Evaluasi Metode *Naive Bayes Classifier* dan *Support Vector Machine* untuk Analisis Sentimen Aplikasi AdaKami di Google Playstore

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Syarat Guna Memperoleh
Gelar Sarjana Strata Satu (S.1) dalam Ilmu Teknologi Informasi

Diajukan Oleh:

Mirza Izzal Musyaffaq

NIM. 2108096040

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG

2025

PERNYATAAN KEASLIAN

Yang bertandatangan dibawah ini:

Nama : Mirza Izzal Musyaffaq

NIM : 2108096040

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

Evaluasi Metode *Naive Bayes Classifier* dan *Support Vector Machine* untuk Analisis Sentimen Aplikasi AdaKami di Google Playstore

Secara keseluruhan adalah hasil penelitian/karya saya sendiri, kecuali bagian tertentu yang dirujuk dari sumbernya.

Semarang, 2 Juni 2025

Pembuat Pernyataan,

Mirza Izzal Musyaffaq

NIM: 2108096040

KEMENTRIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Ngaliyan Semarang
Telp.024-7601259 Fax. 7615387

LEMBAR PENGESAHAN

Naskah skripsi berikut ini:

Judul	:	Evaluasi Metode *Naive Bayes Classifier* dan *Support Vector Machine* untuk Analisis Sentimen Aplikasi AdaKami di Google Playstore
Penulis	:	Mirza Izzal Musyaffaq
NIM	:	2108096040
Jurusan	:	Teknologi Informasi

Telah diujikan dalam sidang *tugas akhir* oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam Ilmu Teknologi Informasi.

Semarang, 2 Juni 2025

DEWAN PENGUJI

Ketua Sidang	Sekretaris Sidang
Dr. Masy Ari Ulinuha, M.T.	Dr. Khotibul Umam, M.Kom.
NIP. 198108122011011007	NIP. 197908272011011007
Penguji Utama I	Penguji Utama II
Dr. Wenty Dwi Yuniarti, M.Kom.	Mukhamad Iklil Mustofa, M.Kom.
NIP. 197706222006042005	NIP. 198808072019031010
Pembimbing I	Pembimbing II
Dr. Khotibul Umam, M.Kom.	Hery Mustofa, M.Kom.
NIP. 197908272011011007	NIP. 198703172019031007

NOTA PEMBIMBING

Semarang. 15 April 2025

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamualaikum. wr. Wb

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan;

Judul : **Evaluasi Metode *Naive Bayes Classifier* dan *Support Vector Machine* untuk Analisis Sentimen Aplikasi AdaKami di Google Playstore**

Penulis : **Mirza Izzal Musyaffaq**
NIM : 2108096040
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang untuk diujikan dalam Sidang Munaqosyah.

Wassalamualaikum, wr. wb.

Pembimbing I,

Dr. Khotibul Umam, M.Kom
NIP. 197908272011011007

NOTA PEMBIMBING

Semarang. 15 April 2025

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamualaikum. wr. Wb

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan;

Judul : **Evaluasi Metode *Naive Bayes Classifier* dan *Support Vector Machine* untuk Analisis Sentimen Aplikasi AdaKami di Google Playstore**

Penulis : **Mirza Izzal Musyaffaq**
NIM : 2108096040
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang untuk diujikan dalam Sidang Munaqosyah.

Wassalamualaikum, wr. wb.

Pembimbing II,

Hery Mustofa, M.Kom
NIP. 198703172019031007

MOTO

"Hidup yang tidak dipertaruhkan tidak akan pernah dimenangkan."

Sutan Sjahrir

ABSTRAK

Perkembangan teknologi informasi telah mendorong kemajuan pesat dalam sektor keuangan, terutama melalui *Financial Technology* (Fintech). Salah satu platform fintech yang sukses di Indonesia adalah AdaKami. Popularitas AdaKami menghasilkan berbagai ulasan pengguna di Google Play Store, mencakup sentimen positif maupun negatif. Penelitian ini bertujuan untuk membandingkan performa algoritma *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) dalam analisis sentimen ulasan pengguna aplikasi AdaKami di *Google Play Store*. Analisis dilakukan dengan mengukur akurasi, presisi, *recall*, *F1-score*, waktu penyelesaian, dan efisiensi sumber daya. Data ulasan dikumpulkan melalui proses *web scraping* dan diproses menggunakan teknik *preprocessing* seperti *case folding*, *normalization*, *tokenizing*, *stopword removal*, dan *stemming*. Pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) digunakan untuk mengekstraksi fitur dari data ulasan. Hasil penelitian menunjukkan bahwa SVM memiliki keunggulan dalam akurasi (0,95), presisi positif (0,95), presisi negatif (0,94), *recall* positif (0,94), *recall* negatif (0,96), *F1-score* positif (0,95), dan *F1-score* negatif (0,95) dibandingkan akurasi (0,94), presisi positif (0,94), presisi negatif (0,93), *recall* positif (0,93), *recall* negatif (0,94), *F1-score* positif (0,93), dan *F1-score* negatif (0,94) NBC. Namun, NBC lebih unggul dalam waktu pelatihan (0,0087 detik), waktu prediksi (0,0006 detik), dan penggunaan memori puncak (0,26 MiB) dibandingkan waktu pelatihan (0,2476 detik), waktu prediksi (0,0485 detik), dan penggunaan memori puncak (1,47 MiB). Penelitian ini memberikan wawasan penting bagi pengembang aplikasi fintech dalam memilih metode analisis sentimen yang sesuai dengan kebutuhan mereka untuk meningkatkan kualitas layanan berdasarkan umpan balik pengguna secara lebih efektif.

Kata Kunci: *Analisis Sentimen, Naive Bayes Classifier (NBC), Support Vector Machine (SVM), AdaKami, Evaluasi Performa.*

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur senantiasa penulis panjatkan ke hadirat Allah SWT atas limpahan nikmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Evaluasi Metode *Naive Bayes Classifier* dan *Support Vector Machine* untuk Analisis Sentimen Aplikasi AdaKami di Google Playstore" dengan baik. Skripsi ini disusun sebagai salah satu syarat kelulusan dalam program sarjana (S1) pada Program Studi Teknologi Informasi di Universitas Islam Negeri Walisongo, Kota Semarang. Selain itu, penelitian ini juga bertujuan untuk memberikan wawasan bagi pembaca terhadap penelitian yang dikaji.

Pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan dan bantuan, baik dalam pelaksanaan penelitian maupun dalam penyusunan skripsi ini. Penulis menyadari bahwa tanpa bimbingan, arahan, serta motivasi dari berbagai pihak, proses penyelesaian skripsi ini tidak akan berjalan dengan lancar. Oleh karena itu, penulis mengucapkan penghargaan dan rasa terima kasih yang mendalam kepada:

1. Bapak Prof. Dr. Nizar, M.Ag selaku Rektor Universitas Islam Negeri Walisongo Semarang.

2. Bapak Prof. Dr. H. Musahadi, M.Ag sekalu Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo Semarang.
3. Bapak Dr. Khotibul Umam, M.Kom selaku Ketua Program studi Teknologi Informasi sekaligus dosen wali dan pembimbing pertama yang telah memberikan bimbingan sejak awal mengikuti perkuliahan hingga skripsi ini terselesaikan.
4. Bapak Hery Mustofa, M,Kom, selaku dosen Pembimbing II yang telah banyak memberi masukan dan arahan yang membuka pikiran penulis dalam penyusunanan skripsi dengan sabar.
5. Staff, karyawan dan dosen di lingkungan Universitas Islam Negeri Walisongo Semarang.
6. Orang tua tercinta dan keluarga yang selalu menemani dalam membantu penulis dan selalu mendo'akan serta memberikan dukungan kepada penulis.
7. Teman-teman Jamaah Al-Remi'ah yang selalu memberi dukungan.
8. Semua pihak yang tidak bisa penulis sebutkan satu persatu yang terlibat dalam pembuatan skripsi ini sehingga dapat terselesaikan dengan baik.

Dalam proses pelaksanaan dan penyusunan skripsi ini, penulis menyadari bahwa masih terdapat berbagai

kekurangan dan ketidaksempurnaan. Oleh karena itu, penulis sangat mengharapkan kritik serta saran yang membangun guna meningkatkan kualitas penulisan ini. Semoga skripsi ini dapat memberikan manfaat bagi semua pihak yang membacanya.

Semarang, 2 Juni 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi informasi dan komunikasi telah mengubah secara mendasar berbagai aspek kehidupan manusia, termasuk sektor keuangan. Di era digital ini, *Financial Technology* (Fintech) muncul sebagai inovasi revolusioner yang mengintegrasikan teknologi dengan layanan keuangan, sehingga memudahkan akses dan penggunaan layanan tersebut oleh masyarakat luas. Di Indonesia, fintech telah menjadi salah satu solusi utama dalam memenuhi kebutuhan keuangan, terutama dalam hal pemberian kredit pinjaman. Sistem peminjaman berbasis digital ini menawarkan kemudahan, kecepatan, dan efisiensi yang tidak dapat ditawarkan oleh lembaga keuangan konvensional, yang pada gilirannya menjadi daya tarik bagi masyarakat untuk beralih ke fintech (Hidayat et al., 2022).

Salah satu contoh sukses fintech di Indonesia adalah AdaKami, sebuah platform pinjaman online yang telah menjadi pilihan utama bagi banyak masyarakat Indonesia. Dengan lebih dari 10 juta pengguna dan *rating* 4,4 di *Google Play Store*, AdaKami berhasil menyalurkan pinjaman sebesar Rp 1,31 triliun, menjadikannya *platform* dengan penyaluran pinjaman

terbanyak ketiga di Indonesia, setelah Lentera Dana Nusantara dan EasyCash. Menurut Roadmap Pengembangan dan Penguatan Layanan Pendanaan Bersama Berbasis Teknologi Informasi (LPBBTI) 2023-2028 yang dikeluarkan oleh Otoritas Jasa Keuangan (OJK), keberhasilan ini menunjukkan tingginya kepercayaan masyarakat terhadap layanan yang ditawarkan oleh AdaKami, serta peran pentingnya dalam inklusi keuangan di Indonesia.

Namun, popularitas yang tinggi ini juga menghadirkan tantangan tersendiri. Seiring dengan meningkatnya jumlah pengguna, semakin banyak ulasan dan komentar yang muncul di *platform-platform* seperti *Google Play Store, App Store*, dan media sosial. Ulasan tersebut mencerminkan beragam pengalaman pengguna, mulai dari komentar positif tentang kemudahan proses pinjaman hingga kritik mengenai bunga yang tinggi dan layanan pelanggan yang dianggap kurang memuaskan. Keberagaman ulasan ini memerlukan perhatian serius dari pihak AdaKami, karena dapat berdampak langsung pada reputasi dan kelangsungan bisnis platform tersebut.

Dalam konteks ini, analisis sentimen muncul sebagai metode yang penting untuk memahami persepsi dan perasaan pengguna terhadap produk atau layanan tertentu. Dengan menggunakan analisis sentimen, perusahaan dapat mengidentifikasi tren sentimen positif dan negatif dari ulasan

pengguna, yang kemudian dapat digunakan untuk mengambil keputusan yang lebih baik dalam meningkatkan kualitas layanan dan kepuasan pengguna. Selain itu, analisis sentimen juga memungkinkan perusahaan untuk merespons umpan balik pengguna secara lebih proaktif dan efisien, serta mengidentifikasi peluang untuk perbaikan yang dapat meningkatkan reputasi dan keberlanjutan bisnis (Kusuma et al., 2024).

Sebagaimana dinyatakan dalam QS. Al-Hujurat ayat 6:

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اِنْ جَاۤءَكُمْ فَاسِقٌۢ بِنَبَاٍ فَتَبَيَّنُوْٓا اَنْ تُصِيْبُوْا قَوْمًاۢ بِجَهَالَةٍ فَتُصْبِحُوْا عَلٰى مَا فَعَلْتُمْ نٰدِمِيْنَ ۝٦

Artinya: "Wahai orang-orang yang beriman, jika seorang fasik datang kepadamu membawa berita penting, maka telitilah kebenarannya agar kamu tidak mencelakakan suatu kaum karena ketidaktahuan(-mu) yang berakibat kamu menyesali perbuatanmu itu" (Q.S. Al-Hujurat: 6).

Ayat ini menggarisbawahi pentingnya memeriksa dan menilai informasi dengan hati-hati sebelum membuat keputusan atau kesimpulan. Prinsip ini sejalan dengan proses analisis sentimen, di mana data dianalisis secara mendalam untuk mendapatkan hasil yang akurat dan dapat diandalkan.

Dalam analisis sentimen, berbagai metode *machine learning* digunakan untuk mengklasifikasikan data ulasan pengguna ke dalam kategori sentimen positif dan negatif. *Naive Bayes* adalah salah satu metode pengklasifikasian yang paling

sederhana namun efektif, yang sering digunakan karena kemudahan penerapannya dan kemampuan menghasilkan hasil yang baik dalam berbagai kasus. Algoritma *Naive Bayes Classifier* bekerja berdasarkan prinsip probabilitas, di mana setiap kata dalam dokumen dianalisis untuk menentukan kemungkinan kata tersebut termasuk dalam kategori positif atau negatif. Pembobotan menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF) membantu dalam memberikan nilai lebih pada kata-kata yang lebih penting dalam dokumen (Siregar et al., 2020).

