. Memulai pendidikan taman kanak-kanak di TK Salsabila pada tahun 2004 di Samarinda dan lulus pada tahun 2006. Kemudian melanjutkan pendidikan sekolah dasar di Sekolah Dasar 042 pada tahun 2006 di Samarinda dan lulus pada tahun 2012. Kemudian melanjutkan pendidikan di Sekolah SMPN 22 Samarinda dan lulus pada tahun 2015. Pada tahun yang sama melanjutkan pendidikan di SMAN 3 Samarinda dan lulus pada tahun 2018. Kemudian melanjutkan pendidikan ke Politeknik Negeri di Samarinda pada tahun 2018 dan lulus pada tahun 2021. Kemudian melanjutkan kembali pendidikan ke Sekolah Tinggi Manajemen Informatika dan Komputer Widya Cipta Dharma di Samarinda pada tahun 2022.

Pada tanggal 01 Februari 2024 sampai dengan 31 Juli 2024. Melakukan Penelitian di Dinas Pendidikan dan Kebudayaan Kota Samarinda.

Dinas Pendidikan kota mempunyai tugas dalam melaksanakan pendidikan sesuai dengan tanggung jawab yang sudah di berikan oleh Pemerintah. Dinas Pendidikan dan Kebudayaan atau Kemendikbudristek juga memberikan kemudahan terhadap para tenaga pengajarnya yaitu berupa sebuah aplikasi bernama Merdeka Mengajar. aplikasi ini diharapkan bisa mendukung penerapan dari Kurikulum Merdeka dengan menyediakan referensi, inspirasi, dan bahan ajar yang sesuai.

KATA PENGANTAR

Puji dan Syukur kepada Tuhan Yang Maha Esa yang telah memberikan rahmat serta kekuatan kepada penulis sehingga Skripsi yang berjudul " Penerapan Algoritma *K-Means Clustering* Pada Ulasan Pengguna Merdeka Mengajar di *Play Store*" dapat peneliti selesaikan sesuai dengan rencana yang diharapkan. Adapun maksud dan tujuan penyusunan Skripsi ini adalah sebagai salah satu bentuk persyaratan untuk memperoleh gelar Sarjana (S.Kom.) pada Jurusan S1 Teknik Informatika STMIK Widya Cipta Dharma.

Ucapan terima kasih kepada semua pihak yang telah memberikan dukungan dan bantuan atas terselesaikannya Skripsi ini. Ucapan terima kasih khusus disampaikan kepada :

1. Allah SWT atas Rahmat, Rezeki dan Pertolongan-Nya yang diberikan sehingga peneliti dapat menyelesaikan Tugas Akhir ini dengan baik dan lancar.
2. Orang Tua dan Keluarga yang selalu memberi dukungan Doa, Moral dan Materi.
3. Bapak H. Tommy Bustomi, S.Kom., M.Kom. selaku Ketua Sekolah Tinggi Manajemen Ilmu Komputer Widya Cipta Dharma Samarinda.
4. Ibu Wahyuni, S.Kom., M.Kom, selaku Ketua Program Studi Teknik Informatika dan Pembimbing Utama yang telah menyetujui permohonan penyusunan Skripsi ini dan bersedia membimbing serta meluangkan waktu

untuk memberikan arahan serta masukan dalam penelitian dan penyusunan skripsi ini hingga selesai..

5. Bapak Pitrasacha Aditya, S.T., M.T., selaku pembimbing Pendamping yang telah bersedia membimbing serta meluangkan waktu untuk memberikan arahan serta masukan dalam penelitian dan penyusunan skripsi ini hingga selesai.
6. Ibu Siti Lailiyah, S.Kom., M.Kom. Selaku Ketua Penguji yang telah banyak memberikan masukkan serta materi dalam penelitian dan penyusunan skripsi ini hingga selesai.
7. Bapak Rizky Zakariyya Rasyad, S.E., M.M. Selaku Anggota Penguji yang telah banyak memberikan masukkan serta materi dalam penelitian dan penyusunan skripsi ini hingga selesai.
8. Dosen, dan Staff Administrasi STMIK Widya Cipta Dharma yang telah membantu dalam segala hal yang berkaitan dengan perkuliahan.
9. Serta semua sahabat dan rekan-rekan mahasiswa Program Studi Teknik Informatika yang telah banyak membantu dan memberikan semangat serta dukungan.

Samarinda, Senin 29 Juli 2024

Nur Syamsu Wais Al Qorni

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1. Latar Belakang Masalah

Dinas Pendidikan dan Kebudayaan adalah instansi pemerintahan kota atau kabupaten yang bertanggung jawab terhadap Pendidikan siswa-siswi yang berada di kota atau kabupaten. Dinas Pendidikan dan Kebudayaan Kota mempunyai tugas dalam melaksanakan pendidikan sesuai dengan tanggung jawab yang sudah di berikan oleh Pemerintah. Salah satu tugas dan tanggung jawab dinas pendidikan adalah mengelola perkembangan pendidikan untuk anak-anak dengan mengetahui perkembangan anak-anak yang dilaporkan langsung oleh guru maka dapat mengetahui tumbuh kembang kemampuan siswa yang dimiliki. (Susanto, 2022). Dinas Pendidikan dan Kebudayaan atau Kemendikbudristek juga memberikan kemudahan terhadap para tenaga pengajar nya yaitu berupa sebuah aplikasi bernama Merdeka Mengajar, dengan aplikasi tersebut, dapat membantu para tenaga pengajar dalam mengajar dengan mudah dan efektif, serta meningkatkan kompetensi mereka melalui pelatihan mandiri dan sumber belajar yang tersedia di aplikasi Merdeka Mengajar. Selain itu, aplikasi ini diharapkan bisa mendukung penerapan dari Kurikulum Merdeka dengan menyediakan referensi, inspirasi, dan bahan ajar yang sesuai. Aplikasi Merdeka Mengajar ini juga dapat ditemukan di *Google Play Store*, yaitu tempat untuk mengunduh aplikasi berbasis *android* dan *App Store* untuk mengunduh aplikasi berbasis *iOS*.

Banyak dari pengguna aplikasi Merdeka Mengajar memberikan rating berupa bintang 5 di *platform Google Play Store*, beberapa pengguna juga meninggalkan kesan menggunakan aplikasi merdeka mengajar dengan memberikan komentar di halaman komentar *Google Play Store*. Ulasan yang ditinggalkan berupa kemudahan dalam menggunakan aplikasi nya seperti akses ke berbagai fitur atau materi, memiliki banyak konten pembelajaran yang relevan serta selalu di *update*, dan komentar positif lainnya. Maka dari itu agar dapat mempertahankan kualitas dari aplikasi merdeka mengajar dan tetap menjadi andalan bagi para pengajar dalam menggunakan media belajar-mengajar serta mendapatkan pemahaman yang mendalam dari respon atau ulasan pengguna aplikasi Merdeka Mengajar di *Google Play Store*, dapat menggunakan teknik *Clustering*. Teknik *Clustering* dilakukan agar dapat mengetahui fitur teknis atau fungsionalitas di bagian manakah yang disukai oleh pengguna aplikasi merdeka mengajar. Teknik *Clustering* dapat mengindetifkasikan pola - pola yang muncul dalam ulasan pengguna dari aplikasi Merdeka Mengajar atau mengelompokan ulasan yang serupa. Agar dapat mengetahui kebutuhan pengguna dan dapat terfokuskan pada kelompok-kelompok tertentu yang memerlukan perhatian lebih lanjut. Dalam melakukan teknik *Clustering* terdapat beberapa model untuk melakukan teknik tersebut, salah satunya *K-Means* yang digunakan dalam penelitian ini. Pemilihan *K-Means Clustering* sebagai metode analisis dalam penelitian ini didasarkan pada pertimbangan yang relatif mudah di implementasikan, algoritma relatif sederhana dan efisien secara komputasional, sehingga cocok digunakan untuk analisis data yang relatif besar dan kompleksitas yang terukur. Dikarenakan banyaknya ulasan pada aplikasi Merdeka

Mengajar di *Google Play Store* tersebut tentunya akan sulit untuk dapat dibaca satu persatu. Dengan menggunakan *text mining* serta algoritma dari *K-Means Clustering* untuk melakukan analisis, data ulasan tersebut dapat di klasterisasi sehingga didapatkan gambaran besar atau menjadi pola-pola tertentu. Yang mana hasil dari klasterisasi yang didapatkan tersebut bermanfaat bagi pihak instansi Dinas Pendidikan dan Kebudayaan agar segera dapat menentukan tindakan perbaikan yang lebih tepat terkait aplikasi Merdeka Mengajar.

Dari latar belakang yang sudah peneliti bahas, maka peneliti melakukan penelitian terkait Menerapkan Algoritma *K-Means Clustering* dari model *Machine Learning* dengan mengenai Analisis Klaster berupa *Clustering* pada ulasan dari pengguna aplikasi Merdeka Mengajar dari Kemendikbudristek di *Google Play Store*.

1.2. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah "Penerapan Algoritma *K-Means Clustering* Pada Ulasan Pengguna Merdeka Mengajar di *Playstore* ?".

1.3. Batasan Masalah

Berdasarkan permasalahan yang sudah dijelaskan, berikut Batasan masalah pada penelitian ini :

1. Analisis Klasterisasi yang dilakukan berdasarkan dari permasalahan yang terdapat pada layanan aplikasi Merdeka Mengajar.
2. Data pada penelitian ini diambil dari ulasan Pengguna terkait aplikasi Merdeka Mengajar, serta hasil data yang diambil dengan format *.csv*.

3. *Tools* atau *library* yang digunakan pada penelitian ini dalam mengambil data atau proses *scrapping* adalah *google-play-scrapper* untuk mengambil data dari aplikasi *Google Play Store*.
4. *Tools* atau *library* yang digunakan pada penelitian ini untuk proses visualisasi data dan algoritma *K Means Clustering* adalah *matplotlib* dari *python*.
5. *Tools* atau *library* yang digunakan pada penelitian ini untuk proses *data frame* atau menyimpan dan mengorganisir data teks yang akan dianalisis adalah *Pandas*.
6. *Tools* atau *library* yang digunkan pada penelitian ini untuk awal pemrosesan sebelum dilakukan analisis data Menggunakan *NLTK*, Sastrawi dan *NLP-ID*.
7. *Tools* atau *library* yang digunakan pada penelitian ini untuk model *machine learning*, algoritma *K-means Clustering* dan proses pembobotan kata pada Teknik TF-IDF, dan *Bag of Word* adalah *Scikit-Learn* serta *Word2Vec* menggunakan *Tensorflow*.
8. Metode Penelitian yang digunakan adalah *CRISP-DM Metode Cross Industry Standard Process for Data*.
9. Pemodelan yang digunakan pada penelitian ini adalah *Machine Learning* dengan Algoritma *K Means Clustering*.
10. Menentukan Jumlah Klaster menggunakan *Silhouette Score* dan *Elbow Method*.
11. Evaluasi terhadap model dan algoritma yang digunakan adalah dengan menggunakan metode *Davies Bouldin Index* dan *Principal Component Analysis*.

12. Melakukan analisis klasterisasi di *Google Colaboratory.*
13. Implementasi atau *deployment* hasil analisis Klaster berupa *website* menggunakan *web service* dari *python* yaitu *Streamlit* dipadukan dengan *GitHub Codespaces*.
14. Bahasa pemrograman yang digunakan pada penelitian ini adalah *Python*.
15. Evaluasi terhadap *website* yang digunakan untuk menampilkan hasil analisis sentimen menggunakan metode *White Box Testing* dan *Black Box Testing*.

1.4. Tujuan Penelitian

Berdasarkan uraian dan latar belakang yang sudah dijelaskan, berikut tujuan dari penelitian ini :

1. Untuk membantu dan mempermudah Pihak instansi Dinas Pendidikan dan Kebudayaan dapat menentukan Tindakan untuk mempertahankan fitur fungsional dan teknis yang disukai oleh pengguna apliaksi serta melakukan evaluasi perbaikan terkait hal yang dikeluhkan pada aplikasi Merdeka Mengajar.
2. Mengetahui hasil analisis klaster pada *Google Play Store* terhadap ulasan pengguna aplikasi Merdeka Mengajar menggunakan Algoritma *K Means Clustering*.

1.5. Manfaat Penelitian

Berdasarkan latar belakang yang sudah dijelaskan, berikut manfaat dari penelitian ini :

1. Bagi peneliti, penelitian ini merupakan eksplorasi terhadap teori yang selama ini dipelajari, menambah wawasan peneliti, ilmu pengetahuan dan pengalaman

terhadap *machine learning*, algoritma *K Means Clustering* dan Analisis Klaster.

2. Bagi Perguruan Tinggi, penelitian ini diharapkan meningkatkan produktifitas dan mutu Pendidikan agar selalu relevan dengan kebutuhan lapangan kerja, serta menjadikan penelitian ini sebagai referensi untuk penelitian berikutnya.
3. Bagi pihak Dinas Pendidikan dan Kebudayaan, penelitian ini memberikan wawasan yang lebih mendalam mengenai kebutuhan dan fitur yang disukai oleh pengguna aplikasi Merdeka Mengajar. Dengan mengidentifikasi area-area yang perlu dipertahankan atau ditingkatan dalam aplikasi, Dinas Pendidikan dan Kebudayaan dapat meningkatkan kepuasan pengguna atau guru pengajar dan mengarahkan pengembangan aplikasi berdasarkan umpan balik pengguna yang konkret dan berbasis data. Hal ini tidak hanya dapat meningkatkan efektivitas layanan yang disediakan, tetapi juga memperkuat citra Dinas Pendidikan dan Kebudayaan selain sebagai Lembaga yang memberikan Pendidikan untuk anak-anak di Indonesia juga responsif terhadap kebutuhan dan masalah yang dihadapi tenaga pengajar secara keseluruhan.

1.6. Sistematika Penelitian

Sistematika penulisan skripsi ini terdiri dari lima bab yang saling berkaitan dengan yang lainnya. Adapun penjabaran sistematika penulisan adalah sebagai berikut :

BAB I PENDAHULUAN

Berisikan tentang latar belakang, rumusan masalah, Batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Berisikan tentang kajian empirik dan kajian teoritik yang menunjang dalam penyusunan Skripsi dalam Penerapan Algoritma *K-Means* pada Ulasan Pengguna di Aplikasi Merdeka Mengajar dari Kemendikbudristek di *Play Store*.

BAB III METODOLOGI PENELITIAN

Pada bab ini menguraikan dan menjabarkan pada metodologi penelitian yang digunakan dalam penelitian ini. Meliputi lokasi dan waktu penelitian, Teknik pengumpulan data, sumber data serta analisis data penelitian.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini peneliti membahas mengenai gambaran umum terkait Dinas Pendidikan dan Kebudayaan dan hasil penerapan metode berbasis *CRISP-DM* dan algoritma *K-Means Clustering* pada ulasan pengguna aplikasi Merdeka Mengajar dari *Google Play Store*.

BAB V PENUTUP

Pada bab ini dipaparkan mengenai kesimpulan serta saran-saran yang disampaikan dari hasil penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1. Kajian Empirik

Dalam kajian empirik ini menganalisis dan membandingkan judul, metode, dan algoritma dari studi sebelumnya yang dilakukan oleh peneliti lain dalam jurnal yang sebanding.Berikut ini adalah penelitian sebelumnya yang sebanding dengan studi ini:

Tabel 2.1 Kajian Empirik
Sumber: Dokumentasi Pribadi

No	Peneliti	Tahun	Judul	Algoritma
1	Gilbert Sanko Sunarko, Wasino, Tru Sutrisno	2023	Klasterisasi Sentimen Ulasan Pengguna Aplikasi Bca Mobile Pada Platform Google Play Store Dengan Algoritma *K-Means Clustering*	*K-Means Clustering*
2	Fitri Dwi Handayani, Isnaini Rosyida	2023	*Clustering Review* Pengguna Aplikasi Zenius pada Layanan *Google Play Store* Menggunakan Metode DBSCAN dan HDBSCAN	*DBSCAN* dan *HDBSCAN*
3	Ahmad Habib Husaini, Rini Mayasari, Susilawati	2022	Pengelompokan Ulasan Aplikasi Pedulilindungi Dengan Algoritma *K-Medoids*	*K-Medoids*

Sumber: Gilbert Sanko Sunarko, Wasino, Tru Sutrisno (2023). Klasterisasi Sentimen Ulasan Pengguna Aplikasi Bca *Mobile* Pada Platform *Google Play Store* Dengan Algoritma *K-Means Clustering*. Fitri Dwi Handayani, Isnaini Rosyida (2023). *Clustering Review* Pengguna Aplikasi *Zenius* pada Layanan *Google Play Store* Menggunakan Metode *DBSCAN* dan *HDBSCAN*. Ahmad Habib Husaini, Rini Mayasari, Susilawati (2022). Pengelompokan Ulasan Aplikasi Pedulilindungi Dengan Algoritma *K-Medoids*.

Penelitian berjudul "Klasterisasi Sentimen Ulasan Pengguna Aplikasi *BCA Mobile* Pada Platform *Google Play Store* Dengan Algoritma *K-Means Clustering*." Data diambil dari ulasan *BCA Mobile* selama 26-30 Desember 2022 menggunakan pustaka *google-play-scrapper* dengan Python. Pertumbuhan pengguna internet selama pandemi *COVID-19* meningkatkan penggunaan aplikasi perbankan seperti *BCA Mobile*. Dengan *k-means clustering*, 662 ulasan dianalisis menjadi sepuluh klaster menggunakan metode *Term Frequency-Inverse Document Frequency* (*tf-idf*). Skor siluet klasterisasi adalah 0,1027277, dengan rata-rata peringkat bintang 2,65. Hasilnya menunjukkan beragam sentimen, termasuk masalah pembaruan aplikasi, kesulitan verifikasi, dan sentimen positif. Analisis ini membantu pengembang aplikasi memahami dan meningkatkan pengalaman pengguna.

Penelitian berjudul "*Clustering Review* Pengguna Aplikasi *Zenius* pada Layanan *Google Play Store* Menggunakan Metode *DBSCAN* dan *HDBSCAN*" bertujuan untuk mengevaluasi ulasan pengguna yang berpengaruh terhadap kualitas aplikasi. Dua metode *clustering*, *DBSCAN* dan *HDBSCAN*, digunakan untuk mengelompokkan ulasan. Hasilnya menunjukkan *HDBSCAN* lebih efektif dengan *Silhouette Coefficient* 0,2941 dibandingkan *DBSCAN* dengan 0,1310. Penelitian ini menemukan bahwa banyak keluhan terkait kesulitan masuk dan masalah teknis, serta beberapa ulasan positif tentang kegunaan aplikasi. Metode *clustering* dan *text mining* membantu dalam menganalisis ulasan dan memahami persepsi pengguna terhadap aplikasi *Zenius*. Penelitian ini menekankan pentingnya evaluasi ulasan pengguna untuk meningkatkan kualitas dan responsivitas aplikasi.

Penelitian berjudul “Pengelompokan Ulasan Aplikasi Pedulilindungi Dengan Algoritma *K-Medoids*” mengevaluasi respons pengguna terhadap aplikasi PeduliLindungi, yang dirilis untuk memantau penyebaran *COVID-19* di Indonesia. Aplikasi ini membantu melacak lokasi pengguna dan memberikan informasi tentang keramaian, mendukung upaya *tracing* dan pencegahan virus.Penelitian menggunakan data ulasan dari *Google Play Store* dan mengelompokkan ulasan tersebut dengan algoritma *k-medoids clustering* serta *Word Embedding FastText*. Metode *KDD* (*Knowledge Discovery in Database*) diterapkan, dimulai dari seleksi data hingga *data mining*. Hasilnya menunjukkan bahwa *clustering* tanpa *stemming*, menggunakan arsitektur *cbow* dan *manhattan distance*, menghasilkan dua cluster dengan nilai *Davies Bouldin Index* (DBI) mendekati nol. Hal ini menunjukkan efektivitas aplikasi PeduliLindungi berdasarkan respons pengguna.

Sedangkan dalam penelitian ini, peneliti mengkaji pemahaman tentang tingkat kepuasan pengguna terhadap aplikasi Merdeka Mengajar dari Kemendikbudristek. Pendekatan yang digunakan melibatkan penerapan algoritma *K-Means Clustering* pada ulasan pengguna yang tersedia di *Google Play Store*. Yang membedakan penelitian ini adalah penggunaan Metodologi *CRISP-DM (Cross Industry Standard Process for Data Mining)* serta beragam *tools* seperti *google-play-scrapper, matplotlib, Pandas,* dan lain sebagainya. Hasil dari upaya ini diharapkan dapat menghasilkan sebuah *dashboard* interaktif yang dibentuk dalam situs *web* agar memudahkan pihak terkait untuk membaca data yang sudah di analisis.

2.2. Kajian Teoritik

Berikut beberapa Kajian Teoritik yang digunakan dalam penelitian ini:

2.2.1. Aplikasi

Menurut Rachmad Hakim S, Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur *Windows* & permainan(*game*), dan sebagainya (Mahardika, 2020). Secara istilah aplikasi adalah program siap pakai yang dibuat untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melakukan tugas tertentu (Jainuri, dkk. 2021).

2.2.2. Dinas Pendidikan dan Kebudayaan

Dinas Pendidikan dan Kebudayaan adalah instansi pemerintahan kota atau kabupaten yang bertanggung jawab atas pendidikan siswa di wilayahnya. Tugas utamanya adalah melaksanakan pendidikan sesuai dengan kewenangan daerah yang diberikan oleh Pemerintah. Salah satu tugas pentingnya adalah mengelola perkembangan pendidikan anak-anak melalui laporan dari guru untuk memantau tumbuh kembang siswa (Susanto, 2022). Fungsi utama pendidikan adalah membebaskan manusia dari kebodohan, penindasan, ketertinggalan, dan kemiskinan. Dinas Pendidikan dan Kebudayaan memiliki beberapa fungsi utama, termasuk merumuskan kebijakan pendidikan, membina dan mengendalikan pendidikan pra-sekolah dan luar sekolah, mengurus pendidikan dasar dan menengah, serta membina anak rawan putus sekolah dengan bantuan psikolog.

Mereka juga merumuskan pelaksanaan pencegahan anak putus sekolah dan membina siswa tentang pentingnya pendidikan (Fitriani, 2018).

2.2.3. *Platform* Merdeka Mengajar

Platform Merdeka Mengajar adalah sebuah aplikasi yang dirancang untuk mendukung guru dalam merencanakan, melaksanakan, dan mengevaluasi pembelajaran. Aplikasi ini menyediakan berbagai fitur, termasuk akses ke materi pembelajaran, alat bantu mengajar, dan berbagai sumber daya pendidikan lainnya. *Platform* Merdeka Mengajar didesain sebagai alat yang membantu guru-guru dalam meningkatkan kualitas pengajaran mereka dengan menyediakan berbagai fasilitas yang mendukung pembelajaran yang efektif (Wardana, Indra, & Ulya, 2023).

Platform Merdeka Mengajar (PMM) memberikan keleluasaan bagi tenaga pendidik untuk belajar dan memajukan kemampuan serta keterampilan yang dimilikinya dalam keadaan kapanpun dan juga dimanapun. *Platform* ini juga menyediakan fitur "Pembelajaran" di dalamnya terdapat fasilitas pelatihan mandiri bagi tenaga pendidik maupun tenaga kependidikan untuk mengakses berbagai sumber atau bahan pelatihan yang berkualitas dan bermutu, dan mereka juga bisa mempelajarinya secara mandiri (Prasetyaningsih, Muiz, & Fatimah, 2024).

2.2.4. *Google Play Store*

Google Play Store adalah toko aplikasi resmi yang dikembangkan oleh *Google* yang memungkinkan pengguna dalam mencari dan mengunduh aplikasi untuk sistem *Android* (Larasati, dkk. 2022).

2.2.5. Ulasan

Teks ulasan adalah teks yang berisi tinjauan atau ringkasan buku atau yang lain untuk koran atau penerbitan. Teks ulasan dapat dikaitkan dengan resensi.

Resensi adalah tulisan atau ulasan mengenai nilai sebuah karya. Tujuan dari ulasan yaitu menyampaikan kepada pembaca mengenai kelayakan dari sebuah hasil karya sastra (Chalidiah, dkk. 2020).

2.2.6. Analisis Univariat

Analisa univariat adalah analisis untuk mengetahui gambaran dari tiap variabel *independen* dan variabel *dependen* data yang telah dipeoleh dari hasil pengumpulan data disajikan dalam bentuk tabel distribusi frekuensi dan teks. Dimana variabel *independen* dan *dependen* (Umami, 2019).

2.2.7. Analisis Multivariat

Metode analisis multivariat adalah suatu metode statistika yang tujuan digunakannya adalah untuk menganalisis data yang terdiri dari banyak variabel serta diduga antar variabel tersebut saling berhubungan satu sama lain (Mewengkang, dkk. 2022).

2.2.8. Bahasa Pemrograman *Python*

Python adalah bahasa pemrograman *freeware* yang bebas digunakan tanpa batasan distribusi atau penyalinan. Dilengkapi dengan *source code*, debugger, profiler, antarmuka untuk fungsi sistem, GUI (*Graphical User Interface*), dan basis data, *Python* memudahkan pembuatan server web hanya dalam tiga baris kode. Filosofi *Python* termasuk *coherence*, yang menekankan kemudahan membaca, menulis, dan memelihara kode. Fitur utamanya meliputi perpustakaan yang luas dengan modul 'siap pakai', tata bahasa yang jelas dan mudah dipelajari, serta aturan layout kode sumber yang memudahkan pengecekan dan penulisan ulang. *Python* juga mendukung pemrograman berorientasi objek. (Clinton & Sengkey, 2019).

