

HALAMAN JUDUL

ANALISIS SENTIMEN ULASAN PENGGUNA TERHADAP APLIKASI SIMANTAP DI GOOGLE PLAY STORE MENGGUNAKAN METODE NAÏVE BAYES

SKRIPSI

Diajukan untuk Memenuhi Salah Satu Persyaratan dalam Menyelesaikan Laporan
Skripsi pada Program Studi Teknologi Informasi Fakultas Sains dan Teknologi
Universitas Ibrahimy



Oleh :

NAUFAL ARIF MAULANA

2021503118

**PROGRAM STUDI TEKNOLOGI INFORMASI FAKULTAS SAINS DAN
TEKNOLOGI UNIVERSITAS IBRAHIMY
SITUBONDO**

2025

HALAMAN PERNYATAAN KEASLIAN TULISAN

Yang bertanda tangan di bawah ini :

Nama : Naufal Arif Maulana

NPM : 2021503118

Prodi : Teknologi Informasi Fakultas : Fakultas Sains dan Teknologi

Dengan ini saya menyatakan bahwa laporan Skripsi dengan judul “ANALISIS SENTIMEN ULASAN PENGGUNA TERHADAP APLIKASI SIMANTAP DI GOOGLE PLAY STORE MENGGUNAKAN METODE NAÏVE BAYES” ini beserta seluruh isinya adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Situbondo, 5 Mei 2025



Naufal Arif Maulana

PERSETUJUAN PEMBIMBING

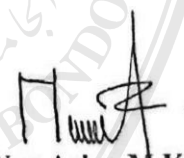
Nama : Naufal Arif Maulana
NPM/NIRM : 2021503118
Judul : Analisis Sentimen Ulasan Pengguna Terhadap
Aplikasi SIMANTAP Di Google Play Store
Menggunakan Metode Naive Bayes.

Telah disetujui oleh:

Pembimbing 1

Pembimbing 2


Firman Santoso, M.Kom
NIDN: 0722129201


Nur Azise, M.Kom
NIDN: 730108802

HALAMAN PENGESAHAN SKRIPSI

**ANALISIS SENTIMEN ULASAN PENGGUNA TERHADAP APLIKASI
SIMANTAP DI GOOGLE PLAY STORE MENGGUNAKAN METODE
NAÏVE BAYES**

NAUFAL ARIF MAULANA

2021503118

Telah dipertahankan di depan dewan penguji Sidang/Munaqasyah Skripsi pada hari Senin, Tanggal 31 Agustus 2025 sebagai salah satu syarat memperoleh gelar sarjana (S.Kom.) pada Fakultas Sains dan Teknologi Universitas Ibrahimiy

Tim Penguji,

Ketua Sidang,

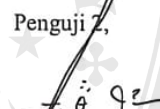
Sekretaris Sidang,


Ahmad Homaidi, M.Kom
NIDN : 0705078901

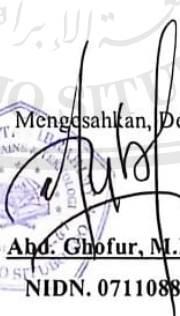

Khoirul Roziqin, S.Kom
NIDN :

Penguji 1,

Zachol Fatah, M.Kom
NIDN : 0715057801

Penguji 2,

Faridun Lazim, M.Tr.T
NIDN : 0711099201

Mengesahkan, Dekan,


Abd. Ghofur, M.Kom.
NIDN. 0711088303

MOTTO

Bismillah



PERSEMBAHAN

Saya persembahkan laporan ini kepada orang – orang yang telah membantu dan berjasa dalam menjalani pembuatan Laporan Skripsi ini :

1. Kepada diri saya sendiri yang sudah tetap hidup apapun masalah didepan
2. Kedua Orang tua yang telah berjuang demi masa depan saya
3. Saudara – saudari saya yang selalu memberikan support pada saya selama ini
4. Sahabat – sahabat yang selalu ada dalam keadaan suka maupun duka
5. Semua orang yang tidak bisa saya sebut satu persatu namanya yang telah mendukung saya sepenuhnya sehingga saya dapat menyelesaikan proposal skripsi ini dengan sebaik baiknya.



KATA PENGANTAR

Dengan memanjatkan puji syukur ke hadirat Allah SWT, atas limpahan rahmat dan hidayah-Nya penulis dapat menyajikan Proposal ini dengan sebaik baiknya, oleh karena itu kami mengucapkan terimakasih kepada :

1. KHR. Ach. Azaim Ibrahimy, S, Sy, M. H. Selaku Pengasuh Pondok Pesantren Salafiyah Syafi'iyah Sukorejo Situbondo.
2. K.H Ach. Fadlail, SH, M.H Selaku Rektor Universitas Ibrahimy.
3. Abd. Ghofur, M. Kom Selaku Dekan Fakultas Sains dan Teknologi Universitas Ibrahimy.
4. Dr. Ach. Khumaidi, M.P Selaku Wakil Dekan 1 Fakultas Sains dan Teknologi Universitas Ibrahimy.
5. Abd. Wafi, M.P Selaku Wakil Dekan 2 Fakultas Sains dan Teknologi Universitas Ibrahimy.
6. Ahmad Lutfi, M. Kom Selaku Wakil Dekan 3 Fakultas Sains dan Teknologi Universitas Ibrahimy.
7. Firman Santoso, M. Kom Selaku Ketua Program Studi Teknologi Informasi Universitas Ibrahimy dan Selaku Dosen Pembimbing 1 Skripsi saya yang telah memberikan arahan dan bimbingan.
8. Nur Azise, M. Kom Selaku Dosen Pembimbing 2 Skripsi saya yang telah memberikan arahan dan bimbingan.
9. Seluruh Dosen Fakultas Sains dan Teknologi yang telah memberikan kami ilmu sehingga sampai pada masa pengerjaan laporan skripsi ini.

Situbondo, 15 Mei 2025

Naufal Arif Maulana

CURRICULUM VITAE

Identitas Diri

Nama Lengkap : Naufal Arif Maulana
NPM : 2021503118
Tempat, Tanggal Lahir : Situbondo, 20 Agustus 2002
Program Studi : Teknologi Informasi

Nama Orang Tua

Ayah : Muhammad Ibrahim
Ibu : Fitriya Nuraini

Latar Belakang Pendidikan

SD/MI : SDIT Nurul Anshar
SMP/MTs : SMP Negeri 1 Panji
SMA/SMK/MA : SMK 1 ZAHA
Latar Organisasi : -

Alamat Rumah

Jalan : Jalan Sucipto RT 02 RW 04
Dusun : Dawuhan Parse
Desa/Kelurahan : Dawuhan Parse
Kabupaten/Kota : Situbondo
Provinsi : Jawa Timur
No. Telepon : 0895410549822
E-mail : naufalarifmaulana@gmail.com

ABSTRAK

Naufal, Arif, Maulana. 2025. **ANALISIS SENTIMEN ULASAN PENGGUNA TERHADAP APLIKASI SIMANTAP DI GOOGLE PLAY STORE MENGGUNAKAN METODE NAÏVE BAYES.**

Karya Tulis Ilmiah, Program Studi Teknologi Informasi,
Universitas Ibrahimy. Pembimbing: (1) Firman Santoso,
M.Kom., (2) Nur Azise, M.Kom.

Penelitian Penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna terhadap aplikasi **Simantap** di Google Play Store menggunakan metode **Naïve Bayes**. Analisis sentimen dilakukan untuk mengidentifikasi opini pengguna, baik positif maupun negatif, sehingga dapat memberikan masukan bagi pengembang aplikasi dalam meningkatkan kualitas dan pengalaman pengguna. Data ulasan dikumpulkan melalui *web scraping* dari Google Play Store, kemudian dilakukan **preprocessing** untuk membersihkan dan menyiapkan data. Selanjutnya, algoritma **Naïve Bayes** diterapkan untuk mengklasifikasikan ulasan berdasarkan sentimen. Hasil pengujian menunjukkan bahwa metode ini mampu mengklasifikasikan sentimen dengan tingkat akurasi yang memadai, sehingga dapat digunakan sebagai dasar pengambilan keputusan dan perbaikan layanan aplikasi. Penelitian ini diharapkan memberikan kontribusi dalam pemanfaatan data ulasan pengguna sebagai sumber informasi strategis untuk pengembangan aplikasi.

Kata Kunci : Analisis Sentimen, Naïve Bayes, Aplikasi Simantap, Google Play Store, Klasifikasi Teks.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN KEASLIAN TULISAN	Error! Bookmark not defined.
PERSETUJUAN PEMBIMBING	iii
HALAMAN PENGESAHAN	iv
MOTTO	v
PERSEMBAHAN	vi
KATA PENGANTAR	vii
CURRICULUM VITAE	viii
ABSTRAK.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Rumusan Masalah	4
1.4 Batasan Masalah.....	4
1.5 Tujuan Penelitian	5
1.6 Manfaat Penelitian.....	5
1.7 Metode Penelitian.....	6
1.7.1 Jenis Penelitian.....	6
1.7.2 Teknik Pengumpulan Data.....	7
1.8 Sistematika Pembahasan.....	7
BAB II TINJAUAN PUSTAKA	9
2.1 Penelitian Terdahulu.....	9
2.1.1 Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode <i>Naïve Bayes</i> Classifier Berbasis Konversi Ikon Emosi.....	9
2.1.2 Analisis Sentimen Ulasan Pengguna Aplikasi <i>Google play</i> Menggunakan <i>Naïve Bayes</i>	10
2.1.3 Implementasi analisis sentimen pada ulasan aplikasi Duolingo di <i>Google</i> <i>playstore</i> menggunakan algoritma <i>Naïve Bayes</i>	12
2.2 Landasan Teori.....	12
2.2.1 Analisis Sentimen.....	12
2.2.2 <i>Text Preprocessing</i>	14
2.2.3 TF-IDF (Term Frequency – Inverse Document Frequency).....	15
2.2.4 <i>Naïve Bayes</i> Classifier.....	16

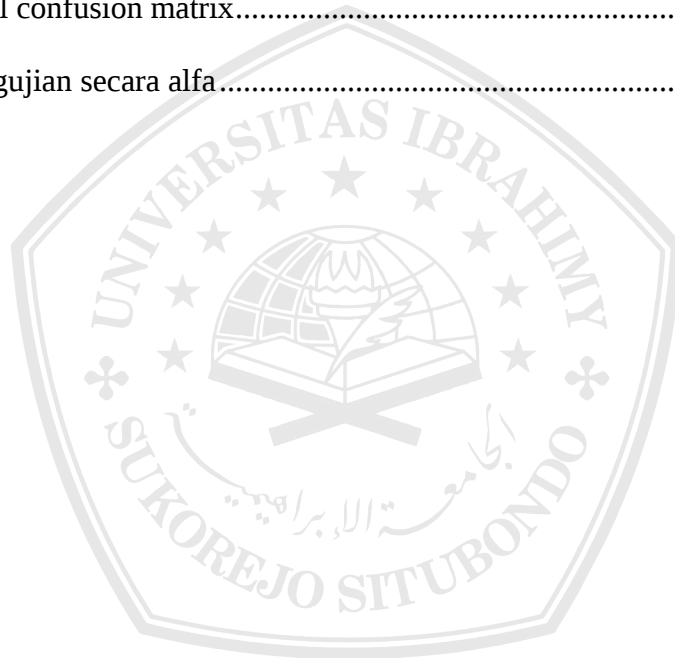
2.2.5	Google Play Store	17
2.2.6	Python	17
2.2.7	Java	18
2.3	Pemodelan.....	18
2.3.1	Flowchart	18
2.4	Perangkat Lunak yang digunakan	20
2.4.1	Netbeans.....	20
2.4.2	Rapidminer.....	20
2.4.3	Visual Studio Code.....	21
BAB III.....		22
ANALISIS DAN PERANCANGAN		22
3.1	Gambaran Umum Objek Penelitian	22
3.1.1	Data scraping Aplikasi SIMANTAP.....	24
3.2	Alur Proses.....	26
3.2.1	Identifikasi dan analisis proses bisnis	27
3.2.2	Analisis Proses Bisnis.....	28
3.2.3	Identifikasi dan analisis kebutuhan	34
3.2.4	Identifikasi Dan Analisis Alternatif Solusi	36
3.2.5	Identifikasi Dan Analisis Desain Proses	38
BAB IV		44
IMPLEMENTASI SISTEM.....		44
4.1	Konstruksi Sistem	44
4.1.1	Kebutuhan Sistem	44
4.1.2	Instalasi Sistem.....	45
4.1.3	Segmen Program	49
4.2	Skenario Pengujian	61
4.2.1	Pengujian Alfa.....	61
4.3	Pengujian	62
4.3.1	Cara Kerja Sistem.....	62
4.3.2	Hasil Pengujian	72
4.4	Maintenance.....	73
BAB V		78
PENUTUP		78
5.1	Kesimpulan	78
5.2	Saran	78
Daftar Pustaka.....		80

DAFTAR GAMBAR

Gambar 3. 1 Rating dan comment dari Aplikasi SIMANTAP	22
Gambar 3. 2 Splash Screen aplikasi SIMANTAP	23
Gambar 3. 3 Halaman Awal SIMANTAP	23
Gambar 3.4 Hasil Data scrapping simantap	25
Gambar 3. 5 Tahapan Preprocessing data	29
Gambar 3. 6 Preprocessing data	39
Gambar 3. 7 Labelling data	39
Gambar 3.8 Naive Bayes.....	40
Gambar 3. 9 Arsitektur Sistem	41
Gambar 3. 10 Context Diagram	43
Gambar 4. 1 Manjalankan Netbeans	46
Gambar 4. 2 Susunan dalam Netbeans (alternatif).....	47
Gambar 4.3 Lokasi Project.....	47
Gambar 4. 4 Buka Project	48
Gambar 4. 5 Tampilan Halaman Project Di Netbeans	48
Gambar 4. 6 Menjalankan Project di Netbeans.....	49
Gambar 4. 7 Halaman Preprocessing	65
Gambar 4. 8 Labelingframe	67
Gambar 4. 9 Halaman NaivebayesFrame.....	70
Gambar 4. 10 Tampilan awan dan upload website	74
Gambar 4. 11 tampilan jendela traning dan Evaluasi	76

DAFTAR TABEL

Tabel 2. 1 Simbol Flowchart.....	19
Tabel 3. 1 identifikasi analisis solusi	36
Tabel 3. 2 Analisis Kelayakan	37
Tabel 4. 1 Pengujian Alfa.....	61
Tabel 4. 2 koding Data Scrapping Python	63
Tabel 4. 3 tabel confusion matrix.....	69
Tabel 4. 4 pengujian secara alfa.....	72



BAB I

PENDAHULUAN

1.1 Latar Belakang

Berdasarkan laporan *Businessofapps*, jumlah aplikasi yang tersedia di *Google Play Store* pada Januari 2024 telah mencapai 3,3 juta aplikasi. Angka ini menunjukkan peningkatan sekitar 15,5% dibandingkan tahun sebelumnya yang berjumlah sekitar 2.597.819 aplikasi. Peningkatan tersebut tentu berdampak pada semakin banyaknya ulasan yang diberikan pengguna terhadap aplikasi di *Google Play Store*. Salah satu aplikasi yang dapat ditemukan di platform tersebut adalah aplikasi Simantap.