Selain itu, *Support Vector Machine* (SVM) merupakan metode yang sangat kuat dalam menangani data yang kompleks dan berjumlah besar, termasuk dalam konteks analisis sentimen. SVM bekerja dengan mencari *hyperplane* optimal yang dapat memisahkan dua kelas data dengan margin maksimum, sehingga memungkinkan klasifikasi yang lebih akurat dan efisien. Dalam konteks analisis sentimen, SVM memungkinkan pemisahan yang jelas antara ulasan positif dan negatif, yang penting untuk menghasilkan *insight* yang bermanfaat bagi pengambilan keputusan strategis (Muttaqin & Kharisudin, 2021).

Sebagaimana disebutkan dalam QS. Az-Zumar ayat 18:

الَّذِيْنَ يَسْتَمِعُوْنَ الْقَوْلَ فَيَتَّبِعُوْنَ اَحْسَنَهٗۗ اُولٰۤىِٕكَ الَّذِيْنَ هَدٰىهُمُ اللّٰهُ وَاُولٰۤىِٕكَ هُمْ اُولُوا الْاَلْبَابِ ﴿١٨﴾

> *Artinya: "(Yaitu) mereka yang mendengarkan perkataan lalu mengikuti apa yang paling baik di antaranya. Mereka itulah orang-orang yang telah diberi petunjuk oleh Allah dan mereka itulah ululalbab (orang-orang yang mempunyai akal sehat)" (Q.S. Az-Zumar: 18).*

Ayat ini menekankan pentingnya menggunakan akal dan penilaian yang bijaksana dalam menerima dan memproses informasi. Hal ini relevan dengan penggunaan metode seperti Naive Bayes dan SVM dalam analisis sentimen, di mana pendekatan yang sistematis dan terukur digunakan untuk memastikan bahwa informasi diolah dengan cara yang paling efektif dan akurat.

Penelitian ini bertujuan untuk membandingkan metode *Naive Bayes* dan SVM dalam menganalisis sentimen pengguna aplikasi AdaKami. Dengan membandingkan kinerja kedua metode ini, penelitian ini diharapkan dapat mengidentifikasi metode mana yang lebih efektif dalam mengklasifikasikan sentimen ulasan pengguna. Hasil dari penelitian ini diharapkan tidak hanya memberikan wawasan yang berharga bagi pengembangan strategi bisnis AdaKami tetapi juga memberikan kontribusi signifikan terhadap literatur akademik dalam bidang analisis sentimen dan penerapan *machine learning*.

B. Identifikasi Masalah

1. Kebutuhan untuk memahami beragamnya persepsi pengguna dalam ulasan aplikasi di *google play store.*
2. Kebutuhan akan metode analisis sentimen yang efektif.

C. Rumusan Masalah

1. Bagaimana kinerja metode *Naive Bayes* dalam menganalisis sentimen ulasan pengguna aplikasi AdaKami di *Google Play Store*?
2. Bagaimana kinerja metode *Support Vector Machine* (SVM) dalam menganalisis sentimen ulasan pengguna aplikasi AdaKami di *Google Play Store*?
3. Di antara metode *Naive Bayes* dan SVM, metode manakah yang lebih efektif dalam menganalisis sentimen ulasan pengguna aplikasi AdaKami?

D. Batasan Masalah

Sebagai bentuk antisipasi agar materi yang dikaji dalam penelitian ini tidak terlalu luas, maka diberikan batasan masalah sebagai berikut:

1. Data Ulasan: Penelitian ini hanya akan menggunakan data ulasan pengguna aplikasi AdaKami yang diambil dari *platform Google Play Store.*
2. Metode Analisis Sentimen: Penelitian ini akan fokus pada penggunaan dua metode analisis sentimen, yaitu *Naive Bayes* dan *Support Vector Machine* (SVM).

3. Klasifikasi Sentimen: Sentimen ulasan akan diklasifikasikan ke dalam dua kategori utama, yaitu positif dan negatif.
4. Periode Pengumpulan Data: Data ulasan yang akan dianalisis akan dibatasi pada bulan september 2024 sampai februari 2025
5. Pengukuran Kinerja: Kinerja kedua metode akan diukur berdasarkan metrik umum dalam machine learning seperti akurasi, presisi, *recall*, dan *F1-score* serta waktu penyelesaian dan efisiensi sumber daya.

E. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menganalisis dan mengukur kinerja metode *Naive Bayes* dalam menganalisis sentimen ulasan pengguna aplikasi AdaKami di *Google Play Store*.
2. Menganalisis dan mengukur kinerja metode *Support Vector Machine* (SVM) dalam menganalisis sentimen ulasan pengguna aplikasi AdaKami di *Google Play Store*.
3. Membandingkan kinerja metode Naive Bayes dan SVM dalam menganalisis sentimen ulasan pengguna, guna menentukan metode yang lebih efektif dan akurat dalam konteks analisis sentimen pada aplikasi fintech.

F. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Manfaat Praktis
   a. Dengan memahami sentimen pengguna, AdaKami dapat lebih responsif terhadap umpan balik pengguna, yang pada akhirnya dapat meningkatkan kualitas layanan dan reputasi perusahaan di pasar.
   b. Hasil analisis sentimen yang akurat akan membantu manajemen AdaKami dalam membuat keputusan yang lebih baik terkait strategi pengembangan produk dan layanan.
2. Manfaat Teoritis
   a. Penelitian ini diharapkan dapat memperkaya literatur dalam bidang analisis sentimen, khususnya dalam penerapan metode NBC dan SVM pada konteks fintech.
   b. Dengan membandingkan dua metode *machine learning* (NBC dan SVM), penelitian ini memberikan wawasan baru tentang keunggulan dan kelemahan masing-masing metode dalam mengklasifikasikan sentimen ulasan pengguna.

BAB II
LANDASAN PUSTAKA

A. Kajian Teori

1. Analisis Sentimen

Analisis sentimen merupakan bagian dari *Natural Language Processing* (NLP) yang memungkinkan komputer bekerja dan berpikir seperti manusia. Dalam penerapannya dibutuhkan *machine learning* sebagai algoritma penyelesaian untuk mengambil keputusan berdasarkan data yang telah diolah (Salsabila, 2022).

Analisis sentimen atau *opinion mining* adalah proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terdapat dalam suatu kalimat opini. Analisis sentimen memiliki tujuan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang, apakah mengarah ke opini positif atau negatif (Zidan, 2022).

Analisis sentimen digunakan untuk menilai teks atau data mengandung sentimen positif, negatif, atau netral. Dalam analisis sentimen menerapkan teknik pemrosesan bahasa alami, pembelajaran mesin, dan algoritma klasifikasi untuk mendeteksi ekspresi emosional dan

penilaian dalam teks seperti ulasan pengguna, tweet, dan artikel berita (Eviyanti et al., 2023).

2. AdaKami

AdaKami merupakan sebuah *platform peer-to-peer lending* yang dinaungi oleh PT Pembiayaan Digital Indonesia. AdaKami menawarkan layanan pinjaman online tanpa agunan dengan proses pengajuan yang mudah dan cepat. Pengguna hanya perlu menyiapkan KTP, akun bank aktif, dan mengisi data pribadi melalui aplikasi, pengguna tinggal menunggu proses persetujuan maksimal 48 jam pengajuan pinjaman (AdaKami, t.thn.).

AdaKami juga menjamin perlindungan data pribadi pengguna sehingga memberikan rasa aman dalam menggunakan layanan pengguna. Transaksi dalam aplikasi AdaKami terjamin karena diawasi oleh OJK (Otoritas Jasa Keuangan) dengan surat nomor KEP-128/D.05/2019 (Syafutri, 2023).

3. *Naive Bayes Classifier* (NBC)

NBC merupakan salah satu metode *machine learning* yang memanfaatkan perhitungan probabilitas dan statistik. cara kerja naive bayes adalah dengan memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya (Zidan, 2022).

NBC tidak bisa mendeteksi gambar, tetapi hanya bisa mendeteksi teks dan numerik. perhitungan probabilitas dalam metode ini menggunakan pendekatan teorema bayes. Teorema bayes adalah pengenalan pola melalui pendekatan statistik yang fundamental. Teorema bayes dapat dideskripsikan seperti probabilitas terjadinya hubungan A dengan syarat hubungan B sudah terjadi, begitupun sebaliknya (Salsabila, 2022).

Klasifikasi naive bayes termasuk pengklasifikasian statistik yang dapat digunakan untuk prediksi probabilitas keanggotaan suatu class, klasifikasi ini berdasarkan teorema bayes yang memiliki kemampuan klasifikasi serupa dengan *decision tree* dan *neural network*. penggunaan klasifikasi ini memiliki waktu pemrosesan yang cepat dan mudah diimplementasikan dengan struktur yang cukup sederhana dan efektif (Fitri et al., 2020).

Rumus naive bayes secara umum adalah sebagai berikut:

$$P(Y|X) = \frac{P(X|Y).P(Y)}{P(X)} \quad (2.1)$$

Keterangan:

Y = Hipotesis data X dari kelas yang spesifik

X = Data dengan kelas yang belum diketahui

P(Y|X) = Probabilitas hipotesis H berdasarkan kondisi X (posterior)

P(X|Y) = Probabilitas X berdasarkan kondisi tersebut (likelihood)

P(X) = Probabilitas hipotesis H (prior)

P(Y) = Probabilitas dari X (Evidance)

4. *Support Vector Machine* (SVM)

Support vector machine (SVM) merupakan suatu teknik klasifikasi yang efisien untuk masalah *nonlinear*. SVM dikenal sebagai teknik pembelajaran mesin untuk mengenali pola dengan menggunakan pasangan data *input* dan data *output* berupa sasaran yang diinginkan. Sederhananya, konsep SVM adalah mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada *input space* dengan memaksimalkan jarak antar kelas (Rokhman et al., 2021).

Support Vector Machine (SVM) merupakan sebuah metode yang baru namun memiliki performa yang lebih baik dalam berbagai bidang aplikasi seperti klasifikasi teks, pengenalan tulisan tangan, bioinformatika, dan lain sebagainya. Metode ini digunakan untuk menganalisa data dan mengenali pola klasifikasi. Dalam penggunaannya, metode ini mengharuskan menggunakan pelabelan, yaitu pelabelan kalimat positif dan negatif (Effendi, 2023).

Support Vector Machine mampu bekerja pada dataset yang berdimensi tinggi dengan menggunakan kernel trik. *Support Vector Machine* hanya menggunakan beberapa titik data terpilih yang berkontribusi (*support vector*) untuk membentuk model yang digunakan dalam proses klasifikasi (Julianto et al., 2022).

Persamaan SVM:

$$f(x) = w.x + b \quad (2.2)$$

dimana:

w : Parameter *hyperplane* yang dicari (garis yang tegak lurus antara garis *hyperplane* dan titik support vector)

x : Titik data masukan *support vector machine*

b : Parameter hyperplane yang dicari (nilai bias)

sehingga diperoleh persamaan sebagai berikut:

$$\begin{gathered}[(w^T.x_i) + b] \geq 1 \; untuk \; y_i = +1 \\ [(w^T.x_i) + b] \leq 1 \; untuk \; y_i = -1 \\ [(w^T.x_i) + b] = 0 \; untuk \; y_i = 0\end{gathered} \quad (2.3)$$

dimana:

x_i : Himpunan data *training*

i : 1,2,.....,n

y_i : Label kelas dari x_i

B. Kajian Penelitian yang Relevan

Penelitian ini membutuhkan rujukan sebagai bahan informasi lain yang bertujuan untuk mendukung penelitian dan mendapatkan perbandingan antara penelitian satu dengan penelitian-penelitian lainnya. Beberapa penelitian yang relevan dengan penelitian ini adalah:

Tabel 2. 1 Penelitian Terdahulu

Penelitian, Tahun	Topik Penelitian	Metode	Keterangan
(Zidan, 2022)	Analisis Sentimen Kenaikan Harga Bahan Bakar Minyak (BBM) Berdasarkan Respon Pengguna Media Sosial Twitter	Naive Bayes	Sentimen negatif mendominasi dengan persentase 53,8%, diikuti oleh sentimen positif sebesar 37,1%, dan sentimen netral sebesar 9,1%. Selain itu, model Naive Bayes juga menunjukkan performa yang baik dengan akurasi 81%, precision 83%, recall 81%, dan F1 score 79%.
(Fitri et al., 2020)	Analisis Sentimen Aplikasi Ruangguru	Random Forest, Naive Bayes, SVM	Metode random forest menunjukkan performa terbaik dengan akurasi 97,16% dan nilai AUC sebesar 0,996. SVM memperoleh akurasi

			96,01% dengan AUC 0,543, sedangkan Naive Bayes mencapai akurasi 94,16% dan AUC 0,999. Metode yang diterapkan dalam penelitian ini menunjukkan bahwa analisis sentimen dapat memberikan wawasan yang berguna untuk meningkatkan kualitas aplikasi berdasarkan umpan balik pengguna.
(Muhammadin & Sobari, 2021)	Analisis Sentimen Pada Ulasan Aplikasi Kredivo	SVM dan NBC	Penelitian ini menganalisis sentimen ulasan aplikasi Kredivo menggunakan algoritma Support Vector Machine (SVM) dan Naive Bayes Classifier (NBC) dengan data dari 10.000 ulasan di Google Play Store. Setelah melalui proses preprocessing, data dibagi menjadi 80% untuk pelatihan dan 20% untuk pengujian. Hasilnya, SVM mencapai akurasi 83,3% dengan presisi 77% untuk kelas positif dan 87% untuk kelas negatif, serta recall

			89% dan 73%, sedangkan NBC memperoleh akurasi 80,8% dengan presisi 81% dan 87% untuk kelas positif dan negatif, serta recall 88% dan 79%. Kesimpulannya, SVM lebih efektif dibandingkan NBC dalam mengklasifikasikan sentimen ulasan aplikasi Kredivo.
(Styawati et al., 2021)	Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter	SVM, Decision Tree	Metode Support Vector Machine (SVM) memiliki akurasi yang lebih tinggi dibandingkan dengan metode Decision Tree dalam mengklasifikasikan ulasan aplikasi transportasi online. Dengan menggunakan data ulasan dari Google Play, penelitian ini mencapai akurasi sebesar 90,20% untuk metode SVM, sedangkan metode Decision Tree hanya mencapai akurasi sebesar 89,80%. Hal ini menunjukkan bahwa SVM lebih efektif dalam mengidentifikasi sentimen

			positif dan negatif dari ulasan pengguna aplikasi transportasi online seperti Gojek.