2.2.9. *Google Colaboratory*

Google Colaboratory (alias *Colab*) adalah proyek yang bertujuan menyebarluaskan pendidikan dan penelitian pembelajaran mesin. *Colab* menyediakan runtime *Python* 2 dan 3 yang telah dikonfigurasikan dengan perpustakaan penting seperti *TensorFlow*, *Matplotlib*, dan *Keras*. Layanan ini menawarkan *runtime* yang dipercepat GPU, juga dikonfigurasikan dengan perangkat lunak yang disebutkan. Infrastruktur *Google Colab* di-host di platform *Google Cloud* (Bagas, dkk. 2020).

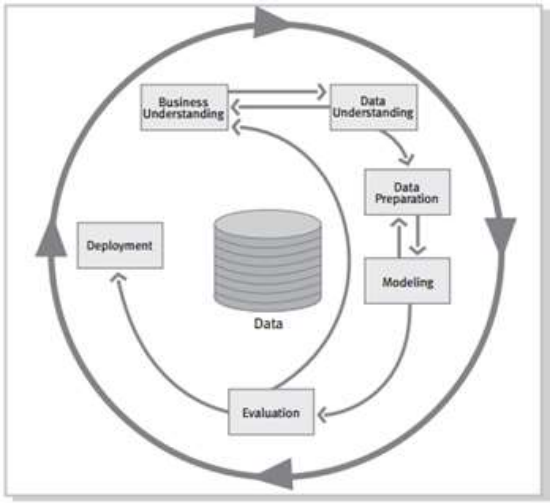
2.2.10. *Data Mining*

Data mining atau *knowledge discovery in database* (*KDD*) adalah proses *resourcing* dan penggunaan data untuk menemukan pola atau hubungan dari sekumpulan data berukuran besar. *Data mining* juga digunakan sebagai proses menemukan informasi yang berguna dari gudang basis data yang besar. *Data mining* juga dapat diartikan sebagai pengekstrakan informasi dari sekumpulan data besar untuk membantu dalam pengambilan keputusan.Hasil dari proses *data mining* dapat digunakan sebagai evaluasi pengambilan keputusan di masa depan (Harahap & Sulindawaty, 2019).

2.2.11. Metodologi *CRISP-DM*

Cross Industry Standard Process for Data Mining (CRISP–DM) digunakan sebagai standard proses *data mining* sekaligus sebagai metode penelitian. Menurut Daniel T. Larose, *Cross-Industry Standart Process for Data Mining (CRISP-DM*) yang dikembangkan tahun 1996 oleh analisis dari beberapa industri seperti *Daimler Chrysler, SPSS dan NCR. CRISP-DM* menyediakan standar proses *data mining* sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian (Rivanthio, dkk. 2020). Proses *data mining* berdasarkan *CRISP-DM (Cross*

Industry Standard Process for Data Mining) terdiri dari enam fase adalah sebagai berikut:



Gambar 2.1 Tahapan Metodologi CRISP-DM
Sumber: *https://binus.ac.id/malang/2022/05/crisp-dm-cross-industry-standard-process-for-data-mining/*

1. *Business Understanding Phase* (Fase Pemahaman Bisnis): Pada fase ini, tujuan proyek dan kebutuhan secara detail dalam lingkup bisnis atau unit penelitian ditentukan secara keseluruhan. Tujuan dan batasan diidentifikasi dan diterjemahkan menjadi formula dari permasalahan *data mining*. Selanjutnya, strategi awal disiapkan untuk mencapai tujuan yang ditentukan, memastikan semua pihak memahami arah, batasan proyek ini.
2. *Data Understanding Phase* (Fase Pemahaman Data): Fase ini melibatkan pengumpulan data yang relevan. Analisis penyelidikan data digunakan untuk mengenali lebih lanjut data yang telah dikumpulkan dan untuk pencarian pengetahuan awal. Evaluasi kualitas data dilakukan untuk memastikan data yang digunakan berkualitas baik. Jika diperlukan, sebagian kecil data yang mungkin mengandung pola dari permasalahan dipilih untuk analisis lebih lanjut.

3. *Data Preparation Phase* (Fase Persiapan Data): Data awal disiapkan dan dikumpulkan untuk digunakan dalam fase-fase berikutnya. Ini merupakan pekerjaan berat yang perlu dilakukan secara intensif. Kasus dan variabel yang akan dianalisis dipilih berdasarkan relevansi dan kecocokan dengan analisis yang akan dilakukan. Beberapa variabel mungkin perlu diubah sesuai kebutuhan. Data awal disiapkan sehingga siap untuk digunakan dalam perangkat permodelan.
4. *Modelling Phase* (Fase Pemodelan): Pada fase ini, teknik pemodelan yang sesuai dipilih dan diterapkan. Aturan model dikalibrasi untuk mengoptimalkan hasil. Beberapa teknik mungkin digunakan untuk permasalahan data mining yang sama, tergantung pada kebutuhan spesifik proyek. Jika diperlukan, proses dapat kembali ke fase pengolahan data untuk memastikan data sesuai dengan spesifikasi kebutuhan teknik data mining yang digunakan.
5. *Evaluation Phase* (Fase Evaluasi): Fase ini mengevaluasi satu atau lebih model yang digunakan dalam fase pemodelan untuk mendapatkan kualitas dan efektivitas sebelum disebarkan untuk digunakan. Ditentukan apakah model yang dihasilkan memenuhi tujuan yang ditetapkan pada fase awal. Fase ini juga menetapkan apakah ada permasalahan penting dari bisnis atau penelitian yang tidak tertangani dengan baik. Keputusan diambil berkaitan dengan penggunaan hasil dari data mining.
6. *Deployment Phase* (Fase Penyebaran): Fase terakhir ini melibatkan penggunaan model yang dihasilkan. Pembentukan model tidak menandakan

penyelesaian proyek; model harus digunakan dalam konteks yang relevan. Contoh sederhana penyebaran adalah pembuatan laporan untuk mendokumentasikan hasil dan temuan proyek (Damayanti, 2021).

2.2.12. *Natural Language Processing* (NLP)

Natural Language Processing (*NLP*) adalah disiplin ilmu dalam bidang kecerdasan buatan yang memfasilitasi interaksi antara mesin dan manusia melalui bahasa alami (Suherlan, dkk. 2023). *NLP* merupakan pemrosesan bahasa, seperti lisan dan tulisan yang dilakukan oleh manusia dalam melakukan percakapan sehari-hari melalui komputer. Dalam prosesnya, *NLP* akan membuat komputer dapat memahami dari setiap perintahperintah atau standar bahasa yang biasa ditulis atau dilakukan oleh menusia (Radhian & Afrianto, 2019). *Preprocessing* merupakan tahapan sebelum proses pengklasifikasian yang diperlukan untuk membersihkan, menghilangkan, mengubah sumber data, baik itu berupa karakter non alfabet maupun kata-kata yang tidak diperlukan. Hal ini bertujuan agar data yang digunakan lebih optimal ketika digunakan pada proses pengklasifikasiannya (Muttaqin & Bachtiar, 2022). *Data Preprocessing* bertujuan untuk mengubah data mentah menjadi data yang berkualitas sehingga data layak untuk diolah pada tahapan selanjutnya (Alghifari & Juardi, 2021). Beberapa proses *preprocessing* yang terdapat pada *NLP*, yaitu:

1. *Case folding*, adalah salah satu proses pengolah kata. Proses ini mengubah huruf besar menjadi huruf kecil. Selain itu, angka dan simbol khusus seperti tanda seru (!), koma (,), garis miring (/), lebih dari (>), kurang dari (danlt;) dll

2. *Tokenisasi*, menggambarkan proses membagi teks menjadi kata-kata, menggunakan spasi sebagai pemisah, dengan tujuan membuat setiap kata berdiri sendiri, tidak terkait dengan kata lain
3. *Part-of-Speech Tagging*, adalah penugasan *tag POS* ke kalimat atau kata. Pengkodean *POS* dapat dilakukan dengan memberikan pengenal pada setiap kata seperti kata kerja, kata benda, kata sifat, dan aturan tata bahasa lainnya
4. *Syntactic Parsing*, adalah proses menganalisis sintaksis kalimat berdasarkan teori gramatikal tertentu. Proses *screening* secara umum dapat dibagi menjadi dua bagian, yaitu kelompok ketergantungan dan kelompok daya tarik.
5. *Stemming* merupakan tahap menciptakan kata dasar dari sebuah kata dengan menghilangkan sufiks dari kata tersebut seperti ”in-”, ”-nya” dll
6. *Stopword Removal* adalah proses *filtering*, sebuah kalimat biasanya mengandung beberapa kata yang tidak lagi memiliki makna substantif, seperti ”ini”, ”ini”, dll.
7. *Stopword* merupakan kumpulan kata umum (*common words*) yang tidak penting namun kerap muncul. Contoh *stopword* pada bahasa Indonesia adalah ”ke”, ”di”, ”yang”, dll. (Furqan, dkk. 2023)

Pada penelitian ini tahap *preprocessing* menggunakan *tools* atau *library NLTK* (*Natural Language Toolkit*) dari pemrograman *python* dan menggunakan kamus sentimen dari Sastrawi. *NLTK* merupakan platform terkemuka yang bersifat *open-source* dan gratis untuk membangun *program Python* dan menyediakan antarmuka yang mudah digunakan dan serangkaian pustaka pemrosesan teks untuk

klasifikasi, tokenisasi, *stemming, tagging, parsing*, penalaran semantik, dan pustaka pemrosesan bahasa alami lainnya (Ademariana, dkk. 2021).

2.2.13. *Term Frequency — Inverse Document Frequency (TF - IDF)*

Metode *TF/IDF* merupakan suatu cara untuk memberikan bobot hubungan suatu kata (term) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu yang disebut *Term Frequency (TF)* dan *inverse* frekuensi dokumen yang mengandung kata yang disebut *Inverse Document Frequency (IDF)* (Sasmita & Falani, 2018). *TF-IDF* adalah salah satu metode yang umum digunakan dalam pemrosesan bahasa alami dan sistem temu kembali informasi untuk meningkatkan akurasi dan relevansi hasil pencarian. Dalam metode *TF-IDF,* nilai *TF* dan *IDF* dikalikan bersama-sama untuk menghasilkan bobot kata (*term weight*) untuk setiap kata dalam dokumen. Bobot ini mencerminkan tingkat pentingnya kata dalam dokumen tersebut dibandingkan dengan koleksi dokumen yang lebih besar. Rumus metode *Term Frequency-Inverse Document Frequency (TF-IDF)*:

$$tf = 0{,}5 + 0{,}5 \text{ x } \frac{tf}{\max(tf)}$$

$$idf_t = \log\left(\frac{D}{df_t}\right)$$

$$W_{d.t} = tf_{d.t} \text{ x } idf_{d.t}$$

Keterangan :

D = dokumen ke-d

t = *term* ke-t dari dokumen

W = bobot ke-d terhadap *term* ke-t

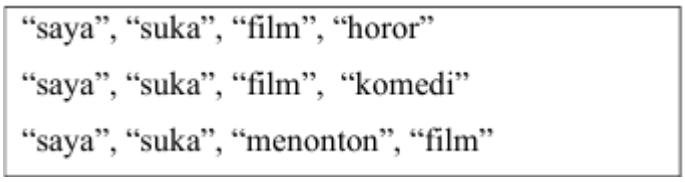
tf = jumlah kemunculan *term* i dalam dokumen

idf = *Inversed Document Frequency*

df = banyak dokumen yang mengandung *term* i (Septiani & Isabela, 2022)

2.2.14. *Bag of Words(BoW)*

Bag-of-Words merupakan sebuah model dari sebuah proses yang ada didalam *Natural Language Processing*, dan banyak digunakan untuk mengambil nilai dari sebuah kata yang sebelumnya diolah pada sebuah model *machine learning*. Model *Bag-of-Words* bekerja dengan cara mempelajari sebuah kata dari pada sebuah dokumen, kemudian menginterpretasikan setiap dokumen dengan menghitung jumlah kemunculan tiap kata dari dokumen tersebut.



Gambar 2.3 Sebelum Penerapan *Bag of Words*

Sumber: Raja Farhan Ramadhan dkk., Implementasi Algoritma *Support Vector Machine* dan Model *Bag-of-Words* dalam Analisis Sentimen mengenai PILKADA 2020 pada Pengguna Twitter, 2022, 4924-4931

saya	suka	film	horor	komedi	menonton
1	1	1	1	0	0
1	1	1	0	1	0
1	1	1	0	0	1

Gambar 2.4 Sesudah Penerapan *Bag of Words*

Sumber: Raja Farhan Ramadhan dkk., Implementasi Algoritma *Support Vector Machine* dan Model *Bag-of-Words* dalam Analisis Sentimen mengenai PILKADA 2020 pada Pengguna Twitter, 2022, 4924-4931

Langkah pertama yang harus dilakukan adalah melakukan *scraping* atau pengambilan data. Selanjutnya adalah proses data *labeling*, yaitu membuat kolom pada data yang sudah berhasil diambil dengan tujuan untuk menentukan polaritas dari sebuah ulasan. Selanjutnya yang dilakukan adalah tahap *data preprocessing* Tujuan dari proses ini adalah untuk mendapatkan bentuk data yang awalnya tidak

terstruktur, menjadi bentuk yang lebih terstruktur, dan dapat disesuaikan dengan kebutuhan proses penelitian selanjutnya (Pohan, Ratnawati, & Arwani, 2022).

2.2.15. *Word2Vec*

Word2Vec merupakan algoritma representasi vektor kata yang mampu mencapai kinerja terbaik dalam *NLP* (*Natural Language Processing*) dengan cara mengelompokan kata yang serupa memiliki vektor yang sama. *Word2Vec* menghitung representasi kata ke dalam vektor menggunakan *neural network*. Vektor kata yang dihasilkan adalah vektor ruang dimensi yang menangkap makna semantik dari kata *Word2Vec* mentransformasikan setiap kata unik sebagai vektor. Kelebihan *Word2Vec* yaitu dapat merepresentasikan kesamaan kontekstual dari dua kata pada vektor yang dihasilkan (Af'idah, Dairoh, & Handayani, 2021).

2.2.16. *Clustering analisis*

Clustering adalah proses pengelompokan sejumlah data atau objek ke dalam *cluster* (kelompok) yang terdiri dari data yang paling mirip satu sama lain, dan berbeda dengan data dalam *cluster* lain. Ada dua metode *clustering* utama: *hierarchical clustering* dan *partitioning*. Dalam *clustering*, data dengan kesamaan dimasukkan ke dalam *cluster* yang sama, sedangkan data yang berbeda dimasukkan ke dalam *cluster* yang berbeda (Lestari, 2019). *Clustering* termasuk salah satu teknik analisis multivariat yang mengelompokkan objek berdasarkan karakteristik yang sama. Kumpulan objek serupa dalam satu kelompok dan berbeda dengan objek dalam kelompok lain disebut *cluster*. *Clustering* adalah alat bantu *data mining* bertujuan mengelompokkan objek ke dalam *cluster* (Riza & Saputro, 2022).

2.2.17. Algoritma *K-Means Clustering*

K-Means Cluster adalah salah satu metode dan *clustering* non hierarki yang berusaha mengelompokan data ke dalam suatu *cluster* sehingga data yang memiliki karakteristik yang sama dikelompokan ke dalam satu *cluster* yang sama. *K-Means* termasuk algoritma clustering dengan proses berulang-ulang. Huruf K diartikan sebagai jumlah *cluster* yang hendak dibuat. Selanjutnya nilai K ditetapkan secara acak. Sedangkan *Means* adalah nilai sementara yang menjadi pusat dari *cluster* atau disebut juga dengan *centroid*. Setiap data yang ada dihitung jaraknya terhadap masing-masing *centroid* dengan memakai rumus *Euclidean* hingga dihasilkan jarak terdekat dari setiap data dengan centroid (Zaki, dkk. 2022). Adapun prosedur perhitungan Algoritma *K-Means Clustering Analysis* diuraikan sebagai berikut:

1. Menentukan banyaknya *cluster,* Dalam menentukan banyaknya cluster diharuskan tidak lebih dari jumlah kriteria yang ada.
2. Menentukan titik pusat *cluster (centroid),* Dalam menentukan titik pusat *cluster (centroid)* dengan mengambil nilai minimal, rata-rata dan maksimal, jika akan dibuat 3 *cluster*, karna pada penelitian ini penulis membuat 2 *cluster* jadi hanya mengambil nilai minimal dan maksimal pada setiap kriteria.
3. Menghitung antara jarak titik data objek ke titik data pusat (*centroid*), Pada tahap perhitungan jarak ini menggunakan rumus dengan *Euclidean Distance* sebagai berikut:

$$d_{Euclidean}(x, y) = \sqrt{\sum_{i} (x_i - y_i)^2}$$

Keterangan :

d(x,y) : jarak antara data pada titik x dan y
x : titik data objek
y : titik data *centroid*
i : jumlah atribut data

4. Pengelompokan data objek untuk menentukan anggota *cluster* berdasarkan jarak minimum Pada proses ini setelah menghitung jarak diambil nilai minimum yang diberi nilai 1 dan lainnya 0, dimana nilai 1 untuk data yang ditempatkan pada ke cluster dan nilai 0 untuk data yang tempatkan ke cluster yang lain hingga akan membentuk sebuah *matriks*.
5. Kembali pada tahap ke 2, Lakukan pengulangan hingga poin 4 hingga nilai pada tiap cluster tidak berubah tempat. Lakukan pengulangan hingga poin 4 hingga nilai pada tiap *cluster* tidak berubah tempat.

2.2.18. *Silhouette Score*

Menurut *silhouette score* atau yang biasa disebut *silhouette coefficient* merupakan sebuah metode untuk melakukan pengukuran terhadap kualitas dan kekuatan *cluster*. Metode ini menggabungkan konsep dari *cohesion* yang mengukur sebuah *cluster* yang memiliki hubungan antar objek dan separation yang mengukur jarak antara *cluster* yang berbeda. Berikut merupakan persamaan mengenai perhitungan *silhouette score*: (Haq, dkk. 2023),

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

1. Mencari jarak rerata di klaster yang sama, diasumsikan data ke-i berada di klaster A. Rumus dari a(i) ditulis dalam persamaan di mana, A = banyaknya data di klaster A

$$a(i) = \frac{1}{|A| - 1} \sum_{jeA, j \neq 1} d(i,j)$$

2. Menghitung nilai b(i) yang merupakan nilai minimum dari jarak rata-rata data ke-i dengan semua data di klaster berbeda. Sekarang, mari asumsikan klaster berbeda selain A dengan klaster C. Maka, perhitungan jarak rata-rata data ke-i dengan semua data di klaster C ditulis sebagai berikut, di mana C = banyaknya data di klaster C.:

$$d(i, C) \coloneqq \frac{1}{|C|} \sum_{jeC} d(i,j)$$

3. Memilih nilai jarak paling minimum sebagai nilai b(i), dengan persamaan Jika klaster B memiliki nilai jarak minimum, maka d(i,B) = b(i) yang disebut sebagai tetangga dari data ke-i dan merupakan klaster terbaik kedua untuk data ke-i setelah klaster A.

$$b(i) \coloneqq \frac{min}{C \neq A} d\ (i,j)$$

4. Menghitung *Silhouette coefficient* sesuai persamaan (2), dengan nilai s(i) berada antara -1 dan 1. Bila s(i) sama atau mendekati 1, maka separasi data ke-i dikatakan baik, bila antara 0 dan 1 maka separasi data ke-i antara klaster A dan B, dan bila s(i) sama atau mendekati -1 maka separasi kategori lemah

2.2.19. *Elbow Method*

Metode *elbow* merupakan salah satu metode yang dapat digunakan untuk menentukan jumlah *cluster* terbaik, yaitu dengan cara melihat presentase setiap *cluster* yang akan membentuk siku pada suatu titik tertentu. Metode *elbow* biasa disajikan dalam bentuk grafik untuk mengetahui lebih jelas siku yang terbentuk. Tujuan dari metode *elbow* adalah untuk memilih nilai k yang kecil dan masih

memiliki nilai *withinss* yang rendah. Nilai k pada kombinasi siku dengan *k-means* adalah grafik hubungan cluster dengan penurunan *error* (Maori, 2023).

2.2.20. *Library Google-Play-Scraper*

Google-Play-Scraper adalah *API* untuk mengekstraksi data aplikasi dan ulasan dari *Google Play Store* tanpa ketergantungan eksternal. Data yang diambil mencakup informasi aplikasi (judul, *URL developer*, kategori, rating, *review*, deskripsi, *thumbnail*, *screenshot*) dan ulasan pengguna (nama, foto, rating, tanggal, *likes*, komentar) (Larasati, dkk. 2022).

2.2.21. *Library Pandas*

Pandas adalah *library Python* populer untuk analisis data, mendukung struktur data yang cepat, fleksibel, dan ekspresif, dirancang untuk bekerja pada data "relasional" atau "berlabel". *Pandas* kini menjadi *library* penting untuk analisis data praktis dengan *Python* (Alfarizi, dkk. 2023)

2.2.22. *Library Matplotlib*

Library yang digunakan untuk visualisasi data. Visualisasi data memiliki peranan penting untuk memahami data secara lebih mendalam sebelum melakukan *data processing* dan melatihnya dalam program *machine learning* (Alfarizi, dkk. 2023).

2.2.23. *Library NLTK*

NLTK adalah paket *program open source* yang menyediakan modul untuk komputasi linguistik. *NLTK* mencakup pemrosesan bahasa alami simbolik dan statistik, serta berorientasi pada korpus (Rifano, dkk. 2020). Fungsi NLTK meliputi mendeteksi tanda baca sebagai karakter terpisah, memisahkan koma dan tanda petik

jika diikuti spasi, serta memisahkan titik-titik di akhir baris. Menggunakan *API* yang dibuat dengan *Python*, penelitian ini bertujuan membantu pengguna menemukan informasi terkait *Poltekpar* secara *real time* (Furqan, dkk. 2023).

2.2.24. *Library Python Scikit-Learn*

Library ini menyediakan banyak algoritma pembelajaran tanpa pengawasan dan pengawasan. Itu dibangun di atas beberapa teknologi yang mungkin sudah dikenal, seperti *NumPy, panda*, dan *Matplotlib* (Alfarizi, dkk. 2023).

2.2.25. *Library Streamlit*

Streamlit merupakan *library* yang ada di bahasa pemrograman *Python* yang dapat dengan mudah untuk *deploy machine learning* maupun *data science. Streamlit* sangat cocok bagi mereka yang terjun di dunia data dan mereka ingin membuat sebuah *project* yang tergabung dalam sebuah tim(Irawansyah, dkk. 2010).

2.2.26. *GitHub*

GitHub adalah *platform hosting* untuk proyek *open source* yang menggunakan alat *Git* dan berfungsi juga sebagai *web hosting*. *Git* adalah *tool system control* yang memfasilitasi kolaborasi proyek dalam pemrograman dan pengembangan perangkat lunak, memungkinkan manajemen proyek yang efektif meskipun secara daring. Di bidang pemrograman, *GitHub* sering digunakan sebagai *version control system (VCS)* untuk manajemen kode sumber dan kolaborasi tim. *GitHub* gratis, *open source*, dan mendukung berbagai bahasa pemrograman, menjadikannya alat yang sangat populer di kalangan developer.

GitHub menawarkan *hosting* gratis untuk proyek *open source*, memungkinkan kerja tim tanpa batasan lokasi, serta memiliki fitur seperti sosial

media untuk mengikuti profil *programmer* lain dan memonitor *repository*. Pengguna dapat menerima notifikasi untuk setiap perubahan pada *repo*, melihat sejarah perubahan, membagikan proyek, dan memperbaiki kode sumber secara bersama-sama. Dengan demikian, *GitHub* menjadi situs *repository* yang ideal untuk pendistribusian *source code* secara terbuka, memonitor proyek dari awal, dan mempercepat manajemen proyek perangkat lunak (Sumber: GitHub).

2.2.27. Evaluasi *White Box Testing*

White box testing adalah pengujian perangkat lunak pada tingkat alur kode program, apakah masukan dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. dan pengujian yang didasarkan pada pengujian design program secara prosedural, secara structural, pengujian berbasis logika atau pengujian berbasis kode. Tes ini bertujuan untuk menganalisis kebenaran struktur program yang dibuat dan kinerja program (Pratala, Asyer, Prayudi, & Saifudin, 2020). Pada *white box* testing terdapat beberapa pengujian yakni data *flow testing, control flow testing, basis path,* dan *loop testing*. (Solissa, dkk. 2023). Pada penelitian ini Peneliti menggunakan Evaluasi *White box* dengan *basis bath*. Tujuan dari teknik ini adalah untuk mengukur kompleksitas dari suatu program dengan cara mengidentifikasi semua jalur yang mungkin dilalui oleh program tersebut . Proses perhitungan dalam teknik *basis path* adalah sebagai berikut :

1. Membuat *Flow Graph*: Buat *flow graph program* yang akan diuji, terdiri dari simpul-simpul (titik keputusan) dan jalur-jalur penghubungnya. Hitung jumlah *edges* (jalur) dan *nodes* (simpul).

2. Menghitung *Cyclomatic Complexity (CC)*: Hitung kompleksitas program dengan rumus V(G) = E - N + 2, di mana E adalah jumlah *edges* dan N adalah jumlah *nodes*. Nilai *CC* menunjukkan tingkat kompleksitas program.
3. Menyusun Skenario Pengujian: Susun skenario pengujian berdasarkan jalur-jalur dalam *flow graph* untuk memastikan semua bagian program diuji (Zen & Nuryasin, 2024).

2.2.28. Evaluasi *Black Box Testing*

Pengujian *Black box* adalah pengujian yang hanya menguji bagian luar dari perangkat lunak, yang berfokus pada kebutuhan fungsional pada perangkat lunak, berdasarkan pada spesifikasi kebutuhan perangkat lunak (Pratama, dkk. 2023). *Black Box testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. (Nurdiansah & Irmawati, 2020) . Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan, tidak ada upaya untuk mengetahui kode program apa yang *output* pakai. Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan (Baktiar, Mulainsyah, Sasmoro, & Sumiati, 2021).