SIMANTAP (Sistem Informasi Manajemen Pembayaran Uang Tahun Pesantren) adalah aplikasi berbasis Android gratis yang dikembangkan oleh Pondok Pesantren Salafiyah Syafiiyah Sukorejo Situbondo, dirancang untuk mempermudah proses pembayaran bagi para siswa pesantren. Aplikasi ini berfokus pada memfasilitasi pembayaran UTAP, SPP, dan biaya lain yang terkait dengan pendidikan siswa di Pondok Pesantren Salafiyah Syafiiyah Sukorejo Situbondo. Aplikasi ini menyediakan sistem pembayaran yang nyaman dan efisien yang memungkinkan orang tua atau wali siswa untuk melakukan pembayaran dari mana saja dan kapan saja. Dengan SIMANTAP, pengguna dapat dengan mudah melihat dan melacak riwayat pembayaran, dan melakukan pembayaran.

Ulasan pengguna di *Google Play Store* khusus nya pada aplikasi SIMANTAP dapat mencerminkan kepuasan atau ketidakpuasan terhadap suatu aplikasi. Komentar tersebut biasanya berisi pengalaman positif maupun negatif, seperti

keluhan terkait fitur atau performa aplikasi. Namun, sering kali terjadi ketidaksesuaian antara penilaian yang diberikan pengguna dengan isi ulasan yang diberikan. Misalnya, beberapa pengguna memberikan peringkat bintang 5, tetapi komentar yang ditulis justru mengandung ulasan negative.

Berdasarkan permasalahan tersebut, diperlukan metode analisis sentimen berbasis teks, salah satunya menggunakan *Naïve Bayes* untuk mengklasifikasikan ulasan pengguna aplikasi SIMANTAP di *Google Play Store*. Metode ini tidak hanya mengandalkan penilaian melalui jumlah bintang, tetapi juga memperhatikan isi dari ulasan yang diberikan pengguna. Analisis sentimen sendiri merupakan bagian dari penelitian *Text Mining* yang berfokus pada proses pengklasifikasian teks. Melalui teknik pemrosesan bahasa alami, analisis sentimen dapat mengekstraksi opini, emosi, serta evaluasi tertulis seseorang terhadap suatu topik tertentu[2]. Termasuk analisis sentimen mengenai aplikasi SIMANTAP yang tersedia di *Google Play Store*. Aplikasi ini bertujuan untuk mempermudah pengguna dalam mengakses layanan dipondok pesantren sukorejo. Tapi, sejauh mana aplikasi ini diterima oleh masyarakat dapat dilihat dari ulasan yang diberikan pengguna di *platform* tersebut.

Salah satu metode yang sering digunakan dalam analisis sentimen adalah *Naïve Bayes*, sebuah algoritma klasifikasi berbasis probabilitas yang efektif dalam mengolah data teks. Metode ini bekerja dengan menghitung kemungkinan suatu teks memiliki sentimen tertentu berdasarkan distribusi kata-kata dalam *Dataset* pelatihan. Keunggulan utama *Naïve Bayes* terletak pada efisiensinya dalam menangani data teks dalam jumlah besar serta kemampuannya dalam

mengidentifikasi pola sentimen dengan tingkat keakuratan yang kompetitif [3].

Beberapa penelitian telah membuktikan bahwa *Naïve Bayes* mampu memberikan hasil yang baik dalam tugas analisis sentimen. Dalam penelitian berjudul "Klasifikasi Sentimen *Twitter* menggunakan Pengawasan Jarak Jauh" menunjukkan bahwa model berbasis *Naïve Bayes* dapat mengklasifikasikan sentimen teks dengan cukup baik dan memiliki keunggulan dalam efisiensi komputasi [4]. Selain itu, penelitian yang berjudul "Sebuah Perbandingan Model Peristiwa Untuk Klasifikasi Teks *Naïve Bayes*" menunjukkan bahwa *Naïve Bayes*, khususnya dengan model Multinomial, memberikan hasil akurasi yang cukup baik dalam klasifikasi teks. Dalam penelitian tersebut, algoritma Multinomial *Naïve Bayes* mencapai akurasi sekitar 80% - 86% tergantung pada *Dataset* dan konfigurasi yang digunakan. Model ini terbukti efektif dalam klasifikasi teks, terutama pada data teks pendek dan terstruktur seperti email spam dan dokumen berita.[5].

Walaupun sejumlah penelitian sebelumnya telah menelaah efektivitas *Naïve Bayes* dalam melakukan analisis sentimen, sebagian besar kajian tersebut masih terbatas pada pengolahan teks dengan label sentimen yang sudah ditentukan. Masih terdapat permasalahan terkait adanya ketidaksesuaian antara rating bintang yang diberikan pengguna dengan isi ulasan sesungguhnya. Oleh karena itu, penelitian ini berupaya untuk mengisi celah tersebut dengan melakukan analisis ulasan pengguna aplikasi SIMANTAP menggunakan algoritma *Naïve Bayes*, dengan titik fokus pada ketidaksesuaian antara peringkat bintang dan konten ulasan. Melalui penerapan metode ini, penelitian diharapkan

dapat mengklasifikasikan ulasan ke dalam kategori sentimen positif maupun negatif berdasarkan isi teks, bukan hanya dari nilai bintang semata.

Hasil penelitian ini diharapkan dapat memberikan wawasan mengenai persepsi pengguna terhadap aplikasi SIMANTAP serta membantu pengembang dalam meningkatkan kualitas layanan berdasarkan data ulasan yang dianalisis.

1.2 Identifikasi Masalah

Berdasarkan latar belakang penelitian ini terdapat beberapa identifikasi permasalahan sebagai berikut:

- a. Banyak ulasan di aplikasi SIMANTAP yang tidak selaras antara peringkat bintang dan isi ulasan, misalnya pengguna memberikan rating bintang 5 namun menyampaikan kritik atau keluhan dalam teks ulasannya.
- b. Dibutuhkan metode *Naïve Bayes* untuk menganalisis ulasan pengguna yang memberikan peringkat tinggi tapi ulasan nya tidak sesuai.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah, maka rumusan masalah pada penelitian ini antara lain:

- a. Bagaimana metode *Naïve Bayes* diimplementasikan menggunakan *Java* untuk menganalisis sentimen ulasan pengguna terhadap aplikasi SIMANTAP di *Play store*.
- b. Bagaimana Performa metode *Naïve Bayes* dalam menganalisis sentimen ulasan pengguna terhadap aplikasi SIMANTAP.

1.4 Batasan Masalah

Batasan untuk peneltian ini adalah sebagai berikut :

- a. Data pada penelitian ini adalah hasil ulasan dari para pengguna aplikasi SIMANTAP di *Play store* yang kemudian dilakukan *scrapping data* pada aplikasi tersebut.
- b. Data yang digunakan adalah dokumen berupa hasil ulasan yang telah dilakukan *scrapping data* terhadap aplikasi SIMANTAP di *Play store*, kemudian dilakukan eksperimen menggunakan algoritma *naïve bayes*.

1.5 Tujuan Penelitian

Dari rumusan masalah diatas, diambil lah tujuan dari penelitian ini sebagai berikut:

1. Mengetahui hasil performa algoritma *Naive Bayes* dan hasil analisis sentimen pada Aplikasi SIMANTAP.
2. Mendapatkan hasil visualisasi pemetaan positif dan negatif pada masing masing sentiment ulasan dari aplikasi SIMANTAP.

1.6 Manfaat Penelitian

Adapun manfaat pada penelitian ini antara lain :

- a. Mengembangkan dan menerapkan ilmu yang diperoleh dari dosen serta tugas tugas kuliah atau ujian selama masa perkuliahan berlangsung
- b. Menambah pengetahuan dalam bidang data dan *machine learning* untuk analisis sentimen sehingga bermanfaat.
- c. Memberikan wawasan kepada pengembang aplikasi mengenai persepsi pengguna berdasarkan analisis sentimen, sehingga dapat dijadikan bahan evaluasi dalam meningkatkan kualitas layanan.
- d. Membantu pengembang dalam mengidentifikasi ketidaksesuaian antara

peringkat bintang dan isi ulasan pengguna, sehingga dapat mengurangi ketidaksesuaian dalam penilaian aplikasi.

1.7 Metode Penelitian

Metode penelitian merupakan langkah atau cara yang ditempuh untuk memperoleh pengetahuan baru maupun menyelesaikan suatu permasalahan melalui proses pengumpulan, analisis, serta interpretasi data. Adapun metode yang digunakan dalam penelitian ini adalah sebagai berikut:

1.7.1 Jenis Penelitian

a. Pendekatan Kuantitatif

Pendekatan kuantitatif adalah metode penelitian yang berlandaskan pada paradigma postpositivisme dalam pengembangan ilmu pengetahuan. Ciri utama dari pendekatan ini antara lain menggunakan data numerik untuk proses pengumpulan dan analisis, menerapkan strategi survei maupun eksperimen, melakukan pengukuran serta observasi, serta menguji teori melalui analisis statistik. [6].

b. Eksperimen

Eksperimen yang digunakan dalam penelitian ini adalah menggunakan Bahasa pemrograman *Java* untuk menganalisis penggunaan algoritma *Naïve Bayes* dan pengujian nya menggunakan *Rapid Miner*.

c. Library Research

Penelitian kepustakaan adalah jenis penelitian yang dilakukan dengan menelusuri berbagai sumber tertulis. Riset ini berlandaskan pada beragam karya ilmiah, baik yang telah dipublikasikan maupun yang belum dipublikasikan [7].

1.7.2 Teknik Pengumpulan Data

a. Studi Literatur

Studi pustaka merupakan metode pengumpulan data dengan menelusuri serta mengumpulkan berbagai referensi tertulis yang relevan dengan pengembangan sistem informasi, baik berupa karya ilmiah, jurnal, skripsi, maupun tugas akhir yang disusun oleh peneliti sebelumnya.

b. Web Scrapping Data SIMANTAP

Web scraping merupakan metode untuk mengakses dan memperoleh informasi dari suatu situs secara otomatis tanpa perlu melakukan penyalinan secara manual. Tujuan utamanya adalah mengekstrak data tertentu kemudian menghimpunnya menjadi kumpulan informasi baru. Proses ini berfokus pada kegiatan pengambilan serta ekstraksi data sehingga informasi yang diperoleh menjadi lebih terarah dan memudahkan pencarian. Penerapan web scraping pada aplikasi umumnya hanya menitikberatkan pada cara memperoleh dan mengekstraksi data dengan ukuran yang dapat bervariasi [8].

1.8 Sistematika Pembahasan

Sistematika pembahasan dalam karya tulis ini dibagi menjadi lima bab, yang memiliki suatu tujuan tertentu dalam setiap bab:

BAB I : PENDAHULUAN

Dalam pendahuluan berisi tentang latar belakang, identifikasi masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika pembahasan.

BAB II : TINJAUAN PUSTAKA

Pada bab dua dipaparkan tentang tinjauan pustaka, hal-hal yang terkait serta referensi penunjang yang sesuai dengan judul laporan yang diangkat, juga memuat landasan teori dan perangkat lunak yang akan digunakan.

BAB III : METODELOGI PENELITIAN

Bab ini menjelaskan proses peneliti dalam memperoleh data yang digunakan pada penelitian, meliputi tahap pengumpulan, pengolahan awal, hingga persiapan sebelum dilakukan analisis lebih lanjut dengan algoritma yang digunakan dalam data mining.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil analisis yang telah dilakukan sebelumnya, berisi uraian serta perhitungan terhadap data yang telah diolah sehingga menghasilkan prediksi beserta tingkat akurasi dari prediksi yang diperoleh.

BAB V : PENUTUP

Kesimpulan dari laporan yang dibuat serta saran atau rekomendasi bagi penulis tentang sistem yang dibuat untuk pengembangan aplikasi, lembaga maupun instansi. Semua proses dalam penyusunan laporan penelitian ini disajikan secara ringkas berdasarkan analisis dan data yang telah diuraikan pada bab sebelumnya.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

2.1.1 Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode *Naïve Bayes Classifier* Berbasis Konversi Ikon Emosi

Tingginya jumlah opini masyarakat di Twitter mengenai toko online JD.id menunjukkan adanya beragam pandangan yang bersifat subyektif. Opini tersebut berbentuk data teks yang dapat dianalisis dan dimanfaatkan untuk memperoleh informasi mengenai sentimen publik terhadap kinerja JD.id di media sosial melalui penerapan teknik *Text Mining*, khususnya analisis sentimen. Sentimen yang muncul dikelompokkan ke dalam kategori positif, netral, dan negatif. Pada penelitian ini, proses klasifikasi sentimen pengguna Twitter terhadap JD.id dimodelkan untuk mengevaluasi layanan maupun produk yang ditawarkan marketplace tersebut. Metode yang digunakan adalah *Naïve Bayes Classifier* (NBC) dengan pengembangan melalui pembobotan TF-IDF serta konversi fitur ikon emosi. Penelitian tersebut menunjukkan bahwa metode *Naïve Bayes Classifier* (NBC) tanpa pembobotan tf-idf dan konversi ikon emosi memiliki akurasi sebesar 96,44%, sedangkan NBC dengan pembobotan tf-idf dan konversi ikon emosi mengalami peningkatan akurasi hingga 98%, dengan peningkatan sebesar 1,56%. Pengembangan aplikasi dalam penelitian ini menggunakan bahasa pemrograman PHP dan HTML, sementara pengambilan data *tweet*

dilakukan dengan RStudio. Eksperimen dilakukan menggunakan *Rapidminer* dengan serangkaian proses text *Preprocessing*, seperti *case folding*, konversi ikon emosi, *cleansing*, *text transformation*, *Stopword Removal*, *tokenizing*, dan *Stemming*. Hasil penelitian juga menunjukkan bahwa sentimen masyarakat terhadap toko online JD.id terbagi menjadi tiga kategori, yaitu positif, netral, dan negative[9].

2.1.2 Analisis Sentimen Ulasan Pengguna Aplikasi Google play Menggunakan Naïve Bayes

membahas peningkatan penggunaan aplikasi mobile dan pentingnya memahami kepuasan serta ketidakpuasan pengguna. *Google Play Store* menyediakan fitur rating dan ulasan yang memungkinkan pengguna memberikan opini terhadap aplikasi yang digunakan. Selama ini, predikat aplikasi terbaik diberikan berdasarkan jumlah unduhan dan rating tertinggi. Salah satu kategori aplikasi yang banyak digunakan di Indonesia adalah dompet digital (e-wallet), seperti OVO, DANA, dan Go-Pay, yang memungkinkan transaksi keuangan non-tunai secara online maupun offline.

Penelitian ini berfokus pada aplikasi DANA, yang menyediakan berbagai layanan keuangan, termasuk pembayaran tagihan, transfer uang, dan investasi. Banyaknya layanan yang ditawarkan menimbulkan berbagai feedback dari pengguna, yang tercermin dalam ulasan di *Google Play Store*. Beberapa kendala yang sering dikeluhkan meliputi kegagalan top-up, tidak diterimanya cashback yang dijanjikan, serta customer service

yang sulit dihubungi. Dengan lebih dari 10 juta unduhan, aplikasi DANA telah mendapatkan lisensi dari Bank Indonesia dan mengusung konsep open platform untuk mendukung transaksi digital dengan berbagai merchant seperti Bukalapak, tix.id, KFC, dan BPJS.

Untuk menilai feedback pengguna terhadap layanan DANA, penelitian ini menggunakan analisis sentimen berbasis *Naïve Bayes Classifier* (NBC). *Naïve Bayes* merupakan algoritma klasifikasi probabilistik yang bekerja dengan mengasumsikan bahwa setiap atribut bersifat independen. Keunggulan metode ini adalah kemampuannya dalam mengolah data pelatihan dalam jumlah kecil untuk menghasilkan estimasi parameter yang akurat. Dalam penelitian ini, proses *Text Mining* dilakukan menggunakan bahasa pemrograman Python, mencakup tahapan data *Preprocessing* hingga klasifikasi sentimen.