Tabel 2.1 menunjukkan bahwa baik *Naive Bayes Classifier* maupun *Support Vector Machine* (SVM) telah digunakan secara luas dalam analisis sentimen di berbagai konteks, dengan SVM umumnya menunjukkan akurasi yang lebih tinggi dibandingkan NBC.

Penelitian ini bertujuan untuk membandingkan kinerja *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) dalam menganalisis sentimen ulasan pengguna aplikasi Adakami yang diambil dari *Google Play Store*. Meskipun studi sebelumnya telah banyak membandingkan algoritma dalam analisis sentimen, sebagian besar hanya berfokus pada metrik umum seperti akurasi, presisi, *recall*, dan *F1 score*. Penelitian ini juga akan membandingkan waktu komputasi serta efisiensi sumber daya. Oleh karena itu, penelitian ini diharapkan dapat memberikan evaluasi yang lebih menyeluruh terhadap kinerja NBC dan SVM dalam konteks data yang beragam dan spesifik untuk ulasan aplikasi Adakami.

BAB III
METODE PENELITIAN

A. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan desain penelitian yang bersifat deskriptif. Pendekatan kuantitatif dipilih karena penelitian ini bertujuan untuk mengukur dan menganalisis sentimen pengguna terhadap aplikasi AdaKami berdasarkan data yang dikumpulkan dari ulasan di *Google Play store*. Dengan menggunakan metode ini, peneliti dapat mengumpulkan data numerik yang dapat dianalisis secara statistik. Metode deskriptif memungkinkan peneliti untuk memberikan gambaran yang jelas mengenai karakteristik sentimen pengguna, serta membandingkan hasil antara dua metode klasifikasi yaitu *Naive Bayes* dan *Support Vector Machine* (SVM) dalam menganalisis sentimen.

Kombinasi pendekatan kuantitatif dan desain deskriptif dalam penelitian ini memungkinkan peneliti untuk mengumpulkan dan menganalisis data secara sistematis, serta menarik kesimpulan yang valid berdasarkan hasil analisis statistik. Dengan demikian, penelitian ini diharapkan dapat memberikan wawasan yang komprehensif tentang sentimen pengguna terhadap aplikasi AdaKami dan membandingkan kinerja metode NBC dan SVM dalam menganalisis sentimen.

B. Metode Pengumpulan Data

Pengumpulan data dilakukan melalui dua metode sebagai berikut:

1. Studi Pustaka

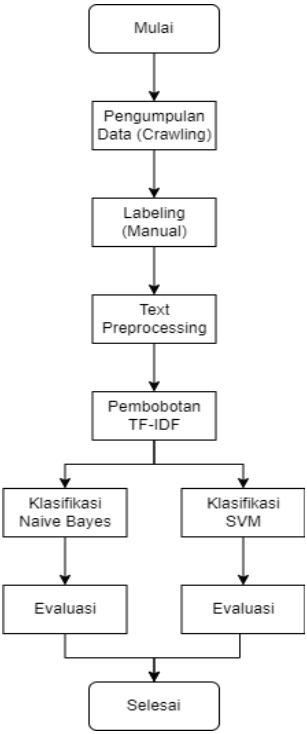
Studi pustaka dilakukan dengan mengumpulkan informasi dari berbagai sumber literatur yang relevan mengenai analisis sentimen, algoritma NBC, dan SVM. Hal ini bertujuan untuk memahami teori dasar serta perkembangan terkini dalam bidang ini.

2. *Crawling* Data

Data ulasan aplikasi AdaKami di *Google Play Store* diambil menggunakan teknik *crawling* dengan periode September 2024 sampai Februari 2025. Proses ini melibatkan pengambilan data secara otomatis dari situs web menggunakan skrip yang dirancang khusus untuk mengumpulkan komentar pengguna beserta *rating* yang diberikan.

C. Perancangan Sistem

Perancangan sistem dilakukan dengan membuat diagram alir yang menggambarkan alur proses analisis sentimen. Sistem ini dirancang untuk memudahkan pengguna dalam memahami langkah-langkah yang diambil dalam penelitian.



Gambar 3. 1 Alur Sistem

Penelitian ini dimulai dengan *crawling* data pada aplikasi AdaKami di *google play store*, data yang diperoleh diberi label secara manual (positif atau negatif). Data yang diperoleh merupakan data mentah yang perlu diolah pada tahap *text processing* meliputi *case folding, normalization, tokenizing, stopword removal,* dan *stemming*. Data yang sudah diolah kemudian diberi bobot tf-idf. Bobot tf-idf digunakan untuk

klasifikasi NBC dan SVM, hasilnya kemudian dievaluasi untuk mengukur kinerja tiap metode.

D. Pelabelan

Pelabelan adalah proses penting dalam analisis sentimen yang melibatkan pemberian label pada data ulasan berdasarkan kategori sentimen yang diinginkan, seperti positif dan negatif. Proses ini dapat dilakukan secara manual atau otomatis. Pelabelan manual biasanya dilakukan oleh ahli bahasa atau peneliti yang memiliki pemahaman mendalam tentang konteks dan nuansa bahasa. Dalam kasus ini, pelabelan manual dilakukan oleh guru bahasa Indonesia yang memiliki keahlian khusus dalam memahami struktur, makna, dan nuansa bahasa, sehingga memastikan akurasi dalam klasifikasi sentimen. Meski pelabelan manual dianggap lebih teliti, metode ini sering kali memakan waktu dan kurang praktis untuk dataset yang besar. Namun, dalam situasi tertentu, pelabelan manual tetap menjadi pilihan utama karena keakuratannya yang tinggi, terutama jika dataset mengandung banyak variasi bahasa yang sulit ditangkap oleh algoritma otomatis.

E. *Preprocessing* Data

Preprocessing data adalah tahap penting dalam analisis teks yang bertujuan untuk menyiapkan data mentah agar dapat digunakan dalam model pembelajaran mesin. Proses ini

melibatkan beberapa langkah yang membantu mengurangi kompleksitas data dan meningkatkan akurasi model. Dalam konteks analisis sentimen, preprocessing bertujuan untuk membersihkan dan menstandarkan data teks, sehingga informasi yang relevan dapat diambil dengan lebih mudah. Langkah-langkah ini meliputi *case folding, normalization, tokenizing, stopwords removal,* dan *stemming*, yang semuanya berkontribusi pada kualitas data yang akan digunakan dalam klasifikasi. Setiap langkah preprocessing memiliki perannya masing-masing yaitu sebagai berikut:

1. Case Folding

Semua huruf akan diubah menjadi *lowercase* atau huruf kecil (Muhammadin & Sobari, 2021). Hal ini penting karena dalam analisis teks, kata-kata yang sama tetapi ditulis dengan huruf besar atau kecil harus dianggap identik. Tanpa *case folding*, model mungkin akan menganggap kedua kata tersebut berbeda, yang dapat menyebabkan kesalahan dalam analisis sentimen.

Tabel 3. 1 Contoh Case Folding

Sebelum	Sesudah
Aplikasi AdaKami proses pngajuannya cpat dan mdah serta tdk membingungkan	aplikasi adakami proses pngajuannya cpat dan mdah serta tdk membingungkan

Proses *case folding* biasanya dilakukan sebelum langkah-langkah *preprocessing* lainnya. Dengan menerapkan *case folding* terlebih dahulu, kita dapat memastikan bahwa semua langkah berikutnya seperti penghapusan *stopwords* dan *tokenizing* dilakukan pada data yang konsisten. Ini sangat membantu dalam meningkatkan akurasi model klasifikasi akhir.

2. Normalization

Normalization adalah proses perbaikan kesalahan yang ada pada kata seperti ejaan yang salah agar kata yang memiliki makna sama menjadi setara (Hendriyanto et al., 2022). Proses ini bertujuan untuk mengubah kata slang ke bentuk baku sehingga meminimalisirkan variasi kata dengan menggunakan kamus yang telah banyak digunakan.

Tabel 3. 2 Contoh Normalization

Sebelum	Sesudah
aplikasi adakami proses pngajuannya cpat dan mdah serta tdk membingungkan	aplikasi adakami proses pengajuannya cepat dan mudah serta tidak membingungkan

Pada penelitian ini proses *normalization* menggunakan ’kamus slang.csv’ untuk mengubah kata slang menjadi kata bakunya.

3. Tokenizing

Tokenizing adalah memisahkan kalimat menjadi kata tunggal dan pengecekan kata dari karakter pertama sampai karakter terakhir (Muhammadin & Sobari, 2021). Proses ini merupakan langkah pertama dalam banyak alur kerja pemrosesan bahasa alami (NLP) karena memungkinkan kita untuk bekerja dengan bagian-bagian kecil dari teks secara individual. *Tokenization* dapat dilakukan pada berbagai tingkat, termasuk pemisahan kalimat (*sentence tokenization*) dan pemisahan kata (*word tokenization*).

Tabel 3. 3 Contoh Tokenizing

Sebelum	Sesudah
aplikasi adakami proses pengajuannya cepat dan mudah serta tidak membingungkan	['aplikasi', 'adakami', 'proses', 'pengajuannya', 'cepat', 'dan', 'mudah', 'serta', 'tidak', 'membingungkan']

Dalam penelitian ini, *tokenizing* dilakukan menggunakan pustaka seperti NLTK atau spaCy, yang menyediakan fungsi-fungsi siap pakai untuk melakukan tokenisasi dengan efisien. Hasil dari tokenisasi adalah daftar kata atau frasa yang akan digunakan dalam langkah-langkah berikutnya seperti penghapusan *stopwords* dan *stemming*. Dengan

memecah teks menjadi token-token ini, kita dapat lebih mudah menganalisis frekuensi kemunculan kata serta hubungan antar kata dalam konteks analisis sentimen.

4. Stopwords Removal

Stopwords removal adalah menghilangkan kata-kata yang tidak penting dalam suatu data. *Stopwords* merupakan kata-kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak berarti (Ramadhan et al., 2023). Kata-kata ini sering kali muncul dalam teks tetapi tidak memberikan informasi tambahan tentang sentimen atau konteks dari ulasan tersebut. Dengan menghapus *stopwords*, kita dapat mengurangi dimensi data dan fokus pada kata-kata yang lebih relevan.

Tabel 3. 4 Contoh Stopword Removal

Sebelum	Sesudah
['aplikasi', 'adakami', 'proses', 'pengajuannya', 'cepat', 'dan', 'mudah', 'serta', 'tidak', 'membingungkan']	['aplikasi', 'adakami', 'proses', 'pengajuannya', 'cepat', 'mudah', 'membingungkan']

Proses ini juga membantu mengurangi noise dalam dataset, sehingga model klasifikasi dapat lebih mudah belajar dari pola-pola yang ada. Beberapa pustaka pemrograman seperti NLTK dan spaCy

menyediakan daftar *stopwords* standar yang dapat digunakan untuk berbagai bahasa, termasuk Bahasa Indonesia. Penggunaan daftar ini memungkinkan peneliti untuk dengan cepat membersihkan dataset mereka tanpa harus membuat daftar *stopwords* dari awal.

5. Stemming

Stemming adalah menghilangkan gabungan kata, dan menyebabkan kata yang telah ditambahkan menjadi kata dasar (Ramadhan et al., 2023). Tujuan utama dari *stemming* adalah untuk menyederhanakan variasi kata sehingga model klasifikasi tidak perlu menangani banyak bentuk morfologis dari satu kata.

Tabel 3. 5 Contoh Stemming

Sebelum	Sesudah
['aplikasi', 'adakami', 'proses', 'pengajuannya', 'cepat', 'mudah', 'membingungkan']	['aplikasi', 'adakami', 'proses', 'aju', 'cepat', 'mudah', 'bingung']

Proses *stemming* sangat berguna dalam analisis sentimen karena sering kali satu konsep dapat diungkapkan dengan berbagai cara. Dengan menggunakan algoritma *stemming* seperti *Porter Stemmer* atau *Snowball Stemmer*, kita dapat memastikan bahwa semua variasi kata diubah menjadi

bentuk dasar yang sama. Hal ini tidak hanya meningkatkan efisiensi pemrosesan tetapi juga meningkatkan akurasi hasil klasifikasi dengan mengurangi kebisingan data akibat variasi bahasa alami.

Dengan menerapkan semua langkah *preprocessing* ini secara sistematis, kami berharap dapat meningkatkan kinerja algoritma NBC dan SVM dalam menganalisis sentimen aplikasi AdaKami di *Google Play Store* secara signifikan.

