

**PERBANDINGAN METODE *SUPPORT VECTOR MACHINE* DENGAN  
*NAÏVE BAYES* UNTUK ANALISIS SENTIMEN PADA ULASAN  
APLIKASI DISCORD DI GOOGLE PLAYSTORE**

**SKRIPSI**

**DISUSUN OLEH**

**Robby Suriansyah**

**NPM. 2109020122**



**UMSU**

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS ISLAM MUHAMMADIYAH SUMATERA UTARA**

**MEDAN**

**2025**

**LEMBAR PENGESAHAN**

Judul Skripsi : PERBANDINGAN METODE *SUPPORT VECTOR MACHINE* DENGAN *NAÏVE BAYES* UNTUK ANALISIS SENTIMEN PADA ULASAN APLIKASI DISCORD DI GOOGLE PLAYSTORE

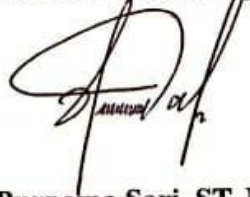
Nama Mahasiswa : ROBBY SURIANSYAH

NPM : 2109020122

Program Studi : TEKNOLOGI INFORMASI

Menyetujui

Komisi Pembimbing



**(Indah Purnama Sari, ST.,M.Kom)**

**NIDN. 0116049001**

Ketua Program Studi



**(Fatma Sari Hutagalung, S.Kom., M.Kom.) (Dr. Al-Khowarizmi, S.Kom., M.Kom.)**

**NIDN. 0117019301**

Dekan



**NIDN. 0127099201**

**PERBANDINGAN METODE *SUPPORT VECTOR MACHINE* DENGAN  
*NAÏVE BAYES* UNTUK ANALISIS SENTIMEN PADA ULASAN  
APLIKASI DISCORD DI GOOGLE PLAYSTORE**

**SKRIPSI**

Saya menyatakan bahwa karaya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 23 Mei 2026

Yang membuat pernyataan



Robby Surtiansyah

NPM. 2109020122

**PERNYATAAN PERSETUJUAN PUBLIKASI  
KARYA ILMIAH UNTUK KEPENTINGAN  
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan di bawah ini:

Nama : Robby Suriansyah  
NPM : 2109020122  
Program Studi : Teknologi Informasi  
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (Non-Exclusive Royalti free Right) atas penelitian skripsi saya yang berjudul:

**PERBANDINGAN METODE *SUPPORT VECTOR MACHINE*  
DENGAN *NAÏVE BAYES* UNTUK ANALISIS SENTIMEN PADA ULASAN  
APLIKASI DISCORD DI GOOGLE PLAYSTORE**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 2026

Yang membuat pernyataan



Robby Suriansyah

NPM. 2109020122

## **RIWAYAT HIDUP**

### **DATA PRIBADI**

Nama Lengkap : Robby Suriansyah  
Tempat dan Tanggal Lahir : Rantau Prapat, 20 agustus 2003  
Alamat Rumah : Komplek Royal Monaco Blok C Jl. Eka  
Surya  
Telepon/Faks/HP : 081244845990  
E-mail : robbysurian03@gmail.com

### **DATA PENDIDIKAN**

SD : SD Negeri 4 Mentawa Baru Hilir TAMAT: 2015  
SMP : SMP Negeri 2 Sampit TAMAT: 2018  
SMK : SMA Negeri 2 Sampit TAMAT: 2021

## KATA PENGANTAR



### Pendahuluan

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Akrim. M.Pd., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Dr. Firahmi Rizky, S.Kom., M.Kom. Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
4. Bapak Mhd. Basri, S.Si, M.Kom. Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
5. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom. Ketua Program Studi Teknologi Informasi.
6. Bapak Okvi Nugroho, S.Kom., M.Kom. Sekretaris Program Studi Teknologi Informasi.
7. Pembimbing saya, yaitu Ibu Indah Purnama Sari, ST.,M.Kom.
8. Kedua orang tua saya yang sudah mendukung saya dalam penyelesaian kuliah ini.
9. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

**PERBANDINGAN METODE *SUPPORT VECTOR MACHINE* DENGAN  
*NAÏVE BAYES* UNTUK ANALISIS SENTIMEN PADA ULASAN  
APLIKASI DISCORD DI GOOGLE PLAYSTORE**

**ABSTRAK**

Riset ini diorientasikan guna mengomparasikan efektivitas klasifikasi antara algoritma *Support Vector Machine* (SVM) dan *Naïve Bayes* dalam memetakan polaritas sentimen pada kolom ulasan pengguna aplikasi Discord di Google Play Store. Korpus data diekstraksi secara digital melalui teknik *web scraping* memanfaatkan pustaka *google-play-scraper* berbasis bahasa pemrograman Python. Melalui prosedur tersebut, berhasil dihimpun sebanyak 952 entitas ulasan berbahasa Indonesia yang terdistribusi ke dalam kategori opini positif dan negatif. Alur pengolahan data diawali dengan serangkaian operasi *preprocessing* yang mengeksekusi tahapan *case folding*, *cleansing*, *tokenizing*, *stopword removal*, hingga *stemming*. Selanjutnya, pembobotan berbasis nilai *TF-IDF* diterapkan untuk merepresentasikan teks ke dalam bentuk numerik sebelum diumpankan pada proses pemodelan klasifikasi. Pengukuran performansi komputasional diuji lewat fungsionalitas *confusion matrix* yang mencakup parameter kalkulasi akurasi, presisi, *recall*, serta *F1-score*. Hasil temuan empiris mengonfirmasi keunggulan arsitektur *Support Vector Machine* atas model *Naïve Bayes*. Secara definitif, SVM menorehkan persentase akurasi di angka 87,96%, presisi 75,68%, *recall* 91,80%, dan capaian *F1-score* 82,97%. Sebagai pembandingan, pengklasifikasi *Naïve Bayes* membukukan raihan akurasi sebesar 83,77%, presisi 58,11%, daya *recall* menyentuh 100%, disertai fiksasi *F1-score* senilai 73,52%. Melalui visualisasi diagnostik ini, ditarik kesimpulan bahwa model *Support Vector Machine* menyajikan tingkat ketangguhan dan kestabilan yang jauh lebih superior untuk mengategorikan tren opini publik pada platform Discord.

**Kata Kunci:** Analisis Sentimen, *Support Vector Machine*, *Naïve Bayes*, Discord, *TF-IDF*.

**COMPARISON OF SUPPORT VECTOR MACHINE AND NAÏVE BAYES  
METHODS FOR SENTIMENT ANALYSIS ON DISCORD APPLICATION  
REVIEWS IN GOOGLE PLAY STORE**

**ABSTRACT**

*This study is oriented toward evaluating and comparing the classification efficiency of the Support Vector Machine (SVM) and Naïve Bayes algorithms in mapping sentiment polarity from user reviews of the Discord application on the Google Play Store. The data corpus was digitally extracted via web scraping techniques using the Python-based google-play-scraper library. This procedure yielded a total of 952 Indonesian-language review entities, categorized into positive and negative opinions. The computational pipeline initiated with data preprocessing operations spanning case folding, cleansing, tokenizing, stopword removal, and stemming phases. Subsequently, TF-IDF vectorization was implemented for text-to-numerical representation before initiating the predictive modeling process. Performance evaluation was conducted using a confusion matrix framework capturing metrics of accuracy, precision, recall, and F1-score. Empirical findings confirm the superiority of the Support Vector Machine architecture over the Naïve Bayes approach. Definitively, SVM recorded an accuracy rate of 87.96%, precision of 75.68%, recall of 91.80%, and an F1-score of 82.97%. In comparison, the Naïve Bayes classifier posted an accuracy of 83.77%, precision of 58.11%, a recall rate reaching 100%, and an F1-score of 73.52%. Based on these diagnostic outcomes, it is concluded that the Support Vector Machine model offers significantly superior robustness and stability for categorizing public opinion trends on the Discord platform.*

*Keywords: Sentiment Analysis, Support Vector Machine, Naïve Bayes, Discord, TF-IDF.*

## DAFTAR ISI

<b>LEMBAR PENGESAHAN</b> .....	Error! Bookmark not defined.
<b>PERNYATAAN ORISINALITAS</b> .....	Error! Bookmark not defined.
<b>PERNYATAAN PERSETUJUAN PUBLIKASI</b> .....	Error! Bookmark not defined.
<b>RIWAYAT HIDUP</b> .....	ii
<b>KATA PENGANTAR</b> .....	vi
<b>ABSTRAK</b> .....	vii
<b>ABSTRACT</b> .....	viii
<b>DAFTAR ISI</b> .....	ix
<b>DAFTAR TABEL</b> .....	xi
<b>DAFTAR GAMBAR</b> .....	xii
<b>BAB I PENDAHULUAN</b> .....	1
1.1. Latar Belakang Masalah .....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah .....	3
1.4. Tujuan Penelitian .....	3
1.5. Manfaat Penelitian .....	4
<b>BAB II TINJAUAN PUSTAKA</b> .....	6
2.1. Analisis Sentimen .....	6
2.2. Ulasan Pengguna di Google Play Store .....	6
2.3. Aplikasi Discord .....	7
2.4. <i>Python</i> .....	9
2.5. <i>Google colab</i> .....	10
2.6. <i>Support Vector Machine</i> .....	11
2.7. <i>Naïve Bayes</i> .....	11
2.8. <i>Confusion Matrix</i> .....	12
2.9. Penelitian Terdahulu.....	13
<b>BAB III METODOLOGI PENELITIAN</b> .....	37
3.1. Pendekatan Penelitian .....	37
3.2. Jenis dan Sumber data.....	37
3.2.1. Jenis Data.....	37
3.2.2. Sumber Data .....	37
3.2.3. Pengumpulan Data.....	38
3.2.4. Pembagian Data .....	38
3.3. Tahapan pemrosesan data .....	38

3.3.1. <i>Case Folding</i> .....	38
3.3.2. <i>Cleansing</i> .....	38
3.3.3. <i>Tokenizing</i> .....	39
3.3.4. <i>Stopword Removal</i> .....	39
3.3.5. <i>Stemming</i> .....	39
3.4. Penerapan metode.....	39
3.4.1 Metode <i>support Vector machine</i> .....	39
3.4.2. Metode <i>Naïve Bayes</i> .....	41
3.5. Waktu dan tempat penelitian.....	44
3.5.1. waktu penelitian.....	44
3.5.2. Tempat Penelitian .....	44
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	46
4.1. Hasil Pengumpulan Data .....	46
4.2. Tahapan Preprocessing Data .....	48
4.3. Transformasi Data dengan TF-IDF .....	51
4.3.1. Training Model <i>Naïve Bayes</i> .....	51
4.3.2. Training Model SVM.....	52
4.4. Hasil Klasifikasi.....	53
4.5. <i>Word Cloud</i> Analisis .....	55
4.5.1. <i>Word Cloud</i> Semua Ulasan .....	56
4.5.2. <i>Word Cloud</i> Ulasan Positif.....	56
4.5.3. <i>Word Cloud</i> Ulasan Negatif.....	57
4.6. Evaluasi Hasil dengan <i>Confusion Matrix</i> .....	57
4.6.1. <i>Naïve Bayes</i> .....	57
4.6.2. SVM.....	58
4.7. Visualisasi Hasil Perbandingan .....	59
<b>BAB V PENUTUP</b> .....	61
5.1. Kesimpulan.....	61
5.2. Saran.....	62
<b>DAFTAR PUSTAKA</b> .....	63
<b>LAMPIRAN</b> .....	66
Lampiran 1 .....	66
Lampiran 2 .....	67