Hasil penelitian memperlihatkan bahwa penerapan *Naïve Bayes Classifier* dengan seleksi fitur TF-IDF mampu menghasilkan akurasi tertinggi sebesar 85%, dengan nilai precision 79%, recall 85%, dan f1-score 80%. Data uji yang digunakan berjumlah 200 data atau sekitar 20% dari total 3000 data, dengan metode pengujian secara acak (*random Training*). Dengan demikian, dibutuhkan metode yang efisien dan akurat untuk menganalisis serta mengelompokkan ulasan pengguna ke dalam kategori positif, negatif, maupun netral, agar dapat memberikan gambaran yang lebih jelas mengenai persepsi masyarakat terhadap aplikasi DANA. [10].

2.1.3 Implementasi analisis sentimen pada ulasan aplikasi

Duolingo di *Google playstore* menggunakan algoritma *Naïve Bayes*

Penelitian ini berfokus pada analisis sentimen terhadap ulasan aplikasi Duolingo dengan menggunakan metode *Naïve Bayes*. Aplikasi Duolingo merupakan contoh penerapan big data dalam pengolahan informasi yang bersifat kompleks dan berskala besar. Melalui fitur ulasan dan penilaian di *Google Play Store*, data dapat dimanfaatkan untuk mendukung pengembangan aplikasi serta memperbaiki aspek-aspek yang masih perlu ditingkatkan. Dalam penelitian ini, analisis sentimen diterapkan untuk secara otomatis mengolah ulasan pengguna di Indonesia dan mengidentifikasi emosi atau perasaan yang terkandung di dalamnya. Metode *Naïve Bayes* digunakan untuk mengelompokkan ulasan ke dalam dua kategori, yaitu positif dan negatif. *Dataset* penelitian terdiri atas 1000 ulasan aplikasi Duolingo yang diambil dari *Google Play Store*, dengan distribusi seimbang yakni 500 ulasan positif dan 500 ulasan negatif, yang terlebih dahulu telah diberi label secara manual sebelum tahap prapemrosesan. Hasil pengujian menunjukkan tingkat akurasi 86%, dengan precision 89%, recall 83%, serta F1-score 86%. [11].

2.2 Landasan Teori

2.2.1 Analisis Sentimen

Analisis sentimen, atau sering disebut juga opinion mining, merupakan proses menganalisis teks dalam jumlah besar untuk mengetahui apakah teks tersebut mengandung sentimen positif, negatif, atau netral. Penerapan analisis

sentimen sangat penting karena dapat memberikan hasil yang lebih objektif terhadap suatu subjek tertentu melalui penggalian ulasan dari berbagai target. Selain itu, analisis ini juga bermanfaat untuk mencapai skalabilitas yang lebih luas dari suatu program yang dikembangkan oleh pihak tertentu, khususnya dalam mendukung pengembangan bisnis para stakeholder. [12].

Untuk dasar penentuan analisis sentiment sendiri dilakukan berdasarkan dasar pengukuran yang jelas agar proses klasifikasi tidak bersifat subjektif. Oleh karena itu, penelitian ini menggunakan kombinasi pendekatan lexicon-based, analisis konteks kalimat, serta aturan dominasi sentimen. Penjelasan dasar penentuan label adalah sebagai berikut.

Pertama, digunakan pendekatan *lexicon-based* dengan mengacu pada daftar kata positif dan negatif dalam Bahasa Indonesia. Kata-kata seperti baik, bagus, cepat, puas, dan memuaskan dikategorikan sebagai sentimen positif, sedangkan kata error, lambat, gagal, hang, buruk, susah, dan kecewa dikategorikan sebagai sentimen negatif. Jika ulasan mengandung lebih banyak kata positif daripada negatif, maka ulasan diberi label positif, begitu pula sebaliknya. [13]

Untuk ulasan positif saya menggunakan 36 kata sedangkan untuk negative menggunakan 103 kata/ frasa.

Kedua, dilakukan analisis konteks kalimat, karena tidak semua ulasan dapat dinilai hanya berdasarkan kemunculan satu kata. Misalnya, ulasan seperti “aplikasinya bagus tapi sering error” lebih condong ke sentimen negatif karena keluhan lebih dominan. Sebaliknya, ulasan “sempat error tapi sekarang sudah bagus” lebih condong ke sentimen positif karena evaluasi akhir menunjukkan

perbaikan.

Ketiga, apabila dalam satu ulasan terdapat campuran kata positif dan negatif, maka digunakan aturan dominasi sentimen, yaitu:

- jika keluhan lebih dominan → diberi label negatif,
- jika pujian lebih dominan → diberi label positif.

Keempat, penentuan label tidak menggunakan rating bintang karena ditemukan banyak ketidaksesuaian antara rating dan isi ulasan pada data aplikasi SIMANTAP. Dengan demikian, proses labeling sepenuhnya berdasarkan isi teks (content) dari ulasan pengguna, bukan pada skor bintang.

Kelima, seluruh hasil pelabelan yang dilakukan secara otomatis tetap diverifikasi secara manual untuk memastikan akurasi, konsistensi, dan kesesuaian dengan konteks ulasan. Dengan dasar pengukuran tersebut, proses labeling pada penelitian ini menjadi lebih terstruktur, objektif, dan dapat dipertanggungjawabkan dalam pengembangan model analisis sentimen.

2.2.2 Text *Preprocessing*

Text Preprocessing adalah proses mengubah data yang masih tidak terstruktur menjadi data terstruktur sehingga dapat diproses oleh algoritma *machine learning*. Dalam analisis sentimen, tahapan ini berfungsi untuk menyaring, menyederhanakan, dan menstandarkan teks agar lebih mudah dianalisis. Umumnya, langkah-langkah *text Preprocessing* meliputi case folding, tokenisasi, *Stopword Removal*, dan *Stemming*. Seluruh tahapan tersebut bertujuan untuk mengurangi noise pada data serta meningkatkan akurasi dalam klasifikasi sentimen. [14].

2.2.3 TF-IDF (Term Frequency – Nnverse Document Frequency)

TF-IDF merupakan salah satu teknik pembobotan kata yang digunakan secara luas dalam proses pengolahan teks untuk menentukan tingkat kepentingan suatu kata di dalam sebuah dokumen relatif terhadap keseluruhan dataset. Pembobotan ini tidak hanya memperhitungkan seberapa sering sebuah kata muncul dalam satu dokumen, tetapi juga mempertimbangkan seberapa jarang kata tersebut muncul pada seluruh dokumen dalam dataset. Dengan demikian, TF-IDF mampu memberikan nilai bobot yang lebih tinggi pada kata-kata yang benar-benar memiliki makna penting dan bukan sekadar kata yang sering muncul secara umum.

Metode ini sangat membantu dalam analisis teks karena memungkinkan proses representasi data teks menjadi lebih informatif, terutama dalam tahap klasifikasi atau pencarian informasi. Untuk memperoleh nilai bobot TF-IDF, dilakukan perhitungan melalui dua komponen utama, yaitu Term Frequency (TF) dan Inverse Document Frequency (IDF), yang kemudian digabungkan.

Perumusan matematis ini disajikan secara lebih rinci melalui persamaan yang ditampilkan pada **rumus 2.1**, yang menjadi dasar dalam menghitung nilai TF-IDF untuk setiap kata pada proses analisis teks. Rumus tersebut memberikan gambaran jelas mengenai bagaimana setiap elemen dihitung dan bagaimana kontribusinya terhadap nilai bobot akhir suatu kata.

Nilai TF-IDF yang lebih tinggi menunjukkan bahwa kata tersebut lebih berarti dalam analisis dibandingkan kata-kata lain. Teknik ini sering digunakan sebelum teks diklasifikasikan menggunakan Naïve Bayes.

2.2.4 Naïve Bayes Classifier

Naïve Bayes merupakan algoritma klasifikasi berbasis probabilitas yang banyak diterapkan dalam pengolahan teks. Algoritma ini bekerja dengan memanfaatkan *Teorema Bayes*, yaitu perhitungan probabilitas suatu kelas berdasarkan fitur-fitur yang terdapat pada data. Dalam penerapannya pada analisis sentimen, *Naïve Bayes* digunakan untuk mengklasifikasikan teks apakah mengandung sentimen positif atau sentimen negatif. Formula dasar dari *Teorema Bayes* adalah rumus dibawah ini (2.3) :

$$P(X|C) = \left(\frac{P(X|C) \cdot P(C)}{P(X)} \right) \quad (2.3)$$

Di mana:

$P(C|X)$ = Probabilitas suatu kelas C (misalnya, positif atau negatif) berdasarkan fitur X (kata-kata dalam ulasan).

$P(X|C)$ = Probabilitas fitur X muncul dalam kelas tertentu.

$P(C)$ = Probabilitas keseluruhan dari kelas tersebut dalam *Dataset*.

$P(X)$ = Probabilitas keseluruhan dari fitur X di semua kelas.

Keunggulan *Naïve Bayes* dalam analisis sentiment antara lain:

- Cepat dan efisien dalam mengolah data teks berukuran besar.
- Mampu bekerja dengan baik meskipun dengan data yang terbatas.
- Tidak terlalu kompleks daripada metode lain seperti SVM atau deep learning.

2.2.5 *Google Play Store*

Google Play Store merupakan salah satu platform digital resmi yang dikembangkan oleh Google, yang sebelumnya dikenal dengan nama *Android Market*. Platform ini berfungsi sebagai toko aplikasi utama bagi perangkat berbasis Android, di mana pengguna dapat mencari serta mengunduh aplikasi yang dibangun menggunakan Android SDK (*Software Development Kit*). Selain sebagai penyedia aplikasi, *Google Play Store* juga berperan sebagai toko media digital yang menawarkan berbagai konten, seperti buku, film, hingga acara televisi. Layanan ini menyediakan beragam aplikasi, baik gratis maupun berbayar, yang dirancang untuk memudahkan pengguna dalam memperoleh aplikasi maupun game untuk perangkat Android. *Google Play Store* secara otomatis telah terinstall di aplikasi bawaan android, meskipun untuk android versi lama, diwajibkan untuk mengupdate aplikasi *Google Play Store* tersebut terlebih dahulu sebelum dapat digunakan.

2.2.6 *Python*

Python merupakan bahasa pemrograman interpretatif yang mudah dipelajari, terutama bagi pemula, serta menekankan pada kode yang sederhana dan mudah dibaca. Bahasa ini dikembangkan oleh Guido van Rossum pada tahun 1990 di Amsterdam, Belanda, sebagai pengembangan dari bahasa pemrograman ABC. Keunggulan Python antara lain memiliki koleksi pustaka yang sangat luas, struktur bahasa yang sederhana, sistem pengelolaan memori otomatis, serta kemampuan untuk dikembangkan bersama bahasa lain seperti C/C++[14].

2.2.7 Java

Java merupakan bahasa pemrograman tingkat tinggi yang bersifat multiplatform, sehingga dapat dijalankan di berbagai sistem operasi. Bahasa ini dikenal sederhana, berorientasi objek, netral terhadap arsitektur, memiliki kinerja tinggi, mendukung multithreading, serta kuat, dinamis, dan aman. *Java* dapat digunakan pada berbagai platform, baik perangkat keras maupun perangkat lunak, seperti *Windows*, *Linux*, *Solaris OS*, dan *Mac OS*. Platform *Java* sendiri terdiri dari dua komponen utama, yaitu *Java Virtual Machine (JVM)* dan *Java Application Programming Interface (API)* [16].

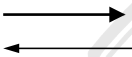
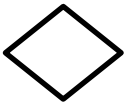
2.3 Pemodelan

2.3.1 Flowchart

Flowchart adalah suatu representasi visual atau penggambaran secara grafik yang menggambarkan langkah-langkah serta urutan-urutan prosedur yang terdapat dalam suatu program atau sistem. Dalam bentuk diagram yang menggunakan simbol-simbol standar tertentu, flowchart menunjukkan aliran logika dan proses yang terjadi dari awal hingga akhir suatu kegiatan atau pemrosesan data. Dengan menggunakan flowchart, seorang analis sistem (analyst) maupun pemrogram (programmer) dapat memperoleh pemahaman yang lebih jelas dan sistematis mengenai struktur dan jalannya suatu proses, sehingga dapat membantu mereka dalam menguraikan atau memecahkan masalah-masalah yang kompleks menjadi bagian-bagian atau segmen-segmen yang lebih kecil dan lebih mudah ditangani. Selain itu, flowchart juga berperan penting dalam membantu proses analisis terhadap berbagai alternatif solusi atau cara kerja lain yang mungkin lebih efektif

dan efisien dalam pengoperasian suatu sistem atau program. Secara umum, penggunaan flowchart dapat memberikan kemudahan dalam memahami dan menyelesaikan suatu permasalahan, terutama untuk masalah-masalah yang memerlukan penelaahan lebih mendalam serta evaluasi yang teliti sebelum dapat ditemukan solusi yang tepat. [17]. Berikut adalah table simbol flowchart 2.1 dibawah ini yang berisi beberapa contoh dari simbol simbol flowchart yang sering digunakan.

Tabel 2. 1 Simbol Flowchart

Nama Simbol	Keterangan
	Flow Simbol yang dipakai untuk menggabungkan antara simbol yang satu dengan simbol yang lain.
	On-Page Reference Simbol untuk keluar, masuk atau penyambungan proses dalam lembar kerja atau project yang sama
	Off-Page Reference Simbol yang digunakan untuk keluar, masuk atau penyambungan proses dalam lembar kerja atau project yang berbeda.
	Terminator Simbol yang menyatakan awal atau akhir dari suatu sistem
	Process Simbol yang menyatakan suatu proses yang dilakukan didalam computer
	Decision Simbol untuk menunjukkan kondisi tertentu yang akan menghasilkan dua output atau kemungkinan jawaban : iya atau tidak.
	Input/Output Simbol yang menyatakan proses input atau output dari project tergantung peralatan

2.4 Perangkat Lunak yang digunakan

2.4.1 Netbeans

NetBeans merupakan *Integrated Development Environment* (IDE) yang banyak dimanfaatkan dalam pengembangan perangkat lunak berbasis *Java*. IDE ini menyediakan sarana terpadu mulai dari penulisan kode, pengujian, hingga distribusi aplikasi. NetBeans dibekali berbagai fitur, seperti editor kode cerdas, debugger, manajemen proyek, serta integrasi dengan beragam tools pengembang. Kelebihannya terletak pada kemampuannya membangun aplikasi *desktop*, *web*, maupun software berbasis *Java Enterprise Edition* (EE). Selain itu, NetBeans juga mendukung pengembangan aplikasi mobile, termasuk Android. Dengan antarmuka yang mudah digunakan serta dukungan terhadap berbagai teknologi *Java*, NetBeans menjadi salah satu IDE populer yang mampu meningkatkan efisiensi dan produktivitas pengembang. [18].

2.4.2 Rapidminer

RapidMiner adalah platform perangkat lunak yang digunakan dalam bidang data science dan *machine learning*. Aplikasi ini menyediakan fitur lengkap mulai dari persiapan data, pemodelan, evaluasi, hingga implementasi, dengan antarmuka yang mudah digunakan. Salah satu keunggulannya adalah fitur *drag-and-drop*, sehingga pengguna dapat membangun dan menguji model tanpa perlu kemampuan pemrograman. RapidMiner juga mendukung berbagai sumber data, seperti file datar, basis data, hingga platform big data (misalnya *Hadoop* dan *Spark*). Selain itu, platform ini dilengkapi dengan banyak operator bawaan yang mendukung proses data mining, termasuk pembersihan data, pemilihan fitur,

hingga pemodelan[19].