BAB III

METODE PENELITIAN

3.1. Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Dinas Pendidikan dan Kebudayaan Kota Samarinda yang beralamat Jl. Biola No.4A, Sungai Pinang Luar, Kec. Samarinda Kota, Kota Samarinda, Kalimantan Timur 75123 dari tanggal 6 Maret 2024 sampai dengan 6 Agustus 2024.

3.2. Teknik Pengumpulan Data

Berikut Teknik Pengumpulan Data yang digunakan dalam melakukan penelitian ini :

3.2.1. Studi Pustaka

Studi Pustaka adalah tahapan yang sangat penting saat melakukan penelitian. Tujuan utama dari studi dari studi ini adalah mengumpulkan informasi yang relevan dengan topik atau masalah yang fokus dalam penelitian. Sumber informasi yang digunakan meliputi buku buku ilmiah, laporan penelitian, artikel akademisi, jurnal ilmiah, baik dalam format cetak maupun elektronik. Dalam konteks era digital, media sosial juga menjadi sumber informasi yang semakin signifikan, terutama untuk menganalisis tren dan menggali opini publik yang terkait dengan subjek penelitian.

3.2.2. Observasi

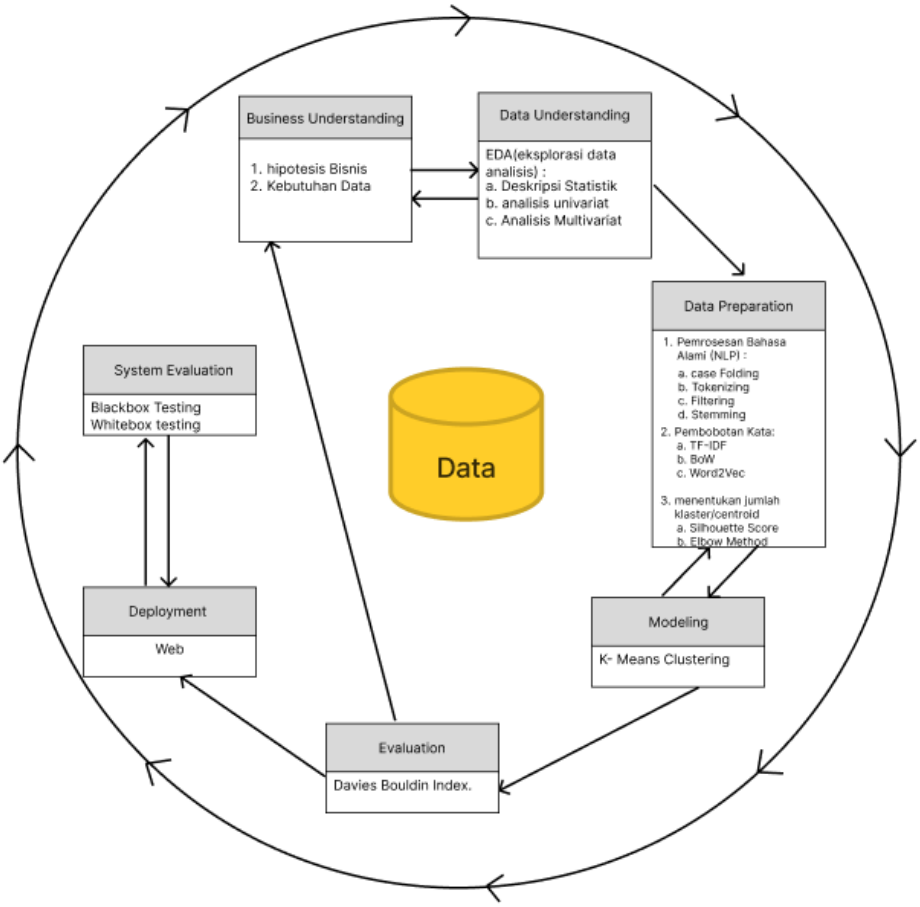
Pada teknik observasi ini peneliti melakukan pengamatan secara daring atau *online* sekaligus mengambil data untuk diteliti pada ulasan dari pengguna aplikasi

Merdeka Mengajar dari Kemendikbudristek di *Google Play Store.*

3.2.3. Wawancara

Pada teknik ini peneliti melakukan kegiatan wawancara langsung dengan pihak Dinas Pendidikan dan Kebudayaan Kota Samarinda mengenai penggunaan aplikasi Merdeka Mengajar.

3.3. Metode Pengembangan Sistem



Gambar 3.1 Tahapan Metode *CRISP-DM* dalam penelitian ini
Sumber: Dokumentasi Pribadi

Pada penelitian ini, metode yang digunakan mengacu pada Kerangka Kerja atau metode *CRISP-DM* (*Cross-Industry Standard Process for Data Mining*). *CRISP-DM* adalah suatu pendekatan yang digunakan untuk

mengelola proyek *data mining*, yang terdiri dari enam tahapan utama, yaitu *business understanding, data understanding, data preparation, modeling, evaluation*, dan *deployment*.

Dengan menggunakan *CRISP-DM* sebagai panduan metodologis, penelitian ini dapat memastikan bahwa langkah-langkah yang terstruktur dan terdokumentasi dengan baik,dan menambahkan tahap terakhir *web service Evaluation* sehingga menghasilkan hasil yang konsisten dan dapat diandalkan dalam pengembangan solusi berbasis data.

Berikut merupakan alur penelitian yang akan dilakukan oleh peneliti pada penelitian ini, berdasarkan metode *CRISP-DM* dan menambahkan tahap terakhir yaitu *Web Service Evaluation* bertujuan uji coba fungsionalitas, kinerja, keamanan, dan responsivitas, untuk memastikan bahwa layanan tersebut dapat digunakan secara efektif oleh pengguna.

3.3.1. *Business Understanding*

Penelitian ini bertujuan mengumpulkan data ulasan pengguna aplikasi Merdeka Mengajar dari *Play Store* untuk memahami pola, topik, dan sentimen utama pengguna. Dengan menggunakan algoritma *K-Means Clustering*, ulasan akan dikelompokkan dalam beberapa kategori relevan. Hipotesis bisnis menyatakan bahwa ulasan positif menyoroti fitur bermanfaat dan apresiasi, sementara ulasan negatif terkait masalah teknis atau performa. Data yang dibutuhkan meliputi teks ulasan, rating, tanggal ulasan, dan teknologi analisis teks serta *clustering*. *Stakeholder* penelitian ini adalah Dinas Pendidikan dan Kebudayaan Kota Samarinda dan pengguna aplikasi Merdeka Mengajar. Kendala yang mungkin

dihadapi termasuk keterbatasan data ulasan, alat dan teknologi analisis teks, serta waktu dan sumber daya.

3.3.2. *Data Understanding*

Tahap *Data Understanding* mencakup beberapa langkah. Pertama, data dikumpulkan menggunakan teknik *crawling* pada *Google Play Store* dengan *library google-play-scraper*. Setelah itu, peneliti melakukan eksplorasi data melalui analisis univariat untuk melihat satu aspek dari ulasan atau komentar pengguna aplikasi Merdeka Mengajar dan analisis multivariat untuk melihat hubungan antara beberapa variabel dalam ulasan tersebut. Peneliti juga mencari wawasan data untuk menemukan pola dan informasi penting dari ulasan, seperti fitur yang paling disukai atau masalah yang paling sering dikeluhkan pengguna aplikasi Merdeka Mengajar di *Google Play Store*.

3.3.3. *Data Preparation*

Pada Tahap ini untuk proses data cleaning dan *text preprocessing* menggunakan *library* dari *NLTK*, *Sastrawi* dan *NLP-ID*. Tahap awal dalam melakukan di tahap ini adalah *Data Cleaning*, termasuk *Case Folding* untuk mengubah semua huruf dalam ulasan menjadi huruf kecil, mengubah karakter *emoji* menjadi teks, dan menghapus kode *HTML*, simbol karakter atau tanda hubung, dan lainnya. Pada tahap *stemming library Sastrawi* dan *NLP-ID* akan dibandingkan, yang lebih baik, maka akan dipilih dalam tahap pemrosesan data. Setelah itu tahap *preprocessing* berikut tahapan-nya :

1. *Normalization*, pada tahap ini meliputi seperti membuat kamus berdasarkan ulasan yang sudah diambil dan diperbaiki secara manual

kata yang kurang benar(jika diperlukan) lalu diterapkan, kemudian memperpanjang kata yang disingkat dan menghapus kata henti atau yang tidak memiliki arti(*Stopword Removal*)

2. *Tokenizing*, mengubah suatu kalimat menjadi kata-kata yang terpisah agar pada data tersebut setiap kata nya memiliki nilai.
3. *Part-of-Speech,* Menandai setiap kata dalam teks dengan label kelas kata seperti *noun, verb, adjective*, dll.
4. *Stemming*, menghilangkan kata yang memiliki imbuhan di awal dan akhir kalimat pada data. Pada tahap proses ini akan menggunakan 2 *library* untuk *NLP*, yaitu *NLTK* dan *NLP-ID*, yang dimana *library* yang memberikan hasil terbaik akan digunakan untuk proses analisis selanjutnya.

Setelah *preprocessing* dilakukan tahap menentukan jumlah Klaster atau *centroid* optimal menggunakan *Silhouette Score* kemudian di maksimalkan dengan menggunakan *Elbow Method.* pembobotan terhadap kata menggunakan beberapa metode seperti *Bag-of-Word*(BoW), *Term Frequency-Inverse Document Frequency*(*TF-IDF*), dan *Word2vec*. Ketiga metode tersebut dilakukan kemudian dibandingkan metode yang mana diantara ketigas metode tersebut yang memiliki hasil lebih baik.

3.3.4. *Modeling*

Pada tahap *modeling* ini, data yang sudah diproses pada tahap *data preparation,* peneliti melakukan proses klasterisasi dengan membangun model

machine learning dan menerapkan algoritma *K-Means Clustering* pada data yang sudah dipersiapkan.

3.3.5. *Evaluation*

Pada tahap ini peneliti melakukan evaluasi hasil dan mengukur kinerja dari model yang sudah dilakukan pada tahap sebelumnya yaitu *Modelling* dengan menggunakan metode *Davies Bouldin Index*.

3.3.6. *Deployment*

Pada tahap *deployment* ini, peneliti mengimplementasikan model dari *machine learning* dengan algoritma *K-Means Clustering* dan hasil klasterisasi yang sudah dilakukan sebelumnya menggunakan tools atau *library web service* dari *python* yaitu *streamlit.* Sebelum dilakukan *deployment*, file dan skrip dari model *clustering* yang sudah dilakukan disalin di *repository* berbasis *online*, yaitu *GitHub*. *Model Clustering* yang sudah disalin dan ditambahkan beberapa skrip pemrograman di *link* kan ke streamlit untuk proses *deployment* agar analisis sentimen yang sudah dibangun dapat ditampilkan secara visual dan dapat dipahami serta digunakan oleh pengguna.

3.3.7. *System Evaluation*

Setelah peneliti melakukan *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation,* dan *Deployment*. Maka pada tahap ini peneliti membuat evaluasi terhadap sistem yang sudah dibangun dan di implementasikan menggunakan *Blackbox testing* dan *Whitebox testing*.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

1. Profil Dinas Pendidikan dan Kebudayaan Kota Samarinda

1) Tentang Dinas Pendidikan dan Kebudayaan Kota Samarinda

Gambar 4.1. Gedung Dinas Pendidikan dan Kebudayaan Kota Samarinda
Sumber: *https://cdn.rri.co.id/berita/17/images/1697237267551-d/9v2exh0m5x1dsz3.jpeg*

Dinas Pendidikan dan Kebudayaan merupakan instansi pemerintah yang memiliki tugas pokok dan fungsi dalam penyelenggaraan dan pengelolaan pendidikan dan kebudayaan di daerah. Pendidikan merupakan salah satu layanan dasar yang perlu ditangani secara profesional oleh pemerintah. Penyelenggaraan pendiidikan menjadi sangat penting dan strategis dalam menetukan masa depan penyelenggaraan pemerintahan dan pembangunan di Kota Samarinda. Oleh karena itu setiap Kabupaten atau Kota memiliki kewenangan untuk mengelola urusan pemerintahan di daerah atas Prakarsa

sendiri, termasuk di dalamnya pengelolaan penyelenggaraan di bidang pendidikan.

Seiring dengan perkembangan kotamadya Samarinda sesuai dengan Peraturan Daerah, maka wilayah Wilayah Samarinda dimekarkan menjadi 10 Kecamatan yang masing-masing terdapat 1 UPTD Pendidikan Cabang Kecamatan, yaitu UPTD Kecamatan Samarinda Ulu, UPTD Kecamatan Samarinda Ilir, UPTD Kecamatan Samarinda Utara, UPTD Kecamatan Samarinda Seberang, UPTD Kecamatan Samarinda Palaran, UPTD Kecamatan Samarinda Kota, UPTD Kecamatan Sungai Kunjang, UPTD Kecamatan Sungai Pinang, UPTD Kecamatan Sambutan, dan UPTD Kecamatan Loa Janan Ilir.

Berdasarkan Peraturan Walikota Samarinda Nomor 9 Tahun 2017 tentang Pembentukan dan Susunan Organisasi Serta Tata Kerja Unit Pelaksana Teknis pada Dinas dan BadanBerdasarkan BAB II pasal 2 di Peraturan, menerangkan bahwa UPT pada Dinas Pendidikan sebagaimana dimaksud pada ayat 1 meliputi: UPT Sanggar Kegiatan Belajar (SKB) dan UPT Pusat Layanan Autis (PLA).

Dinas Pendidikan Kota Samarinda kini membagi penempatan PNS dan Non PNS yang sebelumnya di UPTD ke Kantor Dinas Pendidikan Kota Samarinda dan beberapa instansi di Lingkungan Pemerintah Kota Samarinda.

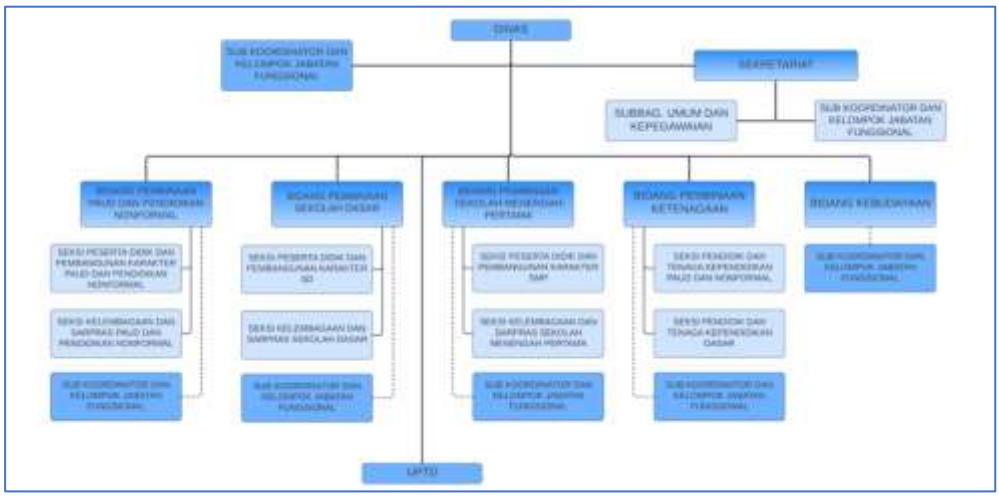
Pada Tahun November 2011, Kantor Dinas Pendidikan Kota Samarinda berpindah lokasi dari Jalan Dahlia No. 09 Kelurahan Bugis Kecamatan

Samarinda Ulu menjadi ke Kantor Baru di Jalan Biola No. 4A Kelurahan Sidodadi Kecamatan Samarinda Ulu, Kota Samarinda Kalimantan Timur.

2) Visi dan Misi Dinas Pendidikan dan Kebudayaan Kota Samarinda

Tabel 4.1 Visi dan Misi Dinas Pendidikan dan Kebudayaan Kota Samarinda
Sumber: Profil – Disdikbud Samarinda (samarindakota.go.id)

Visi	Pendidikan Yang Mewujudkan Masyarakat Bertaqwa, Berkualitas, Sejahtera, Kompetitif dan Berwawasan Lingkungan
Misi	Mengembangkan pendidikan informal, nonformal dan kebudayaan untuk menciptakan Masyarakat yang kreatif dan produktif
	Meningkatkan pemerataan dan pelayanan Pendidikan berkualitas dalam Upaya percepatan penuntasan wajib belajar Pendidikan dasar 9 tahun menuju wajib belajar Pendidikan menengah 12 tahun dan responsive gender.
	Mengembangkan dan meningkatkan mutu sekolah dan pelayanan Pendidikan terhadap peserta didik yang memiliki bakat dan prestasi khusus untuk mendapatkan pengembangan secara optimal sehingga menghasilkan lulusan yang memiliki IMTAQ, menguasai IPTEK, dan mampu bersaing dalam melanjutkan Pendidikan ke jenjang Pendidikan yang lebih tinggi dan memasuki dunia usaha dan industry
	Memberikan kesempatan kepada sekolah untuk mandiri dalam berinisiatif, berkreasi, berinovasi, dan berproduktifitas disertai dengan peningkatan kesejahteraan bagi tenaga pendidik dan tenaga kependidikan



Gambar 4.2 Struktur Organisasi Dinas Pendidikan dan Kebudayaan Kota Samarinda
Sumber: *https://disdikbud.samarindakota.go.id/profil/*

2. Aplikasi Merdeka Mengajar

Platform Merdeka Mengajar dibangun untuk menunjang Implementasi Kurikulum Merdeka agar dapat membantu guru dalam mendapatkan referensi,

inspirasi, dan pemahaman tentang Kurikulum Merdeka. *Platform* ini juga disediakan untuk menjadi teman penggerak bagi guru dan kepala sekolah dalam mengajar, belajar, dan berkarya.

Gambar 4.3 Logo Platform Merdeka Belajar
Sumber: *GTK Kemendikbud | 2023 (kemdikbud.go.id)*

Platform Merdeka Mengajar memberikan kesempatan setara bagi guru untuk belajar dan mengembangkan kompetensinya di mana pun berada. Fitur Belajar mencakup Pelatihan Mandiri yang menyediakan materi pelatihan berkualitas, dan Video Inspirasi yang memberikan akses tidak terbatas ke video inspiratif untuk pengembangan diri dan implementasi kurikulum merdeka. Manfaat *platform* ini antara lain membantu guru dalam mencapai tujuan pembelajaran yang terukur dan menguji pemahaman siswa melalui asesmen. *Platform* ini tersedia di *Google Play Store* untuk perangkat Android minimal versi 5 (*Lollipop*) dan dapat diakses melalui *web browser* di alamat *https://guru.kemdikbud.go.id/.*

Platform Merdeka Mengajar yang dikembangkan diharapkan mampu menjadi partner guru dalam implementasi kurikulum merdeka dengan semangat kolaborasi dan saling berbagi. Konten konten yang dikembangkan oleh kemendikbudristek memberikan pemahaman lebih saat implementasi dan

pembelajaran di satuan Pendidikan yang telah ikut serta dalam implementasi kurikulum merdeka.

Gambar 4.4. *Platform* Merdeka Mengajar di *Google Play*

Sumber: *https://www.detik.com/edu/sekolah/d-6162965/daftar-fitur-menarik-platform-merdeka-mengajar-bukan-untuk-murid-ya*

4.2. Hasil Pengembangan Sistem

4.2.1. *Business Understanding*

Pada tahap ini, peneliti memahami kebutuhan dan tujuan penelitian berupa analisis *clustering* terhadap ulasan aplikasi Merdeka Mengajar dari Kemendikbudristek. Aplikasi ini adalah *platform Edutech* penting bagi tenaga pendidik di Indonesia, diluncurkan pada Oktober 2021, dan telah menerima banyak tanggapan positif. Peneliti melakukan clustering untuk mengelompokkan komentar pengguna terkait fitur yang disukai dan diterima dengan baik.

1. Hipotesis Bisnis

    1) Hipotesis pertama bahwa pengguna memberikan ulasan positif terkait fitur-fitur spesifik dari aplikasi Merdeka Mengajar. Mengetahui fitur yang paling dihargai dapat membantu pengembang meningkatkan dan mempromosikannya. Pengujian dilakukan dengan analisis *clustering* untuk mengidentifikasi fitur yang sering disebutkan dalam ulasan positif.

2) Hipotesis kedua bahwa ulasan pengguna dapat dikelompokkan berdasarkan kesamaan tema atau sentimen. Mengelompokkan ulasan memberikan wawasan tentang kelompok pengguna dan kebutuhan mereka. Pengujian dilakukan dengan algoritma *K-Means* untuk *clustering* ulasan dan menganalisis tema utama dalam setiap *cluster*.

2. Kebutuhan Data

Berikut kebutuhan data dalam penelitian ini :

1) Teks Ulasan: Isi teks ulasan yang diberikan oleh pengguna. Ini adalah komponen utama yang akan dianalisis untuk memahami sentimen dan tema yang diungkapkan oleh pengguna.
2) Rating Bintang: Nilai rating yang diberikan pengguna. Ini memberikan indikasi langsung tentang kepuasan pengguna.
3) Tanggal Ulasan: Tanggal kapan ulasan tersebut diposting. Ini membantu dalam analisis tren ulasan dari waktu ke waktu.
4) Versi Aplikasi: Versi aplikasi Merdeka Mengajar yang digunakan saat ulasan diberikan. Ini membantu memahami apakah ulasan terkait dengan versi tertentu dari aplikasi.

4.2.2. *Data Understanding*

1. Pengumpulan Data(*Scraping*)

```
reviews_all_app = reviews_all(
    'id.belajar.app',
    lang='id', # defaults to 'en'
    country='id', # defaults to 'us'
    filter_score_with=None # defaults to None (means all score)))
# Mengubah data review menjadi DataFrame pandas
df_reviews_all = pd.DataFrame(reviews_all_app)
```

Gambar 4.5 Sintak program *scraping data*
Sumber: Penelitian Sendiri

Sintak *Python* pada Gambar 4.5 mengambil ulasan pengguna aplikasi dari Merdeka Mengajar dari *Google Play Store* menggunakan fungsi `*reviews_all*` dengan parameter bahasa Indonesia dan negara Indonesia. Semua ulasan diambil tanpa filter. Kemudian data ulasan kemudian diubah menjadi *DataFrame Pandas* untuk memudahkan analisis lebih lanjut. Manfaatnya adalah mempermudah pengambilan dan pengolahan ulasan pengguna untuk analisis sentimen dan *clustering*. Serta, dapat ditampilkan dengan sintak "*df_reviews_all.head()*".

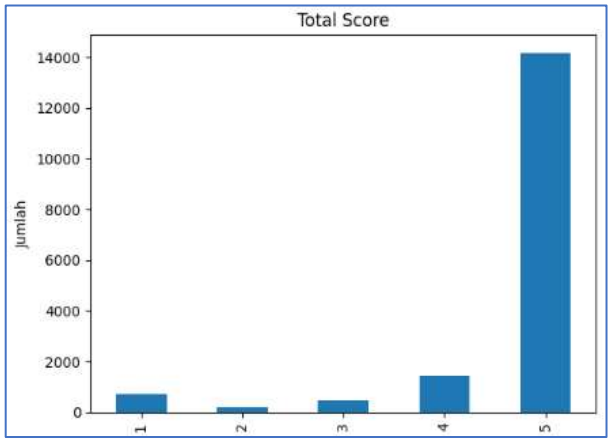
2. Analisis Univariat

Analisis univariat dalam penelitian ini bertujuan untuk memahami distribusi banyaknya rating ulasan dan banyaknya ulasan diberikan di *google play store* per-4 bulan nya seperti dibawah ini :

```
0 | df_reviews_all['score'] = df_reviews_all['score'].astype(int)
1 | score_counts = df_reviews_all['score'].value_counts().sort_index()
2 | score_counts.plot(kind='bar')
3 | plt.title('Total Score')
4 | plt.xlabel('Skor')
5 | plt.ylabel('Jumlah')
6 | plt.xticks(ticks=[0, 1, 2, 3, 4], labels=[1, 2, 3, 4, 5])  # Ensure x-
ticks are labeled correctly
7 | plt.show()
```

Gambar 4.6 Sintak Program banyaknya rating ulasan berdasarkan komentar
Sumber: Penelitian Sendiri

Sintak Program di atas mengubah kolom `*score*` di *DataFrame* `*df_reviews_all*` menjadi tipe *integer*, menghitung frekuensi setiap nilai skor, dan membuat grafik batang (*bar chart*) untuk menunjukkan distribusi total skor ulasan. Sintak program ini berguna untuk membantu memahami distribusi skor ulasan pengguna aplikasi Merdeka Mengajar, memudahkan identifikasi tren dan pola dalam skor ulasan, serta menyediakan visualisasi yang jelas untuk analisis lebih lanjut.