## DAFTAR TABEL

Tabel 2.1. Penelitian Terdahulu.....	13
Tabel 3.1. Jadwal Penelitian.....	44
Tabel 4.1. Ringkasan Metrik Evaluasi Naïve Bayes dan SVM.....	59

## DAFTAR GAMBAR

Gambar 2.1 Logo Aplikasi Discord .....	8
Gambar 2.2 Logo Phyton .....	9
Gambar 3.1 Model SVM.....	40
Gambar 3.2 Model Naïve bayes.....	42
Gambar 4.1 Proses Web Scraping .....	47
Gambar 4.2 Hasil Web Scraping .....	47
Gambar 4.3 Preprocessing data .....	49
Gambar 4.4 Hasil dari Preprocessing data .....	50
Gambar 4.5 Pembobotan TF-IDF .....	51
Gambar 4.6 Hasil Training Model Naïve Bayes .....	52
Gambar 4.7 Hasil Training Model SVM .....	52
Gambar 4.8 Input Confusion Matrix.....	53
Gambar 4.9 Output Confusion Matrix .....	54
Gambar 4.10 Perbandingan Kinerja Naïve Bayes vs SVM .....	55
Gambar 4.11 Word Cloud Semua Ulasan Discord.....	56
Gambar 4.12 Word Cloud Ulasan Positif.....	56
Gambar 4.13 Word Cloud Ulasan Negatif .....	57
Gambar 4.14 Distribusi Sentimen Ulasan Aplikasi Discord.....	59

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Pesatnya kemajuan teknologi informasi saat ini telah memicu lonjakan pemanfaatan perangkat lunak digital untuk menunjang kegiatan harian manusia. Salah satu platform yang kini sangat populer, khususnya di kalangan anak muda dan para pegiat komunitas *game online*, adalah Discord. Perangkat lunak ini memfasilitasi pertukaran pesan teks, panggilan suara, hingga konferensi video terpadu melalui ruang komunikasi virtual yang sering disebut *server*. Lonjakan popularitas ini berbanding lurus dengan peningkatan volume komentar dan opini pengguna yang membanjiri etalase aplikasi, terutama di Google Play Store.

Kumpulan opini dari para pengguna ini sebenarnya merupakan aset informasi krusial untuk memetakan bagaimana publik menilai kualitas suatu aplikasi. Namun, volume data teks yang sangat masif dan bersifat tidak terstruktur (*unstructured data*) membuat proses analisis secara manual menjadi langkah yang lambat dan sama sekali tidak efisien. Untuk merespons kendala operasional ini, dibutuhkan sebuah pendekatan sistem yang mampu mengkategorikan sentimen publik secara presisi, cepat, dan otomatis. Di sinilah algoritma analisis sentimen diimplementasikan sebagai solusi praktis atas tantangan tersebut.

Dalam disiplin ilmu *data mining* dan *machine learning*, terdapat beragam metode klasifikasi teks yang sering diimplementasikan, dua di antaranya adalah *Support Vector Machine* (SVM) dan *Naïve Bayes* (NB). Kedua algoritma ini sering dijadikan andalan dalam berbagai riset analisis sentimen karena rekam jejaknya yang mumpuni dalam memproses dataset berbasis tekstual. Algoritma *Support Vector Machine* dikenal luas memiliki keunggulan dalam menentukan margin pemisah kelas secara optimal, terutama ketika dihadapkan pada data dengan dimensi fitur yang sangat tinggi layaknya teks. Di sisi lain, *Naïve Bayes* menawarkan pendekatan klasifikasi berbasis probabilitas yang lebih ringkas namun tetap kompeten untuk memecahkan masalah teks yang distribusi kerumitannya tidak terlalu ekstrem.

Meskipun kedua pendekatan ini lazim digunakan, hasil akhir klasifikasinya sangat bergantung pada sifat bawaan dataset, kelengkapan proses *preprocessing*, dan teknik penyusunan fitur yang diaplikasikan. Berangkat dari premis tersebut, penelitian ini digagas untuk menguji dan membandingkan secara komprehensif kapabilitas *Support Vector Machine* dengan *Naïve Bayes* dalam mendeteksi polaritas sentimen ulasan Discord pada Google Play Store. Kualitas prediksi model nantinya akan dievaluasi menggunakan serangkaian parameter pengukuran matriks kebingungan (*confusion matrix*), yang meliputi tingkat akurasi, presisi, *recall*, serta nilai *F1-score*.

Riset ini berupaya membedah rasio perbandingan kinerja antara algoritma SVM dan *Naïve Bayes* ketika menganalisis opini pengguna Discord, baik itu sentimen yang berkonotasi positif maupun negatif. Melalui pengujian terstruktur dengan metrik evaluasi tersebut, diharapkan dapat ditemukan rumusan algoritma klasifikasi mana yang paling handal dan paling relevan untuk diimplementasikan pada studi kasus ini.

## **1.2. Rumusan Masalah**

Berdasarkan pemaparan latar belakang di atas, rumusan permasalahan dalam penelitian ini dapat didefinisikan sebagai berikut:

1. Sejauh mana tingkat kehandalan algoritma *Support Vector Machine* dalam memetakan polaritas sentimen pengguna terhadap aplikasi Discord di platform Google Play Store?
2. Bagaimana efektivitas model *Naïve Bayes* saat diimplementasikan untuk mengkategorikan dataset sentimen yang identik?
3. kedua algoritma tersebut (SVM dan *Naïve Bayes*), manakah pendekatan yang mampu menghasilkan performa paling unggul apabila diukur menggunakan parameter evaluasi standar

### 1.3. Batasan Masalah

Agar pembahasan masalah yang dilakukan tetap terfokus dari pokok permasalahan, maka ditetapkan beberapa parameter yang akan dibatasi sebagai pembahasan berikut:

1. Data yang diambil dari ulasan pengguna aplikasi Discord hanya berbahasa Indonesia dan tidak dibatasi kemungkinan menggunakan bahasa *Slank* pada ulasan Google Play Store
2. Penelitian sentimen hanya difokuskan pada pendekatan klasifikasi biner, yaitu hanya mengidentifikasi kelas positif dan negatif. Model sentimen yang berindikasi netral disertakan dalam proses pengujian.
3. Tahapan pra-pemrosesan data (*preprocessing*) mencakup serangkaian prosedur standar pemrosesan teks, meliputi penyeragaman huruf (*case folding*), pembersihan data dari entitas yang tidak relevan (*cleansing*), pemotongan kalimat menjadi kata (*tokenizing*), pembuangan kata umum yang tidak bermakna (*stopword removal*), serta pengembalian kata ke bentuk dasar (*stemming/lemmatization*).
4. Komparasi kinerja model klasifikasi secara eksklusif hanya dilakukan antara *Support Vector Machine* (SVM) dan *Naïve Bayes* (NB).
5. Penelitian ini tidak mencakup analisis terhadap variabel non-teknis, semisal dampak pembaruan versi perangkat lunak tertentu maupun latar belakang demografis pengguna dalam memengaruhi sentimen.

### 1.4. Tujuan Penelitian

Adapun objektif yang ingin dicapai melalui pelaksanaan penelitian ini mencakup:

1. Mengukur dan mengevaluasi kapabilitas metode *Support Vector Machine* saat mengklasifikasikan teks opini pengguna Discord (menjadi sentimen positif maupun negatif) dari platform Google Play Store.

2. Menyajikan perbandingan kinerja komprehensif antara algoritma *Support Vector Machine* dan *Naïve Bayes*, yang diukur secara kuantitatif berdasarkan metrik evaluasi seperti akurasi, presisi, *recall*, serta *F1-score*.
3. Memberikan rekomendasi ilmiah mengenai metode algoritma klasifikasi mana yang paling optimal untuk diimplementasikan dalam menganalisis sentimen teks, dengan studi kasus spesifik pada aplikasi Discord.

### 1.5. Manfaat Penelitian

Penelitian ini diharapkan mampu memberikan kontribusi positif, baik secara praktis maupun teoretis, di antaranya:

1. Manfaat bagi objek kajian:

Menyediakan informasi strategis terkait persepsi publik terhadap layanan Discord, yang nantinya dapat dimanfaatkan oleh pihak pengembang sebagai dasar evaluasi untuk melakukan peningkatan fitur maupun kinerja operasional aplikasi.

2. Bagi penulis :

Menjadi wadah implementasi nyata untuk memperdalam wawasan dan kemampuan praktis seputar pembuatan model *machine learning*, khususnya pada ranah *sentiment analysis*.

3. Bagi pembaca:

Memberikan rujukan literatur komparatif mengenai sejauh mana tingkat integrasi dan efisiensi performa algoritma SVM dibandingkan dengan *Naïve Bayes*. Selain itu, hasil riset ini dapat dialihfungsikan sebagai landasan teoretis untuk pengembangan model klasifikasi teks yang lebih mutakhir di masa mendatang.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Analisis Sentimen**

Analisis yang dilakukan oleh Sugiyono (2019), tahapan analisis dapat dipahami sebagai upaya mendekomposisi sebuah fenomena atau objek menjadi unit-unit yang lebih terperinci, dengan tujuan utama untuk memperjelas relasi, fungsi, dan struktur pembentuknya. Melalui tahapan analisis inilah, kumpulan data mentah diekstraksi sedemikian rupa guna menghasilkan konklusi yang sejalan dengan perumusan masalah penelitian.

Sementara itu, analisis sentimen dikenal sebagai metode komputasi linguistik yang bertugas menambang dan mengklasifikasikan kecenderungan opini pengguna terhadap suatu entitas, seperti layanan atau produk. Berdasarkan muatan sintaksisnya, algoritma akan memetakan opini tersebut ke dalam polaritas sentimen positif, negatif, maupun netral. Sebagaimana dijelaskan oleh Junianto et al. (2024), adopsi *sentiment analysis* kini semakin masif di berbagai platform daring, termasuk pada sistem ulasan aplikasi dan media sosial, guna membedah tingkat penerimaan publik terhadap sebuah layanan.

Secara praktis, luaran dari klasifikasi ini berpotensi besar untuk diolah menjadi bahan rujukan utama dalam mengevaluasi dan merumuskan pembaruan perangkat lunak.