2.4.3 Visual Studio Code

Visual Studio Code adalah text editor ringan yang dikembangkan oleh *Microsoft* dan dapat digunakan pada berbagai sistem operasi seperti *Linux*, *macOS*, dan *Windows*. Editor ini secara bawaan mendukung bahasa pemrograman *JavaScript*, *TypeScript*, dan *Node.js*, serta dapat diperluas dengan plugin untuk bahasa lain seperti *C++*, *C#*, *Python*, *Go*, *Java*, maupun *PHP* melalui *Visual Studio Code Marketplace*[20].

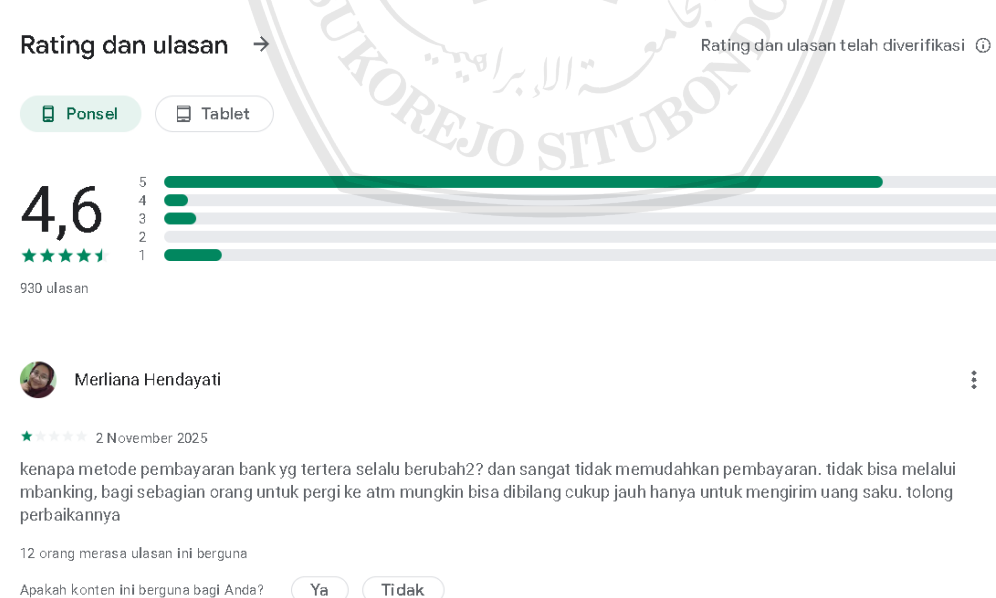


BAB III

ANALISIS DAN PERANCANGAN

3.1 Gambaran Umum Objek Penelitian

Objek penelitian ini adalah aplikasi SIMANTAP (Sistem Informasi Manajemen Administrasi Pondok Pesantren), yaitu aplikasi berbasis Android yang dikembangkan oleh PP Salafiyah Syafi'iyah Sukorejo Situbondo. Aplikasi ini dirancang untuk mempermudah administrasi pembayaran santri dan wali santri, seperti UTAP, SPP, dan biaya pendidikan lainnya. Melalui SIMANTAP, pembayaran dapat dilakukan secara online sehingga lebih praktis dan efisien dibandingkan metode konvensional. Aplikasi tersebut telah di rilis di Play store pada tanggal 02 Mei 2021 dengan total download lebih dari 50.000+ hingga saat ini, serta memiliki Rating 4.6 dari 930 ulasan Seperti gambar yang tercantum dibawah ini :



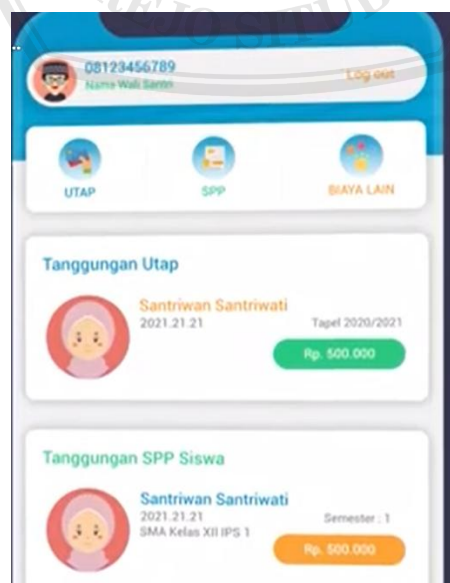
Gambar 3. 1 Rating dan comment dari Aplikasi SIMANTAP

Berikut adalah gambar *Splash Screen* aplikasi SIMANTAP ketika selesai diinstal dan dibuka. Tampilan ini berfungsi sebagai identitas visual aplikasi sebelum pengguna diarahkan ke halaman berikutnya



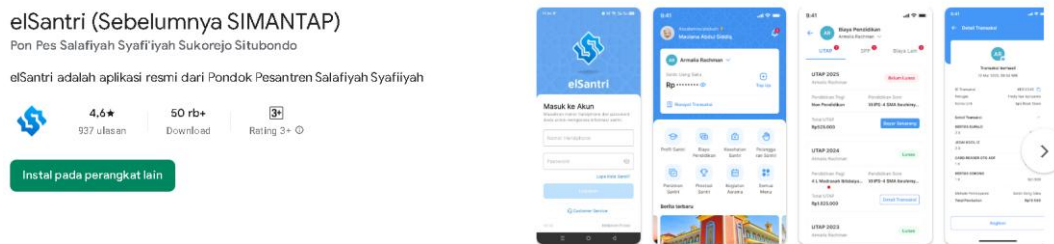
Gambar 3. 2 Splash Screen aplikasi SIMANTAP

Dan berikut tampilan awal dari aplikasi SIMANTAP ketika selesai melakukan proses login pada Aplikasi SIMANTAP.



Gambar 3.3 Halaman Awal SIMANTAP

Dan SIMANTAP telah berubah menjadi elSantri (yang sebelumnya SIMANTAP) pada bulan November tahun 2025 sesuai dengan post akun official Instagram pesantrensukorejo yang dapat dilihat dengan gambar di bawah ini :



Gambar 3.4 Tampilan elSantri di google Play store

Dengan demikian data ulasan yang diambil untuk penelitian ini yaitu ulasan elSantri, tetapi data ulasan tersebut masih terdapat ulasan pengguna dari SIMANTAP sebelumnya.

3.1.1 Data scraping Aplikasi SIMANTAP

Untuk memperoleh data ulasan pengguna terhadap aplikasi SIMANTAP di *Google Play Store*, penelitian ini menggunakan teknik *data scraping*. *Data scraping* adalah proses ekstraksi data dari suatu platform atau *website* secara otomatis menggunakan skrip atau program tertentu. Dalam penelitian ini, data yang dikumpulkan meliputi *reviewid*, *username*, ulasan, bintang, serta tanggal ulasan yang diberikan dari pengguna aplikasi SIMANTAP. Proses *scraping* dilakukan menggunakan *Google play Scraper*, yaitu pustaka Python yang memungkinkan pengambilan data dari *Google Play Store*. Langkah-langkah dalam proses *data scraping* adalah sebagai berikut:

1. Mengidentifikasi aplikasi SIMANTAP di *Google Play Store*

Menggunakan URL resmi aplikasi SIMANTAP untuk memastikan data

yang diambil benar-benar berasal dari aplikasi yang diteliti.

2. Melakukan ekstraksi data menggunakan *Google play Scrape*

Skrip *scraping* akan mengumpulkan data dari halaman ulasan aplikasi. Data yang diambil mencakup:

- Reviewid*
- Username*
- Ulasan
- Rating Bintang (1 – 5)
- Tanggal Ulasan

3. Menyimpan data ke dalam format yang sesuai

Data hasil *scraping* disimpan dalam format CSV atau JSON untuk memudahkan proses analisis lebih lanjut. Berikut adalah data yang didapat dari hasil *data scrapping* ke aplikasi SIMANTAP yang terekspor ke CSV seperti di gambar 3.5 dibawah ini :

reviewid	userName	content	score	at
d372f345-98a7-48f8-Anwar Zadad76		Maaf...klo siang sering eror		4 09/01/2025 07:38
2afc3dae-92b0-4a43-Madinah		Aplikasinya sangat lelet dibuka		1 21/12/2024 14:17
b89d3bd1-51de-42-MJ Mudzakkir Jack		Lumayan membantu		4 20/12/2024 12:50
10495430-8175-45c-Imam Gozali		Terlalu sering di update itu mengganggu juga toh penarr		2 08/12/2024 19:49
6a80ac25-e2ec-48e-Firman Hidayat		Aplikasinya kacau. Sering ceket		1 12/09/2024 13:25
5a507509-6594-48b-Ahmad ahmad		Good		5 05/09/2024 09:16
8078f31a-f0d2-44dc-Muis Misbah		Profesional		5 14/08/2024 08:31
70b329e3-263d-435-Af Rossi		Bagus		5 10/08/2024 12:42
d497a93a-dec2-4be-ABDUL WAFI SDN M.		Sangat membantu wali santri		5 24/07/2024 19:21
d14ec4f2-6f7b-4d6i-SAHIR		Ok		5 24/07/2024 13:42
b519ee0b-bcc8-415-Mahdi Mujiono		Sangat membantu orangbtua dalam mempermudah me		5 16/07/2024 19:32
7623d44e-e150-4ae-Ramzi Situbondo		Bagus		5 03/07/2024 04:53
79340744-2195-477-SOBRI AlfarizyParseh Mantab				5 20/06/2024 08:48
5f4b5ee8-6070-497-Tiga Putra Bintang		Baik		5 17/05/2024 12:56
a4a8454b-5580-474-Myshr 342		Membantu		5 09/05/2024 08:36
b653d0fb-6e06-44c-R. Moh. Nahda		Sangat baik sekali dan sangat membantu kami sebagai w		5 16/04/2024 16:12
eed0287a-7bdb-4a4-bapak gaming		Mantap sekali		5 03/03/2024 15:22
bb231a42-ae68-431-UMI NURUL KHOVIV		aplikasi sangat membantu mempermudah pemba		5 18/02/2024 20:23
204ea14a-e604-4f8l-yunita indrayani		Sangat membantu saya dalam pembayaran biaya putra s		5 18/02/2024 14:57
83738884-30f9-4a3-Imam G		Kenapa harus update terus KL gak ada perubahan yg siki		4 13/02/2024 19:00

Gambar 3.5 Hasil Data scrapping simantap

Data tersebut kemudian diperiksa secara awal untuk melihat kondisi kualitas ulasan sebelum dilakukan proses preprocessing. Berdasarkan hasil scraping, dapat dilihat bahwa setiap baris data mewakili satu ulasan pengguna

lengkap dengan identitas pengguna, isi komentar, skor bintang, serta waktu komentar diberikan. Data ini menunjukkan bahwa ulasan yang masuk sangat beragam, baik dari segi panjang komentar maupun variasi sentimen yang terkandung di dalamnya. Beberapa ulasan berisi pujian terhadap kemudahan penggunaan aplikasi, sementara sebagian lainnya berisi keluhan seperti error, kendala login, atau lamanya proses loading.

Selain itu, ditemukan pula adanya ketidakkonsistenan antara skor bintang dan isi ulasan. Misalnya, beberapa ulasan dengan rating tinggi justru berisi keluhan, sedangkan beberapa ulasan dengan rating rendah memberikan komentar positif. Temuan ini penting karena memperkuat alasan digunakannya analisis sentimen berbasis teks, sehingga penentuan sentimen tidak hanya bergantung pada skor bintang, tetapi benar-benar berdasarkan isi komentar pengguna.

Data yang telah diekspor ke dalam format CSV tersebut menjadi dasar untuk proses analisis selanjutnya, seperti tahapan preprocessing, pelabelan data, pembersihan teks, dan pengelompokan sentimen menggunakan metode Naïve Bayes. Dengan adanya gambaran awal terhadap struktur dan karakteristik data, penelitian ini dapat dilakukan secara lebih terarah dan akurat.

3.2 Alur Proses

Alur proses merupakan gambaran dari suatu proses yang akan dilakukan dalam sebuah pemecahan masalah, proses ini akan menjelaskan secara rinci mengenai tahapan dari penelitian. Dengan adanya proses ini maka akan memudahkan penelitian sehingga berjalan sesuai dengan rancangan yang diinginkan.

3.2.1 Identifikasi dan analisis proses bisnis

Pada bagian ini dilakukan identifikasi terkait dengan objek penelitian yang sedang dilakukan, yaitu ulasan pengguna terhadap aplikasi SIMANTAP di *Google Play Store*. Pada tahap ini akan dijabarkan proses yang terkait sehingga data siap digunakan untuk analisis sentimen.

1. *Preprocessing Data*

Data ulasan yang diperoleh selanjutnya diproses melalui tahap *Preprocessing*, yang mencakup pembersihan teks dari karakter khusus, tokenisasi, penghapusan *Stopword*, *Stemming*, serta normalisasi kata. Tahapan ini dilakukan untuk meningkatkan akurasi analisis sentimen dengan menghilangkan elemen yang tidak relevan.

2. *Labeling Data*

Setelah melalui tahap *Preprocessing*, data ulasan diklasifikasikan ke dalam dua kelas sentimen, yakni positif dan negatif. Proses pelabelan dapat dilakukan secara manual maupun dengan bantuan aplikasi seperti RapidMiner untuk mengenali pola sentimen berdasarkan kata-kata dalam ulasan.

3. Metode Naïve Bayes

Model Penelitian ini menggunakan model klasifikasi Naïve Bayes, yang berfungsi dengan menghitung probabilitas suatu ulasan masuk ke dalam kategori sentimen tertentu berdasarkan fitur yang telah diekstraksi.

4. Evaluasi Model dengan *Confusion matrix*

Sebelum model diekspor dan diimplementasikan, performa model Naïve Bayes dievaluasi menggunakan *confusion matrix*. *Confusion matrix* adalah

tabel yang menunjukkan jumlah prediksi benar dan salah yang dilakukan oleh model klasifikasi, dan terdiri dari empat komponen:

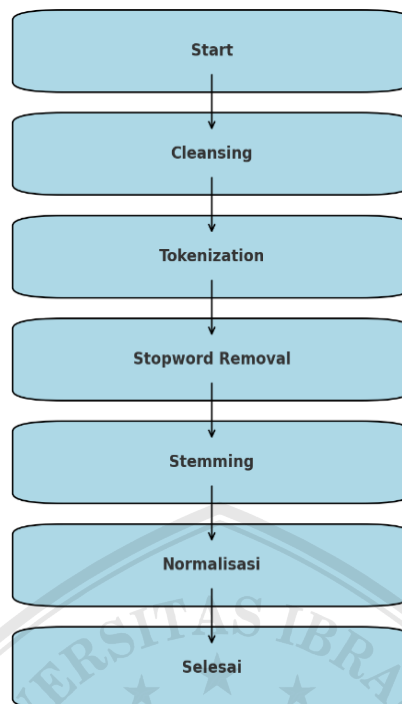
- a True Positive (TP): Ulasan positif yang diprediksi positif.
- b True Negative (TN): Ulasan negatif yang diprediksi negatif.
- c False Positive (FP): Ulasan negatif yang diprediksi positif.
- d False Negative (FN): Ulasan positif yang diprediksi negatif.