F. Pembobotan TF-IDF

Pembobotan TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah teknik yang umum digunakan dalam pemrosesan teks untuk mengevaluasi seberapa penting sebuah kata dalam dokumen relatif terhadap seluruh koleksi dokumen. Dalam konteks analisis sentimen, pembobotan ini sangat berguna untuk menyoroti kata-kata kunci yang memiliki pengaruh besar terhadap klasifikasi sentimen. Rumus TF-IDF yaitu:

$$TF - IDF = TF\ x\ IDF \qquad (3.1)$$

Term Frequency (TF) yaitu semakin tinggi frekuensi kemunculan *term* pada sebuah dokumen maka menjadi semakin tinggi juga nilai bobot untuk *term* itu sendiri. Rumus TF yaitu:

$$TF = \frac{Jumlah\ kemunculan\ kata\ dalam\ dokumen}{Total\ kata\ dalam\ dokumen} \quad (3.2)$$

IDF merupakan kebalikan dari TF, semakin tinggi frekuensi kemunculan *term* maka nilai bobot *term* itu sendiri menjadi semakin kecil (Julianto et al., 2022). Rumus IDF yaitu:

$$IDF = log\left(\frac{Total\ jumlah\ dokumen}{Jumlah\ dokumen\ mengandung\ kata\ tersebut}\right) \quad (3.3)$$

Proses pembobotan TF-IDF dilakukan setelah tahap *preprocessing* dan pelabelan selesai. Data teks yang telah difilter dan dibersihkan akan dikonversi menjadi representasi vektor menggunakan metode TF-IDF. Vektor ini kemudian akan digunakan sebagai *input* untuk algoritma klasifikasi seperti NBC dan SVM. Dengan menggunakan TF-IDF, kita dapat meningkatkan kualitas fitur yang digunakan dalam model klasifikasi, sehingga diharapkan hasil analisis sentimen menjadi lebih akurat dan relevan.

G. Klasifikasi

Klasifikasi adalah tahap di mana data ulasan yang telah diproses dan diberi label akan dianalisis menggunakan algoritma pembelajaran mesin untuk menentukan kategori sentimen masing-masing ulasan. Dalam penelitian ini, dua algoritma klasifikasi utama yang digunakan adalah *Naive Bayes Classifier* dan *Support Vector Machine* (SVM). NBC merupakan metode probabilistik yang didasarkan pada teorema Bayes

dengan asumsi independensi antar fitur. Sementara itu, SVM adalah metode klasifikasi yang lebih kompleks yang mencari hyperplane optimal untuk memisahkan kelas-kelas dalam ruang fitur.

Dengan melakukan perbandingan ini, kita dapat menentukan algoritma mana yang lebih efektif untuk analisis sentimen dalam konteks aplikasi *mobile*.

H. Evaluasi

Evaluasi adalah langkah krusial dalam proses analisis sentimen karena bertujuan untuk menilai kinerja model klasifikasi yang telah diterapkan. Metrik evaluasi umum yang digunakan dalam penelitian ini meliputi waktu penyelesaian, efisiensi sumber daya, akurasi, presisi, *recall,* dan *F1-score*. Akurasi mengukur proporsi prediksi yang benar dari total prediksi yang dilakukan oleh model dengan rumus yaitu:

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \qquad (3.4)$$

dimana:

TP (*True Positive*) : Jumlah prediksi positif benar

TN (*True Negative*) : Jumlah prediksi negatif benar

FP (*False Positive*) : Jumlah prediksi negatif salah

FN (*False Negative*) : Jumlah prediksi negatif salah

Presisi mengukur sejauh mana prediksi positif benar-benar positif dengan rumus yaitu:

$$\text{Presisi} = \frac{TP}{TP + FP} \qquad (3.5)$$

Recall mengukur sejauh mana semua *instance* positif terdeteksi oleh model dengan rumus yaitu:

$$\text{Recall} = \frac{TP}{TP + FN} \qquad (3.6)$$

F1-score merupakan rata-rata harmonik dari presisi dan *recall*, memberikan gambaran menyeluruh tentang keseimbangan antara keduanya (Alfiyanti & Indra, 2024) dengan rumus yaitu:

$$F1\text{-}score = 2 \,.\, \frac{Presisi \,.\, Recall}{Presisi + Recall} \qquad (3.7)$$

Setelah model dilatih menggunakan data latih (*training set*), evaluasi dilakukan dengan menggunakan data uji (*test set*) yang belum pernah dilihat oleh model sebelumnya. Hasil evaluasi akan memberikan wawasan tentang seberapa baik model dapat menggeneralisasi pada data baru dan seberapa efektifnya model dalam menganalisis sentimen ulasan aplikasi AdaKami di *Google Play Store.*

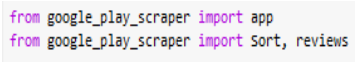
BAB IV
HASIL DAN PEMBAHASAN

A. *Crawling* Data

Pada penelitian ini data diambil dari ulasan pengguna aplikasi AdaKami di *google play store*. Proses pengambilan data menggunakan *google play scraper* yaitu *library* dari python untuk mempermudah pengambilan data dari playstore secara otomatis dengan parameter yang diterapkan. Tahapan yang pertama dilakukan adalah install *library google play scraper* seperti pada gambar 4.1 berikut:

Gambar 4. 1 Code Install google-play-scraper

Setelah *library* berhasil terinstall langkah selanjutnya adalah mengimpor modul app, *Sort, reviews* dari *library google_play_scraper* yang akan digunakan dalam *crawling* data seperti pada gambar 4.2 berikut:



Gambar 4. 2 Code Impor Modul Scraper

Langkah berikutnya adalah melakukan proses *crawling* data dengan parameter id aplikasi AdaKami, bahasa indonesia, negara indonesia, sortir paling relevan, dan jumlah data 5000 seperti pada gambar 4.3 berikut:

```
from google_play_scraper import app
from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'com.adakami.dana.kredit.pinjaman',
    lang='id',
    country='id',
    sort=Sort.MOST_RELEVANT,
    count=5000,
    filter_score_with=None
)
```

Gambar 4. 3 Code Scraper Data

Data mentah yang telah diambil berbentuk list diubah menjadi *dataframe* agar lebih terstruktur dengan cara seperti gambar 4.4 berikut:

```
import pandas as pd
import numpy as np
data = pd.DataFrame(np.array(result), columns=['review'])
data = data.join(pd.DataFrame(data.pop('review').tolist()))
data
```

Gambar 4. 4 Code Dataframe

Data ulasan yang didapatkan berisi 5000 data *username* (nama pengguna), *content* (teks ulasan), *score* (rating), *at* (tanggal), *appversion* (versi aplikasi), dan *thumbsupcount* (like pada ulasan).

	reviewId	userName	userImage	content	score
0	b383a2ad-ffb0-4ce6-bb43-74dbf36f19bb	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	Aplikasi ini Sangat mengecewakan, terlalu rumi...	1
1	36b3f93a-2dec-40a4-baaf-3feea31c46e7	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	wah asik game nya menarik bisa jalani misi dg ...	5
2	b1c82a86-a072-4b6e-bf7e-49aa330cd10a	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	tidak sesuai dengan iklan	1
3	138ea688-4d2a-4750-952c-005f90aefdef	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	aplikasi adakami ini sangat membantu	4
4	46dabcfe-942d-4397-a990-8a6d7c4f7d0e	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	wah mudah sekali untuk meminjam, syaratnya mud...	5
...	...	...	...	...	...
4995	4a0ca12c-4e72-47ae-8de2-992aca63d6d1	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	biaya layanannya sangat besarr	3
4996	bc94dc9b-6011-4253-8ec6-709a0bb2a3c3	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	aplikasi nya bagus, sangat dimengerti caranya,...	5
4997	20ef8d69-e6a1-4723-9922-35ecdb1e9cc6	Adhi Ramdhani	https://play-lh.googleusercontent.com/a/ACg8oc...	Saya nasabah ada kami sudah hampir 2 tahun pem...	1
4998	c3fa78f5-bf98-4346-baa3-fb2be5ef950d	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	Saya pengguna lama dan tidak ada catatan keter...	5
4999	8b11e62e-1faa-4028-bb7e-cefa7a4ac9e9	Pengguna Google	https://play-lh.googleusercontent.com/EGemoI2N...	Bagaimana ini adakami? Saya pelanggan lamaaaaa...	1

5000 rows × 11 columns

Gambar 4. 5 Hasil Scraping Data

score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt	appVersion
1	586	6.2.0	2024-06-14 08:33:10	Hai TemanKami, apabila Kakak memiliki pertanya...	2024-06-15 03:34:35	6.2.0
5	0	6.9.0	2025-04-23 23:48:49	None	NaT	6.9.0
1	1	6.9.0	2025-03-25 03:56:26	Hai TemanKami, apabila Kakak memiliki pertanya...	2025-03-26 03:21:58	6.9.0
4	0	6.9.0	2025-05-08 08:52:15	None	NaT	6.9.0
5	3	6.9.0	2025-04-22 01:57:36	None	NaT	6.9.0
...	...	...	...	...	...	...
3	0	6.9.0	2025-03-20 08:33:29	None	NaT	6.9.0
5	13	6.9.0	2025-04-07 08:09:00	None	NaT	6.9.0
1	0	2.5.1	2022-07-07 13:10:45	Hai TemanKami, mohon maaf apabila terdapat ken...	2022-07-08 07:51:27	2.5.1
5	3	6.7.0	2025-01-09 02:07:11	Hai Teman Kami, mohon maaf apabila terdapat ke...	2024-07-06 07:18:03	6.7.0
1	2	None	2020-11-16 16:48:13	Hai Teman Kami, dapat kami informasikan Untuk ...	2020-11-17 01:17:28	None

Gambar 4. 6 Hasil Scraping Data

Pada penelitian ini hanya akan menggunakan data *at* untuk melihat tanggal ulasan dan *content* untuk diklasifikasikan sentimennya sehingga data selain itu dibuang.

```
data = data[['at','content']]
```

Gambar 4. 7 Code Filter Data

Setelah itu data diurutkan berdasarkan tanggal ulasan seperti gambar 4.8.

```
data = data.sort_values(by='at', ascending=False)
```

Gambar 4. 8 Code Sortir Data

	at	content
3187	2025-03-02 9:12:11	Aplikasi yg sangat bagus👍😊
3188	2025-03-02 8:49:57	Aplikasi yang izin OJK nya patut di pertanyaka...
3189	2025-03-02 8:42:09	Baru download dan masukin no hp.. tapi data su...
3190	2025-03-02 8:37:51	ada kami memang ok proses cepat
3191	2025-03-02 8:18:44	Tanggal 2 maret 2025' saya sudah tidak memakai...
...	...	...
4998	2023-07-15 5:40:47	Baru ngisi no hp buat login doang dan saya mem...
4999	2023-07-15 5:05:16	Aplikasi buruk pembayaran sudah di lunasi pada...
4991	2023-07-15 18:20:28	Hati2 jangan tertatik apk ini.... Gw daftar su...
4992	2023-07-15 16:13:57	Sangat membantu ndak ribet bunga jg wajar pemb...
4993	2023-07-15 12:56:54	Di ACC ,gak tapi pingin di ksh bintang 5, pada...

5000 rows × 2 columns

Gambar 4. 9 Hasil Sortir Data

Data yang diperoleh berada pada rentang tanggal 15 juli 2023 sampai tanggal 2 maret 2025. Pada penelitian ini periode data dibatasi dari pada bulan september 2024 sampai februari 2025 sehingga data selain tanggal tersebut dihapus dan

diperoleh data yang akan digunakan lebih lanjut sebanyak 3177.

	at	content
0	2024-09-01 2:11:32	Aplikasi apaan? Aku ndak pernah nunggak kok kr...
1	2024-09-01 2:39:43	kenapa sekarang adakami gak bisa pinjam lagi y...
2	2024-09-01 2:49:39	Awal nya tiap pinjem di ksh dan saya juga tiap...
3	2024-09-01 3:31:13	Sangat mudah cara pengajuannya semoga amanah d...
4	2024-09-01 5:51:13	Mantap bisa jadi solusi,di saat membutuhkan me...
...	...	...
3172	2025-01-13 9:41:53	proses cepat dan sangat baik, membantu bisnis ...
3173	2025-01-13 8:50:46	Aplikasi mengecewakan baru pengajuan di tolak ...
3174	2025-01-13 7:48:30	pengajuan yg sangat cepat prosesnya
3175	2025-01-12 22:51:13	adakami memang solusi terbaik di saat ada kebu...
3176	2025-01-12 19:58:48	aplikasi yang sangat membantu sekali terima ka...

3177 rows × 2 columns

Gambar 4. 10 Data Mentah

B. Pelabelan

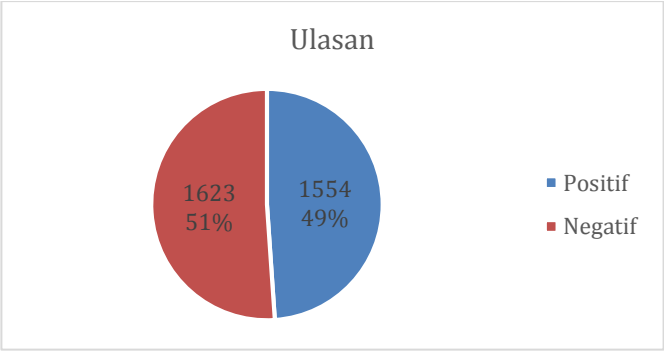
Pada tahap ini setiap data ulasan pengguna diberi label secara manual oleh ahli termasuk dalam kategori positif atau negatif, Pada penelitian ini data ulasan dilabeli oleh ibu Nurul Ainunnisa, S.Pd yang merupakan guru bahasa indonesia di MAN 2 kota Semarang sehingga diperoleh hasil sebagai berikut:

Tabel 4. 1 Hasil Pelabelan

Teks	Label
Sangat mudah cara pengajuannya semoga amanah dan barokah .... Amiiiin🤲	positif

Mantap bisa jadi solusi,di saat membutuhkan mendadak,,proses cepat makasih adakami,	positif
Mengecewakan gak kaya aplikasi lain padahal pembayaran saya GX pernah telat tapi pengajuan kembali susah banget slalu d undur dan d tolak	negatif
cara penagihan sangat tidak ber etitut, sangat sangat kasar dan kurang menyenangkan, penagihan menjurus ke arah ancaman. benar benar aplikasi yang buruk.	negatif
pengajuan sangat cepat dan efesien ,membantu dalam keadaan terdesak ,terimakasih ada kami	positif

Pada proses pelabelan ini diperoleh data positif sebanyak 1554 ulasan dan data negatif sebanyak 1623 ulasan.