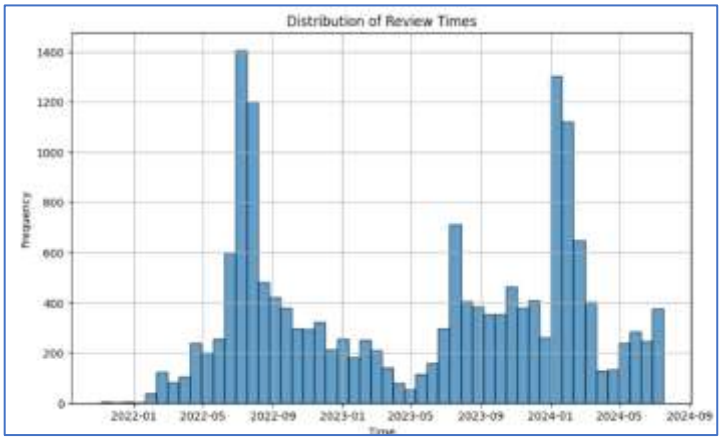


Gambar 4.7 Sintak Program untuk banyaknya rating ulasan berdasarkan komentar
Sumber: Penelitian Sendiri

```
0 | #@title Distribution of 'at' (review times)
1 | plt.figure(figsize=(10, 6))
2 | df_reviews_all['at'].hist(bins=50, edgecolor='k', alpha=0.7)
3 | plt.title('Distribution of Review Times')
4 | plt.xlabel('Time')
5 | plt.ylabel('Frequency')
6 | plt.grid(True)
7 | plt.show()
```

Gambar 4.8 Sintak Program untuk banyaknya ulasan berdasarkan waktu
Sumber: Penelitian Sendiri

Sintak program diatas yaitu membuat histogram untuk mendistribusikan waktu ulasan (*at*) di *DataFrame df_reviews_all*. Sintak program di atas itu juga untuk memahami pola waktu ketika ulasan diberikan, memudahkan identifikasi tren temporal dalam ulasan pengguna aplikasi Merdeka Mengajar, serta menyediakan visualisasi yang jelas untuk analisis lebih lanjut.



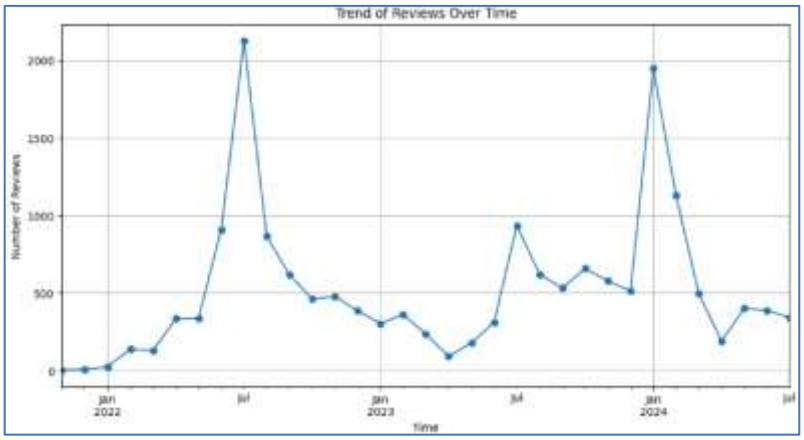
Gambar 4.9 Sintak Program untuk banyaknya ulasan berdasarkan waktu
Sumber: Penelitian Sendiri

3. Analisis Multivariat

```
0 | # @title Trend Ulasan dari Waktu ke Waktu
1 | plt.figure(figsize=(12, 6))
2 | df_reviews_all.set_index('at').resample('M').size().plot(marker='o')
3 | plt.title('Trend of Reviews Over Time')
4 | plt.xlabel('Time')
5 | plt.ylabel('Number of Reviews')
6 | plt.grid(True)
7 | plt.show()
```

Gambar 4.10 Sintak Program Korelasi antara variabel numerik
Sumber: Penelitian Sendiri

Skrip ini membuat plot tren ulasan dari waktu ke waktu dengan menggunakan *DataFrame df_reviews_all*. Manfaatnya adalah untuk memahami tren ulasan, yang bisa membantu melihat periode-periode ketika aplikasi mendapatkan lebih banyak atau lebih sedikit ulasan.

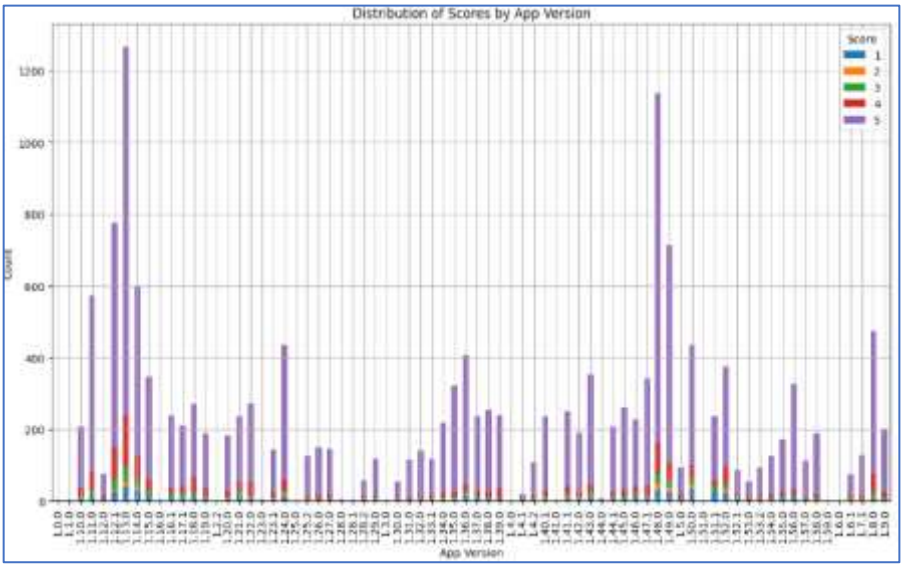


Gambar 4.11 Tampilan Korelasi antara variabel numerik
Sumber: Penelitian Sendiri

```
0 | app_version_scores          =          df_reviews_all.groupby(['appVersion',
'score']).size().unstack().fillna(0)
2 | # Plot the count plot using matplotlib
3 | plt.figure(figsize=(14, 8))
5 | # Create the bar plots
6 | app_version_scores.plot(kind='bar',stacked=True,ax=plt.gca())
8 | # Set the title and labels
9 | plt.title('Distribution of Scores by App Version')
10| plt.xlabel('App Version')
11| plt.ylabel('Count')
13| # Rotate x-axis labels
14| plt.xticks(rotation=90)
16| # Add legend and grid
17| plt.legend(title='Score')
18| plt.grid(True)
20| # Show plot
21| plt.show()
```

Gambar 4.12 Sintak Program Distribusi Score menurut Versi Aplikasi
Sumber: Penelitian Sendiri

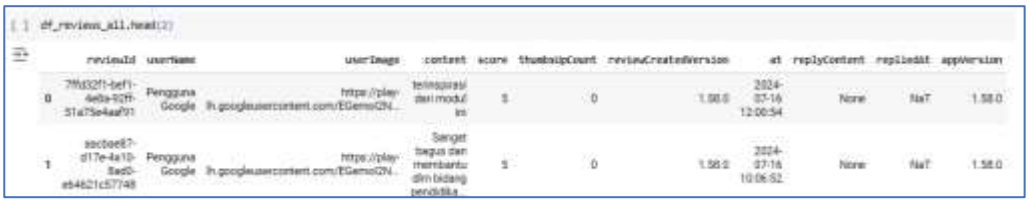
Sintak program di atas yaitu membuat grafik batang bertumpuk untuk menunjukkan distribusi skor ulasan berdasarkan versi aplikasi. Manfaatnya adalah untuk memahami bagaimana skor ulasan bervariasi antara versi aplikasi yang berbeda, membantu mengidentifikasi versi aplikasi mana yang mendapatkan ulasan lebih baik atau lebih buruk.



Gambar 4.13 Tampilan Distribusi *Score* menurut Versi Aplikasi
Sumber: Penelitian Sendiri

4.2.3. *Data Preparation*

1. *Data Collection*



Gambar 4.14 Sintak Program dan Tampilan dari proses *Data Collection*
Sumber: Penelitian Sendiri

Pada tahap *data collection*, sintak program *df_reviews_all.head*(2) menampilkan dua baris pertama dari *DataFrame df_reviews_all*. Manfaatnya adalah untuk memberikan gambaran awal tentang struktur dan isi data dalam

DataFrame untuk memeriksa kolom yang ada, tipe data, dan contoh data yang akan dianalisis lebih lanjut. Berikut sintak dan tampilan dari *drop data* dan *subset data* :

```
0 | # @title drop Data
1 | df_reviews_all_proses = df_reviews_all.copy()
2 | df_reviews_all_proses.drop(columns=['userName','userImage','replyConten
t','repliedAt','reviewCreatedVersion','thumbsUpCount','replyContent','repli
edAt','appVersion','at'], inplace=True)
3 | df_reviews_all_proses   =   df_reviews_all_proses.loc[:,   ['reviewId',
'score', 'content']]
4 | df_reviews_all_proses.head(5)
```

Gambar 4.15 Sintak Program dari *Drop Data*
Sumber: Penelitian Sendiri

Sintak program di atas menyalin *DataFrame df_reviews_all* ke dalam *df_reviews_all_proses*, menghapus kolom yang tidak diperlukan, dan mengubah urutan kolom sesuai keinginan. Tujuannya adalah menyederhanakan *DataFrame* dengan hanya mempertahankan kolom yang relevan, sehingga memudahkan analisis, mengurangi kebisingan data, serta memudahkan pemeriksaan data dengan menampilkan lima baris pertama.

	reviewId	score	content
0	184debdc-3e65-473b-b055-2f57ef56e0bd	5	sangat membantu
1	c72624e4-6615-42e9-9526-a60a75e16788	5	sangat membantu
2	4dbf304b-c8ab-43d2-b700-f375bd7a2cef	5	semangat
3	6927c459-997e-48d8-b514-e47e6a89b705	3	Sangat membantu saya sebagai seorang guru
4	9a86e5aa-bbbd-42d4-9e59-9cc9eb52f77b	5	bagus

Gambar 4.16 Tampilan setelah lokasi *filter* pada *DataFrame* diubah
Sumber: Penelitian Sendiri

2. *Data Cleaning*

1) *Case Folding*

```
1 | import re
2 | import string
5 | emoji_dict = {
6 | "\U0001F600": "senang",  # Grinning Face
7 | "\U0001F601": "sangat senang",  # Grinning Face with Smiling Eyes
9 |     "\U0001F97A": ""  # Folded Hands }
13| def replace_emojis_with_meanings(text):
14|     emoji_pattern = re.compile("|".join(map(re.escape, emoji_dict.keys())))
15|     text_with_meanings = emoji_pattern.sub(lambda match: f"
{emoji_dict.get(match.group(), '')} ", text)
16|     return re.sub(r'[^\w\s,.?!]', '', text_with_meanings)
19| def remove_html_tags(text):
20|     return re.sub('<.*?>', '', text)
23| def hapus_simbol(teks):
24|     return teks.translate(str.maketrans('', '', string.punctuation))
```

```
27| def remove_urls(text):
28|     return re.sub(r'https?://\S+|www\.\S+', '', text)
31| def angka_ke_huruf(angka):
32|     satuan = ["", "satu", "dua", "tiga", "empat", "lima", "enam", "tujuh",
"delapan", "sembilan", "sepuluh", "sebelas"]
33|     if angka < 12:
34|         return satuan[angka]
35|     elif angka < 20:
36|         return satuan[angka - 10] + " belas"
37|     elif angka < 100:
38|         return satuan[angka // 10] + " puluh" + (" " + satuan[angka % 10] if
(angka % 10 != 0) else "")
39|     elif angka < 200:
40|         return "seratus" + (" " + angka_ke_huruf(angka - 100) if (angka > 100)
else "")
41|     elif angka < 1000:
42|         return satuan[angka // 100] + " ratus" + (" " + angka_ke_huruf(angka %
100) if (angka % 100 != 0) else "")
43|     elif angka < 2000:
44|         return "seribu" + (" " + angka_ke_huruf(angka - 1000) if (angka > 1000)
else "")
45|     elif angka < 1000000:
46|         return angka_ke_huruf(angka // 1000) + " ribu" + (" " +
angka_ke_huruf(angka % 1000) if (angka % 1000 != 0) else "")
47|     elif angka < 1000000000:
48|         return angka_ke_huruf(angka // 1000000) + " juta" + (" " +
angka_ke_huruf(angka % 1000000) if (angka % 1000000 != 0) else "")
49|     else:
50|         return "Angka terlalu besar"
52| def remove_pattern(text):
53|     return re.sub(r'\b\d+\b', lambda match: angka_ke_huruf(int(match.group())),
text)
56| def clean_text(text):
57|     text = text.lower()
58|     text = replace_emojis_with_meanings(text)
59|     text = remove_urls(text)
60|     text = remove_html_tags(text)
61|     text = hapus_simbol(text)
62|     text = remove_pattern(text)
63|     return text
66| df_reviews_all_proses['content_cleaning'] =
df_reviews_all_proses['content'].apply(clean_text)
67| df_reviews_all_proses.head(20)
```

Gambar 4.17 Sintak untuk mengubah seluruh kalimat menjadi huruf kecil
Sumber: Penelitian Sendiri

Sintak program di atas menambahkan kolom baru dalam *DataFrame* `*df_reviews_all_proses*` yang berisi teks ulasan (`*content*`) yang diubah menjadi huruf kecil. Selanjutnya, sintak ini menggantikan emotikon dengan arti yang sesuai menggunakan kamus *emoji*, membersihkan emotikon yang tidak dikenal, serta menghapus *tag HTML*, simbol, dan *URL* dari teks. Hasilnya disimpan dalam kolom `*content_cleaning*`. Tujuannya adalah menyamakan format teks ulasan dan meningkatkan kualitas data teks agar lebih bersih dan mudah dianalisis.

	reviewId	score	content	content_cleaning
0	dbf009ad-acef-430b-923b-c6b53a6fa3bd	5	Membantu terkait pembelajaran dan administrasi...	membantu terkait pembelajaran dan administrasi...
1	184aef51-fbb1-4f1f-8bd9-618e8c65d0bd	4	sangat bagus dan membantu	sangat bagus dan membantu
2	24489d71-53fa-45b1-9a87-df6d2bfb7acc	4	terbaik	terbaik
3	e3e58eb4-0eb4-4b36-9161-3607bdeee62b	5	sangat Membagun	sangat membagun
4	2ded9200-736f-4afd-a10b-37b93cf80e6b	5	bagus	bagus
5	dd2e8bcb-86cb-4b01-9ef2-f5f4479b2e4a	5	Sangat menginspirasi	sangat menginspirasi

Gambar 4.18 Tampilan hasil *Case Folding*
Sumber: Penelitian Sendiri

3. *Text Preprocessing*

1) *Normalization*

1. Membuat kamus berdasarkan ulasan

Sintak program di bawah menghitung frekuensi kemunculan kata dalam teks ulasan yang sudah dibersihkan dan menyusun *DataFrame* dengan kata-kata berfrekuensi tertinggi. Selain itu, sintak ini mengganti kata tidak baku dalam teks ulasan dengan kata baku yang sesuai dari kamus yang diunggah secara *online*, serta mengidentifikasi kata-kata tidak baku yang diganti. Tujuannya adalah mengidentifikasi kata-kata paling sering muncul dan memastikan penggunaan bahasa baku, sehingga meningkatkan kualitas teks ulasan untuk analisis lebih lanjut.

```
2  | import pandas as pd
3  | import re
4  | from collections import Counter
7  | text = " ".join(df_reviews_all_proses['content_cleaning'])
8  | tokens = text.split()
11| word_counts = Counter(tokens)
14| top_words = word_counts.most_common(25000)
15| data_kata = pd.DataFrame(top_words, columns=['kata', 'count'])
18| def replace_taboo_words(text, kamus_tidak_baku):
19|     if not isinstance(text, str):
20|         return '', [], [], []
22|     words = text.split()
23|     replaced_words = []
24|     kalimat_baku = []
25|     kata_diganti = []
26|     kata_tidak_baku_hash = []
28|     for word in words:
29|         baku_word = kamus_tidak_baku.get(word, word)
```

```
30|          if isinstance(baku_word, str) and all(char.isalpha() for char
in baku_word):
31|              replaced_words.append(baku_word)
32|              kalimat_baku.append(baku_word)
33|              kata_diganti.append(word)
34|              kata_tidak_baku_hash.append(hash(word))
35|          else:
36|              replaced_words.append(word)
38|      replaced_text = ' '.join(replaced_words)
39|              return     replaced_text,     kalimat_baku,     kata_diganti,
kata_tidak_baku_hash
42| kamus_url                                                                =
"https://github.com/nsyamsu23/kamus/raw/main/kamuskatabaku.xlsx"
43| kamus_data = pd.read_excel(kamus_url)
44| kamus_tidak_baku          =          pd.Series(kamus_data["kata_baku"],
index=kamus_data["tidak_baku"]).to_dict()
47| df_reviews_all_proses['content_cleaning_normalized']                    =
df_reviews_all_proses['content_cleaning'].apply(lambda                    x:
replace_taboo_words(x, kamus_tidak_baku)[0])
49| df_reviews_all_proses.head()
```

Gambar 4.19 Sintak Membuat kamus dari ulasan
Sumber: Penelitian Sendiri

	reviewId	score	content	content_cleaning	content_cleaning_normalized
0	dbf009ad-acef-430b-923b-c6b53a6fa3bd	5	Membantu terkait pembelajaran dan administrasi	membantu terkait pembelajaran dan administrasi	membantu terkait pembelajaran dan administrasi
1	184ae51-8b1-4f1-8bd9-618e8c65d0bd	4	sangat bagus dan membantu	sangat bagus dan membantu	sangat bagus dan membantu
2	24489d71-53fa-45b1-9a87-df6d2bfb7acc	4	terbaik	terbaik	terbaik
3	e3e58eb4-0eb4-4b36-9161-3607bdeee62b	5	sangat Membagun	sangat membagun	sangat membagun
4	2ded5200-736f-4afd-a10b-37b93cf80e6b	5	bagus	bagus	bagus

Gambar 4.20 Tampilan hasil Normalisasi Kata
Sumber: Penelitian Sendiri

2. Normalisasi Kata, Ekspansi Singkatan, dan Penghapusan *Slangword* dan *Stopword*

Sintak ini digunakan untuk memproses teks dalam data ulasan. Sintak mencakup langkah-langkah untuk mengganti kata tidak baku dengan kata baku, memperpanjang kata-kata yang disingkat, dan menghapus kata henti dari teks. Langkah-langkah ini membantu meningkatkan kualitas data teks untuk analisis lebih lanjut.

```
1 | import pandas as pd
2 | import re
3 | from collections import Counter
4 | from NLTK.corpus import stopwords
5 | import NLTK
6 | from NLP-ID.stopword import StopWord
8 | NLTK.download('stopwords')
11| chat_words_mapping = {
12|     'bagu': 'bagus',
13|     'yg': 'yang',
14|      ..............................
15|      "wamil": "wajib Militer"
16|     # tambahkan lainnya sesuai kebutuhan}
```

```
20| def expand_chat_words(text, chat_words_mapping):
21|     return ' '.join([chat_words_mapping.get(word, word) for word in
text.split()])
24| kamus_url                                                          =
"https://github.com/nsyamsu23/kamus/raw/main/kamuskatabaku.xlsx"
25| kamus_data = pd.read_excel(kamus_url)
26| kamus_tidak_baku          =          pd.Series(kamus_data["kata_baku"],
index=kamus_data["tidak_baku"]).to_dict()
29| indonesia_s = stopwords.words('indonesian')
30| stopword_NLP-ID = StopWord().get_stopword()
31| stop_words  =  set(indonesia_s  +  stopword_NLP-ID  +  ["pmm",  "merdeka
mengajar", "nya"])
33| def remove_stop_words(text):
34|     if not isinstance(text, str):
35|         text = str(text)
36|     words = text.split()
37|     return ' '.join([word for word in words if word.lower() not in
stop_words])
40| df_reviews_all_proses['content_cleaning']                           =
df_reviews_all_proses['content_cleaning'].astype(str)
43| df_reviews_all_proses['content_cleaning_normalized'] = (
44|     df_reviews_all_proses['content_cleaning']
45|     .apply(lambda x: expand_chat_words(x, chat_words_mapping))
46|     .apply(remove_stop_words))
50| df_reviews_all_proses                                              =
df_reviews_all_proses[df_reviews_all_proses['content_cleaning_normalized'].
str.strip() != '']
52| df_reviews_all_proses.head()
```

Gambar 4.21 Sintak Normalisasi, Ekspansi Singkatan, Slangword dan *Stopword Removal*
Sumber: Penelitian Sendiri

	reviewId	score	content	content_cleaning	content_cleaning_normalized
0	db6009ad-acef-430b-923b-c6b53a6fa3bd	5	Membantu terkait pembelajaran dan administrasi	membantu terkait pembelajaran dan administrasi	membantu terkait pembelajaran administrasi guru
1	184aef51-8b1-4f1f-8bd9-618e8c65d0bd	4	sangat bagus dan membantu	sangat bagus dan membantu	bagus membantu
2	24489d71-53fa-45b1-9a87-df6d2bfb7acc	4	terbaik	terbaik	terbaik
3	e3e58eb4-0eb4-4b36-9161-3607bdeee62b	5	sangat Membagun	sangat membagun	membagun
4	2dad9200-736f-4afd-a10b-37b93cf80e6b	5	bagus	bagus	bagus

Gambar 4.22 Hasil Normalisasi, Ekspansi Singkatan, Slangword dan *Stopword Removal*
Sumber: Penelitian Sendiri

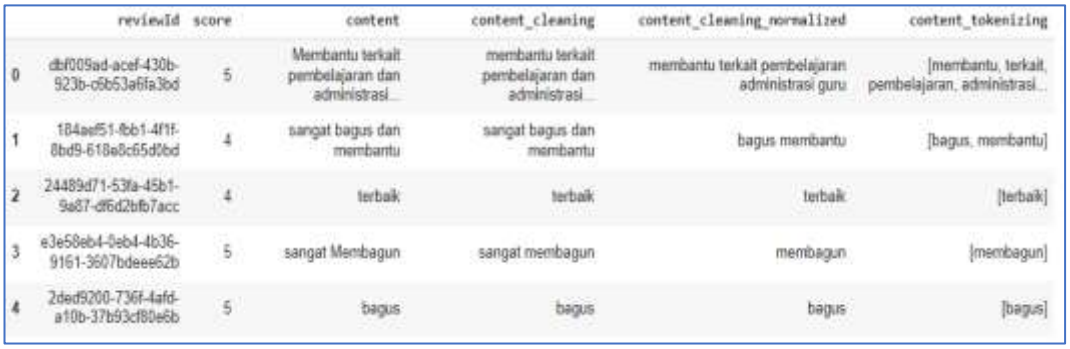
2) *Tokenizing*

```
0 | from NLP-ID.tokenizer import Tokenizer
1 | tokenizer = Tokenizer()
3 | def tokenizing_words(text):
4 | tokens = tokenizer.tokenize(remove_stop_words_NLP-ID (text))
5 | return tokens
7 | df_reviews_all_proses['content_tokenizing']  =
df_reviews_all_proses['content_cleaning_normalized']
.apply(tokenizing_words)
8 | df_reviews_all_proses.head(5)
```

Gambar 4.23 Sintak program dari *Tokenizing*
Sumber: Penelitian Sendiri

Skrip ini bertujuan untuk memberikan token pada setiap kata dalam data ulasan. Dengan menggunakan perpustakaan *NLTK* dan *NLP-ID*, skrip ini

mengunduh *tokenizer* bahasa Indonesia dan melakukan tokenisasi pada setiap kata dalam teks yang dibersihkan. Hasil akhir adalah *DataFrame* yang berisi teks yang telah di-tokenisasi untuk analisis lebih lanjut.



	reviewId	score	content	content_cleaning	content_cleaning_normalized	content_tokenizing
0	dbf009ad-acef-430b-923b-c9b53a6fa3bd	5	Membantu terkait pembelajaran dan administrasi...	membantu terkait pembelajaran dan administrasi...	membantu terkait pembelajaran administrasi guru	[membantu, terkait, pembelajaran, administrasi...
1	184aef51-fbb1-4f1f-8bd9-618e8c65d0bd	4	sangat bagus dan membantu	sangat bagus dan membantu	bagus membantu	[bagus, membantu]
2	24489d71-53fa-45b1-9a87-df6d2bfb7acc	4	terbaik	terbaik	terbaik	[terbaik]
3	e3e58eb4-0eb4-4b36-9161-3607bdeee62b	5	sangat Membagun	sangat membagun	membagun	[membagun]
4	2ded9200-736f-4afd-a10b-37b93cf80e6b	5	bagus	bagus	bagus	[bagus]

Gambar 4.24 Tampilan dari hasil *Tokenizing*
Sumber: Penelitian Sendiri

3) *Part of Speech*

Skrip ini digunakan untuk melakukan *tagging Part-Of-Speech (POS)* pada teks ulasan, menghapus *tag POS* yang tidak diinginkan, dan mengubah teks tersebut menjadi token. Dengan menggunakan *library NLP-ID*, skrip ini pertama-tama menandai setiap kata dalam teks dengan *tag POS*, kemudian menghapus *tag-tag* yang tidak relevan, dan akhirnya mengubah hasilnya menjadi token yang dapat digunakan untuk analisis lebih lanjut.

```
1 | import pandas as pd
2 | import NLTK
3 | from NLP-ID.postag import PosTag
4 | from NLP-ID.tokenizer import Tokenizer
6 | NLTK.download('punkt')
7 | postagger = PosTag()
8 | tokenizer = Tokenizer()
9 | def pos_words(text):
10|     return postagger.get_pos_tag(text)
11| unwanted_tags = {"PR", "RP", "UH", "SC", "SYM", "IN", "DT", "CC",
"FW"}
13| def remove_unwanted_pos_tags(pos_list):
14|     return [(word, tag) for word, tag in pos_list if tag not in
unwanted_tags]
16| def pos_to_tokens(pos_list):
17|     sentence = ' '.join([word.lower() for word, tag in pos_list])
18|     tokens = tokenizer.tokenize(sentence)
19|     return tokens
20| df_reviews_all_proses['content_part_of_speech'] =
df_reviews_all_proses['content_cleaning_normalized'].apply(pos_words)
```

```
21| df_reviews_all_proses['content_part_of_speech'] =
df_reviews_all_proses['content_part_of_speech'].apply(remove_unwanted_p
os_tags)
22| df_reviews_all_proses['content_tokenizing'] =
df_reviews_all_proses['content_part_of_speech'].apply(pos_to_tokens)
23| df_reviews_all_proses.head(5)
```