3.2.2 Analisis Proses Bisnis

Analisis terhadap proses bisnis dilakukan dengan tujuan untuk memperoleh pemahaman yang mendalam mengenai alur atau tahapan-tahapan yang terlibat dalam keseluruhan proses penelitian, yang dimulai dari tahap awal pengumpulan data, yang melibatkan berbagai metode dan teknik untuk memperoleh informasi yang relevan, hingga pada tahap akhir penerapan model analisis sentimen yang dirancang untuk menganalisis dan menginterpretasi data tersebut secara sistematis. Proses ini dijelaskan lebih rinci melalui beberapa tahapan utama yang saling berhubungan dan mendukung, yang akan diuraikan sebagai berikut:

a. *Preprocessing data*

Data yang telah dikumpulkan tadi kemudian melalui proses *Preprocessing* terlebih dahulu yang terdiri dari cleansing, tokenization, stoword removal, stemming, normalisasi yang agar nanti siap untuk dianalisis data datanya. berikut adalah gambar Langkah-langkah dan tahapan *Preprocessing data* yang dapat dilihat di gambar 3.6 dibawah ini :



Gambar 3. 6 Tahapan Preprocessing data

- 1) *Cleansing*: Menghapus karakter khusus, tanda baca, angka, dan tautan yang tidak memiliki kontribusi terhadap penentuan sentimen
- 2) *Tokenization*: Memecah teks ulasan menjadi unit-unit kata terpisah agar dapat dianalisis secara individual.
- 3) *Stopword Removal*: Menghilangkan kata-kata umum seperti "yang", "dan", "adalah", yang tidak memiliki bobot sentimen.
- 4) *Stemming*: Mengembalikan kata ke bentuk dasarnya, misalnya "meningkatkan" menjadi "tingkat", sehingga menyederhanakan representasi kata.
- 5) *Normalisasi*: Mengubah kata tidak baku atau alay seperti "bgt", "ga", "skrg" menjadi bentuk baku seperti "banget", "tidak", "sekarang" untuk menghindari inkonsistensi data.

b. *Labelling data*

Setelah data ulasan berhasil melalui tahap Preprocessing yang meliputi pembersihan dan persiapan data agar siap dianalisis, langkah selanjutnya yang diambil dalam proses analisis adalah tahap Labeling. Pada tahap ini, dilakukan penentuan sentimen dari masing-masing ulasan yang telah diproses sebelumnya. Proses Labeling ini bertujuan untuk mengklasifikasikan setiap ulasan berdasarkan emosi atau opini yang terkandung di dalamnya, apakah itu bersifat positif, negatif, atau dalam beberapa kasus, netral. Penentuan label ini dilakukan dengan menganalisis konteks dan pilihan kata dalam setiap kalimat ulasan yang ada, yang akan mencerminkan apakah sentimen yang diungkapkan cenderung menunjukkan kepuasan atau ketidakpuasan terhadap produk atau layanan yang diulas. Proses ini sangat penting karena label yang tepat akan mempengaruhi kualitas dan akurasi analisis sentimen yang dilakukan oleh model, yang pada akhirnya memberikan gambaran yang lebih jelas mengenai persepsi konsumen terhadap suatu produk atau layanan yang diulas.

Proses *Labeling* dilakukan secara manual berdasarkan konteks dan makna kalimat. Misalnya, jika ulasan mengandung kata-kata seperti "bagus", "puas", atau "cepat", maka ulasan tersebut diberi label positif. Sebaliknya, jika berisi kata seperti "lambat", "error", atau "tidak berfungsi", maka diberi label negatif. Tetapi pelabelan tersebut juga dapat menggunakan pelabelan secara manual apabila ada kekurangan atau kesalahan dari pelabelan otomatis dari aplikasi.

Labeling dapat dilakukan secara manual atau dengan bantuan alat seperti program yang dibuat saat ini atau *Rapidminer*.

c. Proses Klasifikasi menggunakan *Naïve Bayes*

Pada tahap ini, algoritma Multinomial Naïve Bayes diterapkan untuk menghitung probabilitas dari setiap ulasan yang dianalisis dan menentukan apakah ulasan tersebut termasuk ke dalam kategori positif atau negatif. Algoritma ini dipilih karena merupakan salah satu metode klasifikasi yang paling populer dan efektif dalam analisis teks, khususnya pada data yang direpresentasikan dalam bentuk frekuensi kata. Keunggulan utamanya terletak pada kemampuannya menangani data berdimensi tinggi, seperti kumpulan kata dalam dokumen teks, serta proses komputasinya yang efisien sehingga cocok digunakan dalam aplikasi dengan jumlah data yang besar.

Dengan kata lain, algoritma ini memanfaatkan frekuensi kemunculan kata-kata tertentu untuk memperkirakan apakah suatu ulasan lebih mungkin bersentimen positif atau negatif. Setiap kata dihitung probabilitasnya, baik dalam kaitannya dengan kelas positif maupun negatif, kemudian digabungkan untuk menghasilkan nilai probabilitas keseluruhan. Semakin sering kata-kata yang berasosiasi dengan sentimen tertentu muncul dalam ulasan, maka semakin besar pula peluang ulasan tersebut masuk ke dalam kelas tersebut.

Secara umum, mekanisme kerja algoritma ini dapat dijelaskan dan diturunkan secara matematis menggunakan persamaan **Teorema Bayes**, yang menjadi dasar dalam proses penentuan probabilitas kelas. Persamaan tersebut disajikan sebagai berikut:

$$P(C|X) \left(\frac{P(C|X) \cdot P(C)}{P(X)} \right) \quad (3.1)$$

Dimana :

$P(C|X)P(C|X)P(C|X)$ adalah probabilitas bahwa ulasan X termasuk dalam kategori sentimen C .

$P(X|C)P(X|C)P(X|C)$ adalah probabilitas kata-kata dalam ulasan muncul dalam kategori C .

$P(C)P(C)P(C)$ adalah probabilitas keseluruhan kelas sentimen dalam *Dataset*.

$P(X)P(X)P(X)$ adalah probabilitas keseluruhan kata dalam *Dataset*.

Proses klasifikasi ini dilakukan dengan langkah-langkah berikut:

- 1) Probabilitas prior adalah kemungkinan awal suatu kelas sentimen (positif atau negatif) tanpa mempertimbangkan isi ulasan. Rumusnya:

$$P(C) = \frac{\text{Jumlah dokumen kelas } C}{(\text{JumlahTotalDokumen})} \quad (3.2)$$

Misalnya, jika terdapat 300 ulasan positif dari total 500 ulasan, maka:

$$P(\text{Positif}) = \frac{300}{500} = 0.6 \quad (3.3)$$

- 2) Menghitung Probabilitas Kemunculan Kata ($P(w|C)$)

Probabilitas suatu kata w muncul dalam kelas C dihitung menggunakan frekuensi kata dan teknik *Laplace Smoothing* untuk menghindari probabilitas nol. Rumusnya:

$$P(w_i|C) = \frac{n_{w_i,C} + 1}{\sum n_{w,C} + |V|} \quad (3.4)$$

$n_{w_i,C}$ = jumlah kemunculan kata w_i dalam kelas C

$\sum n_{w,C}$ = total seluruh kata dalam kelas C

$|V|$ = jumlah total kata unik (vocabulary)

- 3) Menggunakan hasil probabilitas tersebut untuk menentukan apakah suatu ulasan dari pengguna tersebut memiliki sentimen ulasan positif atau negatif.

d. Confusion Matrix

Untuk mengevaluasi performa model klasifikasi Naïve Bayes dalam penelitian ini, digunakan *confusion matrix* sebagai alat ukur utama. *Confusion matrix* memungkinkan pengamatan terhadap hasil prediksi model dengan membandingkan antara nilai aktual dan nilai prediksi. Dari *confusion matrix*, dapat dihitung beberapa metrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1-score*.

Akurasi menunjukkan proporsi prediksi yang benar terhadap keseluruhan data yang diuji. *Precision* mengukur seberapa tepat model dalam mengklasifikasikan sentimen positif, sedangkan *recall* mengukur kemampuan model dalam menemukan semua data yang benar-benar positif. *F1-score* merupakan nilai rata-rata harmonik dari *precision* dan *recall*, yang digunakan untuk mengevaluasi model ketika data tidak seimbang. Rumus-rumus dari metrik tersebut adalah sebagai berikut:

1) Akurasi

Mengukur seberapa sering model membuat prediksi yang benar

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.5)$$

2) *Precision*

Mengukur proporsi prediksi positif yang benar-benar positif.

$$Presisi = \frac{TP}{TP + FP} \quad (3.6)$$

3) Recall

Mengukur proporsi aktual positif yang berhasil dikenali.

$$Recall = \frac{TP}{TP + FN} \quad (3.7)$$

4) F1 – Score

Rata-rata harmonik antara *precision* dan *recall*, berguna saat data tidak seimbang.

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + recall} \quad (3.8)$$

3.2.3 Identifikasi dan analisis kebutuhan

a. Identifikasi kebutuhan Fungsional

Sebuah proses akan berjalan lancar jika kebutuhan fungsional terpenuhi, sehingga peran kebutuhan fungsional sangatlah penting dalam sebuah penelitian

b. Analisis Kebutuhan fungsional

Analisis kebutuhan fungsional ini bertujuan untuk menentukan fitur utama yang harus ada dalam sistem agar bisa menjalankan analisis sentimen dengan baik. Sistem yang dibuat harus bisa mengambil data ulasan pengguna dari *Google Play Store* terhadap aplikasi SIMANTAP, termasuk rating bintang, isi ulasan, nama pengguna, dan tanggal ulasan. Data yang dikumpulkan nantinya akan disimpan dalam format yang bisa diolah, seperti CSV atau langsung ke dalam *database*. Setelah data

terkumpul, sistem akan membersihkannya melalui tahap *Preprocessing*, seperti menghapus karakter khusus, angka, tanda baca, serta kata-kata umum yang tidak memiliki makna penting dalam analisis sentimen. Selain itu, teks juga akan diproses lebih lanjut dengan tokenisasi untuk memisahkan kata-kata dan *Stemming* agar kata-kata berubah menjadi bentuk dasarnya.

Setelah tahap *Preprocessing* selesai, sistem akan mengelompokkan ulasan pengguna ke dalam kategori sentimen positif atau negatif menggunakan metode Naïve Bayes. Supaya hasilnya lebih akurat, sistem harus dilatih terlebih dahulu dengan *Dataset* latih sebelum diterapkan pada ulasan aplikasi SIMANTAP. Hasil analisis sentimen yang diperoleh akan ditampilkan bersama dengan tingkat akurasi model yang digunakan. Selain itu, sistem juga harus bisa menyimpan hasil analisis ini ke dalam *database* atau format *file* tertentu agar bisa diakses oleh admin atau pengembang aplikasi sesuai kebutuhan.

Agar hasil analisis lebih mudah dipahami, sistem juga perlu menyediakan visualisasi dalam bentuk grafik atau diagram. Visualisasi ini mencakup jumlah ulasan positif dan negatif serta perbandingan antara rating bintang dengan sentimen yang sebenarnya untuk melihat apakah ada ketidaksesuaian. Dengan adanya tampilan seperti ini, pengembang bisa lebih cepat memahami tren opini pengguna terhadap aplikasi SIMANTAP. Selain itu, sistem ini juga dilengkapi dengan fitur ekspor hasil analisis ke format CSV atau laporan PDF agar bisa digunakan lebih

lanjut. Dengan adanya fitur-fitur tersebut, sistem ini bisa memberikan wawasan yang lebih jelas tentang bagaimana pengguna menilai aplikasi dan membantu pengembang dalam meningkatkan layanan mereka.

3.2.4 Identifikasi Dan Analisis Alternatif Solusi

a. Identifikasi alternatif solusi

Proses ini berfungsi untuk menganalisis alternatif apa saja yang digunakan dalam membuat program. Tabel 3.1 adalah tabel yang merupakan identifikasi alternatif solusi. Terdapat dua alternatif solusi dalam penelitian ini.

Tabel 3. 1identifikasi analisis solusi

Karakteristik	Alternatif 1 (Java)	Alternatif 2 (Rapidminer)
Bagian sistem yang terkomputerisasi	Semua kebutuhan fungsional akan terpenuhi	Semua kebutuhan fungsional akan terpenuhi
Keuntungan, deskripsi ringkas keuntungan yang akan direalisasikan	Implementasi lebih fleksibel dan dapat dikustomisasi sesuai kebutuhan sistem	Proses analisis lebih cepat karena menggunakan antarmuka visual
Perangkat lunak	NetBeans	<i>Rapidminer</i>
Alat Input	Data ulasan dari <i>Google Play Store</i> melalui proses <i>scrapping</i> data	Data ulasan dimasukkan dalam format CSV ke <i>Rapidminer</i>
Alat Output	Hasil klasifikasi sentimen ditampilkan dalam antarmuka <i>Java</i>	Hasil analisis dapat langsung diekspor dari <i>Rapidminer</i>
Alat penyimpanan	<i>Database MySQL</i>	<i>Database MySQL</i> atau file CSV

b. Analisis Kelayakan Alternatif Solusi

Proses ini berfungsi untuk menganalisis sistem yang akan dibuat berdasarkan dokumen yang telah dikumpulkan sebelumnya. Dokumen-dokumen tersebut mencakup berbagai informasi terkait kebutuhan,

tujuan, serta persyaratan yang harus dipenuhi oleh sistem yang akan dikembangkan. Analisis ini sangat penting dilakukan untuk memastikan bahwa setiap elemen dalam sistem yang dirancang dapat berfungsi dengan efektif dan efisien sesuai dengan kebutuhan pengguna serta tujuan yang ingin dicapai oleh organisasi. Proses analisis ini juga bertujuan untuk mengidentifikasi potensi risiko dan tantangan yang mungkin dihadapi selama pengembangan dan implementasi sistem, sehingga langkah-langkah mitigasi dapat dipersiapkan sejak dini.

Tabel 3. 2 Analisis Kelayakan

Kriteria Kelayakan	Bobot	Alternatif 1 (Java)	Alternatif 2 (Rapidminer)
Kelayakan Operasional Fungsional	30%	Mendukung seluruh kebutuhan fungsional serta pengembangan lebih fleksibel dan dapat disesuaikan sesuai kebutuhan sistem.	Mendukung seluruh kebutuhan fungsional dengan proses yang lebih mudah dilakukan oleh pemula, tetapi fleksibilitas lebih rendah.
Skor	30	35	30
Kelayakan Teknis Teknologi	20%	Teknologi yang digunakan sudah mendukung pengolahan data dan integrasi ke dalam sistem berbasis <i>Java</i> secara luas.	Teknologi berbasis GUI yang mempermudah analisis data, namun terbatas dalam fleksibilitas pengembangan lanjutan.
Skor	20	30	25
Kelayakan Ekonomi (Biaya & Efisiensi)	20%	Membutuhkan investasi waktu lebih untuk pengembangan, tetapi lebih fleksibel dalam jangka panjang	Lebih efisien dalam proses analisis tanpa banyak pengkodean, tetapi ada keterbatasan fitur dalam versi gratis.
Skor	20	25	30
Kelayakan Ekonomi (Biaya Pengembangan)	15%	Biaya pengembangan lebih rendah karena menggunakan teknologi open- source tanpa lisensi tambahan.	Biaya lebih tinggi jika menggunakan fitur premium atau lisensi komersial.
Skor	10	30	20
Kelayakan Jadwal Implementasi	15%	Membutuhkan waktu lebih untuk pengembangan dan integrasi sistem.	Implementasi lebih cepat karena berbasis antarmuka visual yang mempermudah pengolahan data.
Skor	10	20	30
Total Skor	100%	140	135

3.2.5 Identifikasi Dan Analisis Desain Proses

desain proses merupakan tahapan dalam merancang sebuah sistem.