Gambar 4. 11 Diagram Data Mentah

C. *Preprocessing* Data

Pada tahap ini data mentah yang telah diperoleh diproses untuk menjadi data bersih sehingga pengklasifikasian sentimen dapat dilakukan oleh *machine learning* dengan lebih maksimal. Proses yang dilakukan adalah *case folding, normalization, stopword removal, tokenizing*, dan *stemming*. Data mentah berisi 3177 ulasan dengan total kata sebanyak 67311 kata.

1. Case Folding

Pada tahap *case folding* teks disamaratakan dalam menjadi huruf kecil semua, pada tahap ini juga dilakukan pembersihan sehingga hanya terdapat *text* dengan huruf kecil semua. Untuk langkah yang dilakukan dapat dilihat pada gambar 4.12.

```
def clean_content(content):
    import string, re

    content = content.lower()
    content = re.sub("[^a-z]", " ", content)
    content = re.sub("\t", " ", content)
    content = re.sub("\n", " ", content)
    content = re.sub("\s+", " ", content)
    content = content.strip()

    return content
data["content_clean"] = data["content"].apply(clean_content)
data.head()
```

Gambar 4. 12 Code Case Folding

Case folding dalam penelitian ini dilakukan dengan langkah langkah berikut:

a. content.lower() yaitu mengubah semua huruf menjadi huruf kecil
b. re.sub("[^a-z]", " ", content) yaitu menghapus semua karakter non-alfabet dan mengganti dengan spasi
c. re.sub("\t", " ", content) yaitu menghapus tab dan mengganti dengan spasi
d. re.sub("\n", " ", content) yaitu menghapus baris baru agar teks tetap dalam satu baris
e. re.sub("\s+", " ", content) yaitu menghapus spasi ganda atau lebih dan mengganti dengan satu spasi
f. content.strip() yaitu menghapus spasi di awal dan akhir teks

Data ulasan yang sudah dibersihkan di tahap *case folding* contoh hasilnya sebagai berikut:

Tabel 4. 2 Hasil Case Folding

Sebelum	Sesudah
Sangat mudah cara pengajuannya semoga amanah dan barokah .... Amiiiin🤲	sangat mudah cara pengajuannya semoga amanah dan barokah amiiiin

Mantap bisa jadi solusi,di saat membutuhkan mendadak,,proses cepat makasih adakami,	mantap bisa jadi solusi di saat membutuhkan mendadak proses cepat makasih adakami
Mengecewakan gak kaya aplikasi lain padahal pembayaran saya GX pernah telat tapi pengajuan kembali susah banget slalu d undur dan d tolak	mengecewakan gak kaya aplikasi lain padahal pembayaran saya gx pernah telat tapi pengajuan kembali susah banget slalu d undur dan d tolak
cara penagihan sangat tidak ber etitut, sangat sangat kasar dan kurang menyenangkan, penagihan menjurus ke arah ancaman. benar benar aplikasi yang buruk.	cara penagihan sangat tidak ber etitut sangat sangat kasar dan kurang menyenangkan penagihan menjurus ke arah ancaman benar benar aplikasi yang buruk
pengajuan sangat cepat dan efesien ,membantu dalam keadaan terdesak ,terimakasih ada kami	pengajuan sangat cepat dan efesien membantu dalam keadaan terdesak terimakasih ada kami

Setelah melakukan *case folding* dari 3177 data ulasan, jumlah kata berkurang 91 kata menjadi 67220 kata.

2. *Normalization*

Pada tahap ini teks yang berisi slang atau kata gaul diperbaiki dengan kata yang formal dengan menggunakan kamus slang seperti gambar 4.13.

```
kamus_slang = pd.read_csv('/content/kamus_slang.csv')

kamus_dict = dict(zip(kamus_slang['slang'], kamus_slang['formal']))

def normalisasi_teks(text):
    if isinstance(text, str):
        words = text.split()
        return ' '.join([kamus_dict.get(word, word) for word in words])
    return text

data["text_normalized"] = data["content_clean"].apply(normalisasi_teks)

data.head()
```

Gambar 4. 13 Code Normalization

Langkah pertama adalah dengan membaca *file* yang berisi daftar kata slang dan kata formalnya, daftar kata yang sudah terbaca diubah menjadi kamus untuk menormalisasikan teks ulasan. Setelah itu buat fungsi normalisasi menggunakan kamus slang dan jalankan fungsi tersebut. Hasil dari tahap ini contohnya sebagai berikut:

Tabel 4. 3 Hasil Normalization

Sebelum	Sesudah
sangat mudah cara pengajuannya semoga amanah dan barokah amiiiin	sangat mudah cara pengajuannya semoga amanah dan barokah amin
mantap bisa jadi solusi di saat membutuhkan mendadak proses cepat makasih adakami	mantap bisa jadi solusi di saat membutuhkan mendadak proses cepat terima kasih adakami

mengecewakan gak kaya aplikasi lain padahal pembayaran saya gx pernah telat tapi pengajuan kembali susah banget slalu d undur dan d tolak	mengecewakan enggak kayak aplikasi lain padahal pembayaran saya enggak pernah telat tapi pengajuan kembali susah banget selalu di undur dan di tolak
cara penagihan sangat tidak ber etitut sangat sangat kasar dan kurang menyenangkan penagihan menjurus ke arah ancaman benar benar aplikasi yang buruk	cara penagihan sangat tidak ber etitut sangat sangat kasar dan kurang menyenangkan penagihan menjurus ke arah ancaman benar benar aplikasi yang buruk
pengajuan sangat cepat dan efesien membantu dalam keadaan terdesak terimakasih ada kami	pengajuan sangat cepat dan efesien membantu dalam keadaan terdesak terimakasih ada kami

Setelah melakukan tahap ini jumlah kata dalam keseluruhan data bertambah 109 kata menjadi 67329 kata.

3. Tokenizing

Pada tahap *tokenizing* teks dipisah kata perkata menjadi daftar kata menggunakan modul wordpunct_tokenize dari *library* nltk untuk diproses pada tahap selanjutnya. Untuk langkah penerapannya seperti pada gambar 4.14 berikut:

```
import nltk
nltk.download('punkt')
from nltk.tokenize import wordpunct_tokenize

data['text_tokens'] = data['content_clean'].apply(lambda x: wordpunct_tokenize(x))
data.head(50)
```

Gambar 4. 14 Code Tokenizing

Langkah pertama adalah mengimpor modul wordpunct_tokenize dari *library* nltk.tokenize setelah itu gunakan modul wordpunct.tokenize untuk memisahkan setiap teks menjadi daftar kata. Data yang telah melalui tahap ini contohnya adalah sebagai berikut:

Tabel 4. 4 Hasil Tokenizing

Sebelum	Sesudah
sangat mudah cara pengajuannya semoga amanah dan barokah amin	['sangat', 'mudah', 'cara', 'pengajuannya', 'semoga', 'amanah', 'dan', 'barokah', 'amin']
mantap bisa jadi solusi di saat membutuhkan mendadak proses cepat terima kasih adakami	['mantap', 'bisa', 'jadi', 'solusi', 'di', 'saat', 'membutuhkan', 'mendadak', 'proses', 'cepat', 'terima', 'kasih', 'adakami']
mengecewakan enggak kayak aplikasi lain padahal pembayaran saya enggak pernah telat tapi pengajuan	['mengecewakan', 'enggak', 'kayak', 'aplikasi', 'lain', 'padahal', 'pembayaran', 'saya', 'enggak', 'pernah', 'telat', 'tapi', 'pengajuan',

kembali susah banget selalu di undur dan di tolak	'kembali', 'susah', 'banget', 'selalu', 'di', 'undur', 'dan', 'di', 'tolak']
cara penagihan sangat tidak ber etitut sangat sangat kasar dan kurang menyenangkan penagihan menjurus ke arah ancaman benar benar aplikasi yang buruk	['cara', 'penagihan', 'sangat', 'tidak', 'ber', 'etitut', 'sangat', 'sangat', 'kasar', 'dan', 'kurang', 'menyenangkan', 'penagihan', 'menjurus', 'ke', 'arah', 'ancaman', 'benar', 'benar', 'aplikasi', 'yang', 'buruk']
pengajuan sangat cepat dan efesien membantu dalam keadaan terdesak terimakasih ada kami	['pengajuan', 'sangat', 'cepat', 'dan', 'efesien', 'membantu', 'dalam', 'keadaan', 'terdesak', 'terimakasih', 'ada', 'kami']

Setelah melakukan tahap ini jumlah kata dalam keseluruhan data bertambah 10 kata menjadi 67339 kata.

4. Stopword Removal

Pada tahap *stopword removal* kata-kata yang dianggap tidak berguna dalam analisis sentimen dihapus dengan cara seperti gambar 4.15.

```
import nltk.corpus
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = set(stopwords.words('indonesian'))
data['text_StopWord'] = data['text_tokens'].apply(lambda x: [word for word
                        in x if word.lower() not in stop])
data.head(50)
```

Gambar 4. 15 Code Stopword Removal

Dengan mengimpor modul nltk.corpus untuk mengakses kumpulan data termasuk *stopword* kemudian unduh data *stopword*. Disini menggunakan data *stopword* dalam bahasa indonesia untuk memproses data ulasan. Setiap kata dalam teks ulasan yang ada dalam data *stopword* akan dihapus. Data ulasan yang sudah melalui tahap ini contohnya sebagai berikut:

Tabel 4. 5 Hasil Stopword Removal

Sebelum	Sesudah
['sangat', 'mudah', 'cara', 'pengajuannya', 'semoga', 'amanah', 'dan', 'barokah', 'amin']	['mudah', 'pengajuannya', 'semoga', 'amanah', 'barokah', 'amin']
['mantap', 'bisa', 'jadi', 'solusi', 'di', 'saat', 'membutuhkan', 'mendadak', 'proses', 'cepat', 'terima', 'kasih', 'adakami']	['mantap', 'solusi', 'membutuhkan', 'mendadak', 'proses', 'cepat', 'terima', 'kasih', 'adakami']

['mengecewakan', 'enggak', 'kayak', 'aplikasi', 'lain', 'padahal', 'pembayaran', 'saya', 'enggak', 'pernah', 'telat', 'tapi', 'pengajuan', 'kembali', 'susah', 'banget', 'selalu', 'di', 'undur', 'dan', 'di', 'tolak']	['mengecewakan', 'kayak', 'aplikasi', 'pembayaran', 'telat', 'pengajuan', 'susah', 'banget', 'undur', 'tolak']
['cara', 'penagihan', 'sangat', 'tidak', 'ber', 'etitut', 'sangat', 'sangat', 'kasar', 'dan', 'kurang', 'menyenangkan', 'penagihan', 'menjurus', 'ke', 'arah', 'ancaman', 'benar', 'benar', 'aplikasi', 'yang', 'buruk']	['penagihan', 'ber', 'etitut', 'kasar', 'menyenangkan', 'penagihan', 'menjurus', 'arah', 'ancaman', 'aplikasi', 'buruk']
['pengajuan', 'sangat', 'cepat', 'dan', 'efesien', 'membantu', 'dalam', 'keadaan', 'terdesak', 'terimakasih', 'ada', 'kami']	['pengajuan', 'cepat', 'efesien', 'membantu', 'terdesak', 'terimakasih']

Setelah melakukan tahap ini jumlah kata dalam keseluruhan data berkurang 32171 kata menjadi 35168 kata.