Gambar 4.25 Sintak program dari *Part-of-Speech*
Sumber: Penelitian Sendiri

	reviewId	score	content	content_cleaning	content_cleaning_normalized	content_tokenizing	content_part_of_speech
0	dbf009ad-acef-430b-923b-c5b53a6fa3bd	5	Membantu terkait pembelajaran dan administrasi	membantu terkait pembelajaran dan administrasi	membantu terkait pembelajaran administrasi guru	[membantu, terkait, pembelajaran, administrasi]	[('membantu', 'VB'), ('terkait', 'VB'), ('pembelajaran',
1	184aef51-6b1-4f1f-8bd9-618a8c65d0bd	4	sangat bagus dan membantu	sangat bagus dan membantu	bagus membantu	[bagus, membantu]	[('bagus', 'JJ'), ('membantu', 'VB')]
2	24489d71-53fa-45b1-9a87-df6d2bfb7acc	4	terbaik	terbaik	terbaik	[terbaik]	[('terbaik', 'JJ')]
3	e3e58eb4-0eb4-4b36-9161-3607bdeee62b	5	sangat Membagun	sangat membagun	membagun	[membagun]	[('membagun', 'VB')]
4	2ded9200-736f-4afd-a10b-37b93cf80e6b	5	bagus	bagus	bagus	[bagus]	[('bagus', 'JJ')]

Gambar 4.26 Tampilan dari hasil *Tokenizing*
Sumber: Penelitian Sendiri

4) *Stemming NLP-ID*

Sintak progam diatas adalah untuk melakukan *stemming* pada teks ulasan yang telah di-tokenisasi menggunakan *stemmer* dari *NLP-ID*, mengubah kata ke bentuk dasar dan menghitung jumlah kata yang diubah. Hal itu adalah untuk memudahkan analisis teks dengan menyederhanakan kata-kata ke bentuk dasarnya, sehingga meningkatkan konsistensi data dan memudahkan proses pemrosesan bahasa alami (*NLP*).

```
0 | import pandas as pd
1 | from NLP-ID.stopword import StopWord
2 | from NLP-ID.lemmatizer import Lemmatizer
3 | from tqdm import tqdm
6 | tqdm.pandas()
9 | stopword = StopWord()
10| lemmatizer = Lemmatizer()
13| def lemmatize_wrapper(tokens):
14|     # Lemmatize each token
15|     lemmatized_tokens = [lemmatizer.lemmatize(token) for token in
tokens]
16|     original_vs_lemmatized = list(zip(tokens, lemmatized_tokens))
19|     changed_count = sum(1 for original, lemmatized in
original_vs_lemmatized if original != lemmatized)
20|     return ' '.join(lemmatized_tokens), changed_count  #
Mengembalikan token yang telah di-lemmatize dan jumlah kata yang
diubah
23| total_changed_count = 0
```

```
26| def process_and_count_changes(tokens):
27|     global total_changed_count
28|     lemmatized_tokens, changed_count = lemmatize_wrapper(tokens)
29|     total_changed_count += changed_count
30|     return lemmatized_tokens
32| df_reviews_all_proses['content_proses_stemming_NLP-ID'] =
df_reviews_all_proses['content_tokenizing'].progress_apply(process_and
_count_changes)
33| print(f"Total kata yang diubah: {total_changed_count}")
34| df_reviews_all_proses.head(5)
35|
```

Gambar 4.27 Sintak program dari *Stemming* dengan *NLP-ID*
Sumber: Penelitian Sendiri

Gambar 4.28 Tampilan hasil dari *Stemming* dengan *NLP-ID*
Sumber: Penelitian Sendiri

```
0 | from wordcloud import WordCloud
1 | import matplotlib.pyplot as plt
2 | import pandas as pd
4 | text = '
'.join(df_reviews_all_proses['content_proses_stemming_NLP-
ID'].apply(lambda x: str(x) if isinstance(x, (str, int, float)) else
''))
5 | wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(text)
6 | plt.figure(figsize=(10, 5))
8 | # Menampilkan WordCloud dengan interpolasi gambar bilinear
9 | plt.title("WordCloud Stemming menggunakan NLP-ID", fontsize=18,
fontweight='bold', pad=20)
10| plt.imshow(wordcloud, interpolation='bilinear')
11| plt.axis("off")
12| plt.show()
```

Gambar 4.29 Sintak program dari *wordcloud* dengan *NLP-ID*
Sumber: Penelitian Sendiri

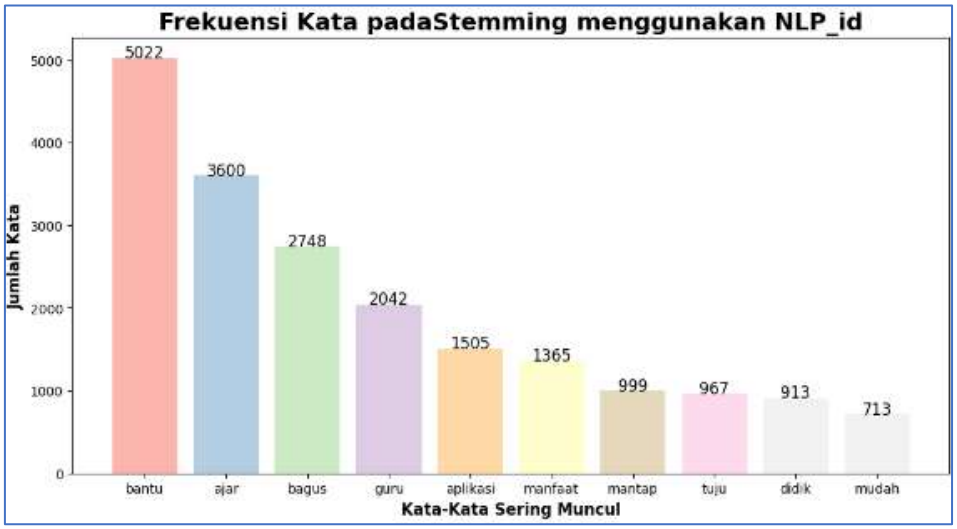
Sintak diatas menghasilkan dan menampilkan *WordCloud* dari teks ulasan yang telah di-*lemmatize* menggunakan *NLP-ID*. Hal itu agar memvisualisasikan kata-kata yang paling sering muncul dalam ulasan, memudahkan identifikasi tema utama dan pola dalam teks ulasan.

Gambar 4.30 Tampilan hasil dari *WordCloud* dengan *NLP-ID*
Sumber: Penelitian Sendiri

```
0 | import matplotlib.pyplot as plt
1 | from collections import Counter
3 | text = "
".join(df_reviews_all_proses['content_proses_stemming_NLP-ID'])
4 | tokens = text.split()
5 | word_counts = Counter(tokens)
6 | top_words = word_counts.most_common(10)
7 | word, count = zip(*top_words)
9 | # Definisikan palet warna dengan warna yang lebih lembut
10| colors = plt.cm.Pastel1(range(len(word)))
12| plt.figure(figsize=(12, 6))
13| bars = plt.bar(word, count, color=colors)
14| plt.xlabel("Kata-Kata Sering Muncul", fontsize=12,
fontweight='bold')
15| plt.ylabel("Jumlah Kata", fontsize=12, fontweight='bold')
16| plt.title("Frekuensi Kata padaStemming menggunakan NLP-ID",
fontsize=18, fontweight='bold')
17| plt.xticks(rotation=0)
20| for bar, num in zip(bars, count):
21|     plt.text(bar.get_x() + bar.get_width() / 2, num + 0.3,
str(num), fontsize=12, color='black', ha='center')
22| plt.show()
```

Gambar 4.31 Sintak program dari Frekuensi kata-kata dengan *NLP-ID*
Sumber: Penelitian Sendiri

Sintak Progam diatas membuat diagram batang untuk menunjukkan frekuensi 10 kata yang paling sering muncul setelah lemmatisasi menggunakan *NLP-ID*. hal tersebut untuk memvisualisasikan kata-kata yang paling sering muncul dalam ulasan, membantu mengidentifikasi tema utama dan pola dalam teks ulasan.



Gambar 4.32 Tampilan hasil dari Frekuensi kata-kata dengan *NLP-ID*
Sumber: Penelitian Sendiri

4. Pembobotan Kata

1) *BoW*, *TF IDF, Word2vec* menggunakan metode *Silhouette Scores*

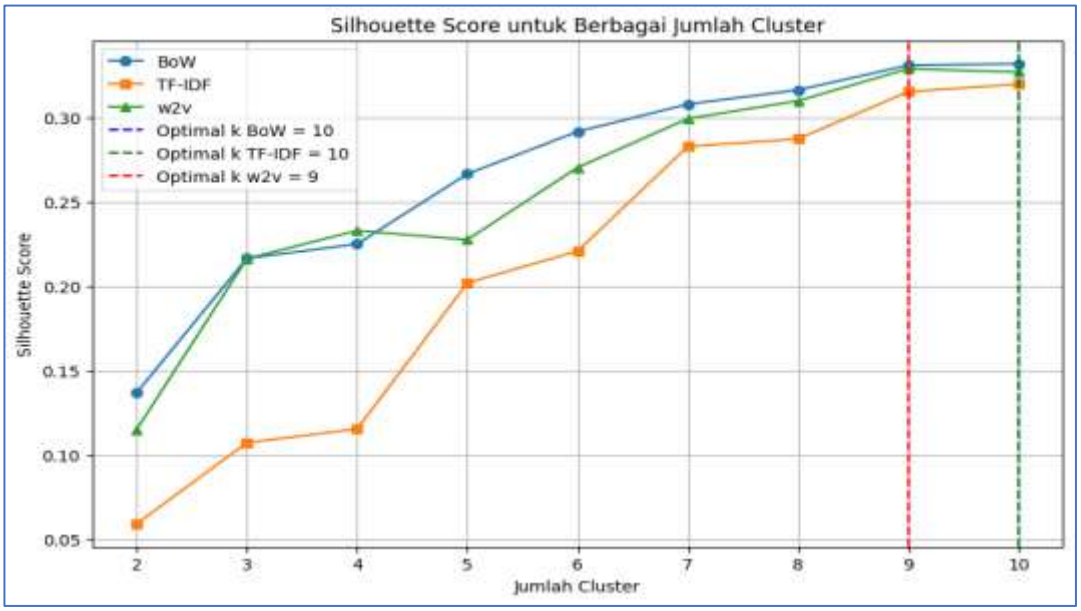
Sintak ini melakukan analisis teks pada ulasan dengan menggunakan tiga teknik: *Bag of Words (BoW), TF-IDF*, dan *Word2Vec*. Setelah melakukan tokenisasi teks, skrip ini mengaplikasikan ketiga teknik tersebut untuk mendapatkan representasi fitur teks dari kolom *content_proses_stemming_NLP-ID*. Selanjutnya, menggunakan metode *Silhouette*, sintak ini menentukan jumlah *cluster* optimal untuk setiap teknik, menambahkan garis vertikal pada titik optimal, dan memvisualisasikan hasilnya dalam satu plot komparatif. Hasil *Silhouette Scores* untuk setiap jumlah *cluster* juga ditampilkan dalam bentuk tabel.

```
# Menginisialisasi objek yang diperlukan
range_n_clusters = range(2, 11)

# Bag of Words (BoW)
vectorizer_bow = CountVectorizer()
X_bow =
vectorizer_bow.fit_transform(df_reviews_all_proses['content_proses_stemming_NLP-ID'])
sums_bow = X_bow.sum(axis=0)
terms_bow = vectorizer_bow.get_feature_names_out()
data_bow = [(term, sums_bow[0, col]) for col, term in enumerate(terms_bow)]
```

```
ranking_bow = pd.DataFrame(data_bow, columns=['term', 'rank']).sort_values('rank', ascending=False)
X_normalized_bow = normalize(X_bow)

# TF-IDF
vectorizer_tfidf = TfidfVectorizer(max_features=1000)
X_tfidf = vectorizer_tfidf.fit_transform(df_reviews_all_proses['content_proses_stemming_NLP-ID'])
sums_tfidf = X_tfidf.sum(axis=0)
terms_tfidf = vectorizer_tfidf.get_feature_names_out()
data_tfidf = [(term, sums_tfidf[0, col]) for col, term in enumerate(terms_tfidf)]
ranking_tfidf = pd.DataFrame(data_tfidf, columns=['term', 'rank']).sort_values('rank', ascending=False)
X_normalized_tfidf = normalize(X_tfidf)

# Word2Vec
keras_tokenizer = KerasTokenizer()
keras_tokenizer.fit_on_texts(df_reviews_all_proses['content_proses_stemming_NLP-ID'])
sequences = keras_tokenizer.texts_to_sequences(df_reviews_all_proses['content_proses_stemming_NLP-ID'])
word_index = keras_tokenizer.word_index
data = pad_sequences(sequences, padding='post')
vocab_size = len(word_index) + 1
embedding_dim = 100
max_length = data.shape[1]

# Latih model Word2Vec dengan TensorFlow
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(input_dim=vocab_size, output_dim=embedding_dim, input_length=max_length),
    tf.keras.layers.GlobalAveragePooling1D(),
    tf.keras.layers.Dense(embedding_dim, activation='relu')
])
model.compile(optimizer='adam', loss='mse')  # Mengubah loss function untuk menghindari ketidaksesuaian dimensi
model.summary()

# Latih model
model.fit(data, np.zeros((data.shape[0], embedding_dim)), epochs=10, verbose=0)  # Mengubah target agar sesuai dengan output

# Ekstrak embedding
embedding_layer = model.layers[0]
embedding_matrix = embedding_layer.get_weights()[0]

# Menghubungkan kata dengan frekuensinya
word_freq = keras_tokenizer.word_counts
data_w2v = [(word, freq) for word, freq in word_freq.items()]
ranking_w2v = pd.DataFrame(data_w2v, columns=['term', 'rank']).sort_values('rank', ascending=False)

# Representasi dokumen sebagai rata-rata vektor kata
def document_vector(doc):
    doc = [embedding_matrix[word_index[word]] for word in doc.split() if word in word_index]
    return np.mean(doc, axis=0) if len(doc) > 0 else np.zeros(embedding_dim)

X_w2v = np.array([document_vector(doc) for doc in df_reviews_all_proses['content_proses_stemming_NLP-ID']])
X_normalized_w2v = normalize(X_w2v)

# Menentukan k yang optimal menggunakan Silhouette Score
def optimal_k(X, method_name):
    silhouette_scores = []
    for n_clusters in range_n_clusters:
        clusterer = KMeans(n_clusters=n_clusters, random_state=10)
        cluster_labels = clusterer.fit_predict(X)
```

```
69|          silhouette_avg = silhouette_score(X, cluster_labels)
70|          silhouette_scores.append(silhouette_avg)
71|      optimal_k_value =
range_n_clusters[silhouette_scores.index(max(silhouette_scores))]
72|      print(f'Jumlah cluster optimal {method_name}: {optimal_k_value}')
73|      return optimal_k_value, silhouette_scores
74|
75| optimal_k_bow, silhouette_scores_bow = optimal_k(X_normalized_bow, "BoW")
76| optimal_k_tfidf, silhouette_scores_tfidf = optimal_k(X_normalized_tfidf, "TF-IDF")
77| optimal_k_w2v, silhouette_scores_w2v = optimal_k(X_normalized_w2v, "w2v")
78|
79| # Buat tabel Silhouette Scores
80| silhouette_scores_table = pd.DataFrame({
81|     'Jumlah Cluster': range_n_clusters,
82|     'BoW': silhouette_scores_bow,
83|     'TF-IDF': silhouette_scores_tfidf,
84|     'w2v': silhouette_scores_w2v
85| })
86|
87| # Plot Silhouette Scores untuk semua metode
88| plt.figure(figsize=(10, 6))
89| plt.plot(range n clusters, silhouette scores bow, marker='o', label='BoW')
90| plt.plot(range_n_clusters, silhouette_scores_tfidf, marker='s', label='TF-IDF')
91| plt.plot(range_n_clusters, silhouette_scores_w2v, marker='^', label='w2v')
92| plt.axvline(x=optimal_k_bow, color='blue', linestyle='--', label=f'Optimal k BoW =
{optimal_k_bow}')
93| plt.axvline(x=optimal_k_tfidf, color='green', linestyle='--', label=f'Optimal k
TF-IDF = {optimal_k_tfidf}')
94| plt.axvline(x=optimal_k_w2v, color='red', linestyle='--', label=f'Optimal k w2v =
{optimal_k_w2v}')
95| plt.title('Silhouette Score untuk Berbagai Jumlah Cluster')
96| plt.xlabel('Jumlah Cluster')
97| plt.ylabel('Silhouette Score')
98| plt.legend()
99| plt.grid(True)
100| plt.show()
```

Gambar 4.33 Sintak Pembobotan kata dengan *BoW, TF-IDF*, dan *Word2Vec* Menggunakan *Silhouette Score*
Sumber: Penelitian Sendiri



Gambar 4.34 Grafik Pembobotan Kata dengan *Silhouette Score*
Sumber: Penelitian Sendiri

Grafik *Silhouette Scores* menunjukkan bahwa jumlah *cluster* optimal untuk metode *Bag of Words (BoW)* dan *TF-IDF* adalah 10, sementara untuk metode *Word2Vec* adalah 9. Skor *Silhouette*, yang mengukur kualitas *clustering*, mencapai nilai tertinggi pada jumlah cluster tersebut, mengindikasikan bahwa *BoW* dan *TF-IDF* akan mengelompokkan data dengan baik jika dibagi menjadi 10 *cluster*, sedangkan *Word2Vec* akan efektif dengan 9 *cluster*. Hasil ini menunjukkan bahwa meskipun *BoW* dan *TF-IDF* memberikan hasil yang sangat mirip dalam *clustering*, *Word2Vec* menunjukkan perbedaan dengan sedikit lebih rendah jumlah *cluster* optimal.

Jumlah Cluster	BoW	TF-IDF	w2v
2	0.136774	0.059158	0.115027
3	0.216474	0.107175	0.216285
4	0.224953	0.115271	0.232870
5	0.266530	0.201816	0.227523
6	0.291557	0.220778	0.270282
7	0.307827	0.282772	0.299238
8	0.316272	0.287244	0.309833
9	0.330895	0.315276	0.328820
10	0.331826	0.319715	0.326873

Gambar 4.35 Tabel performansi terhadap 3 metode pembobotan kata
Sumber: Penelitian Sendiri

Tabel di atas menunjukkan bahwa metode *Bag of Words (BoW)* dan *TF-IDF* mencapai performa *clustering* terbaik pada jumlah *cluster* (k) sebesar 10, dengan Skor *Silhouette* tertinggi masing-masing sebesar 0.331826 dan 0.319715. Sementara itu, metode *Word2Vec* mencapai performa terbaiknya pada jumlah cluster (k) sebesar 9, dengan Skor *Silhouette* sebesar 0.328820. Hal ini menunjukkan bahwa, meskipun *BoW* dan *TF-IDF* lebih cocok dengan 10 *cluster*,

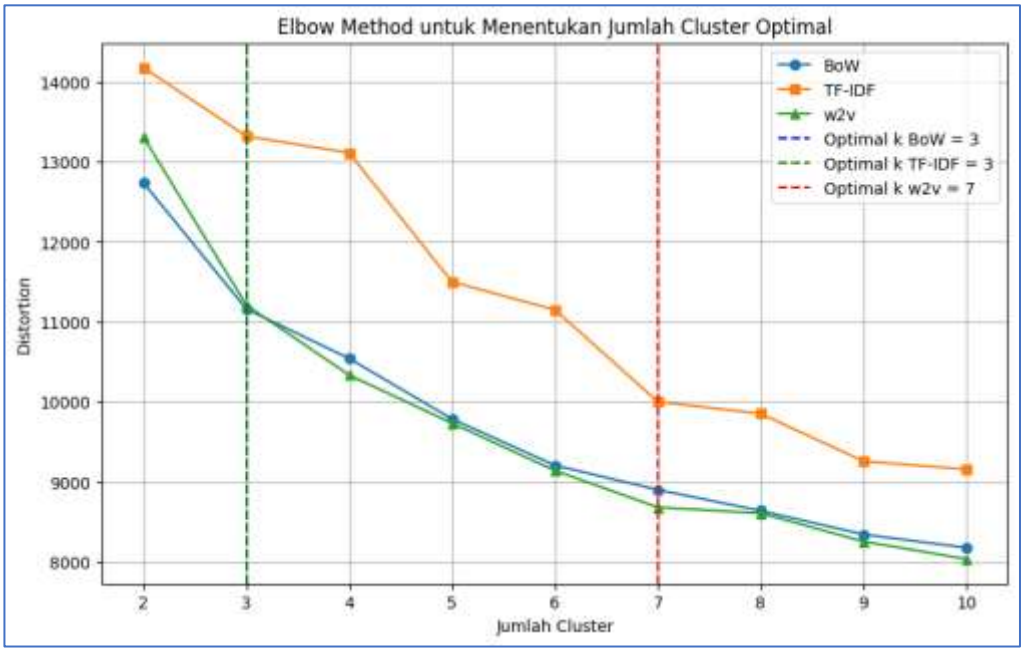
Word2Vec lebih optimal dengan 9 *cluster*. Penurunan Skor *Silhouette* pada k=10 untuk *Word2Vec* mengindikasikan bahwa menambahkan lebih banyak *cluster* tidak meningkatkan kualitas *clustering* untuk metode tersebut. Kesimpulannya, pemilihan jumlah *cluster* yang tepat sangat bergantung pada metode yang digunakan, dan dalam kasus ini, *BoW* dan *TF-IDF* menunjukkan hasil terbaik dengan 10 *cluster*, sedangkan *Word2Vec* dengan 9 *cluster*.

2) *BoW, TF IDF, Word2vec* menggunakan metode *Elbow Method*

```
range_n_clusters = range(2, 11)
vectorizer_bow = CountVectorizer()
X_bow = vectorizer_bow.fit_transform(df_reviews_all_proses['content_proses_stemming_NLP-ID'])
X_normalized_bow = normalize(X_bow)
vectorizer_tfidf = TfidfVectorizer(max_features=1000)
X_tfidf = vectorizer_tfidf.fit_transform(df_reviews_all_proses['content_proses_stemming_NLP-ID'])
X_normalized_tfidf = normalize(X_tfidf)
keras_tokenizer = KerasTokenizer()
keras_tokenizer.fit_on_texts(df_reviews_all_proses['content_proses_stemming_NLP-ID'])
sequences = keras_tokenizer.texts_to_sequences(df_reviews_all_proses['content_proses_stemming_NLP-ID'])
word_index = keras_tokenizer.word_index
data = pad_sequences(sequences, padding='post')
vocab_size = len(word_index) + 1
embedding_dim = 100
max_length = data.shape[1]
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(input_dim=vocab_size, output_dim=embedding_dim, input_length=max_length),
    tf.keras.layers.GlobalAveragePooling1D(),
    tf.keras.layers.Dense(embedding_dim, activation='relu')])
model.compile(optimizer='adam', loss='mse')
model.fit(data, np.zeros((data.shape[0], embedding_dim)), epochs=10, verbose=0)
embedding_layer = model.layers[0]
embedding_matrix = embedding_layer.get_weights()[0]
def document_vector(doc):
    doc = [embedding_matrix[word_index[word]] for word in doc.split() if word in word_index]
    return np.mean(doc, axis=0) if len(doc) > 0 else np.zeros(embedding_dim)
X_w2v = np.array([document_vector(doc) for doc in df_reviews_all_proses['content_proses_stemming_NLP-ID']])
X_normalized_w2v = normalize(X_w2v)
def elbow_method(X):
    distortions = []
    for n_clusters in range_n_clusters:
        kmeans = KMeans(n_clusters=n_clusters, random_state=10)
        kmeans.fit(X)
        distortions.append(kmeans.inertia_)
    return distortions
distortions_bow = elbow_method(X_normalized_bow)
distortions_tfidf = elbow_method(X_normalized_tfidf)
```

```
distortions_w2v = elbow_method(X_normalized_w2v)
def find_optimal_k(distortions):
    deltas = np.diff(distortions)
    second_deltas = np.diff(deltas)
    optimal_k = np.argmin(second_deltas) + 2  # Karena range_n_clusters mulai dari 2
    return optimal_k
optimal_k_bow = find_optimal_k(distortions_bow)
optimal_k_tfidf = find_optimal_k(distortions_tfidf)
optimal_k_w2v = find_optimal_k(distortions_w2v)
plt.figure(figsize=(10, 6))
plt.plot(range_n_clusters, distortions_bow, marker='o', label='BoW')
plt.plot(range_n_clusters, distortions_tfidf, marker='s', label='TF-IDF')
plt.plot(range_n_clusters, distortions_w2v, marker='^', label='w2v')
plt.axvline(x=optimal_k_bow, color='blue', linestyle='--', label=f'Optimal k BoW = {optimal_k_bow}')
plt.axvline(x=optimal_k_tfidf, color='green', linestyle='--', label=f'Optimal k TF-IDF = {optimal_k_tfidf}')
plt.axvline(x=optimal_k_w2v, color='red', linestyle='--', label=f'Optimal k w2v = {optimal_k_w2v}')
plt.title('Elbow Method untuk Menentukan Jumlah Cluster Optimal')
plt.xlabel('Jumlah Cluster')
plt.ylabel('Distortion')
plt.legend()
plt.grid(True)
plt.show()
elbow_results = pd.DataFrame({
    'Jumlah Cluster': range_n_clusters,
    'Score BoW': distortions_bow,
    'Score TF-IDF': distortions_tfidf,
    'Score w2v': distortions_w2v})
print("Metode Elbow selesai dijalankan untuk BoW, TF-IDF, dan w2v.")
print(f'Optimal k BoW: {optimal_k_bow}')
print(f'Optimal k TF-IDF: {optimal_k_tfidf}')
print(f'Optimal k w2v: {optimal_k_w2v}')
```

Gambar 4.36 Sintak Pembobotan kata dengan *BoW, TF-IDF*, dan *Word2Vec* Menggunakan *Elbow Method*
Sumber: Penelitian Sendiri



Gambar 4.37 Grafik Pembobotan Kata dengan *Elbow Method*
Sumber: Penelitian Sendiri

Sintak pada gambar 4.36 melakukan analisis teks pada ulasan dengan menggunakan tiga teknik: *Bag of Words (BoW), TF-IDF*, dan *Word2Vec*. Setelah melakukan tokenisasi teks, sintak ini mengaplikasikan ketiga teknik tersebut untuk mendapatkan representasi fitur teks dari kolom *content_proses_stemming_NLP-ID*. Selanjutnya, menggunakan metode *Elbow*, sintak ini secara otomatis menentukan jumlah *cluster* optimal untuk setiap teknik, menambahkan garis vertikal pada titik optimal, dan memvisualisasikan hasilnya dalam satu *plot* komparatif. Hasil *distortions* untuk setiap jumlah *cluster* juga ditampilkan dalam bentuk tabel.