#### **2.2. Ulasan Pengguna di Google Play Store**

Ulasan pengguna yang tersedia di etalase digital seperti Google Play Store berfungsi sebagai wadah interaktif bagi konsumen untuk menyalurkan umpan balik—baik berupa apresiasi, keluhan, maupun saran penyempurnaan fitur dan antarmuka aplikasi. Pratama & Astuti (2021) menegaskan bahwa jejak digital berupa rekam jejak penilaian dan pemberian *rating* (dari skala 1 hingga 5 bintang) ini merupakan aset esensial bagi pihak pengembang dalam mengukur mutu produk berbasis pengalaman riil audiens. Lebih jauh, konklusi sosial dari ulasan ini sangat krusial karena terbukti mampu memengaruhi persepsi pengguna baru sebelum mereka memutuskan untuk melakukan instalasi perangkat lunak.

Dalam lanskap rekayasa perangkat lunak, Kurniawan et al. (2020) menyoroti bahwa ekstraksi pola dari tumpukan opini pengguna merupakan pilar penting dalam mewujudkan *data-driven development*. Kendati demikian, karakteristik ulasan yang berjumlah raksasa dan bersifat nir-struktur membuat pendekatan analitis secara manual sangat dihindari karena memakan waktu dan rentan bias. Oleh karena itu, Salsabila et al. (2023) merekomendasikan intervensi *machine learning* untuk mengotomatisasi pengolahan bahasa alami tersebut secara lebih cermat.

### 2.3. Aplikasi Discord

Discord adalah sebuah platform komunikasi yang di perkenalkan perdana pada tahun 2015 oleh Jason Citron bersama Stan Vishnevskiy (melalui Hammer & Chisel), Discord pada mulanya dirancang secara spesifik sebagai medium komunikasi VoIP (*Voice over Internet Protocol*) guna memfasilitasi interaksi dalam ekosistem permainan video daring. Seiring evolusi digital, fungsionalitas platform terpadu ini berhasil menarik atensi demografi yang jauh lebih luas. Kini, eksistensi Discord merambah pesat ke ranah profesional dan akademis, mencakup ruang komunitas non-*gaming* seperti kelompok belajar virtual hingga koordinasi bisnis.

Arsitektur ruang kerja Discord bertumpu pada sistem *server* kustom yang menaungi ruang-ruang diskusi tematik. Di dalam infrastruktur *server* tersebut, partisipan dapat berkomunikasi lewat *text channels* maupun *voice channels*, bertukar data, hingga memancarkan siaran layar *real-time* via fitur *Go Live*. Menurut Salsabila et al. (2023), fleksibilitas tingkat tinggi dan antarmuka yang sangat responsif (*user-friendly*) inilah yang mendaulat Discord sebagai salah satu instrumen kolaborasi lintas guna paling efektif saat ini. Fungsionalitas tersebut semakin diperkaya dengan dukungan otomatisasi melalui integrasi bot (seperti bot moderasi, pemutar musik, atau pengingat agenda).



**Gambar 2.1** Logo Aplikasi Discord

(Sumber: (discord, 2025))

Berikut ini Fungsi dan Kegunaan Aplikasi Discord antara lain.

1. Tempat Berkumpulnya Komunitas Online : Discord berfungsi sebagai ruang virtual komunitas (melalui fitur *server*), di mana pengguna dapat berkumpul berdasarkan minat tertentu: gaming, teknologi, musik, pendidikan, seni, dan lainnya. Setiap server bisa dikustomisasi dan dikelola secara mandiri oleh moderator.
2. Kolaborasi Tim: Banyak digunakan oleh tim pengembang, komunitas open-source, hingga kelompok kerja jarak jauh untuk.
  - a. Koordinasi kerja harian.
  - b. Kolaborasi dalam proyek bersama.
  - c. Diskusi ide dan bertukar ide.

Dengan dukungan bot dan integrasi tools seperti Trello, GitHub, dan Google Calendar, Discord bisa dimanfaatkan sebagai hub produktivitas tim.

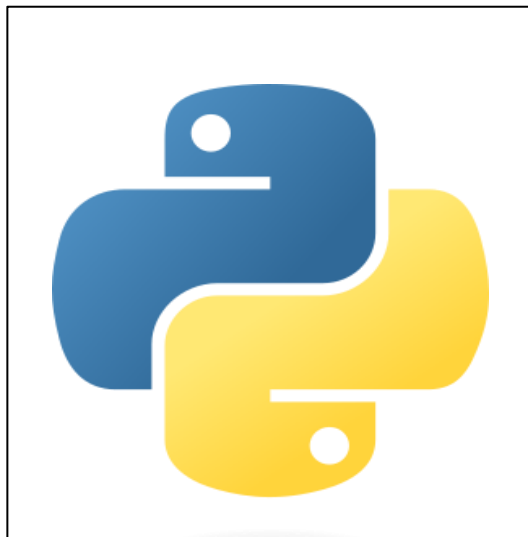
3. Live Streaming dan Siaran Komunitas: Melalui fitur *Go Live*, pengguna dapat menyiarkan layar mereka kepada anggota server lain, misalnya

untuk menampilkan proses pengajaran, permainan, atau presentasi proyek secara real-time.

4. Discord mendukung berbagai bot otomatis yang dapat memperkaya fungsinya, seperti:
  - a. Bot moderasi (misalnya MEE6).
  - b. Bot musik dan audio
  - c. Bot kuis atau pembelajaran interaktif
  - d. Bot pengingat tugas dan manajemen waktu

#### 2.4. Python

*Python* Digagas oleh Guido van Rossum dan dirilis perdana pada tahun 1991, Python tampil sebagai bahasa pemrograman *high-level* bertipe *interpreted* yang memfasilitasi pelbagai paradigma rekayasa perangkat lunak, mulai dari arsitektur fungsional, rasional-prosedural, hingga abstraksi *object-oriented* (Lutz, 2013).



**Gambar 2.2** Logo python

(Sumber: (Python, 2025))

*Python* Tio & Kurniawan (2020) merangkum sejumlah pilar fundamental yang mendaulat Python sebagai standar emas di kalangan akademisi maupun praktisi, di antaranya:

1. struktur pengkodean yang sangat sederhana sehingga mudah dipelajari, fleksibilitas integrasi lintas *platform* (Windows, Linux, macOS).

2. serta dukungan masif komunitas global yang rutin mempublikasikan pustaka (*library*) *open-source*.

dalam dimensi penelitian saintifik dan *data mining*, Sulistyono et al. (2021) menggarisbawahi urgensi Python sebagai perkakas absolut untuk membangun pipeline pemodelan prediktif. Kemampuannya terakselerasi sempurna oleh deretan *library* khusus seperti Pandas, NumPy, dan Scikit-learn yang secara drastis mempermudah manipulasi statistik dan klasifikasi linguistik tekstual.

*Python* telah menjadi bahasa utama dalam berbagai bidang penelitian dan aplikasi industri modern. Fleksibilitas dan ekosistem pustaka yang lengkap membuat *Python* menjadi pilihan, karena kemampuannya dalam menangani berbagai tugas seperti:

1. Pengolahan data dan analisis statistik (menggunakan pustaka seperti *Pandas*, *NumPy*).
2. Pembelajaran mesin dan kecerdasan buatan (*scikit-learn*, *TensorFlow*, *PyTorch*).
3. Web development (*Flask*, *Django*)
4. Otomatisasi dan scripting
5. Visualisasi data (*Matplotlib*, *Seaborn*)

### **2.5. Google colab**

*Google Colab* merupakan pemrograman yang menggunakan *cloud* Sebagai sarana eksekusi komputasional, riset ini mengandalkan platform *Google Colaboratory* (Google Colab). Beroperasi di atas arsitektur komputasi awan (*cloud computing*) adaptasi *Jupyter Notebook*, Colab mengeleminasi hambatan instalasi dan konfigurasi perangkat keras lokal bagi para periset (Google Colab, 2025).

Lebih lanjut, Andarsyah & Yanuar (2024) memaparkan bahwa fasilitas ruang kerja kolaboratif ini tidak hanya menyediakan pustaka bawaan *machine learning* secara utuh, melainkan juga menghibahkan akses akselerasi berbasis GPU gratis, yang terbukti krusial dalam menekan durasi pemrosesan data bervolume raksasa.

## 2.6. Support Vector Machine

*Support Vector Machine* (SVM) diperkenalkan oleh Vladimir Vapnik dan kolega komputasinya pada dekade 90-an sebagai bagian integral dari pendekatan *supervised learning* (Cortes & Vapnik, 1995). Filosofi algoritma ini berpusat pada penemuan batas keputusan linier atau *hyperplane* optimal pada ruang vektor berdimensi tinggi. Secara taktis, SVM mendemarkasi titik-titik data (atau *support vectors*) antar-kelas secara maksimal sedemikian rupa sehingga menciptakan celah margin pemisah yang paling luas.

Oktavia et al. (2023) menekankan bahwa resiliensi SVM dalam menangani dataset berskala kompleks serta mumpuni pada kondisi data non-linier, menjadikannya opsi primadona dalam menangani studi kasus analitik linguistik dan deteksi fitur. Formulasi klasifikasi dasar SVM dinyatakan dengan persamaan matematis berikut:

$$f(x) = (w \cdot x + b)$$

Dimana:

$f(x)$  = Proyeksi fungsi parameter keputusan untuk sampel  $x$

$x$  = titik dan masukan (vektor input)

$w$  = Pembebanan bobot (*weight vector*)

$b$  = Nilai parameter *bias*

## 2.7. Naïve Bayes

Sebagai metode komparatif, *Naïve Bayes* bertumpu pada kalkulasi rasio probabilitas kondisional yang diwariskan dari postulat Teorema Bayes. Sematan diksi "*naïve*" merepresentasikan asumsi penyederhanaan yang diterapkan oleh algoritma ini, yakni anggapan bahwa setiap kemunculan fitur independen dan tidak memiliki dependensi matematis satu sama lain, meskipun dalam praktik nyatanya jarang demikian.

Kendati bersandar pada premis kepolosan absolut tersebut, Zhang (2004) membuktikan bahwa struktur *Naïve Bayes* sangat gesit dan tangguh, serta kerap menghasilkan presisi tebakan yang kompetitif saat difungsikan untuk klasifikasi sentimen berbasis kata (teks). Model probabilitasnya dapat dikalkulasi melalui rumus konvensional di bawah ini:

$$P(Y/X) = \frac{P(X/Y) \cdot P(Y)}{P(X)}$$

Dimana:

$P(Y/X)$  = Nilai probabilitas posterior dari target  $Y$  terhadap variabel bukti  $X$

$P(Y)$  = Kemunculan prioritas kelas  $Y$  secara umum (*prior*)

$P(X/Y)$  = probabilitas kemunculan data  $X$  jika diketahui kelas  $Y$

$P(X)$  = probabilitas keseluruhan dari data  $XXX$  (*evidence*)

## 2.8. Confusion Matrix

Demi menakar presisi empiris dari *output* prediktif yang dicetuskan oleh kedua model algoritma, diterapkan protokol evaluasi matriks kebingungan (*confusion matrix*). Berdasarkan paparan Sari (2020), metode tabulasi ini secara tajam memetakan silang seberapa banyak prediksi label model yang sejajar dengan fakta label aktual dari *testing set*, serta mengisolasi jenis kesalahan (*error*) yang muncul dalam setiap himpunan kategori kelas.