5. *Stemming*

Pada tahap ini semua kata dalam daftar kata diubah menjadi kata dasarnya dengan menghilangkan imbuhan yaitu awalan, sisipan, dan akhiran. *Stemming* bertujuan untuk mengurangi variasi kata sehingga model lebih fokus pada makna utama kata tersebut. Langkah pertama yang dilakukan adalah menginstall *library* sastrawi seperti pada gambar 4.16 berikut:

```
!pip install Sastrawi
```

Gambar 4. 16 Code Install Sastrawi

Sastrawi merupakan *library* python yang dapat digunakan untuk stemming dalam bahasa indonesia. Setelah sastrawi terinstall langkah yang dilakukan berikutnya adalah seperti gambar 4.17 berikut:

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

factory = StemmerFactory()
stemmer = factory.create_stemmer()

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

term_dict = {}

for document in data['text_StopWord']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = stemmed_wrapper(term)
        print(term,":" ,term_dict[term])

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

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

data['text_steam'] = data['text_StopWord'].apply(lambda x: get_stemmed_term(x))
data.head(20)
```

Gambar 4. 17 Code Stemming

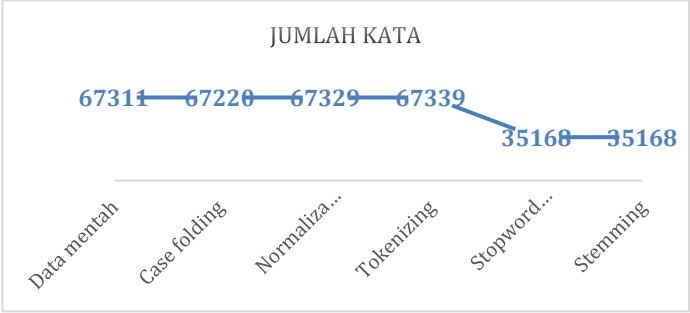
Langkah pertama adalah impor modul StemmerFactory dari *library* sastrawi yang akan digunakan untuk *stemming* kata. Langkah berikutnya adalah membuat objek *stemmer* dari modul StemmerFactory, setelah itu buat fungsi untuk *stemming* setiap kata dan buat kamus sementara untuk menyimpan hasil *stemming* kata. Berikutnya buat fungsi untuk *stemming* kata dalam seluruh data dan menerapkan *stemming* ke *dataframe*. Contoh hasil dari tahap ini sebagai berikut:

Tabel 4. 6 Hasil Stemming

Sebelum	Sesudah
['mudah', 'pengajuannya', 'semoga', 'amanah', 'barokah', 'amin']	['mudah', 'aju', 'moga', 'amanah', 'barokah', 'amin']
['mantap', 'solusi', 'membutuhkan', 'mendadak', 'proses', 'cepat', 'terima', 'kasih', 'adakami']	['mantap', 'solusi', 'butuh', 'dadak', 'proses', 'cepat', 'terima', 'kasih', 'adakami']
['mengecewakan', 'kayak', 'aplikasi', 'pembayaran', 'telat', 'pengajuan', 'susah', 'banget', 'undur', 'tolak']	['kecewa', 'kayak', 'aplikasi', 'bayar', 'telat', 'aju', 'susah', 'banget', 'undur', 'tolak']
['penagihan', 'ber', 'etitut', 'kasar', 'menyenangkan', 'penagihan', 'menjurus',	['tagih', 'ber', 'etitut', 'kasar', 'senang', 'tagih', 'jurus',

'arah', 'ancaman', 'aplikasi', 'buruk']	'arah', 'ancam', 'aplikasi', 'buruk']
['pengajuan', 'cepat', 'efesien', 'membantu', 'terdesak', 'terimakasih']	['aju', 'cepat', 'efesien', 'bantu', 'desak', 'terimakasih']

Setelah melakukan tahap ini jumlah kata dalam keseluruhan data tidak berubah, masih tetap 35168 kata.



Gambar 4. 18 Diagram Kata

Setelah lima tahap *preprocessing* data diperoleh data bersih. Data yang sudah bersih dicek kembali untuk memastikan tidak ada ulasan yang terduplikasi seperti gambar 4.19.

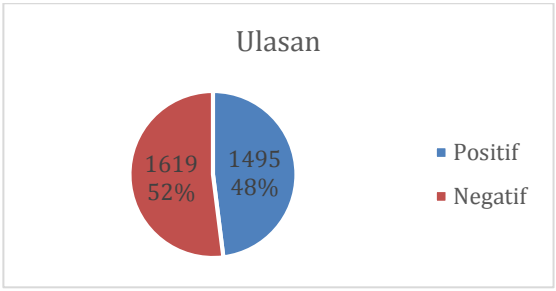
```
data.duplicated(subset=['text']).sum()
```

Gambar 4. 19 Code Cek Duplikasi

Pada data bersih yang digunakan terdapat 63 ulasan yang terduplikasi sehingga harus dihapus agar hasil klasifikasi oleh mesin lebih maksimal.

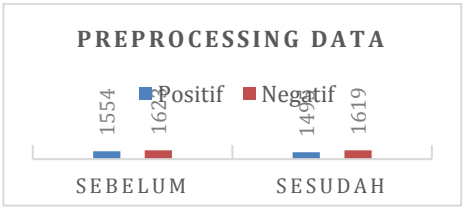
```
data = data.drop_duplicates(subset=['text'])
```

Gambar 4. 20 Code Hapus Duplikasi



Gambar 4. 21 Diagram Data Bersih

Hasil akhir dari *preprocessing* data diketahui bahwa data bersih yang akan digunakan sebanyak 3114 ulasan dengan total kata sebanyak 35168 kata. Dari data bersih tersebut terbagi menjadi 1619 ulasan negatif dan 1495 ulasan positif.



Gambar 4. 22 Diagram Preprocessing Data

Proses *preprocessing* data menyebabkan data ulasan berkurang 63 ulasan dari yang awalnya berjumlah 3177 ulasan menjadi 3114 ulasan. Data ulasan yang berkurang sebagian besar dari ulasan positif dari awalnya 1554 ulasan positif

menjadi 1495 ulasan positif (59 ulasan). Sedangkan data positif awalnya 1623 menjadi 1619 (4 ulasan).

D. Pembobotan TF-IDF

Tahap ini bertujuan untuk mempresentasikan setiap kata dalam data ulasan dalam bentuk numerik agar bisa diterapkan pada model. Sebelum melakukan pembobotan TF-IDF data ulasan dibagi menjadi dua bagian yaitu data latih dan data uji dengan cara seperti gambar 4.23 berikut:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data['text'],
                                   data['sentimen'], test_size = 0.20,
                                   random_state = 0)
```

Gambar 4. 23 Code Pemisahan Data

Pada penelitian ini data ulasan dibagi menjadi 20% data uji (623 ulasan) dan 80% data pelatihan (2491 ulasan). X_train dan X_test berasal dari teks ulasan, sedangkan y_train dan y_test berasal dari label sentimen dari teks ulasan. Setelah membagi data ulasan, teks direpresentasikan dalam bentuk numerik dengan TF-IDF menggunakan modul TfidfVectorizer seperti gambar 4.24 berikut:

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer()
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
tfidf_test = tfidf_vectorizer.transform(X_test)
```

Gambar 4. 24 Code TF-IDF

Pertama impor modul TfidfVectorizer dari *library* sklearn.feature_extraction.text, selanjutnya buat objek

tfidf_vectorizer untuk transformasi data. Setelah itu representasikan data menjadi dua bagian yaitu X_train_tfidf dan X_test_tfidf.

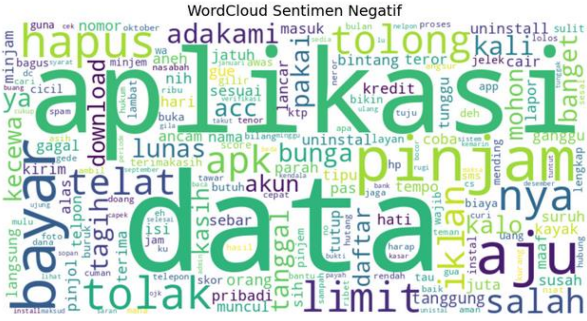
Setiap kata yang sudah diberikan bobot dengan tf-idf dapat divisualisasikan melalui *wordcloud* untuk melihat kata-kata yang dominan berdasarkan pembobotan tf-idf. Penggunaan *wordcloud* dapat dilakukan dengan cara seperti gambar 4.25 berikut:

```
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud

tfidf_df = pd.DataFrame(tfidf_train.toarray(),
                        columns=tfidf_vectorizer.get_feature_names_out())
tfidf_df['sentimen'] = y_train.values

tfidf_positif = tfidf_df[tfidf_df['sentimen'] == 'positif'].drop(columns=
                                                                 ['sentimen'])
tfidf_negatif = tfidf_df[tfidf_df['sentimen'] == 'negatif'].drop(columns=
                                                                 ['sentimen'])

word_freq_positif = tfidf_positif.mean().to_dict()
word_freq_negatif = tfidf_negatif.mean().to_dict()

def generate_wordcloud(word_freq, title):
    wordcloud = WordCloud(width=800, height=400, background_color='white'
                                        ).generate_from_frequencies(word_freq)
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title(title, fontsize=14)
    plt.show()

generate_wordcloud(word_freq_positif, "WordCloud Sentimen Positif")
generate_wordcloud(word_freq_negatif, "WordCloud Sentimen Negatif")
```

Gambar 4. 25 Code Wordcloud

Pertama impor modul pandas dan matplotlib.pyplot serta WordCloud dari *library* wordcloud selanjutnya membuat *dataframe* dari hasil tf-idf dan pisahkan kata-kata berdasarkan label sentimen. Setelah kata-kata dipisahkan, hitung rata rata frekuensi kata dan buat fungsi untuk menampilkan *wordcloud* sehingga diperoleh hasil seperti gambar 4.26 dan gambar 4.27 berikut:

Gambar 4. 26 Wordcloud Positif



Gambar 4. 27 Wordcloud Negatif

E. Klasifikasi

Tahap klasifikasi dilakukan untuk mengelompokan sentimen pengguna terhadap aplikasi AdaKami termasuk dalam kategori positif atau negatif. Pada penelitian ini menggunakan dua model klasifikasi yaitu *Naive Bayes Classifier* dan *Support Vector Machine*.

Sebelum melakukan klasifikasi dengan *naive bayes* dan *support vector machine* perlu untuk menginstall dan mengimpor modul seperti gambar 4.28 berikut:

```
import time
import tracemalloc
from sklearn.metrics import classification_report
```

Gambar 4. 28 Code Impor Modul Klasifikasi

Impor modul time untuk menghitung waktu penyelesaian, modul tracemalloc untuk melacak penggunaan memori, dan modul classification_report dari *library* sklearn.metrics untuk melihat hasil klasifikasi seperti akurasi, presisi, *recall*, dan *f1 score*.

1. Naive Bayes Classifier

Model klasifikasi ini menggunakan probabilitas dalam menentukan kategori sentimen berdasarkan kata-kata yang muncul dalam data ulasan. Data yang telah diolah kemudian

diterapkan dalam model klasifikasi naive bayes dengan cara seperti gambar 4.29.

```
from sklearn.naive_bayes import MultinomialNB

nb = MultinomialNB()

start_train = time.time()
nb.fit(tfidf_train, y_train)
end_train = time.time()
train_time = end_train - start_train

start_pred = time.time()
nb_pred = nb.predict(tfidf_test)
end_pred = time.time()
pred_time = end_pred - start_pred

tracemalloc.start()
nb.fit(tfidf_train, y_train)
nb_pred = nb.predict(tfidf_test)
current, peak = tracemalloc.get_traced_memory()
tracemalloc.stop()
mem_usage = peak / 1024 / 1024

print("=== Laporan Klasifikasi ===")
print(classification_report(y_test, nb_pred))
print("\n=== Hasil Pengukuran ===")
print(f"Waktu Pelatihan: {train_time:.4f} detik")
print(f"Waktu Prediksi: {pred_time:.4f} detik")
print(f"Penggunaan Memori Puncak: {mem_usage:.2f} MiB")
```

Gambar 4. 29 Code Klasifikasi NBC

Langkah pertama adalah mengimpor modul MultinomialNB dari *library* sklearn.naive_bayes sebagai model klasifikasi, model yang sudah diimpor kemudian digunakan untuk membuat objek untuk klasifikasi. Kemudian latih model

dengan data TF-IDF dan label sentimen setelah itu prediksikan sentimen dari data uji.

Tracemalloc digunakan untuk mengukur penggunaan memori puncak selama proses pelatihan dan prediksi. Langkah terakhir yang dilakukan adalah mengevaluasi waktu penyelesaian, penggunaan memori, dan klasifikasi dalam bentuk matriks evaluasi.

2. *Support Vector Machine*

Model klasifikasi ini mencari *hyperplane* optimal untuk memisahkan kelas-kelas (positif dan negatif) dalam ruang fitur. Data yang telah diolah kemudian diterapkan dalam model klasifikasi naive bayes dengan cara seperti gambar 4.30.

```
from sklearn.svm import SVC

svm = SVC(kernel="linear")

start_train = time.time()
svm.fit(tfidf_train, y_train)
end_train = time.time()
train_time = end_train - start_train

start_pred = time.time()
svm_pred = svm.predict(tfidf_test)
end_pred = time.time()
pred_time = end_pred - start_pred

tracemalloc.start()
svm.fit(tfidf_train, y_train)
svm_pred = svm.predict(tfidf_test)
current, peak = tracemalloc.get_traced_memory()
tracemalloc.stop()
mem_usage = peak / 1024 / 1024
```

Gambar 4. 30 Code Klasifikasi SVM

```
print("=== Laporan Klasifikasi ===")
print(classification_report(y_test, svm_pred))
print("\n=== Hasil Pengukuran ===")
print(f"Waktu Pelatihan: {train_time:.4f} detik")
print(f"Waktu Prediksi: {pred_time:.4f} detik")
print(f"Penggunaan Memori Puncak: {mem_usage:.2f} MiB")
```

Gambar 4. 31 Code Klasifikasi SVM

Langkah pertama adalah mengimpor modul SVC dari *library* sklearn.svm sebagai model klasifikasi, model yang sudah diimpor kemudian digunakan untuk membuat objek untuk klasifikasi. Kemudian latih model dengan data TF-IDF dan label sentimen setelah itu prediksikan sentimen dari data uji.

Tracemalloc digunakan untuk mengukur penggunaan memori puncak selama proses pelatihan dan prediksi. Langkah terakhir yang dilakukan adalah mengevaluasi waktu penyelesaian, penggunaan memori, dan klasifikasi dalam bentuk matriks evaluasi.

F. Evaluasi

Setelah melakukan klasifikasi dengan naive bayes dan svm, langkah terakhir adalah mengevaluasi kedua metode tersebut. Pada penelitian ini menggunakan matriks evaluasi untuk mengevaluasi akurasi, presisi, *recall,* dan *f1 score* selain itu juga mengevaluasi waktu penyelesaian dan efisiensi sumber daya.

1. Matriks Evaluasi

```
=== Laporan Klasifikasi ===
              precision    recall  f1-score   support

     negatif       0.93      0.94      0.94       316
     positif       0.94      0.93      0.93       307

    accuracy                           0.94       623
   macro avg       0.94      0.94      0.94       623
weighted avg       0.94      0.94      0.94       623
```

Gambar 4. 32 Hasil Klasifikasi Naive Bayes

Hasil pada gambar 4.32 dapat dihitung melalui *confusion matrix* berikut:

Tabel 4. 7 Confusion Matrix NBC

Confusion Matrix		Label Prediksi	
		Negatif	Positif
Label Sebenarnya	Negatif	298	18
	Positif	22	285

Dari *confusion matrix* diatas dapat dihitung akurasi, presisi *recall,* dan *f1-score* sebagai berikut:

$$\text{Akurasi} = \frac{298+285}{298+285+18+22} = 0{,}936 \quad (4.1)$$

$$\text{Presisi\_positif} = \frac{285}{285+18} = 0{,}940 \quad (4.2)$$

$$\text{Presisi\_negatif} = \frac{298}{298+22} = 0{,}931 \quad (4.3)$$

$$\text{Recall\_positif} = \frac{285}{285+22} = 0{,}928 \quad (4.4)$$

$$\text{Recall\_negatif} = \frac{298}{298+18} = 0{,}943 \quad (4.5)$$

$$\text{F1-Score\_pos} = 2 \cdot \frac{0{,}940 \cdot 0{,}928}{0{,}940 + 0{,}928} = 0{,}933 \quad (4.6)$$

$$\text{F1-Score\_neg} = 2 \cdot \frac{0{,}931 \cdot 0{,}943}{0{,}931 + 0{,}943} = 0{,}936 \quad (4.7)$$

Pada klasifikasi Naive Bayes mencapai akurasi 94% dalam mengklasifikasikan sentimen positif dan negatif dari 623 sampel. Presisi kelas negatif (0.93) dan positif (0.94) menunjukkan ketepatan model dalam memprediksi ulasan dengan tingkat kesalahan rendah. *Recall* kelas negatif (0.94) dan positif (0.93), ini menunjukkan bahwa *false negatives* rendah. *F1-score* untuk kelas negatif (0.94) dan positif (0.93), nilai *F1-score* yang hampir seimbang antara kedua kelas menunjukkan bahwa model tidak bias terhadap salah satu kelas tertentu.