Grafik *Elbow* pada gambar 4.37 menunjukkan bahwa jumlah *cluster* optimal berbeda-beda untuk masing-masing metode yang digunakan. Metode *Bag of Words (BoW)* dan *TF-IDF* keduanya mencapai performa *clustering* terbaiknya pada jumlah *cluster* (k) sebesar 3, yang ditandai dengan penurunan paling tajam dalam distortions hingga k=3. Ini mengindikasikan bahwa *BoW* dan *TF-IDF* mengelompokkan data secara efektif saat dibagi menjadi 3 cluster. Sebaliknya, metode *Word2Vec* menunjukkan jumlah *cluster* optimal pada k=7, yang ditandai dengan penurunan yang signifikan dalam *distortions* hingga k=7. Hal ini menunjukkan bahwa *Word2Vec* mengelompokkan data dengan lebih baik saat dibagi menjadi 7 *cluster*. Penentuan jumlah *cluster* optimal ini penting untuk memastikan kualitas *clustering* yang baik, yang berarti *cluster* lebih homogen di dalamnya dan lebih heterogen antar *cluster*. Dengan demikian, *BoW* dan *TF-IDF* lebih cocok dengan 3 *cluster*, sedangkan *Word2Vec* lebih optimal dengan 7 *cluster* dalam analisis ini. Dalam gambar tersebut, ditemukan bahwa untuk model *Bag of Words (BoW)* dan *TF-IDF*, jumlah *cluster* optimal adalah 3, di mana skor *distortion*

masing-masing adalah 11,158.93 dan 13,319.75. Skor *distortion* pada 3 *cluster* menunjukkan penurunan tajam sebelum melambat, menandakan bahwa 3 *cluster* adalah pilihan terbaik untuk kedua metode ini. Sementara itu, untuk model *Word2Vec*, jumlah *cluster* optimal adalah 7, dengan skor *distortion* sebesar 8,678.10 pada 7 *cluster*, menunjukkan penurunan yang konsisten hingga titik ini sebelum mulai melambat.

Jumlah Cluster	Score BoW	Score TF-IDF	Score w2v
2	12736.411451	14175.729595	13308.588933
3	11158.931190	13319.750090	11207.654202
4	10539.607202	13110.981282	10332.258009
5	9782.881480	11496.504334	9730.880963
6	9202.721939	11147.964187	9138.476297
7	8897.547764	9998.390130	8678.096412
8	8635.410855	9850.206755	8604.351220
9	8339.491024	9254.413261	8251.033848
10	8172.578444	9154.309076	8031.196110

Gambar 4.38 Tabel performansi terhadap 3 metode pembobotan kata
Sumber: Penelitian Sendiri

Berdasarkan metode *Silhouette* dan *Elbow*, analisis klasterisasi menunjukkan bahwa jumlah *cluster* optimal untuk model *Bag of Words (BoW)* dan *TF-IDF* adalah 3. Kedua metode ini menghasilkan nilai *Silhouette Score* dan skor *distortion* yang konsisten, menunjukkan keseimbangan terbaik antara pemisahan dan konsolidasi klaster. Sebaliknya, model *Word2Vec* menunjukkan hasil terbaik pada 7 *cluster* dengan skor *distortion* lebih rendah, meskipun tidak sebaik 3 *cluster* pada *BoW* dan *TF-IDF*. Oleh karena itu, dalam konteks penelitian ini, *modeling* dilakukan dengan menggunakan 3 cluster dan pembobotan kata menggunakan

Word2Vec, karena pendekatan ini memberikan kinerja yang lebih baik dan lebih konsisten secara umum.

4.2.4. *Modeling*

```
1 | import pandas as pd
2 | import re
3 | import numpy as np
4 | import tensorflow as tf
5 | from tensorflow.keras.preprocessing.text import Tokenizer
6 | from tensorflow.keras.preprocessing.sequence import pad_sequences
7 | from sklearn.cluster import KMeans
8 | import matplotlib.pyplot as plt
9 | from wordcloud import WordCloud
10| #memilih pembobotan optimal
11| clust_num =3
12| X = X_w2v
13| X_normalized = X_normalized_w2v
14| #@title Clustering
15| kmeans = KMeans(n_clusters=clust_num, random_state=0)
16| kmeans.fit(X_normalized)
17| labels = kmeans.labels_
18| df_reviews_all_modelling  = df_reviews_all_proses
[['reviewId','content_proses_stemming_NLP-ID','score']].copy()
19| df_reviews_all_modelling['cluster'] = labels
20| df_reviews_all_modelling.rename(columns={'content_proses_stemming_NLP-ID':
'review'}, inplace=True)
22| df_reviews_all_modelling.head()
```

Gambar 4.39 Sintak *Modeling* dari *K-Means*
Sumber: Penelitian Sendiri

Sintaks di atas melakukan *clustering* pada ulasan pengguna menggunakan algoritma *K-Means*. Pertama, berbagai *library* yang diperlukan seperti *pandas*, *numpy*, *tensorflow*, *sklearn, matplotlib*, dan *wordcloud* diimpor. Kemudian, jumlah *cluster* optimal *(clust_num)* ditentukan dan data input (X) serta data yang telah dinormalisasi (*X_normalized*) disiapkan. Algoritma *KMeans* diterapkan pada data yang telah dinormalisasi untuk melakukan *clustering* dengan jumlah *cluster* yang ditentukan. Label *cluster* hasil *clustering* ditambahkan ke dalam *DataFrame* ulasan pengguna (*df_reviews_all_modelling*) sebagai kolom baru *'cluster'*, dan nama kolom *'content_proses_stemming_NLP-ID'* diubah menjadi *'review'* untuk kemudahan analisis lebih lanjut. Akhirnya, program menampilkan lima baris pertama dari *DataFrame* hasil *clustering*.

reviewId	review	score	cluster
cb99ff8f-0042-40be-8bfa-f38a250bab8e	bagus	5	0
dbf009ad-acef-430b-923b-c6b53a6fa3bd	bantu kait ajar administrasi guru	5	1
184aef51-fbb1-4f1f-8bd9-618e8c65d0bd	bagus bantu	4	1
24489d71-53fa-45b1-9a87-df6d2bfb7acc	baik	4	2
e3e58eb4-0eb4-4b36-9161-3607bdeee62b	membagun	5	2

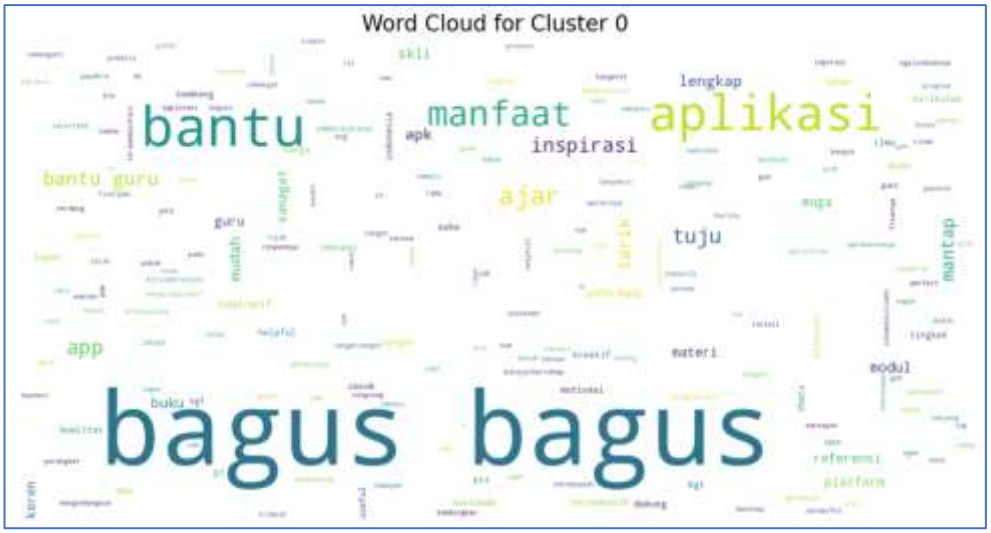
Gambar 4.40 Hasil dari *Clustering* dengan *K-Means*
Sumber: Penelitian Sendiri

```
1 | import pandas as pd
2 | import matplotlib.pyplot as plt
3 | from wordcloud import WordCloud
4 | from tensorflow.keras.preprocessing.text import Tokenizer
7 | clust_num1 = list(set(df_reviews_all_modelling['cluster']))
10| overall_word_freq = {}
13| for cluster in clust_num1:
14|                                          cluster_data                    =
df_reviews_all_modelling[df_reviews_all_modelling['cluster'] == cluster]
15|     tokenizer = Tokenizer()
16|     tokenizer.fit_on_texts(cluster_data['review'])
17|     word_counts = tokenizer.word_counts
19|     for word, count in word_counts.items():
20|         if word in overall_word_freq:
21|             overall_word_freq[word].append((cluster, count))
22|         else: overall_word_freq[word] = [(cluster, count)]
26| reassigned_clusters = {word: max(freq_list, key=lambda x: x[1])[0] for word,
freq_list in overall_word_freq.items()}
29| reassigned_df = df_reviews_all_modelling.copy()
31| for index, row in reassigned_df.iterrows():
32|     words = row['review'].split()
33|     cluster_counts = {cluster: 0 for cluster in clust_num1}
34|     for word in words: if word in reassigned_clusters:
36|             cluster_counts[reassigned_clusters[word]] += 1
37|     new_cluster = max(cluster_counts, key=cluster_counts.get)
38| reassigned_df.at[index, 'cluster'] = new_cluster, for cluster in clust_num1:
42|     cluster_data = reassigned_df[reassigned_df['cluster'] == cluster]
43|     text = ' '.join(cluster_data['review'])
46|                wordcloud      =      WordCloud(width=1600,      height=800,
background_color='white').generate(text)
47|     plt.figure(figsize=(16, 8))
48|     plt.imshow(wordcloud, interpolation='bilinear')
49|     plt.axis('off')
50|     plt.title(f'WordCloud for Cluster {cluster}', fontsize=20)
51|     plt.show()
54|     tokenizer.fit on texts(cluster data['review'])
55|     word_counts = tokenizer.word_counts
56|         word_freq   =   pd.DataFrame(word_counts.items(),   columns=['word',
'count']).sort_values(by='count', ascending=False)
59|     plt.figure(figsize=(12, 8))
60|        bars   =   plt.barh(word_freq['word'][:10],   word_freq['count'][:10],
color='blue')
61|     plt.title(f'Word Frequency for Cluster {cluster}', fontsize=20)
62|     plt.xlabel('Frequency', fontsize=16)
63|     plt.ylabel('Words', fontsize=16)
65|     for bar in bars: width = bar.get_width()
67|             plt.text(width,  bar.get_y()  +  bar.get_height()/2,  int(width),
va='center', fontsize=14)
69|     plt.gca().invert_yaxis() plt.show()
72| print("Kata-kata  telah  dipindahkan  ke  cluster  berdasarkan  frekuensi
tertinggi.")
```

Gambar 4.41 Sintak WorkCloud untuk hasil dari Clustering
Sumber: Penelitian Sendiri

Sintak di atas menganalisis ulasan pengguna dengan mengaitkan kata-kata ke *cluster* berdasarkan frekuensi tertinggi dan menampilkan hasilnya melalui *WordCloud* dan grafik frekuensi kata. Pertama, program menghitung frekuensi kata dalam setiap *cluster* dan menyusun *DataFrame* baru dengan *cluster* yang diperbarui. Selanjutnya, program menghasilkan *WordCloud* dan grafik frekuensi kata untuk setiap *cluster*, yang membantu dalam visualisasi distribusi kata-kata yang paling sering muncul dalam setiap *cluster*, sehingga memudahkan pemahaman pola dan tema dominan dalam ulasan pengguna.

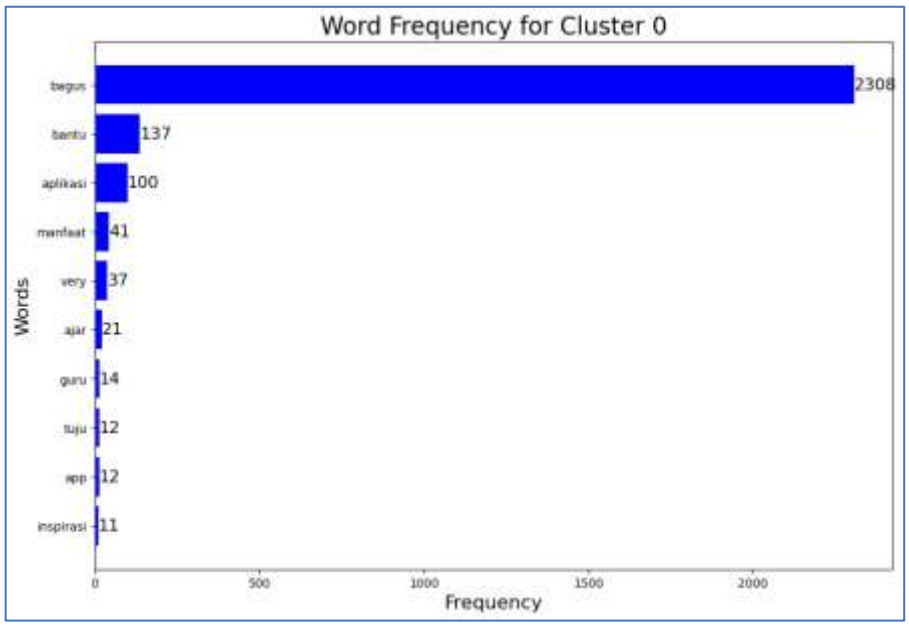
1. *Cluster* 0 :



Gambar 4.42 *Workcloud* dari *Cluster* 0
Sumber: Penelitian Sendiri

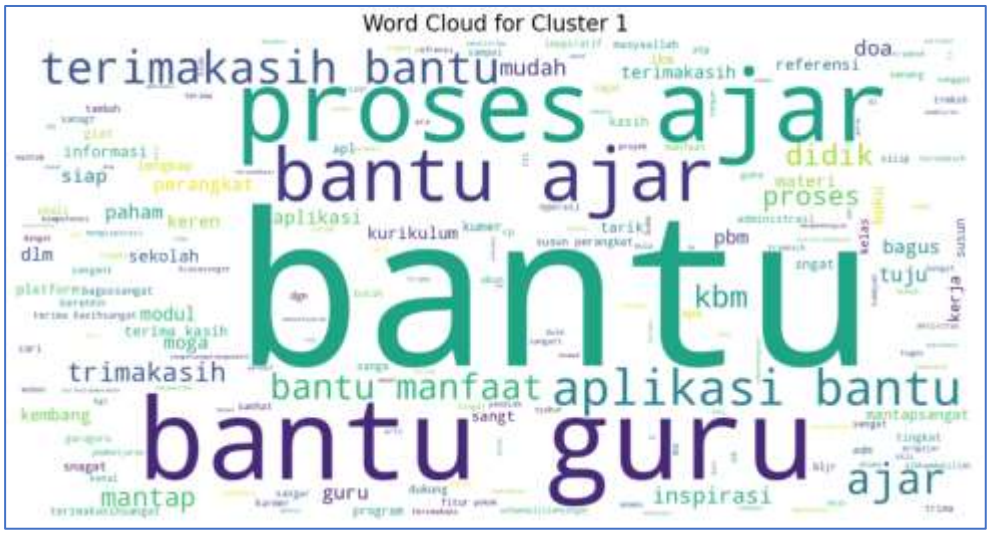
Grafik menunjukkan hasil *clustering* ulasan pengguna untuk *Cluster* 0 melalui *WordCloud* dan grafik frekuensi kata. *WordCloud* menampilkan kata-kata yang sering muncul dengan ukuran mencerminkan frekuensinya, seperti "bagus," "bantu," dan "aplikasi." Grafik batang horizontal memperlihatkan sepuluh kata teratas, di mana "bagus" muncul 2308 kali, "bantu" 137 kali, dan "aplikasi" 100

kali. Visualisasi ini memberikan gambaran tentang topik dominan dan persepsi pengguna terhadap aplikasi dalam *Cluster* 0.



Gambar 4.43 Frekuensi Kata dari *Cluster* 0
Sumber: Penelitian Sendiri

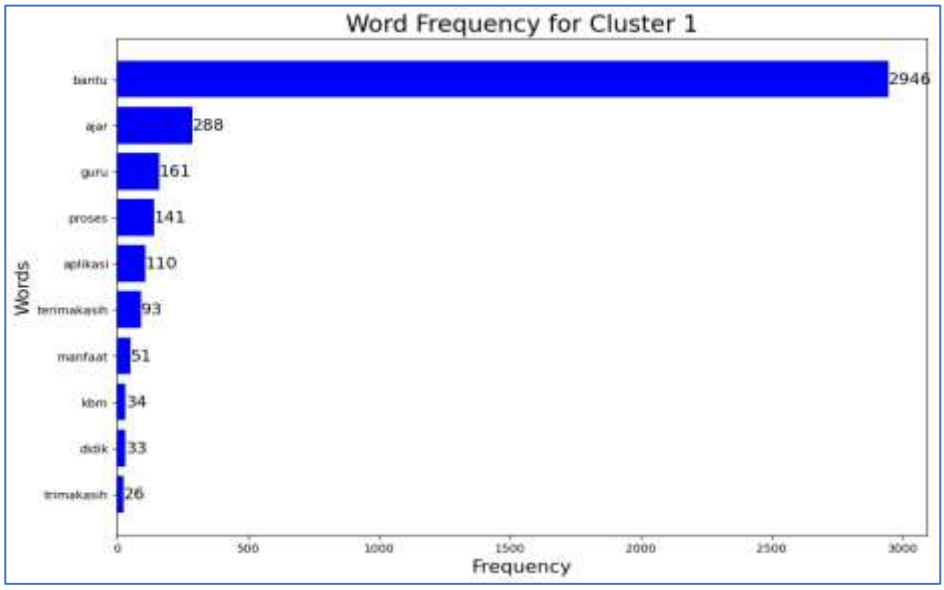
2. *Cluster* 1



Gambar 4.44 *Workcloud* dari *Cluster* 1
Sumber: Penelitian Sendiri

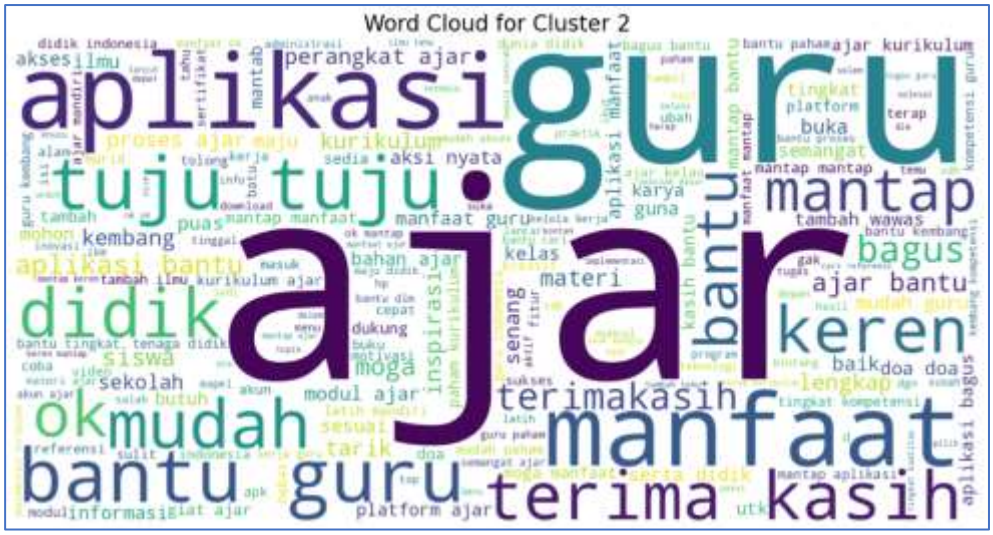
Grafik menunjukkan hasil *clustering* ulasan pengguna untuk *Cluster* 1 melalui *WordCloud* dan grafik frekuensi kata. *WordCloud* menampilkan kata-kata

yang sering muncul dengan ukuran mencerminkan frekuensinya, seperti "bantu," "proses," "ajar," "guru," dan "terima kasih." Grafik batang horizontal memperlihatkan sepuluh kata teratas, di mana "bantu" muncul 2946 kali, "ajar" 288 kali, dan "guru" 161 kali. Visualisasi ini memberikan gambaran tentang topik dominan dan persepsi pengguna terhadap aplikasi dalam *Cluster* 1.



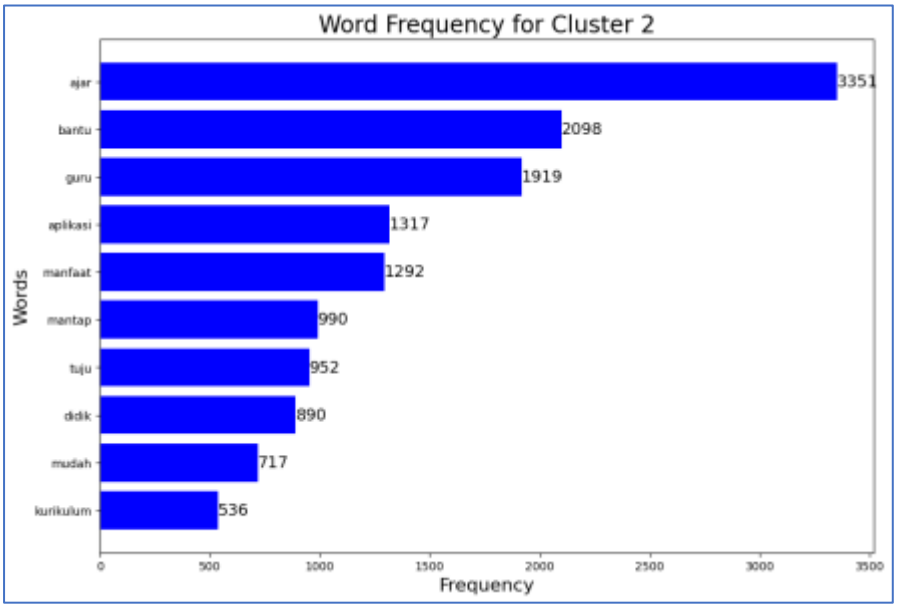
Gambar 4.45 Frekuensi Kata dari *Cluster* 1
Sumber: Penelitian Sendiri

3. Cluster 2



Gambar 4.46 *Workcloud* dari *Cluster* 2
Sumber: Penelitian Sendiri

Gambar menunjukkan hasil *clustering* ulasan pengguna untuk *Cluster* 2 melalui *WordCloud* dan grafik frekuensi kata. *WordCloud* menampilkan kata-kata yang sering muncul dengan ukuran mencerminkan frekuensinya, seperti "ajar," "guru," "aplikasi," "tuju," dan "bantu." Grafik batang horizontal memperlihatkan sepuluh kata teratas, di mana "ajar" muncul 3351 kali, "bantu" 2098 kali, dan "guru" 1919 kali. Visualisasi ini memberikan gambaran tentang topik dominan dan persepsi pengguna terhadap aplikasi dalam *Cluster* 2.