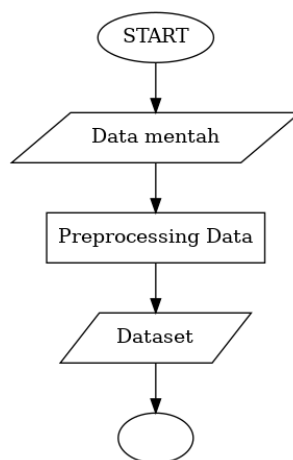
Tahapan ini dilakukan agar proses sistem berjalan sesuai dengan alur yang diinginkan.

a. Identifikasi Desain Proses

Pada identifikasi desain proses menjelaskan tentang alur dari analisis yang dilakukan. Bagian ini fokus pada proses yang akan dilakukan yakni proses input data, perhitungan dengan metode dan evaluasi hasil dari metode. Berikut adalah desain proses yang akan dilakukan.

b. *Preprocessing* data

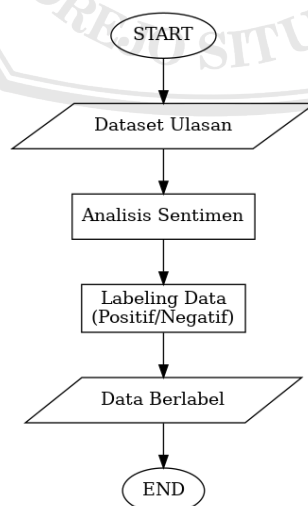
Proses ini dilakukan untuk membersihkan dan mempersiapkan data ulasan pengguna dari *Google Play Store* agar dapat digunakan dalam analisis sentimen. Data yang diperoleh akan melalui beberapa tahap, seperti penghapusan karakter khusus, angka, dan tanda baca, serta penghapusan kata-kata yang tidak memiliki makna signifikan (*Stopwords*). Selain itu, dilakukan proses tokenisasi untuk memisahkan kata-kata dalam teks dan *Stemming* untuk mengubah kata ke bentuk dasarnya. Hasil akhir dari *Preprocessing* ini adalah *Dataset* dalam format yang siap untuk diproses oleh algoritma klasifikasi. Berikut adalah gambar proses *Preprocessing* data 3.3 dibawah ini :



Gambar 3. 7 Preprocessing data

c. *Labeling Data*

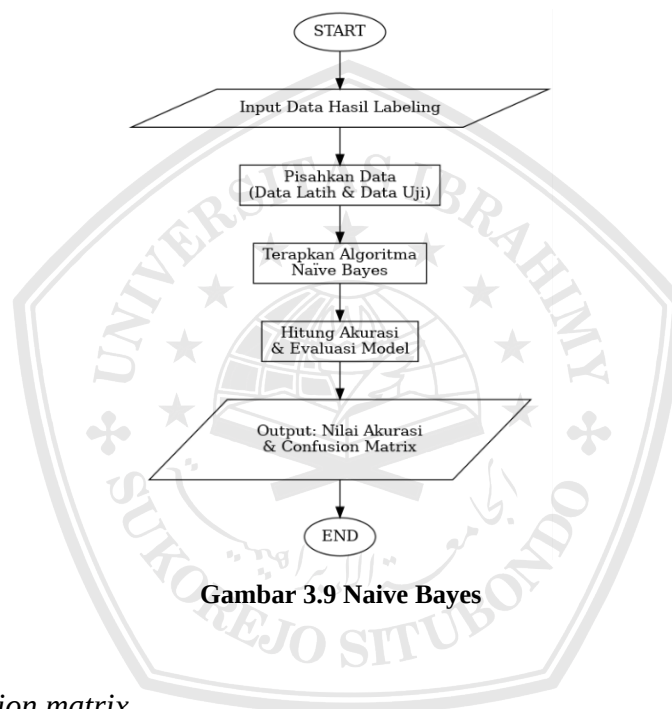
Setelah proses *Preprocessing* selesai, tahap berikutnya adalah pemberian label pada data ulasan. Ulasan yang telah dibersihkan akan dikategorikan ke dalam dua kelas utama, yaitu sentimen positif dan sentimen negatif, berdasarkan isi teks, bukan hanya berdasarkan rating bintang yang diberikan oleh pengguna. *Labeling* dapat dilakukan secara manual atau semi-otomatis untuk memastikan bahwa model mendapatkan data yang valid untuk proses pelatihan.



Gambar 3. 8 Labelling data

d. Metode *Naïve Bayes*

Tahap ini merupakan inti dari analisis sentimen. *Dataset* yang telah diberi label akan dibagi menjadi data latih dan data uji untuk mengukur performa model. Model *Naïve Bayes* kemudian diterapkan untuk mengklasifikasikan ulasan pengguna berdasarkan distribusi kata dalam teks. Proses ini menghasilkan output berupa prediksi sentimen serta metrik evaluasi, seperti akurasi dan *confusion matrix*, yang digunakan untuk mengukur kinerja model dalam mengklasifikasikan data dengan benar.



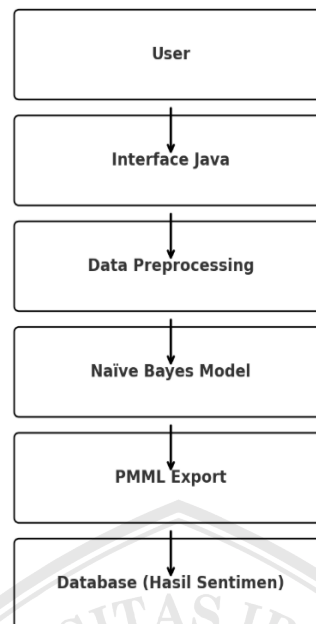
Gambar 3.9 Naive Bayes

e. *Confusion matrix*

performa klasifikasi dievaluasi menggunakan *confusion matrix*. *Confusion matrix* adalah tabel yang menggambarkan kinerja algoritma klasifikasi dengan menunjukkan perbandingan antara hasil prediksi model dan label sebenarnya.

f. Arsitektur system

Arsitektur sistem berfungsi sebagai penjelasan dari pola sistem analisis yang akan dilakukan, berikut adalah arsitektur sistem yang akan dilakukan. Gambar 3.6 menjelaskan arsitektur sistem.



Gambar 3. 10 Arsitektur Sistem

h. Alur Flowchart

Alur flowchart dibawah ini menjelaskan tahapan lengkap dalam proses analisis sentimen terhadap ulasan pengguna aplikasi SIMANTAP menggunakan metode Naïve Bayes. Proses dimulai dari pengumpulan data ulasan melalui tahap scraping dari Google Play Store. Data yang telah diperoleh kemudian disimpan dalam bentuk dataset mentah dan diunggah ke dalam aplikasi berbasis Java sebagai masukan utama sistem.

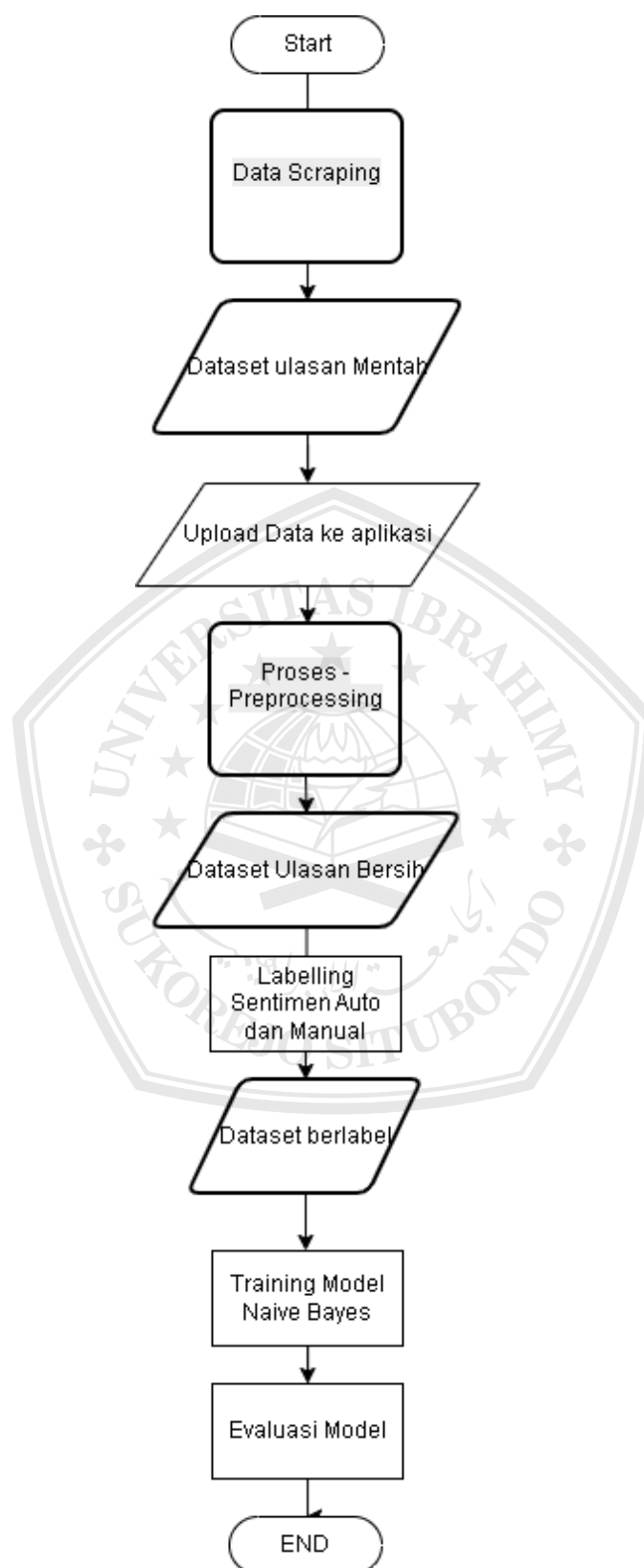
Sebelum dilakukan analisis lebih lanjut, sistem menjalankan serangkaian proses preprocessing yang meliputi pembersihan teks, normalisasi huruf, tokenisasi, penghapusan stopwords, hingga stemming. Pada tahap ini, pengguna juga dapat mengunggah kamus stopwords tambahan untuk menyesuaikan kebutuhan analisis. Hasil preprocessing menghasilkan data ulasan yang telah bersih dan siap digunakan dalam proses pelabelan.

Selanjutnya, sistem melakukan pemeriksaan apakah dataset sudah

memiliki label sentimen. Jika belum, peneliti dapat melakukan proses pelabelan secara manual maupun otomatis.

Model yang telah dibangun kemudian digunakan untuk melakukan prediksi terhadap data uji, dan sistem menghasilkan evaluasi melalui perhitungan confusion matrix, precision, recall, F1-score, dan akurasi. Tahap ini memberikan gambaran mengenai performa model dalam mengklasifikasikan sentimen secara benar. Proses ini memberikan keluaran berupa hasil analisis sentimen yang dapat digunakan sebagai dasar pemahaman persepsi pengguna terhadap aplikasi SIMANTAP.

Secara keseluruhan, alur ini dirancang untuk memastikan bahwa setiap tahapan—mulai dari pengumpulan data, pembersihan teks, pelabelan, pelatihan model, hingga evaluasi, menghasilkan analisis sentimen yang akurat dan dapat dimanfaatkan untuk mendukung pengambilan keputusan terkait pengembangan aplikasi SIMANTAP. Berikut adalah alur flowchart untuk analisis sentiment ulasan pengguna terhadap aplikasi SIMANTAP di google play store menggunakan metode naïve bayes yang dapat dilihat di flowchart dibawah ini :



Gambar 3. 11 Alur Flowchart

BAB IV

IMPLEMENTASI SISTEM

4.1 Konstruksi Sistem

4.1.1 Kebutuhan Sistem

Sistem analisis sentimen ini dikembangkan dengan menggunakan bahasa pemrograman *Java* serta memanfaatkan data yang telah diproses melalui tahap *Preprocessing*. Dalam pengembangannya, sistem memerlukan dukungan perangkat lunak, perangkat keras, dan sumber daya manusia agar dapat dijalankan serta dikelola secara optimal. Adapun rincian kebutuhan perangkat lunak dan perangkat keras yang digunakan dalam analisis sentimen pada ulasan aplikasi SIMANTAP adalah sebagai berikut.

a. *Hardware* (Perangkat Keras)

Perangkat keras (*hardware*) adalah komponen fisik dari sistem komputer yang dapat dilihat maupun disentuh, dan berfungsi mendukung jalannya proses komputerisasi. Spesifikasi perangkat keras yang digunakan dalam pengujian sistem informasi ini adalah sebagai berikut:

1. Laptop
2. RAM minimal 1 GB
3. *Processor* 1.4 GHz Intel Pentium III
4. *Hardisk* Atau SSD(*SolidStateDrive*)
5. *Keyboard* dan *Mouse*

b. *Software* (Perangkat Lunak)

Perangkat lunak (*software*) adalah komponen penting dalam pengembangan

sistem analisis sentimen. Komponen ini mencakup program komputer, pustaka, dokumentasi, serta data digital lain yang berfungsi mendukung proses pengolahan data dan penerapan algoritma Naïve Bayes. Pada tahap pengujian dan pengembangan sistem analisis sentimen untuk ulasan pengguna aplikasi SIMANTAP, perangkat lunak yang digunakan adalah sebagai berikut:

- 1) Netbeans IDE
- 2) *Visual Studio Code*

c. *Brainware*

Brainware merupakan komponen manusia yang berperan penting dalam pengembangan dan pengoperasian sistem analisis sentimen. *Brainware* mencakup seluruh pengguna sistem, mulai dari operator, pengelola data, hingga pengembang sistem. Peran *Brainware* sangat krusial untuk memastikan bahwa proses analisis ulasan pengguna aplikasi SIMANTAP dapat berjalan dengan baik, mulai dari tahapan *Preprocessing*, pelabelan data, hingga implementasi metode Naive Bayes dan evaluasi hasil klasifikasi.

4.1.2 Instalasi Sistem

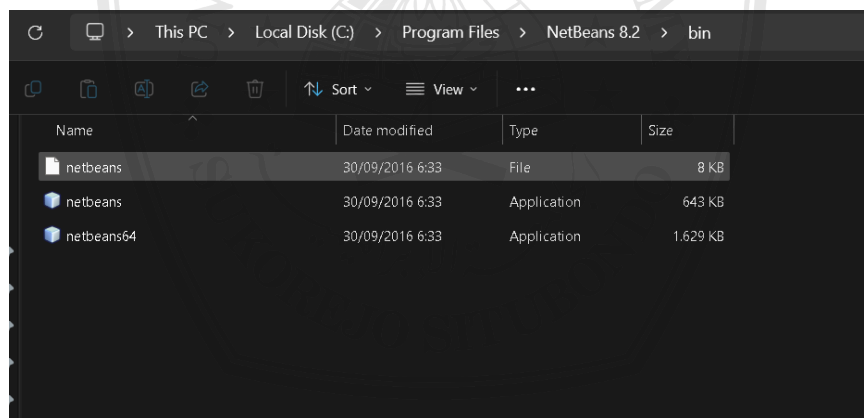
Instalasi sistem merupakan tahapan implementasi dari sistem analisis sentimen yang telah dikembangkan, agar dapat dijalankan dan digunakan pada perangkat komputer. Proses instalasi ini mencakup penyiapan lingkungan pemrograman, pemasangan pustaka yang dibutuhkan, serta konfigurasi sistem agar dapat memproses data ulasan pengguna aplikasi SIMANTAP secara optimal. Langkah-langkah instalasi sistem analisis sentimen yang dilakukan adalah sebagai berikut:

a. Persiapan Awal

Langkah pertama sebelum menjalankan atau mengoperasikan sistem analisis sentimen ini adalah memastikan bahwa seluruh perangkat lunak pendukung telah terinstal dengan baik. Dalam hal ini, software utama yang digunakan adalah NetBeans IDE beserta *Java Development Kit (JDK)*.