Untuk mengkuantifikasi parameter performa sistem—yakni merangkum kalkulasi nilai Akurasi, Presisi, *Recall*, dan harmonisasi *F1-Score*—digunakan formulasi dasar matriks berikut:

matrix antara lain:

- Akurasi dapat dihitung dengan rumus:

$$accuracy: \frac{TP + TN}{P + TN + FP + FN}$$

- Presisi didefinisikan sebagai:

$$precision = \frac{TP}{TP + FP}$$

- *Recall* atau sensitivitas diukur dengan rumus:

$$Recall = \frac{TP}{TP + FN}$$

- F1-Score:

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

## 2.9. Penelitian Terdahulu

Dalam menyusun landasan studi ini, penulis menghimpun serangkaian literatur terdahulu sebagai komparasi dan acuan fundamental. Referensi tersebut diuraikan secara ringkas pada Tabel 2.1 di bawah ini guna menegaskan distingsi metodologis riset ini.

**Tabel 2.1. Penelitian Terdahulu**

No	Judul	Penulis	Kelebihan	Kekurangan
1	Analisis Sentimen Kualitas Aplikasi Discord Menggunakan Algoritma Naïve Bayes dan Support Vector Machine.	Fareza, I. D., & Handayani, E. T. E.	menunjukkan bahwa SVM menghasilkan akurasi lebih tinggi (88%) dibandingkan dengan Naïve Bayes (78%). Hal ini menunjukkan bahwa SVM lebih efektif dalam menangani data dengan kelas-kelas sentimen, sehingga memberikan informasi yang lebih baik kepada pengembang Discord	Meskipun algoritma Support Vector Machine (SVM) menunjukkan akurasi yang lebih tinggi dibandingkan Naïve Bayes, masih terdapat peluang untuk meningkatkan akurasi keseluruhan model melalui pengembangan lebih lanjut dan penggunaan algoritma

No	Judul	Penulis	Kelebihan	Kekurangan
				klasifikasi lainnya, seperti deep learning yaitu CNN, LSTM, atau Transformer
2	Implementasi Algoritma SVM dan Naive Bayes untuk Analisis Sentimen pada Ulasan Pengguna Aplikasi OneDrive.	Widodo, A. E., Wati, F. F., & Hidayati, N.	Jurnal ini menggunakan dua algoritma yang terkenal dalam analisis sentimen, yaitu Naive Bayes dan Support Vector Machine (SVM). Kedua metode ini dapat memberikan perbandingan yang jelas mengenai hasil klasifikasi sentimen	Meskipun Naive Bayes adalah metode yang baik, akurasi (75,79%) lebih rendah dibandingkan dengan SVM (83,60%). Hal ini menunjukkan bahwa ada kemungkinan karakteristik data yang mungkin lebih cocok untuk algoritma SVM sebagaimana metode ini diimplementasikan dalam sistem.
3	Analisis Sentimen Aplikasi X Pada Google Play Store Menggunakan Algoritma Naive Bayes Dan Support	Eskiyaturrofikoh, E., & Suryono, R. R.	Penggunaan teknik SMOTE (Synthetic Minority Over-sampling Technique) untuk menangani ketidakseimbangan	Meskipun Naive Bayes adalah metode yang cepat dan sederhana, hasilnya tetap tidak berubah setelah

No	Judul	Penulis	Kelebihan	Kekurangan
	Vector Machine (SVM)		kelas menunjukkan keinginan untuk meningkatkan akurasi model. Hasil menunjukkan bahwa SVM mengalami peningkatan akurasi yang signifikan setelah penerapan SMOTE, dari 75,5% menjadi 81%	penerapan SMOTE (75,5%). Ini menunjukkan bahwa Naïve Bayes mungkin kurang efisien dalam menangani ketidakseimbangan kelas yang kompleks
4.	Evaluasi Aplikasi Raileo Melalui Analisis Sentimen Ulasan Playstore Dengan Metode Naive Bayes.	Junianto, H., Arsi, P., Kusuma, B. A., & Saputra, D. I. S.	Jurnal ini melaporkan tingkat akurasi analisis sentimen menggunakan metode Naive Bayes mencapai 94%, yang menandakan efektivitas dalam pengklasifikasian ulasan pengguna	Meskipun akurasi tinggi secara keseluruhan, terdapat keberatan dalam mengklasifikasikan beberapa ulasan negatif dengan benar, yang menyebabkan false positives dalam analisis
5.	Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media Sosial Twitter Menggunakan Algoritma Support	Oktavia, D., Ramadahan, Y. R., & Minarto, M.	Sistem e-Tilang diharapkan dapat mengurangi praktik pungutan liar yang sering terjadi dalam penegakan hukum lalu lintas di Indonesia, dengan	Penelitian menunjukkan bahwa sentimen masyarakat terhadap e-Tilang cenderung netral, yang mungkin menunjukkan

<b>No</b>	<b>Judul</b>	<b>Penulis</b>	<b>Kelebihan</b>	<b>Kekurangan</b>
	Vector Machine		memberikan sanksi yang lebih transparan dan akuntabel	bahwa masih ada kekurangan dalam menyampaikan informasi atau implementasi sistem tersebut

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1. Pendekatan Penelitian**

Penelitian Studi ini menerapkan metode komparatif dengan pendekatan kuantitatif guna melihat perbandingan kinerja antara algoritma *Naïve Bayes* dan *Support Vector Machine* (SVM) pada klasifikasi sentimen ulasan Discord di Google Play Store. Alasan penggunaan pendekatan kuantitatif adalah karena fokus riset bertumpu pada pemrosesan angka serta pengujian kinerja model lewat parameter ukur yang spesifik. Di sisi lain, komparasi ditujukan untuk mengevaluasi selisih luaran prediksi dari kedua metode yang diuji.

Alur pengujian mencakup serangkaian proses, berawal dari penarikan data, pra-pemrosesan, pemisahan dataset untuk latih dan uji, pembuatan model, sampai pada tahap pengukuran metrik (akurasi, *recall*, presisi, serta *F1-score*) yang berbasis pada tabel *confusion matrix*.

#### **3.2. Jenis dan Sumber data**

##### **3.2.1. Jenis Data**

Teks opini pengguna aplikasi Discord dari platform Google Play Store difungsikan sebagai bahan utama riset ini. Pada tahap awal, data tersebut memiliki karakteristik tidak terstruktur (*unstructured*), yang nantinya akan dikonversi menjadi susunan numerik terstruktur agar dapat dipahami dan diolah oleh algoritma *machine learning*.

##### **3.2.2. Sumber Data**

Beberapa data observasi diekstraksi secara langsung dari halaman antarmuka aplikasi Discord menggunakan bantuan skrip *web scraping*. Proses pengumpulan data dibatasi hanya pada opini yang ditulis menggunakan bahasa Indonesia atau kemungkinan penggunaan kata slank. Label sentimen kemudian diekstrak sebagai sentimen bernada positif atau negatif bersandar pada parameter *rating* bintang pengguna atau melalui prosedur pelabelan manual oleh peneliti.

### 3.2.3. Pengumpulan Data

Ratusan hingga ribuan baris ulasan ditarik secara massal menggunakan perangkat otomatisasi (*scraping tool*) guna menjamin efisiensi durasi dan kecepatan akuisisi. Kumpulan teks kotor tersebut kemudian disalurkan menuju fase sterilisasi (*preprocessing*) agar siap disuapkan ke dalam agenda pelatihan model prediktif.

### 3.2.4. Pembagian Data

Dataset dari ulasan pengguna yang telah disterilisasikan, selanjutnya dipecah menjadi dua kompartemen dengan rasio yang proporsional. Sebanyak 80% dari total data yang ada akan dialokasikan sebagai data latih (*training set*) untuk menyusun kerangka model, sementara sisa 20% didedikasikan sepenuhnya sebagai data uji (*testing set*), hal tersebut guna memvalidasi kemampuan prediksi algoritma pada data yang belum pernah dijumpai sebelumnya.

## 3.3. Tahapan pemrosesan data

Fase prapemrosesan data (*preprocessing*) bertindak sebagai langkah fundamental untuk mengonversi rentetan teks yang semrawut menjadi format terstruktur yang baku, sehingga tahap analisis sentimen dapat dieksekusi dengan akurasi maksimal. Rincian dari tahapan sterilisasi data ini meliputi:

### 3.3.1. Case Folding

*Case folding* Merupakan proses standarisasi ortografi dengan mereduksi seluruh susunan karakter kapital di dalam dokumen teks menjadi huruf kecil abjad (*lowercase*). Penyeragaman ini vital agar sistem tidak membaca kata yang memiliki makna identik sebagai dua entitas berbeda hanya akibat variasi kapitalisasi.

### 3.3.2. Cleansing

Pembersihan teks dari anomali atau elemen *noise* (gangguan linguistik) yang tidak menyumbang bobot opini. Karakter yang direduksi mencakup rentetan angka, simbol tanda baca, entitas *emoticon*, tautan internet (URL), serta karakter non-alfabetik lainnya.

### 3.3.3. *Tokenizing*

untaian kalimat akan di buat ulang atau di modifikasi menjadi satuan-satuan unit kata tunggal (atau yang lazim disebut token). Pemecahan entitas kata ini sangat krusial agar model komputasi dapat membedah bobot linguistik pada level leksem.

### 3.3.4. *Stopword Removal*

Eksekusi pemangkasan pada kelas kata hubung atau kata imbuhan yang dinilai sangat umum (misalnya: "yang", "dan", "di", "dll"). Kata-kata ini dieliminasi karena meskipun frekuensinya tinggi, mereka tidak membawa arti kata sentimen yang esensial.

### 3.3.5. *Stemming*

*Stemming* bertujuan mengembalikan tiap-tiap token ke bentuk morfologis dasarnya. Sebagai gambaran, awalan bentuk kata seperti "membeli" atau "pembelian" akan secara otomatis disederhanakan kembali menjadi kata dasar "beli".