```
=== Laporan Klasifikasi ===
              precision    recall  f1-score   support

     negatif       0.94      0.96      0.95       316
     positif       0.95      0.94      0.95       307

    accuracy                           0.95       623
   macro avg       0.95      0.95      0.95       623
weighted avg       0.95      0.95      0.95       623
```

Gambar 4. 33 Hasil Klasifikasi SVM

Hasil pada gambar 4.33 dapat dihitung melalui *confusion matrix* berikut:

Tabel 4. 8 Confusion Matrix SVM

Confusion Matrix		Label Prediksi	
		Negatif	Positif
Label Sebenarnya	Negatif	302	14
	Positif	19	288

Dari *confusion matrix* diatas dapat dihitung akurasi, presisi, *recall*, dan *f1-score* sebagai berikut:

$$\text{Akurasi} = \frac{302+288}{302+288+14+19} = 0{,}947 \quad (4.8)$$

$$\text{Presisi\_positif} = \frac{288}{288+14} = 0{,}954 \quad (4.9)$$

$$\text{Presisi\_negatif} = \frac{302}{302+19} = 0{,}940 \quad (4.10)$$

$$\text{Recall\_positif} = \frac{288}{288+19} = 0{,}938 \quad (4.11)$$

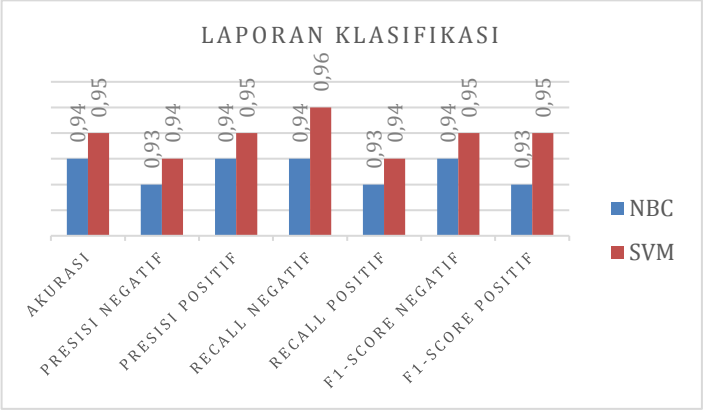
$$\text{Recall\_negatif} = \frac{302}{302+14} = 0{,}956 \quad (4.12)$$

$$\text{F1-Score\_pos} = 2 \,.\, \frac{0{,}954 \,.\, 0{,}938}{0{,}954 + 0{,}938} = 0{,}946 \quad (4.13)$$

$$\text{F1-Score\_neg} = 2 \,.\, \frac{0{,}940 \,.\, 0{,}956}{0{,}940 + 0{,}956} = 0{,}948 \quad (4.14)$$

Pada klasifikasi SVM mencapai akurasi 95% dalam mengklasifikasikan 623 sampel ulasan. Presisi kelas negatif (0.94) dan positif (0.95) menunjukkan ketepatan prediksi dengan sedikit *false positive*. *Recall* untuk kelas negatif (0.96) dan positif (0.94) menunjukkan bahwa jumlah *false negatives* yang dihasilkan cukup rendah. *F1-score* untuk kedua kelas

bernilai 0.95, yang menunjukkan model memiliki performa yang stabil dalam mendeteksi sentimen negatif maupun positif tanpa terlalu condong ke salah satu kelas tertentu.



Gambar 4. 34 Diagram Laporan Klasifikasi

Akurasi klasifikasi naive bayes hanya 94% sedangkan svm 95% yang menunjukan bahwa svm lebih unggul untuk mengklasifikasi data secara keseluruhan.

Dari presisi dan *recall* baik positif maupun negatif, svm menunjukan performa yang lebih unggul dari naive bayes dengan perbedaan paling banyak di *recall* negatif yaitu berbeda 0,02. Pada *F1-score* juga svm lebih unggul daripada naive bayes, *F1-score* positif dan negatif dari svm yang sama menunjukan

kestabilan dalam mendeteksi sentimen positif dan negatif tanpa condong ke kelas tertentu.

Secara keseluruhan klasifikasi naive bayes lebih unggul daripada klasifikasi svm dalam semua matriks evaluasi meskipun perbedaannya tidak terlalu besar.

2. Waktu Penyelesaian

Pada penelitian ini juga membandingkan waktu penyelesaian model dalam melakukan proses klasifikasi yaitu dari waktu pelatihan dan waktu prediksi.

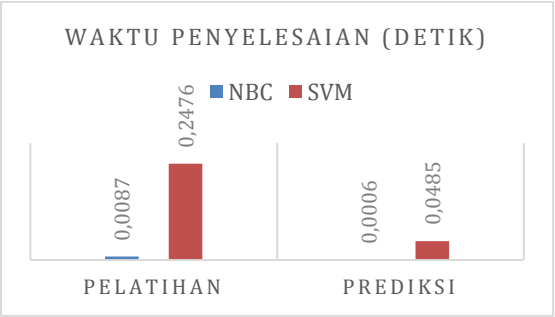
a. Waktu pelatihan

Waktu pelatihan menunjukan seberapa cepat model klasifikasi untuk belajar dari data latih. Pada klasifikasi naive bayes membutuhkan waktu 0.0087 detik untuk menyelesaikan pelatihan sedangkan pada klasifikasi svm membutuhkan waktu 0.2476 detik untuk menyelesaikan pelatihan.

b. Waktu prediksi

Waktu prediksi menunjukan seberapa cepat model klasifikasi untuk mengklasifikasikan data baru (data *test*). Pada klasifikasi naive bayes membutuhkan waktu 0.0006 detik untuk menyelesaikan proses prediksi sedangkan pada

klasifikasi svm membutuhkan waktu 0.0485 detik untuk menyelesaikan proses prediksi.



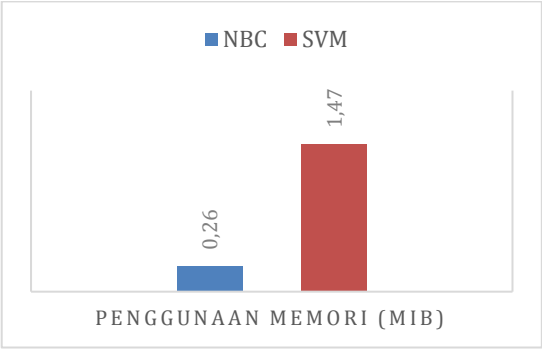
Gambar 4. 35 Diagram Waktu Penyelesaian

Pada penelitian ini klasifikasi svm membutuhkan waktu 0,2389 detik lebih lama daripada klasifikasi naive bayes untuk menyelesaikan proses pelatihan dari 2491 data sedangkan untuk menyelesaikan proses prediksi klasifikasi svm membutuhkan waktu 0,0479 detik lebih lama daripada klasifikasi naive bayes.

3. Efisiensi Sumber Daya

Pada penelitian ini, efisiensi sumber daya dilihat dari penggunaan memori puncak yaitu memori terbanyak yang digunakan dalam seluruh proses klasifikasi oleh model.

Pada klasifikasi naive bayes penggunaan memori puncak sebesar 0.26 MiB sedangkan klasifikasi svm penggunaan memori puncak sebesar 1.47 MiB.



Gambar 4. 36 Diagram Penggunaan Memori

Hasil yang diperoleh menunjukan bahwa klasifikasi svm menggunakan memori 1,21 MiB lebih banyak daripada klasifikasi naive bayes.

BAB V
PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan beberapa hal sebagai berikut:

1. Kinerja *Naive Bayes Classifier*: Metode NBC mencapai akurasi sebesar 94% dalam mengklasifikasikan sentimen positif dan negatif dari 623 sampel ulasan. Presisi positif (0,94), presisi negatif (0,93), *recall* positif (0,93), *recall* negatif (0,94), *F1-score* positif (0,93), dan *F1-score* negatif (0,94) menunjukkan bahwa model memiliki performa yang cukup baik dalam mendeteksi sentimen dengan tingkat kesalahan yang rendah.
2. Kinerja *Support Vector Machine*: Metode SVM memperoleh akurasi sebesar 95%, sedikit lebih tinggi dibandingkan Naive Bayes. Presisi positif (0,95), presisi negatif (0,94), *recall* positif (0,94), *recall* negatif (0,96), *F1-score* positif (0,95), dan *F1-score* negatif (0,95) menunjukkan bahwa model ini memiliki performa yang stabil dalam mendeteksi sentimen negatif maupun positif.
3. Perbandingan Kinerja: Meskipun SVM memiliki akurasi yang lebih tinggi dibandingkan NBC, dari segi waktu

eksekusi dan efisiensi sumber daya, NBC lebih unggul. Waktu pelatihan untuk NBC hanya 0.0087 detik dibandingkan dengan SVM yang membutuhkan 0.2476 detik. Begitu pula dengan waktu prediksi, di mana NBC hanya membutuhkan 0.0006 detik sementara SVM memerlukan 0.0485 detik. NBC juga lebih efisien dalam penggunaan memori, dengan puncak penggunaan sebesar 0.26 MiB dibandingkan dengan SVM yang mencapai 1.47 MiB.

Secara keseluruhan, SVM lebih unggul dalam akurasi dan performa klasifikasi, tetapi NBC lebih efisien dalam waktu dan penggunaan sumber daya. Pemilihan metode terbaik tergantung pada kebutuhan spesifik, apakah lebih mengutamakan akurasi atau efisiensi komputasi.

B. Saran

Berdasarkan hasil penelitian ini, beberapa saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Eksplorasi Metode *Hybrid*: Menggabungkan kedua metode, seperti *ensemble learning* atau *hybrid approach*, dapat dieksplorasi untuk meningkatkan akurasi tanpa mengorbankan efisiensi waktu dan sumber data.
2. Studi Komparatif dengan Model Lain: Penelitian selanjutnya dapat memperluas studi dengan membandingkan metode

lain seperti *Decision Tree, Random Forest,* atau *Neural Networks* untuk menemukan model yang paling optimal untuk analisis sentimen pada fintech.

3. Analisis Sentimen Multikategori: Penelitian selanjutnya dapat mengembangkan model yang mampu mengklasifikasikan sentimen ke dalam lebih dari dua kategori, seperti positif, negatif, netral, atau bahkan lebih spesifik berdasarkan aspek tertentu.

DAFTAR PUSTAKA

AdaKami. (t.thn.). *AdaKami*. Diambil kembali dari Pinjaman Uang Online Cepat Tanpa Jaminan Terdaftar OJK: https://www.adakami.id/

Akbar, A. (2015). *[Repository GitHub]*. Diambil dari GitHub:https://github.com/aliakbars/bilp/blob/master/singkatan-lib.csv

Alfiyanti, D., & Indra. (2024). Penerapan Algoritma Multinomial Naïve Bayes dan Support Vector Machine (SVM) untuk Analisis Sentimen terhadap Ulasan Google Maps di Taman Mini Indonesia Indah (TMII). *Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, *15*(1), 85–102.

Effendi, A. M. (2023). *ANALISIS SENTIMEN PADA JUDUL BERITA ONLINE EKONOMI DENGAN MENGGUNAKAN SUPPORT VECTOR MACHINE* [Skripsi]. UIN MAULANA MALIK IBRAHIM.

Eviyanti, Irawan, B., & Bahtiar, A. (2023). PENGGUNAAN ALGORITMA NAÏVE BAYES DALAM MENGANALISIS SENTIMEN ULASAN APLIKASI ADAKAMI DI GOOGLE PLAY STORE. *Jurnal Mahasiswa Teknik Informatika*, *7*(6), 3879–3885.

Fitri, E., Yuliani, Y., Rosyida, S., & Gata, W. (2020). Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine. *TRANSFORMTIKA*, *18*(1), 71–80.

Hidayat, A., Azizah, N., & Ridwan, M. (2022). Pinjaman Online dan Keabsahannya Menurut Hukum Perjanjian Islam. *Jurnal Indragiri Penelitian Multidisiplin*, *2*(1), 1–9. https://www.jurnalindrainstitute.com/index.php/jipm

Julianto, Y., Setiabudi, D. H., & Rostianingsih, S. (2022). Analisis Sentimen Ulasan Restoran Menggunakan Metode Support Vector Machine. *Jurnal Infra*, *10*(1), 1–7.

Kusuma, F. W., Dermawan, P., Samsie, I., Salman, N., Ghani, ST. A. D., & Rachman, S. R. D. (2024). TINJAUAN SENTIMEN TERHADAP ULASAN APLIKASI PEMINJAMAN ONLINE DENGAN METODE SUPPORT VECTOR MACHINE (SVM). *Journal of Artificial Intelligence and Data Science*, *4*(1), 13–21.

Koto, Fajri & Rahmaningtyas, Gemala. (2017). InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs. 10.1109/IALP.2017.8300625.

Muhammadin, A., & Sobari, I. A. (2021). Analisis Sentimen Pada Ulasan Aplikasi Kredivo Dengan Algoritma SVM Dan NBC. *Jurnal Rekayasa Perangkat Lunak*, *2*(2).

Muttaqin, M. N., & Kharisudin, I. (2021). Analisis Sentimen Pada Ulasan Aplikasi Gojek Menggunakan Metode Support Vector Machine dan K Nearest Neighbor. *UNNES Journal of Mathematics*, *10*(2), 22–27. http://journal.unnes.ac.id/sju/index.php/ujm