Gambar 4.47 Frekuensi Kata dari *Cluster* 2
Sumber: Penelitian Sendiri

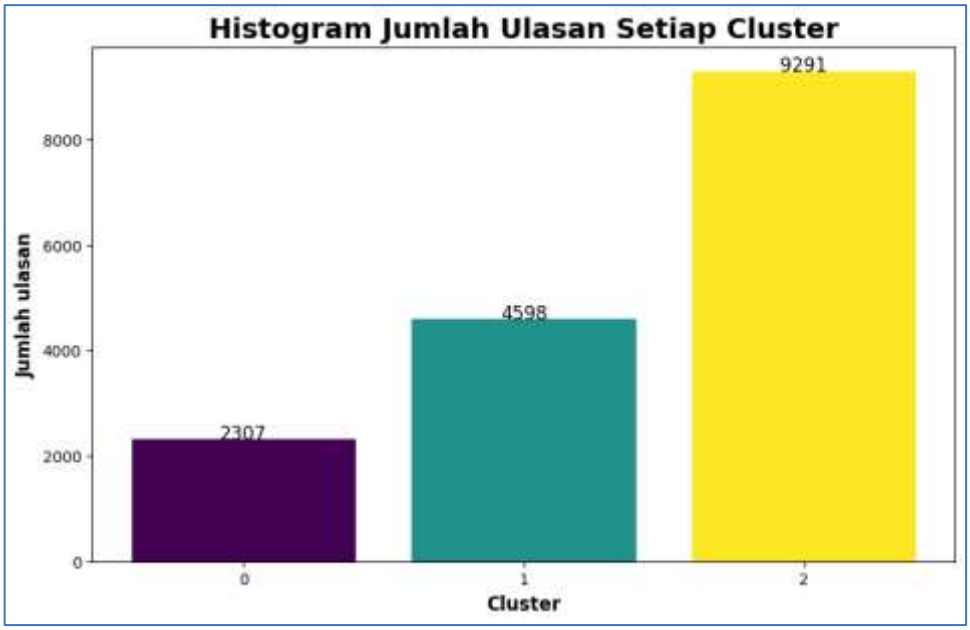
4.2.5. *Evaluation* dengan *Principal Component Analysis* (PCA)

```
2 | import matplotlib.pyplot as plt
3 | from sklearn.decomposition import PCA
5 | cluster_counts = df_reviews_all_modelling['cluster'].value_counts()
7 | pca = PCA(n_components=2)
8 | X_pca = pca.fit_transform(X_normalized)
10| colors = plt.cm.get_cmap('viridis', clust_num)
12| plt.figure(figsize=(10, 6))
13| bars = plt.bar(cluster_counts.index, cluster_counts.values,
color=colors(cluster_counts.index))
14| plt.xlabel('Cluster', fontsize=12, fontweight='bold')
15| plt.ylabel('Jumlah ulasan', fontsize=12, fontweight='bold')
16| plt.title('Histogram Jumlah Ulasan Setiap Cluster', fontsize=18,
fontweight='bold')
17| plt.xticks(cluster_counts.index, fontsize=10)
```

```
20| for bar, count in zip(bars, cluster_counts.values):
21|     plt.text(bar.get_x() + bar.get_width() / 2, count + 0.3, str(count),
fontsize=12, color='black', ha='center')
24| plt.show()
26| plt.figure(figsize=(10, 6))
27| for i in range(clust_num):
28|     plt.scatter(X_pca[labels == i, 0], X_pca[labels == i, 1],
color=colors(i), label=f'Cluster {i}')
29| plt.title('Visualisasi Cluster dengan PCA')
30| plt.xlabel('Komponen Utama 1')
31| plt.ylabel('Komponen Utama 2')
32| plt.legend()
33| plt.show()
```

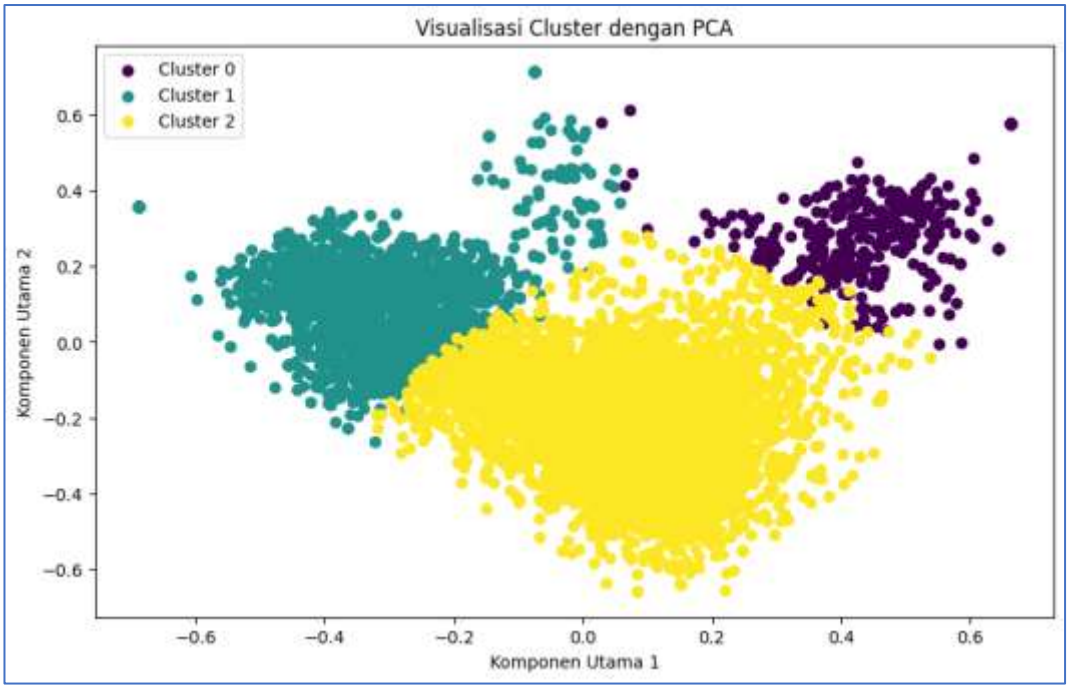
Gambar 4.48 Sintak dari *Principal Component Analysis* (PCA)
Sumber: Penelitian Sendiri

Sintak di atas melakukan visualisasi hasil *clustering* ulasan pengguna. Histogram menunjukkan distribusi jumlah ulasan per *cluster*, sedangkan *scatter plot* menggunakan *PCA* untuk menampilkan hasil *clustering* dalam dua dimensi. *PCA* membantu mereduksi dimensi data, memudahkan pemahaman struktur dan distribusi data dalam setiap *cluster*. *Scatter plot* memperlihatkan pengelompokan ulasan berdasarkan dua komponen utama, membantu mengevaluasi kualitas dan interpretabilitas hasil *clustering*.



Gambar 4.49 Histogram Total Ulasan dari Setiap *Cluster*
Sumber: Penelitian Sendiri

Histogram di atas menunjukkan jumlah ulasan yang termasuk dalam setiap *cluster*. *Cluster* 0 memiliki 2307 ulasan, *Cluster* 1 memiliki 4598 ulasan, dan *Cluster* 2 memiliki 9291 ulasan. Ini memberikan gambaran tentang distribusi jumlah ulasan di setiap *cluster*.



Gambar 4.50 Visualisasi Setiap *Cluster* dengan *PCA*
Sumber: Penelitian Sendiri

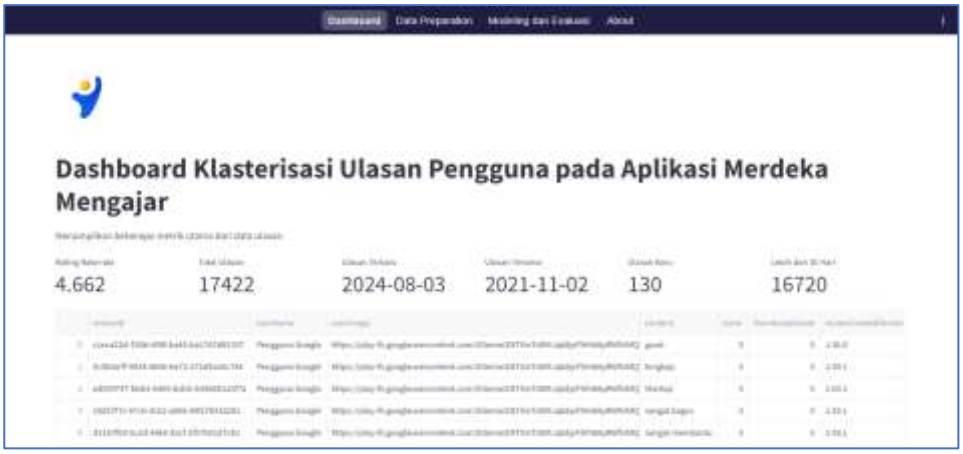
Gambar 4.50 adalah *scatter plot* hasil clustering ulasan pengguna menggunakan *PCA (Principal Component Analysis)* untuk reduksi dimensi. Setiap titik mewakili ulasan yang telah dikelompokkan ke dalam tiga *cluster*: *Cluster* 0 (ungu), *Cluster* 1 (hijau), dan *Cluster* 2 (kuning), dengan dua komponen utama dari *PCA* digunakan untuk memproyeksikan data ke dalam dua dimensi. *Plot* ini memperlihatkan distribusi dan struktur data dalam setiap *cluster*, membantu memahami pola dan karakteristik unik masing-masing *cluster* serta mengevaluasi kualitas dan interpretabilitas hasil *clustering*.

4.2.6. ***Deployment***

Pada tahap ini yang dilakukan yaitu melakukan *deployment* terhadap *clustering* yang sudah dibangun dalam bentuk *web*, Dalam penelitian ini peneliti menggunakan *web service* dari *Streamlit* dengan *server GitHub Codespaces*. Berikut tahapan *deployment* dari *machine learning* dan model yang sudah diterapkan :

1. Membuat akun *GitHub*
2. buat *repository* di *GitHub* agar memiliki *repositor server* dari *website streamlit* yang akan Peneliti gunakan
3. Membuat *file* seperti *file* utama untuk *website* nya, *file* yang berisikan sintak dari kumpulan fungsi yang terdapat dalam *website* untuk melakukan *Clustering K-Means*, dan kumpulan data atau kata *slang word*, arti dari singkatan, dan kata sejenisnya yang mendukung untuk proses teks *Processing* dan data utama.
4. Setelah file dibuat di *repository* maka, Peneliti menyalin keseluruhan kode program yang sudah dilakukan sebelumnya ke *codespaces* di *github*, maka peneliti menambahkan *library streamlit* pada sintak program agar dapat ditampilkan di *website streamlit*
5. Untuk menampilkan sintak program kedalam bentuk *website*, peneliti menggunakan terminal yang tesedia di *server GitHub* dengan mengetikan "*streamlit run app.py*". Kemudian setelah muncul keterangan berupa *url* maka akan secara otomatis muncul notifikasi seperti gambar dibawah ini. Klik tombol "*Open in Browser*" atau "*Make Public*"

6. Berikut Tampilan dari halaman *Dashboard*



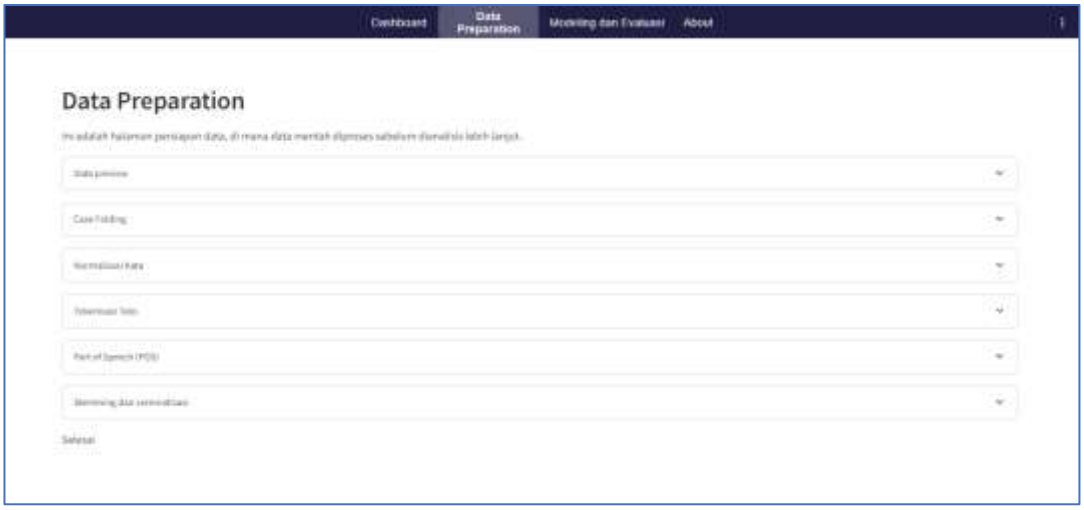
Gambar 4.51 Halaman Dashboard W*ebsite*
Sumber: Penelitian Sendiri

Dashboard ini menampilkan berbagai analisis dan visualisasi mengenai ulasan pengguna pada aplikasi Merdeka Mengajar, termasuk jumlah ulasan, rating rata-rata, tanggal ulasan terbaru, dan ulasan tertua. Terdapat tabel yang menampilkan data ulasan pengguna. Analisis ulasan mencakup distribusi total skor, distribusi waktu ulasan, distribusi skor menurut versi aplikasi, dan tren ulasan dari waktu ke waktu. Klasterisasi data ulasan menunjukkan distribusi ulasan dalam setiap *cluster*, dengan tabel hasil klasterisasi. Selain itu, terdapat visualisasi frekuensi kata dalam bentuk *WordCloud* dan grafik batang untuk setiap *cluster*, memberikan gambaran tentang sentimen, frekuensi kata, dan distribusi ulasan dari pengguna aplikasi.

7. Berikut Tampilan dari halaman *Data Preparation*

Pada halaman *Data Preparation* dari sebuah *dashboard* aplikasi yang digunakan untuk mempersiapkan data sebelum dianalisis lebih lanjut. Halaman ini memiliki menu persiapan data dalam bentuk tombol akordeon, seperti *Data Preview, Case Folding*, Normalisasi Kata, Tokenisasi Teks, *Part of Speech (POS),*

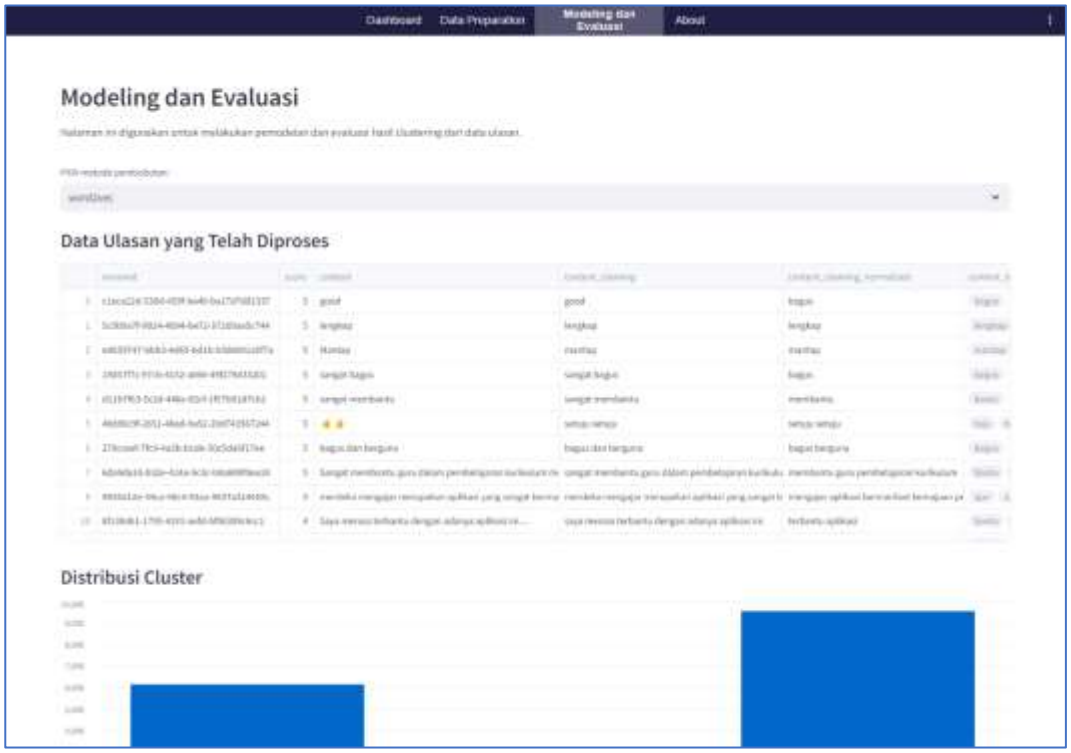
serta *Stemming* dan *Lemmatization*. Bagian atas halaman memiliki navigasi menu yang terdiri dari *Dashboard, Data Preparation, Modeling* dan Evaluasi, serta *About*. Halaman ini membantu pengguna dalam memproses data mentah melalui beberapa tahap penting sebelum analisis atau pemodelan.



Gambar 4.52 Halaman *Data Preparation* dari *Website*
Sumber: Penelitian Sendiri

8. Berikut Tampilan dari halaman *Modeling* dan Evaluasi

Halaman Pada *Modeling* dan Evaluasi dari *dashboard* ini digunakan untuk melakukan pemodelan dan evaluasi hasil *clustering* dari data ulasan. Terdapat pilihan metode pembobotan seperti *word2vec*, dan tabel yang menunjukkan data ulasan yang telah diproses. Visualisasi meliputi distribusi *cluster*, distribusi skor berdasarkan *cluster*, dan *plot PCA* untuk hasil *clustering*. Tabel hasil klasterisasi data ulasan pengguna serta frekuensi kata dan cluster ditampilkan dengan visualisasi *WordCloud* dan grafik batang. Halaman ini dirancang untuk memberikan gambaran menyeluruh tentang analisis dan hasil *clustering* ulasan pengguna.



Gambar 4.53 Halaman *Modeling* dan Evaluasi dari *Website*
Sumber: Penelitian Sendiri

9. Berikut Tampilan Dari halaman *About*

Gambar 4.54 Halaman About dari *Website*
Sumber: Penelitian Sendiri

Halaman "*About*" menjelaskan bahwa aplikasi Merdeka Mengajar dari Kemendikbudristek mendukung guru dengan fitur *asesmen* murid, perangkat ajar, *video* inspirasi, dan pelatihan mandiri. Analisis ulasan pengguna membantu meningkatkan

aplikasi dan merumuskan strategi komunikasi, memastikan aplikasi ini efektif mendukung pendidikan di Indonesia.

4.2.7. *System Evaluation*

1. *Whitebox Testing*

Pada tahapan *Whitebox testing*, peneliti menguji *source code* dengan menggunakan teknik *basis path* untuk mengukur kompleksitas dari suatu program dengan cara mengidentifikasi semua jalur yang mungkin dilalui oleh program di bagian *deployment*.

Tabel 4.2 Whitebox testing dengan Kode Program
Sumber: Dokumentasi Pribadi

NO	Kode Program

1	0 | pages = ["Dashboard", "Data Preparation", "Modeling dan Evaluasi", "About"]
	1 | styles = {
	2 | "nav": {
	3 | "background-color": "rgb(32, 30, 67)",
	4 | },
	5 | "div": {
	6 | "max-width": "32rem",
	7 | },
	8 | "span": {
	9 | "border-radius": "0.5rem",
	10| "color": "rgb(238, 238, 238)",
	11| "margin": "0 0.125rem",
	12| "padding": "0.4375rem 0.625rem",
	13| },
	14| "active": {
	15| "background-color": "rgba(255, 255, 255, 0.25)",
	16| },
	17| "hover": {
	18| "background-color": "rgba(255, 255, 255, 0.35)",
	19| },
	20| }
	21|
	22| # Load and process data
	23| reviews_df = global_data.reviews_all_app()
	24| reviews_df["at"] = pd.to_datetime(reviews_df["at"])
	25| kamus_tidak_baku = global_data.data_dict()
	26| chat_words_mapping = global_data.chat_words_mapping()
	27|
	28| if 'df_reviews_all_proses' not in st.session_state:
	29| st.session_state.df_reviews_all_proses = prepare_data(reviews_df, kamus_tidak_baku, chat_words_mapping)
	30| df_reviews_all_proses = st.session_state.df_reviews_all_proses
	31|
	32| # Create the navigation bar
	33| page = st_navbar(pages, styles=styles)
	34|
	35| # Get user inputs for clustering
	36| if 'clust_num' not in st.session_state:
	37| st.session_state.clust_num = 3
	38| if 'metode_pembobotan' not in st.session_state:
	39| st.session_state.metode_pembobotan = 'word2vec'

```
40|
41| clust_num = st.session_state.clust_num
42| metode_pembobotan = st.session_state.metode_pembobotan
43|
44| X,       X_normalized      =      pembobotan_kata(df_reviews_all_proses,
metode_pembobotan)
45| df_hasil = clustering_k_means(df_reviews_all_proses, metode_pembobotan,
clust_num)
46| df_combined = combine_dataframes(reviews_df, df_hasil)
47| cluster_counts = df_hasil['cluster'].value_counts()
48| cluster_summary = summarize_clusters(df_combined)
49|
```

```
0 | if page == "About":
```

```
0 |     st.header("About")
1 |    st.write("Ini adalah halaman tentang aplikasi ini, yang menjelaskan tujuan dan cara penggunaan aplikasi.")
```

```
0 | elif page == "Data Preparation":
```

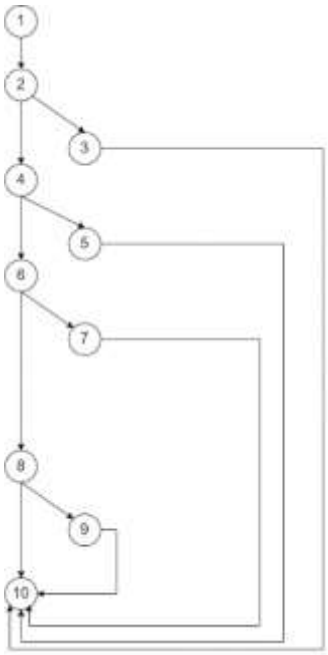
```
0 |     st.header("Data Preparation")
1 |      st.write("Ini adalah halaman persiapan data, di mana data mentah
diproses sebelum dianalisis lebih lanjut.")
2 |     df_reviews_all_proses = st.session_state.df_reviews_all_proses
3 |
4 |     with st.expander("Data preview"):
5 |         st.subheader("Data preview")
6 |         st.write("Tampilan awal dari data ulasan yang telah dimuat.")
7 |         st.dataframe(reviews_df.head(5), use_container_width=True)
8 |
9 |     with st.expander("Case Folding"):
10|         st.subheader("Case Folding")
11|          st.write("Mengubah huruf besar ke kecil, mengubah emotikon ke
teks, dan menghapus kode HTML, URL, dan simbol-simbol.")
12|               df_reviews_all_proses_cf   =  df_reviews_all_proses.loc[:,
['reviewId', 'content', 'content_cleaning']]
13|                          st.dataframe(df_reviews_all_proses_cf.head(5),
use_container_width=True)
14|
15|     with st.expander("Normalisasi Kata"):
16|         st.subheader("Normalisasi Kata")
17|         st.write("Proses normalisasi kata untuk mengubah kata tidak baku
menjadi kata baku.")
18|               df_reviews_all_proses_nk   =  df_reviews_all_proses.loc[:,
['reviewId', 'content', 'content_cleaning_normalized']]
19|                           st.dataframe(df_reviews_all_proses_nk.head(5),
use_container_width=True)
20|
21|     with st.expander("Tokenisasi Teks"):
22|         st.subheader("Tokenisasi Teks")
23|          st.write("Proses memecah teks menjadi token-token yang lebih
kecil.")
24|             df_reviews_all_proses_token  =  df_reviews_all_proses.loc[:,
['reviewId', 'content', 'content_tokenizing']]
25|                        st.dataframe(df_reviews_all_proses_token.head(5),
use_container_width=True)
26|
27|     with st.expander("Part of Speech (POS)"):
28|         st.subheader("Part of Speech (POS)")
29|         st.write("Proses penandaan bagian dari ucapan (POS) pada setiap
token dalam teks.")
30|              df_reviews_all_proses_pos  =  df_reviews_all_proses.loc[:,
['reviewId', 'content', 'content_part_of_speech']]
31|                          st.dataframe(df_reviews_all_proses_pos.head(5),
use_container_width=True)
32|
33|     with st.expander("Stemming dan Lemmatisasi"):
34|         st.subheader("Stemming dan Lemmatisasi")
35|         st.write("Proses mengubah kata ke bentuk dasarnya.")
36|             df_reviews_all_proses_stem  =  df_reviews_all_proses.loc[:,
['reviewId', 'content', 'content_proses_stemming_NLP-ID']]
37|      st.dataframe(df_reviews_all_proses_stem.head(5), use_container_width=True)
```