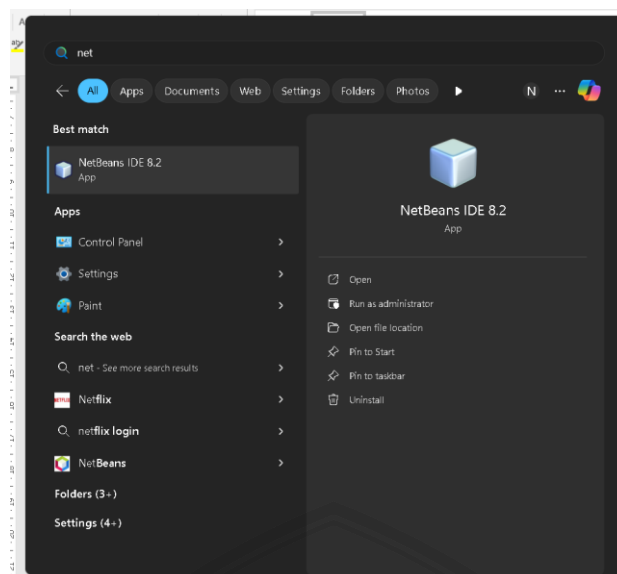
1. Klik File Explorer, Lalu pergi ke Local disk
C:\ProgramFiles\Netbeans\Bin
2. Klik dua kali untuk membuka Aplikasi Netbeans.

Gambar 4.1 menunjukkan tampilan untuk memulai menjalankan Netbeans dibawah ini.



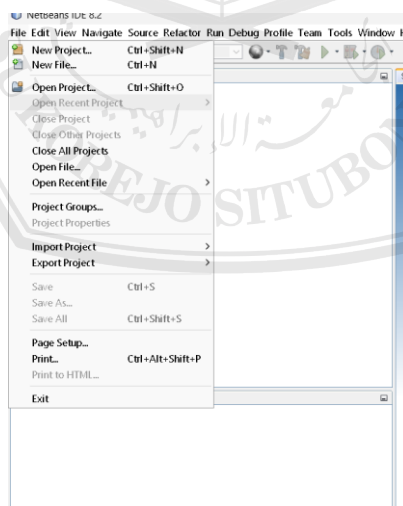
Gambar 4. 1 Manjalankan Netbeans

- b. Atau Cara mudahnya apabila bingung dengan lokasi instalasi file. yaitu tinggal klik Windows di keyboard Lalu ketikkan Netbeansm dan tinggal klik aplikasi tersebut seperti gambar 4.2 dibawah ini.



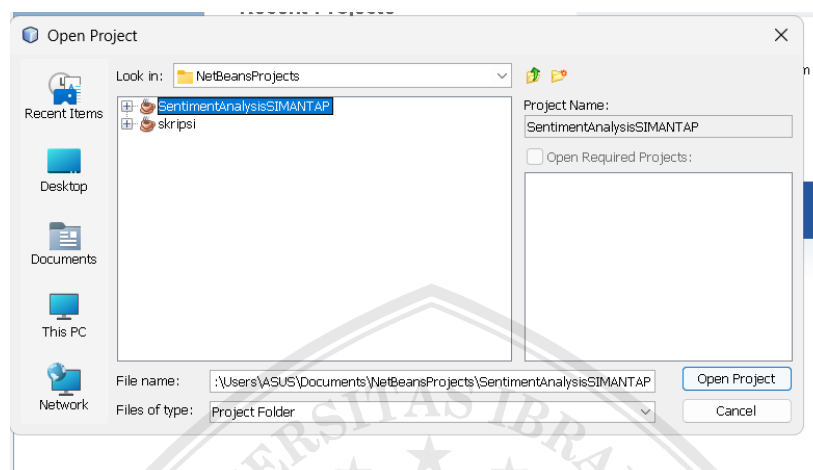
Gambar 4. 2 Susunan dalam Netbeans (alternatif)

- c. Klik File, Lalu Open *Project* dan nanti akan di arahkan tempat *project* berada. Berikut adalah lokasi Menu File dan Open *Project* seperti gambar 4.3 dibawah ini



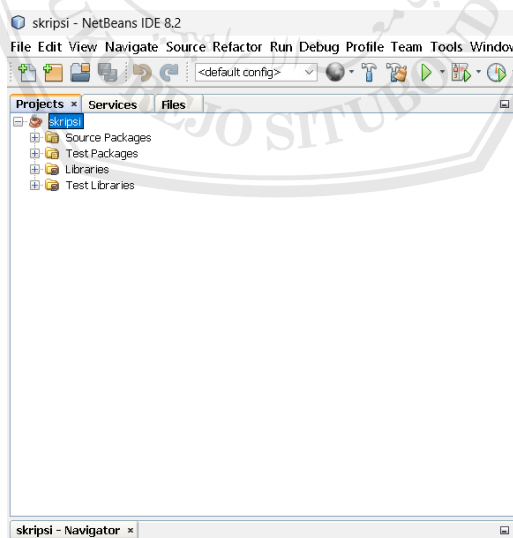
Gambar 4.3 Lokasi *Project*

- d. Langkah selanjutnya yaitu mengarahkan File *project* tempat menyimpan *project* tersebut. Bisa dilihat seperti gambar 4.4 dibawah ini. Lalu Pilih *project* dan Tinggal Klik Open *project*.



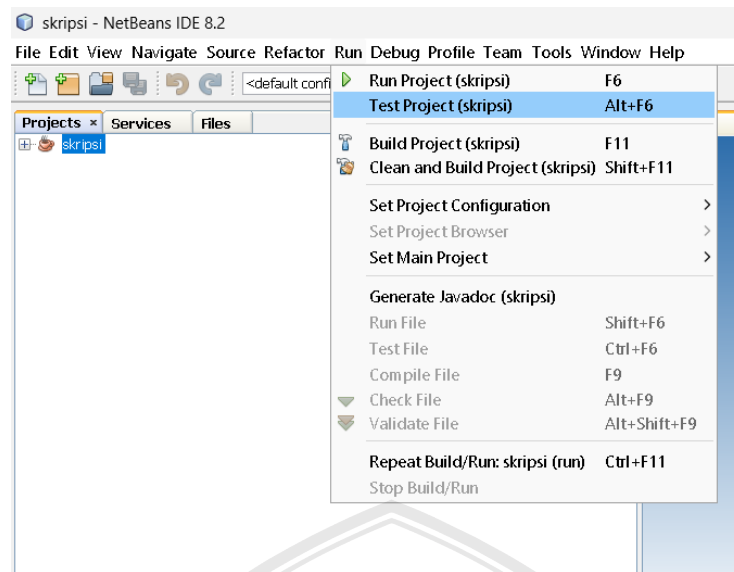
Gambar 4. 4 Buka Project

- e. Setelah Mengklik Open *Project*. Maka *Project* pun selesai terbuka seperti yang. Ditunjukkan pada gambar 4.5



Gambar 4. 5 Tampilan Halaman Project Di Netbeans

- f. Setelah Itu Klik Run di Menu. Lalu pilih *Run Project* Seperti Gambar 4.6 di bawah ini. Dan *Project* pun sedang Berjalan



Gambar 4. 6 Menjalankan Project di Netbeans

4.1.3 Segmen Program

Segmen program adalah kumpulan pernyataan atau deklarasi yang ditulis menggunakan bahasa pemrograman komputer dan dapat dibaca maupun dipahami oleh manusia.

a. Segmen *PreprocessingFrame*

Class *PreprocessingFrame.Java* berfungsi sebagai antarmuka pengguna (GUI) untuk melakukan tahapan *Preprocessing data* pada ulasan pengguna aplikasi SIMANTAP. Tahapan ini sangat penting karena bertujuan untuk membersihkan data sebelum dilakukan pelabelan dan klasifikasi sentimen. Konfigurasi *PreprocessingFrame* yang telah dirancang ke Java class Di Netbeans ditunjukkan pada segmen *program 4.1* di bawah ini.

Segmen program 4. 1 *PreprocessingFrame*

```
package skripsi;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.event.*;
import java.io.*;
import java.util.*;
```

```
import Java.util.regex.*;
public class PreprocessingFrame extends JFrame {
    private JTable table;
    private DefaultTableModel tableModel;
    private JButton loadCSVButton, loadKamusButton,
    preprocessButton, saveButton, LabelingButton;
    private List<String> originalUlasan = new
    ArrayList<>();
    private List<String> cleanedUlasan = new
    ArrayList<>();
    private Set<String> Stopwords = new
    HashSet<>(Arrays.asList(
        "yang", "dan", "di", "ke", "dari", "ini", "itu",
        "dengan", "untuk", "pada", "adalah", "karena", "saya",
        "juga", "tidak", "ya", "jadi", "bisa", "dapat",
        "harus", "lagi", "akan"
    ));
    private Set<String> kataDasar = new HashSet<>();
    public PreprocessingFrame() {
        setTitle("Preprocessing Ulasan");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(950, 550);
        setLayout(null);
        loadCSVButton = new JButton("Load CSV");
        loadCSVButton.setBounds(30, 20, 120, 30);
        add(loadCSVButton);
        loadKamusButton = new JButton("Load Kamus");
        loadKamusButton.setBounds(160, 20, 130, 30);
        add(loadKamusButton);
        preprocessButton = new JButton("Proses
        Preprocessing");
        preprocessButton.setBounds(300, 20, 180, 30);
        add(preprocessButton);
        saveButton = new JButton("Simpan Hasil");
        saveButton.setBounds(490, 20, 150, 30);
        add(saveButton);
        LabelingButton = new JButton("Lanjut ke
        Labeling");
        LabelingButton.setBounds(650, 20, 180, 30);
        add(LabelingButton);
        tableModel = new DefaultTableModel(new
        String[]{"Ulasan Asli", "Hasil Preprocessing"}, 0);
        table = new JTable(tableModel);
        JScrollPane scrollPane = new JScrollPane(table);
        scrollPane.setBounds(30, 70, 870, 400);
        add(scrollPane);
        // Action listeners
        loadCSVButton.addActionListener(e -> loadCSV());
        loadKamusButton.addActionListener(e ->
        loadKamus());
        preprocessButton.addActionListener(e ->
        preprocessData());
        saveButton.addActionListener(e -> saveCSV());
        LabelingButton.addActionListener(e -> new
        LabelingFrame());
        setVisible(true);
    }
}
```

```

    }
    private void loadCSV() {
        JFileChooser fileChooser = new JFileChooser();
        int result = fileChooser.showOpenDialog(this);
        if (result == JFileChooser.APPROVE_OPTION) {
            File file = fileChooser.getSelectedFile();
            originalUlasan.clear();
            tableModel.setRowCount(0);
            try (BufferedReader br = new
BufferedReader(new FileReader(file))) {
                String headerLine = br.readLine();
                if (headerLine == null) {
                    JOptionPane.showMessageDialog(this,
"File kosong.");
                    return;
                }
                String[] headers = headerLine.split(";");
                int contentIndex = -1;

                for (int i = 0; i < headers.length; i++)
                {
                    if
(headers[i].trim().equalsIgnoreCase("content")) {
                        contentIndex = i;
                        break;
                    }
                }
                if (contentIndex == -1) {
                    JOptionPane.showMessageDialog(this,
"Kolom 'content' tidak ditemukan.");
                    return;
                }
                String line;
                while ((line = br.readLine()) != null) {
                    String[] parts = line.split(";", -1);
                    if (parts.length > contentIndex) {
                        String ulasan =
parts[contentIndex].trim().replaceAll("^\\\"|\\\"$", "");
                        originalUlasan.add(ulasan);
                        tableModel.addRow(new
Object[]{ulasan, ""});
                    }
                }
                JOptionPane.showMessageDialog(this, "Berhasil memuat
kolom 'content' dari file!");
            } catch (IOException ex) {
                JOptionPane.showMessageDialog(this,
"Gagal membaca file: " + ex.getMessage());
            }
        }
    }
    private void loadKamus() {
        JFileChooser fileChooser = new JFileChooser();
        int result = fileChooser.showOpenDialog(this);
        if (result == JFileChooser.APPROVE_OPTION) {
            File kamusFile =

```

```

fileChooser.getSelectedFile();
kataDasar.clear();
        try (BufferedReader br = new
BufferedReader(new FileReader(kamusFile))) {
String line; while ((line = br.readLine()) != null)
kataDasar.add(line.trim().toLowerCase());
        }

        JOptionPane.showMessageDialog(this,
"Kamus berhasil dimuat! Total kata: " +
kataDasar.size());
        } catch (IOException ex) {
        JOptionPane.showMessageDialog(this,
"Gagal membaca kamus: " + ex.getMessage());
        }
    }

    private void preprocessData() {
        cleanedUlasan.clear();
        for (int i = 0; i < originalUlasan.size(); i++) {
            String original = originalUlasan.get(i);
            String clean = Preprocessing(original);
            cleanedUlasan.add(clean);
            tableModel.setValueAt(clean, i, 1);
        }
        JOptionPane.showMessageDialog(this,
"Preprocessing selesai!");
    }

    private String Preprocessing(String text) {
        text = text.toLowerCase();
        text = text.replaceAll("[^a-z\\s]", " ");
        String[] tokens = text.split("\\s+");

        StringBuilder result = new StringBuilder();
        for (String token : tokens) {
            if (token.length() > 1 && !Stopwords.contains(token)) {
                if (kataDasar.contains(token)) {
                    result.append(token).append(" ");
                }
            }
        }

        return result.toString().trim();
    }

    private void saveCSV() {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setSelectedFile(new
File("preprocessed.csv"));
        int result = fileChooser.showSaveDialog(this);
        if (result == JFileChooser.APPROVE_OPTION) {
            File saveFile =
fileChooser.getSelectedFile();
            try (BufferedWriter bw = new
BufferedWriter(new FileWriter(saveFile))) {
                bw.write("ulasan_asli,ulasan_bersih\n");
                for (int i = 0; i <
originalUlasan.size(); i++) {
                    String line = "\"" +

```

```
originalUlasan.get(i) + "\",\"" + cleanedUlasan.get(i) +
\"\\n";

        bw.write(line);
    }
    JOptionPane.showMessageDialog(this,
"Berhasil menyimpan hasil Preprocessing!");
    } catch (IOException ex) {
        JOptionPane.showMessageDialog(this,
"Gagal menyimpan file: " + ex.getMessage());
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(PreprocessingFrame::new);
}
}
```

b. Segmen *LabellingFrame*

Class ini bertanggung jawab terhadap proses pelabelan data ulasan. Setelah data hasil *Preprocessing* dimuat, pengguna dapat memberikan label sentimen (positif atau negatif) secara manual. Hasil pelabelan ini akan digunakan sebagai data latih untuk metode Naive Bayes. Konfigurasi *LabelingFrame* Di Netbeans ditunjukkan pada segmen *program* 4.2 di bawah ini.