## 3.4. Penerapan metode

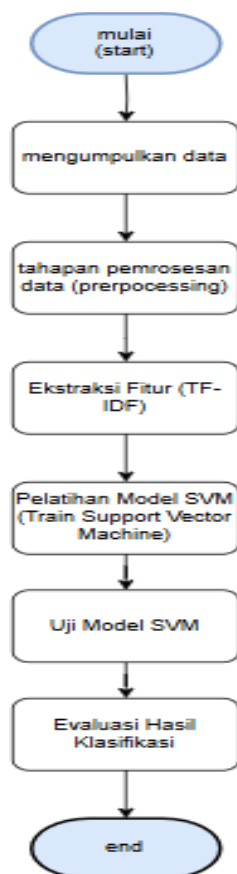
### 3.4.1 *Metode support Vector machine*

*Support Vector Machine* (SVM) adalah Algoritma *Support Vector Machine* (SVM) difungsikan sebagai salah satu pendekatan *supervised learning* untuk mengklasifikasikan kelas sentimen ulasan secara tegak lurus. Mekanisme operasinya berporos pada strategi pencarian bidang pemisah ruang keputusan (*hyperplane*) paling optimal, yang mampu membelah distribusi data kelas positif dan kelas negatif dengan bentangan margin paling maksimum. SVM secara luas diakui sangat tangguh untuk meringkus *dataset* yang memiliki parameter dimensi raksasa (seperti koleksi teks ulasan pengguna) setelah direpresentasikan ke dalam bobot numerik melalui fungsi *TF-IDF*.

Proses pengumpulan data dilakukan melalui pendekatan studi literatur, yang mencakup penelusuran dan analisis sumber-sumber ilmiah terkait Sentiment Analysis, pemanfaatan Twitter API, teknik crawling data, proses prapemrosesan,

penerapan algoritma SVM dengan kernel linear, serta evaluasi model menggunakan K-Fold Cross Validation (Winarto et al., 2024).

Pada obesrvasi ini, rancangan model untuk mengklasifikasikan sentimen dari ulasan pengguna aplikasi Discord ke dalam dua kelas, yaitu positif dan negatif.



Gambar 3.1 Proses Model SVM

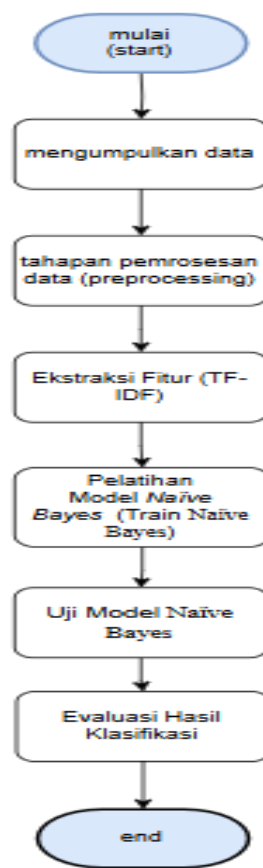
Pada gambar 3.1 memiliki beberapa proses, yaitu:

1. Mulai,
  - Proses dimulai.
2. Pengumpulan Data Ulasan,
  - Tahapan pertama adalah mengumpulkan data berupa ulasan pengguna dari aplikasi google playstore (Discord).
3. Tahapan pemrosesan (preprocessing),
  - Case Folding: Mengubah seluruh huruf menjadi huruf kecil.

- Tokenizing: Memecah kalimat menjadi potongan kata.
  - Stopword Removal: Menghapus kata-kata umum yang tidak memiliki makna penting (seperti "yang", "dan", "di", dll.).
  - Cleansing : Cleansing adalah proses pembersihan teks dari karakter atau simbol yang tidak dibutuhkan
  - Stemming: Mengubah kata ke bentuk dasarnya, misalnya “bermain” menjadi “main”.
4. Pelabelan Data Secara Manual,
    - Positif
    - Negatif
  5. Pelatihan Model,
    - Pelatihan Model SVM: Menggunakan data latih untuk membuat model klasifikasi berdasarkan margin optimal antar kelas.
    - Pelatihan Model Naïve Bayes: Menggunakan probabilitas dan teori Bayes untuk memodelkan distribusi kata dalam masing-masing kelas sentimen.
  6. Evaluasi Model,
    - Akurasi
    - Presisi
    - Recall
    - F-1 model

### 3.4.2. Metode *Naïve Bayes*

Selain SVM, algoritma klasifikasi *Naïve Bayes* turut diimplementasikan berkat rekam jejak efisiensinya dalam memproses dataset *text mining* berdimensi fitur yang padat. Dengan berlandaskan formulasi probabilitas dari Teorema Bayes serta premis hipotesis bahwa tiap-tiap atribut (token kata) berdiri secara independen, sistem akan mengkalkulasi probabilitas kondisional suatu ulasan. Meskipun bersandar pada asumsi naif yang mengabaikan korelasi leksikal antar kata, metode ini lazim mendemonstrasikan hasil klasifikasi yang sangat presisi meskipun suplai data pelatihan (*training data*) yang tersedia bersifat minim atau terbatas pada pelatihan dan pengujian model (Harajoh et al., 2024).



Gambar 3.2 Proses Model Naive Bayes

Pada gambar 3.2 memiliki beberapa proses, yaitu:

1. Mulai,
  - Proses dimulai.
2. Pengumpulan Data Ulasan,
  - Tahapan pertama adalah mengumpulkan data berupa ulasan pengguna dari aplikasi google playstore (Discord).
3. Tahapan pemrosesan (preprocessing),
  - Case Folding: Mengubah seluruh huruf menjadi huruf kecil.
  - Tokenizing: Memecah kalimat menjadi potongan kata.
  - Stopword Removal: Menghapus kata-kata umum yang tidak memiliki makna penting (seperti "yang", "dan", "di", dll.).
  - Cleansing : Cleansing adalah proses pembersihan teks dari karakter atau simbol yang tidak dibutuhkan

- Stemming: Mengubah kata ke bentuk dasarnya, misalnya “bermain” menjadi “main”.
4. Pelabelan Data Secara Manual,
    - Positif
    - Negatif
  5. Pelatihan Model,
    - Pelatihan Model SVM: Menggunakan data latih untuk membuat model klasifikasi berdasarkan margin optimal antar kelas.
    - Pelatihan Model Naïve Bayes: Menggunakan probabilitas dan teori Bayes untuk memodelkan distribusi kata dalam masing-masing kelas sentimen.
  6. Evaluasi Model,
    - Akurasi
    - Presisi
    - Recall
    - F-1 model

### 3.5. Waktu dan tempat penelitian

#### 3.5.1. waktu penelitian

**Tabel 3.5 Waktu Penelitian**

Kegiatan	Bulan							
	Januari	Februari	Maret	April	Mei	Jun	Juli	Agustus
Pengajuan judul								
Observasi								
Pengumpulan Data								
Seminar Proposal								

#### 3.5.2. Tempat Penelitian

Keseluruhan aktivitas observasi, pengembangan sistem, hingga penyusunan analisis tidak terbatas pada satu lokasi fisik karena proses pengumpulan data, pemrosesan, serta analisis dilakukan secara digital. Adapun rincian tempat penelitian sebagai berikut:

1. Data yang digunakan dalam penelitian ini berupa ulasan pengguna terhadap aplikasi Discord yang diambil dari platform Google Play Store. Data diambil melalui proses *web scraping*.
2. Penelitian ini menggunakan Google Colaboratory (Google Colab) sebagai media pemrograman dan eksperimen karena menyediakan dukungan komputasi awan (cloud-based).
3. Waktu dan Durasi Penelitian dilakukan selama rentang waktu 15 juni sampai 11 juli

## BAB IV HASIL DAN PEMBAHASAN

### 4.1. Hasil Pengumpulan Data

Akuisisi data observasi dieksekusi dengan menambang opini pengguna aplikasi Discord secara langsung dari etalase *Google Play Store*. Teknik *web scraping* diimplementasikan dengan memanfaatkan fungsi pustaka *google-play-scraper* di dalam lingkungan pemrograman Python. Korpus yang berhasil dihimpun terdiri atas teks ulasan berdialek Indonesia beserta skor bintang yang mewakili tingkat kepuasan audiens. Secara keseluruhan, terkumpul sebanyak 952 baris ulasan.

Tiap baris data memuat dua atribut krusial:

- *Content* → yang menampung untaian teks opini
- *Score* → ang merepresentasikan nilai *rating* 1 hingga 5

Selanjutnya, pemberian skor pada bintang dikonversi menjadi label sentimen sebagai berikut:

- Pemberian skor 4 dan 5 dianotasikan secara ekuivalen ke dalam kelas sentimen positif
- skor 1 dan 2 dipetakan sebagai sentimen negatif
- Nilai tengah berupa skor 3 dialokasikan sebagai kelas netral

Mengingat model eksperimen ini hanya dibangun secara eksklusif pada klasifikasi biner, himpunan ulasan dengan label netral sengaja tidak diikutsertakan dari arena pemodelan, agar algoritma pembelajaran mesin dapat terfokus sepenuhnya pada penandaan antara kelas positif dan negatif secara ekstrem.

Berikut cara memproses dari data untuk mengkalsifikasikan sentimen positif atau negatif dengan menggunakan web scraping untuk mengumpulkan data, dapat dilihat pada **gambar 4.1**, berikut:

```

▶ from google_play_scraper import reviews, Sort
import pandas as pd

# Ambil 951 ulasan terbaru aplikasi Discord
result, _ = reviews(
    'com.discord',
    lang='id',
    country='id',
    sort=Sort.NEWEST,
    count=1000
)

df = pd.DataFrame(result)[['content', 'score']]

# Buat label sentimen
def label_sentiment(score):
    if score >= 4:
        return "positif"
    elif score <= 2:
        return "negatif"

df['label'] = df['score'].apply(label_sentiment)

# Buang data netral (hanya positif & negatif)
df = df[df['label'] != 'netral']

print("Jumlah data:", len(df))
df.head()

```

**Gambar 4.1** Proses *Web Scraping*

Sementara hasil dari data web scraping (*raw data*) yang telah dikumpulkan dalam penelitian diperlihatkan pada **gambar 4.2**, berikut:

↔ Jumlah data: 952

	content	score	label
0	aplikasi apan ini mau login aja susah bener pa...	1	negatif
1	apk gaje gw sengaja kasih bintang 5 biar dibac...	5	positif
2	good	5	positif
3	this really help	5	positif
4	aplikasi ini sangat bagus dan hebat saya bisa ...	5	positif

**Gambar 4.2** Hasil *Web Scraping*

## 4.2. Tahapan Preprocessing Data

Setelah melalui tahap pengambilan data, selanjutnya teks mentah ini disalurkan ke dalam fase sterilisasi atau *preprocessing*. Tujuan utama dari rangkaian pembersihan ini adalah mereduksi anomali (*noise*) dan merekonstruksi teks kotor menjadi bentuk yang baku, sehingga komputasi klasifikasi dapat berjalan dengan tingkat akurasi optimal. Rute sterilisasi data ini melingkupi serangkaian aksi berurutan:

Tahap preprocessing yang diterapkan dalam penelitian ini terdiri atas beberapa proses berikut:

### 1. Case Folding

Proses penyeragaman ortografi untuk memastikan seluruh rangkaian karakter diformat menjadi huruf kecil (*lowercase*).

Contoh:

“Aplikasi Ini Sangat Bagus” → “aplikasi ini sangat bagus”

### 2. Cleansing

Tahapan eliminasi terhadap seluruh simbol yang miskin relevansi semantik, seperti angka, tautan situs web (URL), emotikon, hingga ragam tanda baca.