6	0 | elif page == "Modeling dan Evaluasi":
7	0 | st.header("Modeling dan Evaluasi") 1 | st.write("Halaman ini digunakan untuk melakukan pemodelan dan evaluasi hasil clustering dari data ulasan.") 2 | col1input, col2input = st.columns(2) 3 | with col1input: 4 | st.session_state.clust_num = st.number_input("Masukkan jumlah cluster:", min_value=2, value=3) 5 | with col2input: 6 | st.session state.metode pembobotan = st.selectbox("Pilih metode pembobotan:", ('bag_of_words', 'tfidf', 'word2vec'), index=2) 7 | 8 | df_reviews_all_proses = st.session_state.df_reviews_all_proses 9 | 10| # Menampilkan data ulasan yang telah diproses 11| st.subheader("Data Ulasan yang Telah Diproses") 12| st.dataframe(df_reviews_all_proses, use_container_width=True) 13| 14| # Menampilkan distribusi cluster 15| st.subheader("Distribusi Cluster") 16| st.bar_chart(cluster_counts) 17| 18| # Membagi hasil clustering dan evaluasi clustering dalam beberapa kolom 19| col1, col2 = st.columns(2) 20| 21| with col1: 22| st.subheader("Distribusi Skor dari Data yang Telah Dikombinasikan") 23| plot_score_distribution(df_combined) 24| 25| with col2: 26| st.write("### Davies-Bouldin Index untuk Berbagai Jumlah Cluster") 27| display_dbi_scores(X_normalized, clust_num) 28| st.write("### Plot PCA untuk Clusters") 29| plot_pca_clusters(X_normalized, df_hasil['cluster']) 30| 31| # Membagi ringkasan cluster dan data yang telah dikombinasikan dalam beberapa kolom 32| st.subheader("Data dengan Hasil Clustering") 33| st.dataframe(df_combined, use_container_width=True) 34| col3, col4 = st.columns(2) 35| 36| with col1: 37| st.title("Data Klaster:") 38| st.write("Hasil klasterisasi data ulasan pengguna.") 39| st.dataframe(df_hasil, use_container_width=True) 40| process_and_display_clusters(df_hasil) 41| 42| with col2: 43| st.write("Ringkasan ulasan:") 44| for i, summary in enumerate(cluster_summary["Ringkasan ulasan"]): 45| st.subheader(f"Clustering {i}") 46| st.write(summary)
8	0 | elif page == "Dashboard":
9	0 | st.image(image="https://play-lh.googleusercontent.com/jBzUQv7vmYT_AUDOt-WQX3Uh4lupq6omQaL2nCdzlG4zNmZUJ2PaqCGpc 03-FBw7w",width=100,use_column_width=100) 1 | st.title('Dashboard Klasterisasi Ulasan Pengguna pada Aplikasi Merdeka Mengajar') 2 | 3 | # Calculate and display metrics 4 | st.write("Menampilkan beberapa metrik utama dari data ulasan.") 5 | average_rating = f"{reviews_df['score'].mean():.3f}" 6 | col1, col2, col3, col4, col5, col6 = st.columns(6) 7 | col1.metric(label="Rating Rata-rata", value=average_rating)

```
8 |     col2.metric(label="Total Ulasan", value=len(reviews_df))
9 |                          col3.metric(label="Ulasan            Terbaru",
value=reviews_df['at'].max().strftime('%Y-%m-%d'))
10|                          col4.metric(label="Ulasan            Terlama",
value=reviews_df['at'].min().strftime('%Y-%m-%d'))
11|                            col5.metric(label="Ulasan             Baru",
value=len(reviews_df[reviews_df['at']     >     (reviews_df['at'].max()     -
pd.Timedelta(days=5))]))
12|                col6.metric(label="Lebih       dari       30       Hari",
value=len(reviews_df[reviews_df['at']     <     (reviews_df['at'].max()     -
pd.Timedelta(days=30))]))
13|     st.dataframe(reviews_df.head(5), use_container_width=True)
14|
15|     col1input, col2input = st.columns(2)
16|     with col1input:
17|          st.session_state.clust_num = st.number_input("Masukkan jumlah
cluster:", min_value=2, value=3)
18|     with col2input:
19|         st.session_state.metode_pembobotan = st.selectbox("Pilih metode
pembobotan:", ('bag_of_words', 'tfidf', 'word2vec'), index=2)
20|
21|     # Ensure the score column is of type integer
22|     reviews_df['score'] = reviews_df['score'].astype(int)
23|
24|     st.title('Analisis Ulasan Aplikasi')
25|      st.write("Analisis yang mendalam dari ulasan aplikasi berdasarkan
berbagai metrik dan distribusi.")
26|
27|     # Layout for Total Score and Distribution of 'at' (review times)
28|     col1, col2 = st.columns(2)
29|
30|     with col1:
31|         st.subheader('Total Score')
32|         st.write("Distribusi total skor dari ulasan pengguna.")
33|         score_counts = reviews_df['score'].value_counts().sort_index()
34|         fig, ax = plt.subplots()
35|         score_counts.plot(kind='bar', ax=ax)
36|         ax.set_title('Total Score')
37|         ax.set_xlabel('Skor')
38|         ax.set_ylabel('Jumlah')
39|         ax.set_xticks(range(5))
40|         ax.set_xticklabels([1, 2, 3, 4, 5])
41|         st.pyplot(fig)
42|
43|     with col2:
44|         st.subheader("Distribusi Waktu Ulasan")
45|         st.write("Distribusi ulasan berdasarkan waktu.")
46|         fig, ax = plt.subplots(figsize=(10, 6))
47|         reviews_df['at'].hist(bins=50, edgecolor='k', alpha=0.7, ax=ax)
48|         ax.set_title('Distribusi Waktu Ulasan')
49|         ax.set_xlabel('Waktu')
50|         ax.set_ylabel('Frekuensi')
51|         ax.grid(True)
52|         st.pyplot(fig)
53|
54|     # Distribusi Skor menurut Versi Aplikasi dan Trend Ulasan dari Waktu
ke Waktu
55|     col1, col2 = st.columns(2)
56|
57|     with col1:
58|         st.subheader('Distribusi Skor menurut Versi Aplikasi')
59|         st.write("Distribusi skor ulasan berdasarkan versi aplikasi.")
60|               app_version_scores   =   reviews_df.groupby(['appVersion',
'score']).size().unstack().fillna(0)
61|         fig, ax = plt.subplots(figsize=(14, 8))
62|         app_version_scores.plot(kind='bar', stacked=True, ax=ax)
63|         ax.set_title('Distribusi Skor berdasarkan Versi Aplikasi')
64|         ax.set_xlabel('Versi Aplikasi')
65|         ax.set_ylabel('Jumlah')
66|         ax.legend(title='Skor')
```

	67| ax.grid(True) 68| plt.xticks(rotation=90) 69| st.pyplot(fig) 71| with col2: 72| st.subheader('Trend Ulasan dari Waktu ke Waktu') 73| st.write("Tren jumlah ulasan dari waktu ke waktu.") 74| fig, ax = plt.subplots(figsize=(12, 6)) 75| reviews_df.set_index('at').resample('M').size().plot(marker='o', ax=ax) 76| ax.set_title('Trend of Reviews Over Time') 77| ax.set_xlabel('Waktu') 78| ax.set_ylabel('Jumlah Ulasan') 79| ax.grid(True) 80| st.pyplot(fig) 82| st.title("Cluster:") 83| st.write("Distribusi dari setiap cluster.") 84| st.bar_chart(cluster_counts) 85| # Data Cluster and Ringkasan Ulasan 86| col1, col2 = st.columns(2) 88| with col1: 89| st.title("Data Klaster:") 90| st.write("Hasil klasterisasi data ulasan pengguna.") 91| st.dataframe(df_hasil, use_container_width=True) 92| process_and_display_clusters(df_hasil) 93| 94| with col2: 95| st.write("Ringkasan ulasan:") 96| for i, summary in enumerate(cluster_summary["Ringkasan ulasan"]): 97| st.subheader(f"Clustering {i}") 98| st.write(summary)
10	0 | st.write("Selesai")

Setelah dibuat tabel di atas untuk membantu melakukan pengujian *White Box testing*. selanjutnya yang dilakukan adalah dengan membangun *flowgraph*:



Gambar 4.55 *Whitebox Testing* dengan *Flowgraph*
Sumber: Dokumentasi Pribadi

Selanjutnya yang dilakukan adalah menghitung *Cyclomatix Complexity* V(G) untuk menentukan jumlah path :

$$V(G) = E – N + 2$$
$$V(G) = 13 - 10 + 2$$
$$V(G) = 5$$

Terdapat 6 Path, Selanjutnya yang dilakukan adalah membuat *Test Case*.

Tabel 4.3 *White Box Testing* dengan *Test Case*
Sumber: Dokumentasi Pribadi

No	Alir Program	Keterangan	Hasil Pengujian
1	1-2-3-10	Menampilkan tampilan halaman “About”	Berhasil
2	1-2-4-5-10	Menampilkan tampilan halaman “Data Preparation” dan melakukan proses Data *Cleaning, Drop, Case Folding, Text Preprocessing*	Berhasil
3	1-2-4-6-7-10	Menampilkan tampilan halaman “Modeling dan Evaluasi”	Berhasil
4	1-2-4-6-8-9-10	Menampilkan tampilan halaman “Dashboard”	Berhasil
5	1-2-4-6-8-10	Program Aplikasi berhasil dijalan secara keseluruhan	Berhasil

2. *Blackbox Testing*

Tabel 4.4 *Black Box Testing*
Sumber: Dokumentasi Pribadi

No	Kegiatan Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Tombol Dashboard	Ketika di Klik, maka akan berpindah ke Halaman Dashboard dan menampilkan semua isi dari halaman dashboard	Berhasil
2	Tombol Data Preparation	Ketika di Klik, maka akan berpindah ke Halaman Data Preparation dan menampilkan semua isi dari halaman Data Preparation	Berhasil
3	Tombol Modeling	Ketika di Klik, maka akan berpindah ke Halaman Modeling dan Evaluasi serta	Berhasil

	dan Evaluasi	menampilkan semua isi dari halaman Modeling dan Evaluasi	
4	Tombol About	Ketika di Klik, maka akan berpindah ke Halaman About dan menampilkan semua isi dari halaman About	Berhasil
5	Tombol Sliding pada Halaman Data Preparation	Ketika di klik, maka akan mengeksekusi dan menampilkan hasil dari proses dari tombol yang di klik	Berhasil
6	Drop Menu untuk Memilih Model Pembobotan Kata	Ketika di pilih inputan salah satu dalam drop menu maka model dari proses klasterisasi akan berubah sesuai dengan model yang dipilih	Berhasil

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pembahasa dari Penerapan Algoritma *K-Means Clustering* Pada Ulasan Pengguna Merdeka Mengajar di *Play Store,* Maka dapat ditarik kesimpulan:

1. Sistem untuk penerapan *K-Means* untuk *Clustering* terhadap Pada Ulasan Pengguna Merdeka Mengajar di *Play Store* dibuat dengan berbasis *website* menggunakan *streamlit* dan *Github Codespaces*
2. Proses membangun model *clustering* menggunakan metode *CRISP-DM* yang meliputi tujuh tahapan: *Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, Deployment*, dan *Evaluation*, ditambah dengan tahapan evaluasi sistem, yaitu *Evaluation System*, telah berhasil diterapkan. Metode *CRISP-DM* terbukti efektif dalam membimbing seluruh proses pembangunan model, dari pemahaman bisnis hingga evaluasi akhir sistem, memastikan hasil yang konsisten dan berkualitas tinggi.
3. Pada Penelitian ini di tahap *Data Preparation, library NLP-ID* lebih baik dalam segi kecepatan memproses data ulasan ketimbang *sastrawi* dengan jumlah yang sama untuk data yang proses, terutama pada tahap *stemming*.
4. Berdasarkan analisis menggunakan metode *Silhouette* dan *Elbow*, jumlah *cluster* optimal untuk model *Bag of Words (BoW)* dan *TF-IDF* adalah 3,

menunjukkan keseimbangan terbaik antara pemisahan dan konsolidasi klaster. Sebaliknya, model *Word2Vec* menunjukkan hasil terbaik pada 7 *cluster*, meskipun tidak sebaik 3 *cluster* pada *BoW* dan *TF-IDF*. Oleh karena itu, penelitian ini menggunakan 3 cluster dengan pembobotan kata menggunakan *Word2Vec*, karena pendekatan ini memberikan kinerja lebih baik dan konsisten.

5. Pada Proses Klasterisasi, ditemukan hanya 3 klaster yang terdiri dari beberapa tema pokok pembahasan. Klaster 0 Membahas terkait "Aplikasi sangat membantu untuk Guru", Klaster 1 Membahas terkait "Memudahkan membuat Administrasi bagi Guru, contoh: Modul Ajar, Materi, dan *Webinar*", dan Klaster 2 membahas terkait "Pelatihan secara *online* maupun *offline*".

5.2. Saran

Terdapat beberapa saran untuk pengembangan sistem selanjutnya dengan saran-saran di bawah:

1. Diharapkan terkait pengambilan data tidak hanya dari satu sumber yaitu *Google Play Store*, tetapi juga dari beberapa media sosial lain seperti *Twitter, Facebook*, dan *Instagram* untuk mendapatkan perspektif yang lebih luas.
2. Diharapkan untuk ke depannya, tidak hanya membangun sistem berbasis *website* tetapi juga dapat membangun sistem berbasis *Android*

agar pengguna dapat mengakses analisis sentimen dengan lebih mudah melalui perangkat *mobile*.

3. Diharapkan pengembangan tidak hanya fokus pada *clustering*, tetapi juga dilakukan hingga tahap analisis sentimen untuk memahami lebih dalam mengenai sentimen positif, negatif, dan netral dari ulasan pengguna.
4. Diharapkan dapat mengeksplorasi model pembobotan kata lain seperti *FastText* atau *BERT* untuk meningkatkan akurasi dan kinerja model klasterisasi.
5. Diharapkan untuk melakukan evaluasi berkala terhadap model dan sistem yang telah dibangun untuk memastikan tetap relevan dan akurat seiring dengan perubahan perilaku pengguna dan perkembangan teknologi.
6. Diharapkan untuk meningkatkan fitur *dashboard* interaktif yang sudah ada, serta menambahkan fitur laporan otomatis untuk memudahkan pemangku kepentingan dalam mengambil keputusan berdasarkan hasil analisis.

DAFTAR PUSTAKA

Af'idah, D. I., Dairoh, & Handayani, S. F. (2021). Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen. *Jurnal Informatika: Jurnal pengembangan IT (JPIT)*, 156-121.

Alghifari, F., & Juardi, D. (2021). Penerapan Data Mining Pada Penjualan Makanan Dan Minuman Menggunakan Metode Algoritma Naïve Bayes. *Jurnal Ilmiah Informatika (JIF), 09*(02), 1-7.

Baktiar, A. R., Mulainsyah, D., Sasmoro, E. C., & Sumiati, E. (2021). Pengujian Menggunakan Black Box Testing dengan Teknik State Transition Testing Pada Perpustakaan Yayasan Pendidikan Islam Pakualam Berbasis Web . *Jurnal Kreativitas Mahasiswa Informatika* , 142-145.

Clinton, R. M., & Sengkey, R. (2019). Purwarupa Sistem Daftar Pelanggaran Lalulintas Berbasis Mini-Komputer Raspberry Pi. *Jurnal Teknik Elektro dan Komputer*, 181-192.

Damayanti, S. E. (2021). ANALISIS DAN IMPLEMENTASI FRAMEWORK CRISP-DM (CROSS INDUSTRY STANDARD PROCESS FOR DATA MINING) UNTUK CLUSTERING PERGURUAN TINGGI SWASTA. *ejournal STT Bandung*, 1-5.

Fitriani, F. (2018). PERAN DINAS PENDIDIKAN DALAM PERUMUSAN PELAKSANAAN PENCEGAHAN ANAK RAWAN PUTUS SEKOLAH DI KOTA PEKANBARU TAHUN 2013-2015 . *JOM FISIP*, 1-13.

Harahap, P. N., & Sulindawaty. (2019). Implementasi Data Mining Dalam Memprediksi Transaksi Penjualan Menggunakan Algoritma Apriori (Studi Kasus PT.Arma Anugerah Abadi Cabang Sei Rampah). *MATICS : Jurnal Ilmu Komputer dan Teknologi Informasi* , 46-50.

Lestari, W. (2019). Clustering Data Mahasiswa Menggunakan Algoritma K-Means Untuk Menunjang Strategi Promosi (Studi Kasus : STMIK Bina Bangsa Kendari) . *SIMKOM*, 35-48.

Mahardika, B. T. (2020). PERANCANGAN SISTEM INFORMASI MANAGEMENT SISWA BERPRESTASI BERBASIS ANDROID PADA SMK PGRI RAWALUMBU. *Program Studi Teknologi Informasi Universitas Darma Persada*, 30-39.

Maori, N. A. (2023). METODE ELBOW DALAM OPTIMASI JUMLAH CLUSTER PADA K-MEANS CLUSTERING. *Jurnal SIMETRIS*, 277-287.

Martias, L. D. (2021). STATISTIKA DESKRIPTIF SEBAGAI KUMPULAN INFORMASI. *FIHRIS: Jurnal Ilmu Perpustakaan dan Informasi*, 40-59.

Muhammad, A. F. (2022). KLASTERISASI PROSES SELEKSI PEMAIN MENGGUNAKAN ALGORITMA K-MEANS (Study Kasus : Tim Hockey Kabupaten Kendal). *Jurusan Teknik Informatika FIK UDINUS*, 1-5.

Muttaqin, F. A., & Bachtiar, A. M. (2022). IMPLEMENTASI TEKS MINING PADA APLIKASI PENGAWASAN PENGGUNAAN INTERNET ANAK "DODO KIDS BROWSER". *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, 112-116.

Novian, A. (2014). FAKTOR YANG BERHUBUNGAN DENGAN KEPATUHAN DIIT PASIEN HIPERTENSI (Studi Pada Pasien Rawat Jalan di Rumah Sakit Islam Sultan Agung Semarang Tahun. *Unnes Journal of Public Health*, 1-9.

Nurdiansah, & Irmawati. (2020). Android Aplikasi Titip Gadai Elektronik Berbasis Android pada CV. Irsaf Maspul Sejahtera. *PROSIDING SEMINAR ILMIAH SISTEM INFORMASI DAN TEKNOLOGI INFORMASI*, 23-32.

Pohan, R. F., Ratnawati, D. E., & Arwani, I. (2022). Implementasi Algoritma Support Vector Machine dan Model Bag-of-Words dalam Analisis Sentimen mengenai PILKADA 2020 pada Pengguna Twitter. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* , 2924-4931.

Prasetyaningsih, N., Muiz, A., & Fatimah. (2024). Penggunaan Platform Merdeka Mengajar (PMM) untuk Peningkatan Kompetensi Guru di Sekolah Dasar. *JURNAL BASICEDU, 8*(1), 789-798.

Pratala, C. T., Asyer, E. M., Prayudi, I., & Saifudin, A. (2020). Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path . *Jurnal Informatika Universitas Pamulang*, 111-119.

Radhian, D., & Afrianto, I. (2019). PEMBANGUNAN APLIKASI CHATBOT SEBAGAI MEDIA PENCARIAN INFORMASI DALAM BIDANG PETERNAKAN. *E-Library UNIKOM*, 1-10.

Rahmayanti, D. A., Juita, R., & Suhendra, C. D. (2022). Penerapan Metode K-Means untuk Clustering Data Anak Berdasarkan Kepemilikan Akta Kelahiran dan KIA. *Informatics Journal*, 210-219.

Riza, A. A., & Saputro, D. R. (2022). CLUSTERING DATA NUMERIK MENGGUNAKAN ALGORITME X-MEANS. *Seminar Nasional Matematika, Geometri, Statistika, dan Komputasi*, 30-35.

Sasmita, R. A., & Falani, A. Z. (2018). PEMANFAATAN ALGORITMA TF/IDF PADA SISTEM INFORMASI ECOMPLAINT HANDLING. *JURNAL LINK*, 27-33.

Septiani, D., & Isabela, I. (2022). ANALISIS TERM FREQUENCY INVERSE DOCUMENT FREQUENCY (TF-IDF) DALAM TEMU KEMBALI INFORMASI PADA DOKUMEN TEKS. *SINTESIA: Jurnal Sistem dan Teknologi Informasi Indonesia*, 81-88.

Susanto, H. (2022). Tugas dan Tanggung Jawab Dinas Pendidikan Menuju Tertib Administrasi Pendidikan Yang Efisien dan Produktif . *Journal of Innovation in Teaching and Instructional Media*, 116-121.

Umami, D. A. (2019). HUBUNGAN MEDIA PEMBELAJARAN DAN MINAT TERHADAP MOTIVASI MAHASISWI TINGKAT IIIKEBIDANAN WIDYA KARSA JAYAKARTA. *Journal Of Midwifery*, 6-16.

Wahyudi, B., & Kuswandi, I. (2022). Prediksi Peringkat Aplikasi di Google Play Menggunakan Metode Random Forest . *Jurnal Nasional Teknologi Komputer*, 38-47.

Wardana, M. A., Indra, D. P., & Ulya, C. (2023). ANALISIS PENGGUNAAN APLIKASI MERDEKA BELAJAR OLEH GURU BAHASA INDONESIA DI SMP SURAKARTA SEBAGAI AKSELERASI IMPLEMENTASI KURIKULUM MERDEKA . *Jurnal Pendidikan Bahasa dan Sastra Indonesia*, 209-220.

Zen, H. R., & Nuryasin, I. (2024). PENERAPAN WHITEBOX TESTING PADA PENGUJIAN SISTEM MENGGUNAKAN TEKNIK BASIS PATH. *JOISIE (Journal Of Information Systems And Informatics Engineering)*, 101-111.

LAMPIRAN

Lampiran 1 Jadwal Penelitian

NO	Kegiatan	Bulan/Tahun																			
		April 2024				Mei 2024				Juni 2024				Juli 2024				Agustus 2024			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Konsultasi Proposal																				
2	Prosesi Bimbingan																				
3	Persetujuan Proposal Seminar I																				
4	Seminar I																				
5	Pembuatan Program																				
6	Revisi Hasil Seminar I																				
7	Persiapan Seminar II																				
8	Persetujuan Maju Seminar II																				
9	Seminar II																				
10	Revisi Hasil Seminar II																				
11	Persiapan Ujian Pendadaran																				
12	Ujian Pendadaran																				
13	Revisi Hasil Pendadaran																				
14	Pendaftaran Wisuda																				
15	Prosesi Yudisium dan Wisuda																				

- Sudah Dilaksanakan
- Sedang Dilaksanakan
- Akan Dilaksanakan

Lampiran 2 Surat Penelitian

PEMERINTAH KOTA SAMARINDA
DINAS PENDIDIKAN DAN KEBUDAYAAN
Jalan Biola Nomor 4A, Sungai Pinang Luar, Samarinda Kota, Samarinda 75123
Telepon/Faksimile (0541) 742368; Telepon Pengaduan 082252656265
Laman https://disdik.samarindakota.go.id; Pos-el : disdik.samarindakota@gmail.com

SURAT IZIN
NOMOR : 500.10.30/2589/100.01
TENTANG
IZIN MELAKSANAKAN PENELITIAN

Dasar : a. Surat dari STMIK Widya Cipta Dharma Nomor: 092/Um-Ks/VI/2024, Tanggal 10 Juni 2024

MEMBERIKAN IZIN

Nama	:	**Nur Syamsu Wais Al Qorni**
NIM	:	22.43.905
Jenjang Studi	:	S1
Program Studi	:	Teknik Informatika
Keperluan	:	Melaksanakan Penelitian (*Research*) di lingkungan Dinas Pendidikan dan Kebudayaan Kota Samarinda tentang Aplikasi PMM (*Platform* Merdeka Mengajar)

Samarinda, 2 Juli 2024
Kepala

Dr. H. Asli Nuryadin, S.Pd., MM.
Pembina Utama Muda /IV/c
Nip [illegible] 198601 1 001

Catatan:
Setelah melaksanakan Penelitian, agar menyampaikan
Hasil penelitian dalam bentuk soft copy (PDF)
Dikirimkan ke link : bit.ly/hasilpenelitiandisdik

Lampiran 3 Laporan Wawancara dengan Pihak Instansi

LAPORAN KETERANGAN WAWANCARA

Tanggal Wawancara : Juli 2024

Waktu Wawancara :

Lokasi Wawancara : Kantor Dinas Pendidikan dan Kebudayaan Kota Samarinda (Jl. Biola No.4A, Sungai Pinang Luar, Kec. Samarinda Kota, Kota Samarinda, Kalimantan Timur 75123)

Profil Narasumber

Nama : Maruna Dalis Mufa

NIK : 647105 2007 99 0001

Jabatan : Staf TU

E-Mail :

Nomor HP :

Dalam daftar pertanyaan wawancara dibawah ini bertujuan sebagai salah satu bahan untuk pendukung penelitian pada skripsi peneliti yang berjudul "Penerapan Algoritma *K-Means Clustering* Pada Ulasan Pengguna aplikasi Merdeka Mengajar di *Play Store*". Berikut daftar pertanyaan wawancara untuk pihak Dinas Pendidikan dan Kebudayaan Kota Samarinda :

Dipindai dengan CamScanner

Daftar Pertanyaan :

1. Bagaimana pendapat guru di kota ini tentang penggunaan aplikasi digital untuk mengajar dan belajar?

   Sangat membantu dan memudahkan guru dalam mengembangkan kemampuan di dunia pendidikan.

2. Apa saja masalah utama yang dihadapi guru di kota ini saat menggunakan aplikasi digital untuk mengajar?

   Materi yang di tampilkan agak sulit untuk di organisir dan harus bersabar jika terjadi masalah.

3. Apakah Dinas Pendidikan sering menerima keluhan atau masukan tentang aplikasi digital untuk mengajar? Jika ya, apa masalah yang paling sering dihadapi?

   Ada masalah dalam hal materi yang kurang berpengaruh dengan jaringan.

4. Bagaimana Dinas Pendidikan menanggapi dan menyelesaikan keluhan atau masalah yang dilaporkan oleh guru tentang aplikasi digital?

   Kami melakukannya pelatihan tambahan dan selalu berkoordinasi dengan pengembang aplikasi

Dipindai dengan CamScanner

5. Apakah ada fitur dalam aplikasi digital yang sering dilaporkan bermasalah oleh guru di kota ini?

Fitur yang bermasalah yaitu validasi yang lama dan fitur evaluasi yang sulit.

6. Bagaimana Dinas Pendidikan menggunakan data dari ulasan pengguna untuk memperbaiki aplikasi digital?

Kami mengambil ulasan dari para pengguna dan selalu berkomunikasi dengan pengembang terkait dengan kekurangan yang di keluhkan pengguna.

7. Seberapa penting umpan balik dari guru dalam pengembangan dan perbaikan aplikasi digital untuk mengajar?

Sangat penting karena mampu memenuhi kebutuhan secara langsung.

8. Apakah Dinas Pendidikan pernah melakukan analisis khusus terhadap ulasan pengguna aplikasi digital di *Google Play Store* ? Jika ya, apa temuan utamanya?

Dari kami mendapatkan bahwa pengguna menghargai fungsi dasar dan menginginkan perbaikan kecepatan dan lainnya.

Dipindai dengan CamScanner

9. Apa rekomendasi utama dari Dinas Pendidikan berdasarkan ulasan pengguna di *Google Play Store* untuk meningkatkan aplikasi digital untuk mengajar?

Rekomendasi kami yaitu upgrade stabilitas aplikasi, proses validasi & proses login.

10. Saat ini saya selaku Mahasiswa Sekolah Tinggi, sedang melakukan penelitian skripsi terkait analisis *clustering* terhadap ulasan salah satu aplikasi dari Kemendikbudristek yaitu Merdeka Mengajar di *Google Play Store*, bagaimana tanggapan Bapak/Ibu mengenai itu ?

Kami sangat mendukung penelitian ini karena membantu Dinas Pendidikan dalam mengetahui review dalam hal aplikasi PMM secara menyeluruh.

Samarinda, 24 Juli 2024

Narasumber

PEMERINTAH KOTA SAMARINDA

Nama Jelas Muda.

Dipindai dengan CamScanner

DOKUMENTASI WAWANCARA

<table>
<tr><td>  </td><td>  </td></tr>
<tr><td></td><td></td></tr>
<tr><td>  </td><td>  </td></tr>
<tr><td></td><td></td></tr>
</table>