Ramadhan, T. D., Wahiddin, D., & Awal, E. E. (2023). Klasifikasi Sentimen Terhadap Pinjaman Online (Pinjol) Menggunakan Algoritma Naive Bayes. *Scientific Student Journal for Information, Technology and Science*, *IV*(1), 82–87.

Rokhman, K. A., Berlilana, & Arsi, P. (2021). PERBANDINGAN METODE SUPPORT VECTOR MACHINE DAN DECISION TREEUNTUK ANALISIS SENTIMEN REVIEW KOMENTAR PADA APLIKASI TRANSPORTASI ONLINE. *JURNAL OF INFORMATION SYSTEM MANAGEMENT*, *2*(2), 1–7.

Salsabila, N. A. (2022). *ANALISIS SENTIMEN PADA MEDIA SOSIAL TWITTER TERHADAP TOKOH GUS DUR MENGGUNAKAN METODE NAÏVE BAYES DAN SUPPORT VECTOR MACHINE (SVM)* [Skripsi]. UIN SYARIF HIDAYATULLAH.

Siregar, N. C., Siregar, R. R. A., & Sudirman, M. Y. D. (2020). Implementasi Metode Naive Bayes Classifier (NBC) Pada Komentar Warga Sekolah Mengenai Pelaksanaan

Pembelajaran Jarak Jauh (PJJ). *Jurnal Teknologia Aliansi Perguruan Tinggi (APERTI) BUMN*, *3*(1), 102–110.

Styawati, Hendrastuty, N., Isnain, A. R., & Rahmadhani, A. Y. (2021). Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine. *Jurnal Pengembangan IT*, *6*(3), 150–155.

Hendriyanto, M. D., Ridha, A. A., & Enri, U. (2022). Analisis Sentimen Ulasan Aplikasi Mola Pada Google Play Store Menggunakan Algoritma Support Vector Machine Sentiment Analysis of Mola Application Reviews on Google Play Store Using Support Vector Machine Algorithm. *J. Inf. Technol. Comput. Sci*, *5*(1), 1-7.

Syafutri, G. E. (2023). *PERLINDUNGAN HUKUM BAGI DEBITUR PINJAMAN FINTECH (FINTECH LENDING) YANG DI RUGIKAN DALAM TRANSAKSI PINJAMAN UANG SECARA ONLINE PADA APLIKASI "ADA KAMI"* [Skripsi]. UNIVERSITAS BATANGHARI.

Zidan, M. (2022). *Analisis Sentimen Kenaikan Harga Bahan Bakar Minyak (BBM) Berdasarkan Respon Pengguna Media Sosial Twitter Di Indonesia Menggunakan Metode Naive Bayes* [Skripsi]. UIN Walisongo.

LAMPIRAN

Lampiran 1: Contoh hasil crawling data

no	at	Content
1	2023-07-15 5:05:16	Aplikasi buruk pembayaran sudah di lunasi padahal belom jatoh tempo.. Tapi malah tidak di kasih pinjam lagi sama sekali.. Aplikasi ga sesuai iklan nya..
2	2023-07-15 5:40:47	Baru ngisi no hp buat login doang dan saya memutuskan engga jadi melanjutkan proses,eh setelah ini banyak sms yg pinjol lain ke nomor hp . Tidak bisa melindungi keamanan data !!!!
3	2023-07-15 6:13:32	AdaKami sangat membantu disaat saya benar benar membutuhkan.AdaKami bunganya sangat rendah ini baru namanya aplikasi yg membantu.Semoga lancar untuk pembayarannya.Mohon ditingkatkan lagi performa dri AdaKami,seperti pinjaman lancar tanpa harus menunggu beberapa hari/minggu untuk pinjaman kembali.Terimakasih
4	2023-07-15 8:47:38	Tidak sesuai apa yg sudah tertera di apk. Limitnya dan tidak bisa d cairkan. Sudah menunggu lama tapi tidak bisa d cairkan. Kalau memang tidak bisa d cairkan tolong hapus data saya.
5	2023-07-15 9:30:00	Barusan ada kode otp masuk lewat sms padahal saya belum mengajukan pinjaman lagi. Sekedar mengingatkan . Khusus untuk pihak adakami ada

		pihak pihak yg sengaja mau menggunakan data saya
...		
4996	2025-03-01 1:29:58	Cepat dan mudah
4997	2025-03-01 1:29:34	Aplikasi yang sangat cepat cash - pinjam online dan lend easy
4998	2025-03-01 1:23:54	Belum masuk masuk ini pinjaman nya
4999	2025-03-01 0:19:45	Mantap pencairan cepat,kagak fafifu 👆□
5000	2025-03-01 0:02:55	Aplikasi sangat baguss

Lampiran 2: Contoh teks yang sudah dilabeli

no	at	Content	Label
1	2024-09-01 2:11:32	Aplikasi apaan? Aku ndak pernah nunggak kok kredit skor kurang,,, penilaianmu dari mana dasarnya??	Negatif
2	2024-09-01 2:39:43	kenapa sekarang adakami gak bisa pinjam lagi ya,,pdhl bayar selalu tepat waktu beda bngtt pas di awal" „tolong dong in kan aplikasi pinjol masa begini sihh	Negatif
3	2024-09-01 2:49:39	Awal nya tiap pinjem di ksh dan saya juga tiap pembayaran gak prnh telat justru tunggakan tinggl 3/4 lg saya sllu lunasin tp Krn skrg pas saya udh lunasin di situ ad limit trs saya mau pinjem LG malah gak BS..di suruh nunggu 30 hri lg	Negatif
4	2024-09-01 3:31:13	Sangat mudah cara pengajuannya semoga amanah dan barokah .... Amiiiin	Positif
5	2024-09-01 5:51:13	Mantap bisa jadi solusi,di saat membutuhkan mendadak,,proses cepat makasih adakami,	Positif
6	2024-09-01 7:09:23	Aplikasinya bagus buat yang memerlukan uang dalam keadaan mendesak boleh nih di coba	Positif
7	2024-09-01 11:00:04	Baru mau pinjam sudah gagal perhari ini saya uninstal apk adakami data yg sudah masuk bila ada pihak yg menggunakan bukanlah	Negatif

		saya,apabila tersebar data saya.akan saya tuntut secara hukum.	
8	2024-09-01 11:16:12	pengajuan sangat cepat dan efesien ,membantu dalam keadaan terdesak ,terimakasih ada kami	Positif
	2024-09-01 12:04:00	Jgn prnh download apk ini...bahasa dri DC nya barokah bgt,pembayaran telat 1hari aja kaya dah telat 1bulan padahal sebelum nya sy ga prnh telat, Dikarnakan ada keterlambatan dri gajian aja sy telat,setelah pinjeman lunas kapok sy pinjem ke adakami lagi,mksh sblm nya dri pihak adakami SDH di ACC	Negatif
	2024-09-01 12:45:04	terimakasih adakami, sangat membantu dan pencairan nya cepat	Positif
...			
3168	2025-01-14 0:36:51	sangat membantu untuk kebutuhan mendadak	Positif
3169	2025-01-13 17:28:12	pinjaman dana online,dana rupiah ini sangat bagus karena ga tipu tipu langsung cair	Positif
3170	2025-01-13 12:42:30	mengganggu kenyaman muncul diiklan terus ..gak butuh aku	Negatif
3171	2025-01-13 11:41:16	Tolong donk ini iklan nya ganggu banget ampir tiap detik muncul saat oprasikan handphone ..... Mengganggu !	Negatif
3172	2025-01-13 10:50:52	Proses cepat dan mudah. Limit pengguna baru lumayan besar. Tenor	Positif

		tergantung kita, sangat meringankan kita sebagai peminjam. Semua tertera secara detail. Saya sudah banyak mencoba aplikasi lain, tpi bagi saya App Ada kami sangat membantu dlm proses peminjaman, maupun pelunasan.	
3173	2025-01-13 9:41:53	proses cepat dan sangat baik, membantu bisnis berjalan terus.	Positif
3174	2025-01-13 8:50:46	Aplikasi mengecewakan baru pengajuan di tolak saya tunggu waktu untuk pengajuan lagi pas waktu pengajuan menghitung limit malah yang kluar limit tidak cukup	Negatif
3175	2025-01-13 7:48:30	pengajuan yg sangat cepat prosesnya	Positif
3176	2025-01-12 22:51:13	adakami memang solusi terbaik di saat ada kebutuhan mendesak trima kasih adakami..	Positif
3177	2025-01-12 19:58:48	aplikasi yang sangat membantu sekali terima kasih adakami	Positif

Lampiran 3: Dokumentasi penyerahan dataset pada ahli bahasa untuk dilabeli

Lampiran 4: Contoh data pendukung "kamus_slang"

no	slang	formal
1	woww	wow
2	aminn	amin
3	met	selamat
4	netaas	menetas
5	keberpa	Keberapa
6	eeeehhhh	eh
7	kata2nyaaa	kata-katanya
8	hallo	halo
9	kaka	kakak
10	ka	kak
11	daah	dah
12	aaaaahhhh	ah
13	yaa	ya
14	smga	semoga
15	slalu	selalu
...		
15002	gataunya	enggak taunya
15003	gtau	enggak tau
15004	gatau	enggak tau
15005	fans2	fan-fan
15006	gaharus	enggak harus

Lampiran 5: Contoh teks hasil preprocessing

no	Content	Hasil Preprocessing
1	Aplikasi apaan? Aku ndak pernah nunggak kok kredit skor kurang,,, penilaianmu dari mana dasarnya??	aplikasi indak nunggak kredit skor nilai dasar
2	kenapa sekarang adakami gak bisa pinjam lagi ya,,pdhl bayar selalu tepat waktu beda bngtt pas di awal" ,,tolong dong in kan aplikasi pinjol masa begini sihh	adakami pinjam ya bayar beda banget pas tolong aplikasi pinjol sih
3	Awal nya tiap pinjem di ksh dan saya juga tiap pembayaran gak prnh telat justru tunggakan tinggl 3/4 lg saya sllu lunasin tp Krn skrg pas saya udh lunasin di situ ad limit trs saya mau pinjem LG malah gak BS..di suruh nunggu 30 hri lg🙏🙏	nya pinjem kasih bayar telat tunggak tinggal lunasin pas lunasin situ limit pinjem suruh tunggu
4	Sangat mudah cara pengajuannya semoga amanah dan barokah .... Amiiiin🤲	mudah aju moga amanah barokah amin
5	Mantap bisa jadi solusi,di saat membutuhkan mendadak,,proses cepat makasih adakami,	mantap solusi butuh dadak proses cepat terima kasih adakami
...		
3173	proses cepat dan sangat baik, membantu bisnis berjalan terus.	proses cepat bantu bisnis jalan
3174	Aplikasi mengecewakan baru pengajuan di tolak saya tunggu waktu untuk pengajuan lagi pas waktu	aplikasi kecewa aju tolak tunggu aju pas aju hitung limit limit

	pengajuan menghitung limit malah yang kluar limit tidak cukup	
3175	pengajuan yg sangat cepat prosesnya	aju cepat proses
3176	adakami memang solusi terbaik di saat ada kebutuhan mendesak trima kasih adakami..	adakami solusi baik butuh desak terima kasih adakami
3177	aplikasi yang sangat membantu sekali terima kasih adakami	aplikasi bantu terima kasih adakami

Lampiran 6: Daftar riwayat hidup

Riwayat Hidup

A. Identitas Diri

1.	Nama Lengkap	:	Mirza Izzal Musyaffaq
2.	Tempat & Tanggal Lahir	:	Tegal, 25 Januari 2003
3.	Alamat	:	Ds. Pedagangan, RT 03, RW 02, Kec. Dukuhwaru, Kab. Tegal
4.	HP	:	085157555751
5.	Email	:	Im25tgl@gmail.com

B. Riwayat Pendidikan

1. Sekolah Dasar (SD) Negeri Kudaile 05
2. Sekolah Menengah Pertama (SMP) Negeri 1 Slawi
3. Sekolah Menengah Atas (SMA) Negeri 1 Slawi

Semarang, 2 Juni 2025

Pembuat Pernyataan,

Mirza Izzal Musyaffaq

NIM: 2108096040