Contoh:

“Discord mantap!! 🤩” → “discord mantap”

### 3. Tokenizing

Operasi pembedahan klausa atau kalimat utuh menjadi kepingan-kepingan kata penyusun (*token*).

Contoh:

“discord mantap sekali” → [“discord”, “mantap”, “sekali”]

### 4. Stopword Removal

Eliminasi terhadap kelas kata tugas atau partikel yang frekuensi kemunculannya terlampau sering namun tidak menyumbang konotasi sentimen yang esensial, seperti “yang”, “dan”, atau “di”.

Contoh: “discord yang sangat bagus” menjadi “discord bagus”

## 5. Stemming

SMekanisme pemotongan afiksasi atau imbuhan untuk mengembalikan morfem turunan ke wujud kata asalnya, yang dioperasikan dengan memanggil pustaka Sastrawi.

Contoh: “*bermain*”, “*bermainlah*”, “*bermainan*” menjadi “*main*”

Berikut Proses Demonstrasi skrip Python dalam mengeksekusi tahapan prapemrosesan didokumentasikan pada **Gambar 4.3**.

```
import re
import nltk
from nltk.tokenize import word_tokenize
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFacto

nltk.download('punkt')
nltk.download('punkt_tab') # Download the missing resource

# Case folding + cleansing
def clean_text(text):
    text = str(text).lower()
    text = re.sub(r"http\S+|www\S+|https\S+", "", text) # hapus URL
    text = re.sub(r'^a-z\s', '', text) # hapus angka, simbol, tanda baca
    text = re.sub(r'\s+', ' ', text).strip()
    return text

df['cleaned'] = df['content'].apply(clean_text)

# Tokenizing
df['tokenized'] = df['cleaned'].apply(lambda x: word_tokenize(x))

# Stopword removal
stop_factory = StopWordRemoverFactory()
stopword = stop_factory.create_stop_word_remover()

df['no_stopwords'] = df['cleaned'].apply(lambda x: stopword.remove(x).split())

# Stemming
stemmer = StemmerFactory().create_stemmer()
df['stemmed'] = df['no_stopwords'].apply(lambda tokens: [stemmer.stem(word) for word in tokens])

# Gabungkan kembali setelah stemming
df['final_text'] = df['stemmed'].apply(lambda x: " ".join(x))

df[['content', 'cleaned', 'tokenized', 'no_stopwords', 'stemmed', 'label']].head()
```

**Gambar 4.3** Proses *Preprocessing data*

Berikut hasil akhir dari preprocessing dapat dilihat pada **Gambar 4.4**, dimana teks ulasan yang sebelumnya masih mengandung banyak noise telah berubah menjadi teks yang lebih bersih, sederhana, dan terstruktur.

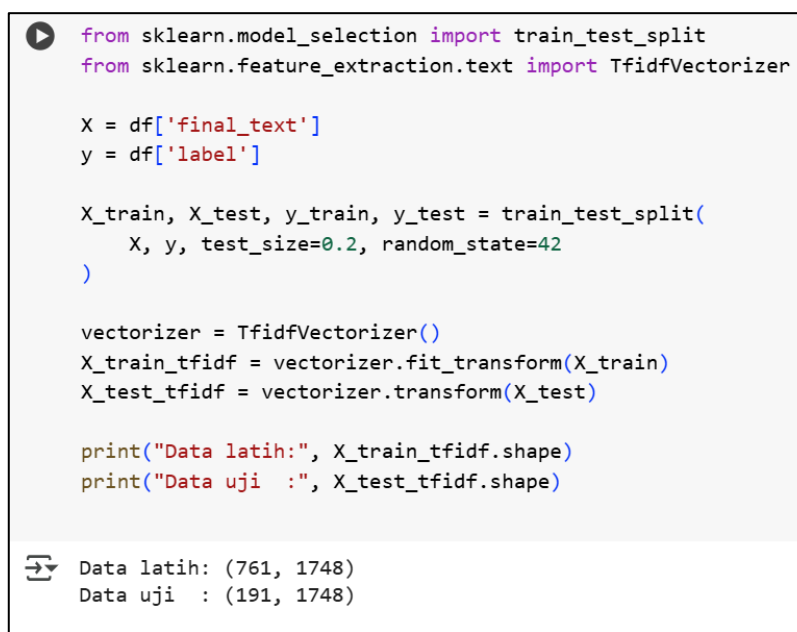
	content	cleaned	tokenized	no_stopwords	stemmed	label
0	aplikasi apan ini mau login aja susah bener pa...	aplikasi apan ini mau login aja susah bener pa...	[aplikasi, apan, ini, mau, login, aja, susah, ...	[aplikasi, apan, mau, login, aja, susah, bener...	[aplikasi, apan, mau, login, aja, susah, bener...	negatif
1	apk gaje gw sengaja kasih bintang 5 biar dibac...	apk gaje gw sengaja kasih bintang biar dibaca ...	[apk, gaje, gw, sengaja, kasih, bintang, biar,...	[apk, gaje, gw, sengaja, kasih, bintang, biar,...	[apk, gaje, gw, sengaja, kasih, bintang, biar,...	positif
2	good	good	[good]	[good]	[good]	positif
3	this really help	this really help	[this, really, help]	[this, really, help]	[this, really, help]	positif
4	aplikasi ini sangat bagus dan hebat saya bisa ...	aplikasi ini sangat bagus dan hebat saya bisa ...	[aplikasi, ini, sangat, bagus, dan, hebat, say...	[aplikasi, sangat, bagus, hebat, bisa, main, g...	[aplikasi, sangat, bagus, hebat, bisa, main, g...	positif

**Gambar 4.4 Hasil dari *Preprocessing data***

### 4.3. Transformasi Data dengan TF-IDF

Setelah preprocessing kumpulan korpus yang telah disterilisasi dari tahapan sebelumnya, sebanyak 951 ulasan kemudian diekstraksi bobot linguistiknya menggunakan rumusan term *TF-IDF* (*Term Frequency-Inverse Document Frequency*). Pemetaan matriks pembobotan kata inilah yang nantinya diumpankan sebagai *input* kepada mesin klasifikasi *Naïve Bayes* dan *Support Vector Machine* (SVM).

Guna memastikan validasi model berjalan tanpa bias, populasi dataset dipartisi menjadi dua kompartemen dengan porsi: 80% didedikasikan sebagai data latih (*training data*) untuk membangun arsitektur sistem, dan 20% sisanya diisolasi sebagai data uji (*testing data*). (Proses instansiasi *TF-IDF Vectorizer* dapat dilihat pada **Gambar 4.5**).



```

▶ from sklearn.model_selection import train_test_split
  from sklearn.feature_extraction.text import TfidfVectorizer

X = df['final_text']
y = df['label']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

print("Data latih:", X_train_tfidf.shape)
print("Data uji :", X_test_tfidf.shape)

↵ Data latih: (761, 1748)
   Data uji  : (191, 1748)

```

**Gambar 4.5** Pembobotan TF-IDF

#### 4.3.1. Training Model Naïve Bayes

langkah Selanjutnya adalah pelatihan model menggunakan algoritma Berlandaskan bobot kata matang dari hasil *TF-IDF*, pengklasifikasi *Naïve Bayes* dilatih memanfaatkan pendekatan teori probabilitas statistika (*Bayes Theorem*).

Hasil evaluasi pelatihan pada **Gambar 4.6** menunjukkan bahwa model ini mampu menorehkan tingkat penemuan (*recall*) yang nyaris sempurna untuk mendeteksi sentimen positif.

```
[19] from sklearn.naive_bayes import MultinomialNB
      from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report

      nb = MultinomialNB()
      nb.fit(X_train_tfidf, y_train)
      y_pred_nb = nb.predict(X_test_tfidf)

      print("=== Naive Bayes ===")
      print("Akurasi :", accuracy_score(y_test, y_pred_nb))
      print("Presisi :", precision_score(y_test, y_pred_nb, pos_label="positif"))
      print("Recall  :", recall_score(y_test, y_pred_nb, pos_label="positif"))
      print("F1-Score:", f1_score(y_test, y_pred_nb, pos_label="positif"))

      ==> === Naive Bayes ===
      Akurasi : 0.837696335078534
      Presisi : 1.0
      Recall  : 0.581081081081081
      F1-Score: 0.7350427350427351
```

**Gambar 4.6 Hasil Training Model Naïve Bayes**

Namun, kesederhanaan asumsi dalam algoritma ini masih rentan menghasilkan bias saat membedah sejumlah teks ulasan berkonotasi negatif.

### 4.3.2. Training Model SVM

Pelatihan Dalam skenario komparatif, implementasi *training* menggunakan algoritma *Support Vector Machine* (SVM) mempertontonkan lonjakan kualitas performa yang jauh lebih stabil di berbagai metrik.

Parameter akurasi, presisi, hingga nilai *F1-Score* mengonfirmasi bahwa penarikan bidang keputusan linier (margin) oleh SVM beroperasi sangat handal untuk menajamkan segregasi antara opini sentimen Discord. (Log pelatihan klasifikasi SVM terlampir pada **Gambar 4.7**).

```
from sklearn.svm import SVC

svm = SVC(kernel='linear')
svm.fit(X_train_tfidf, y_train)
y_pred_svm = svm.predict(X_test_tfidf)

print("=== SVM ===")
print("Akurasi :", accuracy_score(y_test, y_pred_svm))
print("Presisi :", precision_score(y_test, y_pred_svm, pos_label="positif"))
print("Recall  :", recall_score(y_test, y_pred_svm, pos_label="positif"))
print("F1-Score:", f1_score(y_test, y_pred_svm, pos_label="positif"))

==> === SVM ===
Akurasi : 0.8795811518324608
Presisi : 0.9180327868852459
Recall  : 0.7567567567567568
F1-Score: 0.8296296296296296
```

**Gambar 4.7 Hasil Training Model SVM**

#### 4.4. Hasil Klasifikasi

Hasil pengujian Memasuki fase evaluasi, matriks uji dibedah secara komprehensif menggunakan kerangka kerja tabulasi *Confusion Matrix*. Instrumen diagnostik ini beroperasi memetakan silang sebaran kesalahan tipe I dan tipe II, serta kalkulasi tebakan akurat dari kedua klasifikator, yang terpecah ke dalam empat spektrum komponen sentral:

Pada penelitian ini, Confusion Matrix terdiri dari empat komponen utama, yaitu:

- **True Positive (TP):** Frekuensi opini yang secara faktual positif dan berhasil diklasifikasikan secara benar oleh algoritma sebagai sentimen positif.
- **True Negative (TN):** Himpunan keluhan atau opini berskala negatif yang juga sukses dideteksi tepat sasaran sebagai sentimen negatif.
- **False Positive (FP):** Bias fatal (Type 1) ketika mesin keliru memprediksi dan melabeli data sentimen negatif layaknya ulasan positif.
- **False Negative (FN):** Kegagalan tebakan (Tipe 2) di mana algoritma secara keliru melempar sentimen aktual yang aslinya positif masuk ke kategori negatif.

Input confusion matrix pada proses klasifikasi dapat dilihat pada **Gambar 4.8**, yang menunjukkan data hasil prediksi dari model terhadap data uji.

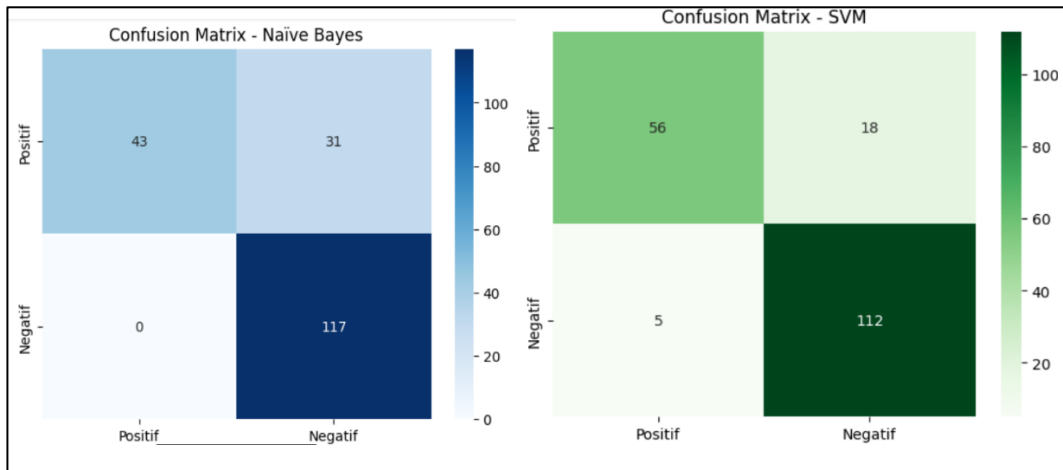
```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

# Naïve Bayes
cm_nb = confusion_matrix(y_test, y_pred_nb, labels=["positif", "negatif"])
sns.heatmap(cm_nb, annot=True, fmt="d", cmap="Blues",
            xticklabels=["Positif", "Negatif"], yticklabels=["Positif", "Negatif"])
plt.title("Confusion Matrix - Naïve Bayes")
plt.show()

# SVM
cm_svm = confusion_matrix(y_test, y_pred_svm, labels=["positif", "negatif"])
sns.heatmap(cm_svm, annot=True, fmt="d", cmap="Greens",
            xticklabels=["Positif", "Negatif"], yticklabels=["Positif", "Negatif"])
plt.title("Confusion Matrix - SVM")
plt.show()
```

### Gambar 4.8 Input Confusion Matrix

Sedangkan hasil output confusion matrix dari kedua algoritma dapat dilihat pada **Gambar 4.9**, yang menunjukkan distribusi hasil klasifikasi Naïve Bayes dan SVM.



**Gambar 4.9 Output Confusion Matrix**

hasil Confusion Matrix antara perbandingan algoritma Naïve Bayes dan SVM., berikut adalah interpretasinya:

#### 1. Naïve Bayes

- True Positif (TP): 43
- True Negatif (TN): 117
- False Positif (FP): 31
- False Negatif (FN): 0

#### 2. SVM

- True Positif (TP): 56
- True Negatif (TN): 112
- False Positif (FP): 18
- False Negatif (FN): 5

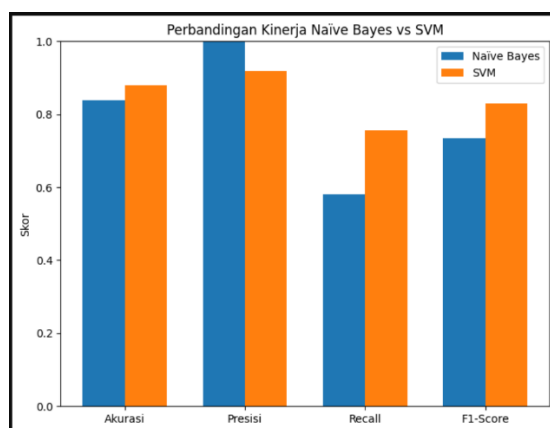
#### 3. Perbandingan

- True Positif: SVM mengklasifikasikan lebih banyak ulasan positif dengan benar (56) dibandingkan Naive Bayes (43).
- True Negatif: Naive Bayes mengklasifikasikan sedikit lebih banyak ulasan negatif dengan benar (117) dibandingkan SVM (112).

- False Positif: SVM memiliki jumlah False Positif yang jauh lebih rendah (18) dibandingkan Naive Bayes (31). Ini berarti SVM lebih baik dalam menghindari salah mengklasifikasikan ulasan negatif sebagai positif.
- False Negatif: Naive Bayes memiliki 0 False Negatif, artinya tidak ada ulasan positif yang salah diklasifikasikan sebagai negatif. SVM memiliki 5 False Negatif.

Secara keseluruhan, SVM menghasilkan output yang lebih baik dalam mengidentifikasi ulasan positif (TP lebih tinggi) dan lebih sedikit melakukan kesalahan False Positif. Naive Bayes unggul dalam mengidentifikasi ulasan negatif (TN sedikit lebih tinggi) dan tidak memiliki False Negatif. Pilihan model terbaik akan tergantung pada prioritas apakah lebih penting untuk meminimalkan False Positif atau False Negatif dalam kasus penggunaan spesifik ini.

Proses perbandingan kinerja model dapat dilihat pada **Gambar 4.10**, yang menampilkan tabel output grafik perbandingan performa antara kedua algoritma berdasarkan keempat metrik evaluasi



**Gambar 4.10 Perbandingan Kinerja Naive Bayes vs SVM**

#### 4.5. Word Cloud Analisis

Sebagai Guna memperkaya wawasan diagnostik, dikonstruksi visualisasi *Word Cloud* sebagai peta agregasi kemunculan diksi. Kerapatan frekuensi kata kunci tertentu memberikan abstraksi konseptual tentang isu sentral (kualitas atau





Berbekal rekaman matriks probabilitas (TP=43, TN=117, FP=31, FN=0), algoritma *Naïve Bayes* membukukan tingkat *Akurasi prediksi keseluruhan sebesar 83,77%*. Rasio presisi tertahan di level 58,11%, diimbangi oleh nilai sensitivitas *Recall* absolut yang membentur angka 100%. Rata-rata penyeimbang harmonik (*F1-Score*) yang diperoleh klasifikator probabilitas ini bermuara pada angka **73,52%**. Berikut caranya:

- True Positive (TP) = 43
- True Negative (TN) = 117
- False Positive (FP) = 31
- False Negative (FN) = 0

Perhitungan metrik evaluasi:

- **Akurasi** =  $\frac{TP+TN}{TP+TN+FP+FN} = \frac{43+117}{43+117+31+0} = \frac{160}{191} = 83,77\%$
- **Presisi** =  $\frac{TP}{TP+FP} = \frac{43}{43+31} = \frac{43}{74} = 58,11\%$
- **Recall** =  $\frac{TP}{TP+FN} = \frac{43}{43+0} = \frac{43}{43} = 100\%$
- **F1-Score** =  $2 \times \frac{Precision \cdot Recall}{Precision+Recall} = 2 \times \frac{0.5811 \times 1.0}{0.5811 + 1.0} = 73,52\%$

#### 4.6.2. SVM

SVM menghasilkan distribusi prediksi (TP=56, TN=112, FP=18, FN=5). Kombinasi parameter margin optimal mengerek rasio *Akurasi SVM menuju 87,96%*. Penekanan drastis pada kasus *False Positive* sukses membangkitkan tingkat presisinya menjadi 75,68%, didampingi sensitivitas *Recall* sekuat 91,80%. Agregat perhitungan ini membukukan performa keseimbangan sentral *F1-Score* pada takaran **82,97%**.

- True Positive (TP) = 56
- True Negative (TN) = 112
- False Positive (FP) = 18
- False Negative (FN) = 5

Perhitungan metrik evaluasi:

- **Akurasi** =  $\frac{TP+TN}{TP+TN+FP+FN} = \frac{56+112}{56+112+18+5} = \frac{168}{191} = 87,96\%$
- **Presisi** =  $\frac{TP}{TP+FP} = \frac{56}{56+18} = \frac{56}{74} = 75,68\%$
- **Recall** =  $\frac{TP}{TP+FN} = \frac{56}{56+5} = \frac{56}{61} = 91,80\%$
- **F1-Score** =  $2 \times \frac{Precision \cdot Recall}{Precision+Recall} = 2 \times \frac{0.7568 \times 0.9180}{0.7568 + 0.9180} = 82,97\%$

#### 4.7. Visualisasi Hasil Perbandingan

Jumlah dataset menyingkap adanya ketimpangan keseimbangan distribusi sentimen. Korpus 952 baris data observasi rupanya didominasi lebat oleh 586 ulasan afirmatif negatif (berkisar pada 61,55%), sedangkan ulasan berdialek positif terbatas pada populasi 366 entitas (38,45%). Rasio ketidakseimbangan yang diilustrasikan pada **Gambar 4.14** ini menginisiasi adanya potensi bias performa klasifikasi saat model dieksekusi.

```

sentiment_counts = df['label'].value_counts()
total_reviews = len(df)

print("Jumlah Ulasan:")
print(sentiment_counts)
print("\nPresentase Ulasan:")
print((sentiment_counts / total_reviews) * 100)

```

---

```

Jumlah Ulasan:
label
negatif    586
positif    366
Name: count, dtype: int64

Presentase Ulasan:
label
negatif    61.554622
positif    38.445378
Name: count, dtype: float64

```

**Gambar 4.14** Distribusi Sentimen Ulasan Aplikasi Discord

Berdasarkan komparasi lintas parameter evaluasi (**Tabel 4.1** dan ringkasan grafis pada **Gambar 4.10**), didapati konklusi diagnostik bahwa:

**Tabel 4.1.** Ringkasan Metrik Evaluasi Naïve Bayes dan SVM

Algoritma	Akurasi	Presisi	Recall	F1-Score
Naïve Bayes	83,77%	58,11%	100%	73,52%
SVM	87,96%	75,68%	91,80%	82,97%

Hasil Analisis:

- Naïve Bayes unggul pada Recall (100%), namun lemah pada presisi sehingga menghasilkan banyak false positive.
- SVM lebih unggul secara keseluruhan karena memiliki metrik yang seimbang, dengan F1-Score lebih tinggi (82,97%).

## BAB V PENUTUP

### 5.1. Kesimpulan

Berdasarkan hasil analisis komputasional dan serangkaian uji evaluasi terkait sentimen pengguna aplikasi Discord di ekosistem Google Play Store, penelitian ini mendeduksi beberapa konklusi utama sebagai berikut:

1. Total ulasan yang berhasil diakuisisi mencapai 952 entitas ulasan, di mana kelas sentimen negatif mendominasi dengan 586 data (61,55%) berbanding 366 data bersentimen positif (38,45%). Opini dengan muatan netral (skor 3 bintang) sengaja dieksklusi guna mematuhi parameter rancangan klasifikasi biner yang menjadi fondasi riset ini.
2. Tahapan sterilisasi data (*preprocessing*) telah dieksekusi secara berurutan melalui lima tahap esensial, yakni penyeragaman karakter (*case folding*), reduksi derau linguistik (*cleansing*), segmentasi frasa (*tokenizing*), pemangkasan partikel kata umum (*stopword removal*), dan penarikan bentuk kata dasar (*stemming*).
3. Rekonstruksi teks ini selanjutnya dikonversi menjadi representasi numerik melalui fungsi pembobotan *TF-IDF* untuk memfasilitasi algoritma klasifikasi. Berdasarkan hasil pengujian dengan Confusion Matrix dan perhitungan metrik evaluasi, diperoleh hasil:
  - Naïve Bayes menghasilkan tingkat Akurasi 83,77%, tingkat Presisi 58,11%, daya *Recall* 100%, serta rerata harmonik *F1-Score* sebesar 73,52%
  - Di sisi lain, *Support Vector Machine* (SVM) memvalidasi keunggulannya dengan mencetak Akurasi 87,96%, rasio Presisi 75,68%, sensitivitas *Recall* 91,80%, yang diimbangi dengan perolehan *F1-Score* 82,97%.
4. Algoritma Naïve Bayes unggul pada nilai recall (100%) sehingga lebih baik dalam mendeteksi ulasan positif. Namun, presisinya rendah (58,11%) karena menghasilkan banyak false positive.

5. *hyperplane* dari model SVM mempresentasikan keseimbangan diagnostik yang jauh lebih tangguh di seluruh lini pengujian. Puncak parameter akurasi dan harmonisasi metrik *F1-Score* secara empiris membuktikan bahwa arsitektur klasifikasi *Support Vector Machine* mutlak mengungguli efisiensi *Naïve Bayes* untuk pemecahan studi kasus analisis opini pada platform Discord.

## 5.2. Saran

Berdasarkan hasil penelitian pada ulasan Discord, penulis memberikan beberapa saran:

1. Untuk Penelitian Selanjutnya:
  - Gunakan dataset dengan jumlah ulasan lebih banyak agar hasil lebih representatif.
  - Sertakan juga ulasan netral sehingga klasifikasi menjadi tiga kelas (positif, negatif, netral), sehingga hasil lebih mendekati kondisi nyata.
  - Coba algoritma lain seperti Random Forest, Logistic Regression, K-Nearest Neighbor, atau Deep Learning (LSTM, CNN, Transformer) agar bisa dibandingkan kinerjanya dengan SVM.
2. Untuk Pengembang Discord:
  - Hasil analisis sentimen dapat dijadikan alat monitoring opini pengguna secara otomatis.
  - visualisasi melalui *Word Cloud* mengisyaratkan bahwa kata kunci seperti "bagus", "ringan", "mantap", dan "jernih" menjadi pilar keunggulan fitur yang harus dipertahankan. Sebaliknya, akumulasi keluhan lekat pada terminologi seperti "error", "lemot", dan kesulitan "login". Pemetaan ini mengindikasikan urgensi mendesak bagi pihak pengembang untuk mengalokasikan prioritas perbaikan pada sektor pemeliharaan peladen (stabilitas jaringan) dan akselerasi operasional peranti lunak.

## DAFTAR PUSTAKA

- Andarsyah, R., & Yanuar, A. (2024). Sentimen Analisis Aplikasi Posaja Pada Google Playstore Untuk Peningkatan Pospay Superapp Menggunakan Support Vector Meachine. *Jurnal Teknik Informatika*, 16(2), 1-7.
- Discord Inc. (2023). *What is Discord?* Retrieved from: <https://discord.com>
- Eskiyaturrofikoh, E., & Suryono, R. R. (2024). Analisis Sentimen Aplikasi X Pada Google Play Store Menggunakan Algoritma Naïve Bayes Dan Support Vector Machine (SVM). *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 9(3), 1408-1419.
- Oktavia, D., Ramadahan, Y. R., & Minarto, M. (2023). Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (SVM). *KLIK: Kajian Ilmiah Informatika dan Komputer*, 4(1), 407-417.
- Fareza, I. D., & Handayani, E. T. E. (2025). *Analisis Sentimen Kualitas Aplikasi Discord Menggunakan Algoritma Naïve Bayes dan Support Vector Machine*. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(3), 4564–4571.
- Hajarah, H., Suprpti, T., & Narasati, R. (2024). Implementasi Algoritma Naive Bayes Untuk Analisis Sentimen Ulasan Produk Makanan Dan Minuman Di Tokopedia. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(1), 111-118.
- Junianto, H., Arsi, P., Kusuma, B. A., & Saputra, D. I. S. (2024). Evaluasi Aplikasi Raileo Melalui Analisis Sentimen Ulasan Playstore Dengan Metode Naive Bayes. *SINTECH (Science and Information Technology) Journal*, 7(1), 27-40.
- Kurniawan, B., Mahendra, R., & Hidayanto, A. N. (2020). *Exploring User Reviews for Mobile Application Enhancement: A Text Mining Approach*. *Jurnal Teknologi dan Sistem Komputer*, 8(3), 165–174.
- Lutz, M. (2013). *Learning Python* (5th ed.). Sebastopol: O'Reilly Media.


- Nurhidayat, R., & Dewi, K. E. (2023). Penerapan Algoritma K-Nearest Neighbor Dan Fitur Ekstraksi N-Gram Dalam Analisis Sentimen Berbasis Aspek. *Komputa: Jurnal Ilmiah Komputer dan Informatika*, 12(1), 91-100.
- Pratama, R. A., & Astuti, E. S. (2021). *Analisis Sentimen pada Ulasan Aplikasi Google Play Store Menggunakan Metode Machine Learning*. *Jurnal Teknologi Informasi dan Komputer*, 7(2), 113–120.
- Salsabila, S. N., Sari, B. N., & Mayasari, R. (2023). Klasifikasi Ulasan Pengguna Aplikasi Discord Menggunakan Metode Information Gain Dan Naïve Bayes Classifier. *INFOTECH journal*, 9(2), 383-392.
- Saputra, D. R., Fitriani, R., & Hidayat, A. (2022). *Pemanfaatan Ulasan Pengguna di Google Play Store untuk Evaluasi Aplikasi Menggunakan Text Mining*. *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, 10(1), 44–52.
- Sari, R. (2020). Analisis sentimen pada review objek wisata dunia fantasi menggunakan algoritma K-Nearest Neighbor (k-nn). *EVOLUSI: Jurnal Sains Dan Manajemen*, 8(1).
- Sulistiyono, A., Nugraha, D., & Herlambang, T. (2021). *Penerapan Python dalam Klasifikasi Sentimen pada Media Sosial*. *Jurnal Teknologi dan Sistem Komputer*, 9(2), 105–112.
- Sugiyono. (2019). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta.
- Tio, D., & Kurniawan, D. (2020). *Pengantar Pemrograman Python untuk Sains Data*. Jakarta: Penerbit Andi.
- Widodo, A. E., Wati, F. F., & Hidayati, N. (2024). Implementasi Algoritma SVM dan Naive Bayes untuk Analisis Sentimen pada Ulasan Pengguna Aplikasi OneDrive. *Indonesian Journal on Software Engineering (IJSE)*, 10(2), 131-138.
- Winarto, W., Musdar, I. A., & Hasniati, H. (2024). Analisis Sentimen Bakal Calon Presiden 2024 Menggunakan Algoritma Support Vector Machine Pada Twitter. *KHARISMA Tech*, 19(1), 86-98.

Zhang, H. (2004). *The Optimality of Naive Bayes*. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference.

## LAMPIRAN

## Lampiran 1

## Surat Penetapan Dosen Pembimbing



MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH

**UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAH-PT/Akred/PT/11/2019  
 Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003  
<http://fku@umsu.ac.id> [fku@umsu.ac.id](mailto:fku@umsu.ac.id) [umsu](https://www.facebook.com/umsu) [umsu](https://www.instagram.com/umsu) [umsu](https://www.youtube.com/umsu) [umsu](https://www.tiktok.com/umsu)

**PENETAPAN DOSEN PEMBIMBING  
 PROPOSAL/SKRIPSI MAHASISWA  
 NOMOR : 659/II.3-AU/UMSU-09/F/2025**

*Assalamu'alaikum Warahmatullahi Wabarakatuh*

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

**Program Studi** : Teknologi Informasi  
**Pada tanggal** : 11 Juni 2025

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

**Nama** : Robby Suriansyah  
**NPM** : 2109020122  
**Semester** : VIII (Delapan)  
**Program studi** : Teknologi Informasi  
**Judul Proposal / Skripsi** : Perbandingan Metode Support Vector Machine, Naïve Bayes, Dan Logistic Regression Untuk Analisis Sentimen Pada Ulasan Aplikasi Discord Di Google Playstore


**Dosen Pembimbing** : Indah Purnama Sari, S.T, M.Kom


Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi dinyatakan " BATAL "** bila tidak selesai sebelum Masa Kadaluarsa tanggal : 11 Juni 2026
4. Revisi judul.....

*Wassalamu'alaikum Warahmatullahi Wabarakatuh.*




Ditetapkan di : Medan  
 Pada Tanggal : 15 Dzulhijjah 1446 H  
 11 Juni 2025 M





Dekan  
**Indah Purnama Sari, S.T, M.Kom.**  
 NIDN : 0127099201

Cc. File

## Lampiran 2

### Cek Plagiarisme Turnitin

5280ab482c71c165\_Turnitin

ORIGINALITY REPORT

25%

SIMILARITY INDEX

PRIMARY SOURCES

1	<a href="https://repository.umsu.ac.id">repository.umsu.ac.id</a> Internet	846 words — 11%
2	<a href="https://fikti.umsu.ac.id">fikti.umsu.ac.id</a> Internet	76 words — 1%
3	<a href="https://jurnal.stkipggritulungagung.ac.id">jurnal.stkipggritulungagung.ac.id</a> Internet	70 words — 1%
4	Evip Athirah Salsabila, Mellysa Pratama, Putri Wahyuni, Vina Purwaningrum, Mufid Faruq Aziz. "Analisis Sentimen Ujaran Kebencian Pada Kolom Komentar Di Instagram", Journal of Integrated Innovation Science, 2025 Crossref	44 words — 1%
5	<a href="https://ojs.stmik-banjarbaru.ac.id">ojs.stmik-banjarbaru.ac.id</a> Internet	44 words — 1%
6	<a href="https://repository.unhas.ac.id">repository.unhas.ac.id</a> Internet	42 words — 1%
7	<a href="https://repository.itpln.ac.id">repository.itpln.ac.id</a> Internet	32 words — < 1%
8	<a href="https://djournals.com">djournals.com</a> Internet	31 words — < 1%
9	<a href="https://kc.umn.ac.id">kc.umn.ac.id</a> Internet	31 words — < 1%