Segmen program 4. 2 *LabelingFrame*

```
package skripsi;
import javax.swing.*.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;
public class LabelingFrame extends JFrame {
    private JTable table;
    private DefaultTableModel tableModel;
    private JButton loadButton, saveButton,
naivebayesButton;
    // Daftar kata negatif untuk deteksi otomatis
    private final Set<String> kataNegatif = new
HashSet<>(Arrays.asList(
        "buruk", "jelek", "error", "gagal", "lemot",
"parah", "susah", "lama", "tidak", "kecewa",
"eror", "kacau", "update",
        "hang", "bug", "crash", "lelet", "bermasalah",
"kurang", "tidak bisa", "tidak mau"
    ));
}
```

```

public LabelingFrame() {
    setTitle("Labeling Data Ulasan");
    setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    setSize(900, 550);
    setLayout(null);

    loadButton = new JButton("Load Preprocessed
CSV");
    loadButton.setBounds(30, 20, 200, 30);
    add(loadButton);

    saveButton = new JButton("Simpan Hasil
Labeling");
    saveButton.setBounds(250, 20, 200, 30);
    add(saveButton);

    naivebayesButton = new JButton("Lanjut ke Naive
Bayes");
    naivebayesButton.setBounds(650, 20, 180, 30);
    add(naivebayesButton);

    tableModel = new DefaultTableModel(new
String[]{"Ulasan Asli", "Ulasan Bersih", "Label"}, 0) {
        @Override
        public boolean isCellEditable(int row, int
column) {
            return column == 2; // Hanya kolom Label yang bisa diedit
        }
        @Override
        public Class<?> getColumnClass(int
columnIndex) {
            if (columnIndex == 2) return
String.class;
            return super.getColumnClass(columnIndex);
        }
    };
    table = new JTable(tableModel);
    table.setRowHeight(28);

    // Dropdown di kolom label
    JComboBox<String> comboBox = new JComboBox<>(new
String[]{"Positif", "Negatif"});
    table.getColumnModel().getColumn(2).setCellEditor(new
DefaultCellEditor(comboBox));
    JScrollPane scrollPane = new JScrollPane(table);
    scrollPane.setBounds(30, 70, 820, 400);
    add(scrollPane);
    // Action listener
    loadButton.addActionListener(e ->
loadPreprocessedCSV());
    saveButton.addActionListener(e ->
saveLabeledCSV());
    naivebayesButton.addActionListener(e -> new
NaiveBayesFrame()); //
    setVisible(true);

```

```

    }
    private void loadPreprocessedCSV() {
        JFileChooser fileChooser = new JFileChooser();
        int result = fileChooser.showOpenDialog(this);
        if (result == JFileChooser.APPROVE_OPTION) {
            File file = fileChooser.getSelectedFile();
            tableModel.setRowCount(0);
            try (BufferedReader br = new
BufferedReader(new FileReader(file))) {
                String line;
                br.readLine(); // skip header
                while ((line = br.readLine()) != null) {
                    String[] parts = line.split(",", -1);
                    if (parts.length >= 2) {
                        String asli =
parts[0].replaceAll("^\\|\\$", "").trim();
                        String bersih =
parts[1].replaceAll("^\\|\\$", "").trim();
                        String label =
deteksiLabel(bersih);
                        tableModel.addRow(new
Object[]{asli, bersih, label});
                    }
                }
                JOptionPane.showMessageDialog(this,
"Berhasil memuat data ulasan dan memberi label
otomatis!");
            } catch (IOException ex) {
                JOptionPane.showMessageDialog(this,
"Gagal membaca file: " + ex.getMessage());
            }
        }
    }
    private String deteksiLabel(String ulasan) {
        ulasan = ulasan.toLowerCase();
        for (String kata : kataNegatif) {
            if (ulasan.contains(kata)) {
                return "Negatif";
            }
        }
        return "Positif";
    }
    private void saveLabeledCSV() {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setSelectedFile(new
File("labeled_data.csv"));
        int result = fileChooser.showSaveDialog(this);
        if (result == JFileChooser.APPROVE_OPTION) {
            File saveFile =
fileChooser.getSelectedFile();
            try (BufferedWriter bw = new
BufferedWriter(new FileWriter(saveFile))) {
                bw.write("ulasan_asli,ulasan_bersih,label\n");
                for (int i = 0; i <
tableModel.getRowCount(); i++) {
                    String asli =

```

```

tableModel.getValueAt(i, 0).toString().replaceAll("\\\"",
""");
        String bersih =
tableModel.getValueAt(i, 1).toString().replaceAll("\\\"",
""");
        String label =
tableModel.getValueAt(i, 2).toString();
        bw.write("\\\"" + asli + "\",\\\"" +
bersih + "\",\" + label + "\\n");
    }
    JOptionPane.showMessageDialog(this,
"Berhasil menyimpan file Labeling!");
    } catch (IOException ex) {
        JOptionPane.showMessageDialog(this,
"Gagal menyimpan file: " + ex.getMessage());
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(LabelingFrame::new);
}
}

```

c. Segmen *Naivebayesclassifier*

Segmen Class ini berperan sebagai logika algoritma Naive Bayes. Seluruh proses perhitungan probabilitas, pelatihan model (*Training*), dan prediksi sentimen dilakukan dalam class ini tanpa bantuan library eksternal seperti Weka atau *Rapidminer*. Class ini murni dibuat dengan menggunakan bahasa pemrograman *Java*. Konfigurasi *Naivebayesclassifier* yang telah dirancang ke *Java* class Di Netbeans ditunjukkan pada segmen **program 4.3** di bawah ini.

Segmen program 4. 3 *NaivebayesClassifier*

```

package skripsi;
import java.util.*;
public class NaiveBayesClassifier {
    private Map<String, Map<String, Integer>>
wordCountPerClass = new HashMap<>();
    private Map<String, Integer> classDocCount = new
HashMap<>();
    private Map<String, Integer> totalWordCountPerClass
= new HashMap<>();
    private Set<String> vocabulary = new HashSet<>();
    // Training
    public void train(List<String> texts, List<String>
labels) {
        for (int i = 0; i < texts.size(); i++) {
            String text = texts.get(i);

```

```

        String label = labels.get(i).toLowerCase();

        classDocCount.put(label,
classDocCount.getDefault(label, 0) + 1);
        String[] words = text.split("\\s+");

        for (String word : words) {
            vocabulary.add(word);

            wordCountPerClass.putIfAbsent(label,
new HashMap<>());
            Map<String, Integer> wordCounts =
wordCountPerClass.get(label);
            wordCounts.put(word,
wordCounts.getDefault(word, 0) + 1);

            totalWordCountPerClass.put(label,
totalWordCountPerClass.getDefault(label, 0) + 1);
        }
    }
    // Prediksi
    public String predict(String text) {
        double maxProb = Double.NEGATIVE_INFINITY;
        String bestClass = null;

        for (String label : classDocCount.keySet()) {
            double logProb =
Math.log(classDocCount.get(label) / (double)
getTotalDocs());

            String[] words = text.split("\\s+");
            for (String word : words) {
                int wordCount =
wordCountPerClass.get(label).getDefault(word, 0);
                int totalWords =
totalWordCountPerClass.get(label);
                double prob = (wordCount + 1.0) /
(totalWords + vocabulary.size());
                logProb += Math.log(prob);
            }
            if (logProb > maxProb || bestClass == null)
            {
                maxProb = logProb;
                bestClass = label;
            }
        }
        return bestClass.substring(0, 1).toUpperCase()
+ bestClass.substring(1); // Kapitalisasi
    }
    private int getTotalDocs() {
        return
classDocCount.values().stream().mapToInt(Integer::intValue)
.sum();
    }
}
}

```

d. Segmen *Naïvebayesframe*

Segmen Class ini merupakan tampilan antarmuka pengguna (GUI) yang digunakan untuk menjalankan proses pelatihan dan klasifikasi menggunakan algoritma Naive Bayes. Di sinilah pengguna dapat memuat *Dataset* yang telah dilabeli, melatih model, dan menguji prediksi sentimen terhadap data baru. Konfigurasi *Naïvebayesframe* yang telah dirancang ke *Java* class Di Netbeans ditunjukkan pada segmen *program 4.4* di bawah ini

Segmen program 4. 4 *Naïvebayesframe*

```
package skripsi;
import javax.swing.*.*;
import javax.swing.table.DefaultTableModel;
import java.awt.event.*;
import java.io.*;
import java.util.*;
public class NaiveBayesFrame extends JFrame {
    private JTable table;
    private DefaultTableModel tableModel;
    private JButton loadButton, trainButton,
    exportToRapidminerButton;;
    private JTextArea resultArea;
    private List<String> texts = new ArrayList<>();
    private List<String> labels = new ArrayList<>();
    private NaiveBayesClassifier classifier = new
    NaiveBayesClassifier();
    public NaiveBayesFrame() {
        setTitle("Naïve Bayes Sentiment Classifier");
        setSize(950, 600);
        setLayout(null);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        loadButton = new JButton("Load Data Labeling");
        loadButton.setBounds(30, 20, 170, 30);
        add(loadButton);
        trainButton = new JButton("Train & Evaluate");
        trainButton.setBounds(220, 20, 170, 30);
        add(trainButton);
        exportToRapidminerButton = new JButton("Ekspor ke
        Rapidminer");
        exportToRapidminerButton.setBounds(410, 20, 200, 30);
        add(exportToRapidminerButton);
        tableModel = new DefaultTableModel(new
        String[]{"Ulasan", "Label Asli", "Prediksi"}, 0);
        table = new JTable(tableModel);
        JScrollPane scrollPane = new JScrollPane(table);
        scrollPane.setBounds(30, 70, 880, 350);
        add(scrollPane);
    }
}
```

```

        resultArea = new JTextArea();
        resultArea.setEditable(false);
        JScrollPane resultScroll = new
JScrollPane(resultArea);
        resultScroll.setBounds(30, 430, 880, 110);
        add(resultScroll);
        loadButton.addActionListener(e -> loadCSV());
        trainButton.addActionListener(e ->
trainAndEvaluate());
        exportToRapidminerButton.addActionListener(e ->
exportToRapidminerCSV());
        setVisible(true);
    }
    private void loadCSV() {
        JFileChooser fileChooser = new JFileChooser();
        int res = fileChooser.showOpenDialog(this);
        if (res == JFileChooser.APPROVE_OPTION) {
            File file = fileChooser.getSelectedFile();
            tableModel.setRowCount(0);
            texts.clear();
            labels.clear();
            try (BufferedReader br = new
BufferedReader(new FileReader(file))) {
                String line;
                br.readLine(); // skip header
                while ((line = br.readLine()) != null) {
                    String[] parts = line.split(",", -1);
                    if (parts.length >= 3) {
                        String text =
parts[1].replaceAll("^\\|\\$", "").trim().toLowerCase();
                        String label =
parts[2].replaceAll("^\\|\\$", "").trim();
                        texts.add(text);
                        labels.add(label);
                    }
                }
                JOptionPane.showMessageDialog(this, "Data berhasil
dimuat: " + texts.size() + " baris");
            } catch (IOException ex) {
                JOptionPane.showMessageDialog(this, "Gagal
membaca file: " + ex.getMessage());
            }
        }
    }
    private void trainAndEvaluate() {
        if (texts.isEmpty() || labels.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Data belum
dimuat!");
            return;
        }

        classifier = new NaiveBayesClassifier(); // reset
        classifier.train(texts, labels);
        int tp = 0, tn = 0, fp = 0, fn = 0;
        tableModel.setRowCount(0);
        for (int i = 0; i < texts.size(); i++) {

```

```

        String text = texts.get(i);
        String actual = labels.get(i);
        String predicted = classifier.predict(text);
        tableModel.addRow(new Object[]{text, actual,
predicted});
        if (actual.equalsIgnoreCase("Positif") &&
predicted.equals("Positif")) tp++;
        else if (actual.equalsIgnoreCase("Negatif") &&
predicted.equals("Negatif")) tn++;
        else if (actual.equalsIgnoreCase("Negatif") &&
predicted.equals("Positif")) fp++;
        else if (actual.equalsIgnoreCase("Positif") &&
predicted.equals("Negatif")) fn++;
    }

    int total = tp + tn + fp + fn;
    double accuracy = (tp + tn) * 100.0 / total;
    double precision = tp + fp == 0 ? 0 : tp * 100.0 /
(tp + fp);
    double recall = tp + fn == 0 ? 0 : tp * 100.0 /
(tp + fn);
    double f1 = (precision + recall == 0) ? 0 : 2 *
precision * recall / (precision + recall);
    resultArea.setText(String.format(
        "Confusion matrix:\n" +
        "TP = %d, TN = %d, FP = %d, FN = %d\n\n" +
        "Accuracy: %.2f%%\n" +
        "Precision: %.2f%%\n" +
        "Recall: %.2f%%\n" +
        "F1 Score: %.2f%%\n",
        tp, tn, fp, fn, accuracy, precision, recall,
f1
    ));
}

private void exportToRapidminerCSV() {
    JFileChooser fileChooser = new JFileChooser();
    fileChooser.setSelectedFile(new
File("rapidminer_Dataset.csv"));
    int result = fileChooser.showSaveDialog(this);
    if (result == JFileChooser.APPROVE_OPTION) {
        File file = fileChooser.getSelectedFile();
        try (BufferedWriter bw = new BufferedWriter(new
FileWriter(file))) {

            bw.write("ulasan_bersih,label\n"); //
header
            for (int i = 0; i < tableModel.getRowCount(); i++) {
                String bersih = tableModel.getValueAt(i,
1).toString().replaceAll("\\\"", ""); // ulasan_bersih
                String label = tableModel.getValueAt(i,
2).toString(); // label
                bw.write "\"" + bersih + "\", " + label +
"\n");
            }
            JOptionPane.showMessageDialog(this, "Berhasil
mengekspor file ke Rapidminer!");
        }
    }
}

```

```

    } catch (IOException ex) {
        JOptionPane.showMessageDialog(this, "Gagal ekspor:
" + ex.getMessage());
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(NaiveBayesFrame::new);
}
}

```

4.2 Skenario Pengujian

Skenario pengujian adalah pengujian program perangkat lunak yang lengkap dan terimplementasi agar mendapatkan hasil yang memuaskan.

4.2.1 Pengujian Alfa

Pengujian alfa dilakukan dengan teknik pengujian diri sendiri yaitu dengan mencoba aplikasi oleh diri sendiri untuk mengetahui *standart* dari aplikasi apakah sudah sesuai kebutuhan yang diperlukan atau tidak sesuai dengan kebutuhan yang diperlukan. Skenario pengujian Analisis Sentimen SIMANTAP dapat dilihat pada tabel 4.1 dibawah ini.

Tabel 4. 1 Pengujian Alfa

No	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Input Data Ulasan	Memasukkan file CSV ulasan	Data terbaca dan ditampilkan pada tabel	Data berhasil terbaca dan tampil	Berhasil
2	<i>Preprocessing</i>	Menjalankan tokenisasi, <i>Stopword Removal</i> , dan <i>Stemming</i>	Teks ulasan terproses sesuai tahapan	Hasil sesuai, teks bersih dari kata umum dan terpotong kata dasar	Berhasil
3	<i>Labeling Data</i>	Melabeli ulasan positif/negatif	Label tersimpan sesuai input	Label tersimpan dengan benar	Berhasil
4	Pelatihan Model Naïve Bayes	Menjalankan proses <i>Training</i> model	Model terlatih dan menghasilkan output probabilitas	Model terlatih sesuai data	Berhasil
5	Klasifikasi Ulasan	Mengklasifikasikan ulasan baru	Output sentimen positif/negatif sesuai prediksi	Output sesuai hasil perhitungan	Berhasil
6	<i>Confusion matrix</i>	Menampilkan akurasi, precision, <i>recall</i>	Nilai metrik sesuai hasil perhitungan manual	Nilai sesuai	Berhasil
7	Ekspor Ke <i>Rapidminer</i>	Mengekspor model terlatih	FileCSV terbentuk untuk rapidminer	File berhasil diekspor	Berhasil

4.3 Pengujian

Setelah melalui tahapan-tahapan sebelumnya, langkah selanjutnya adalah melakukan tahap pengujian terhadap sistem analisis sentimen yang telah dibangun. Tujuan dari pengujian ini adalah untuk memastikan bahwa seluruh fungsi dalam sistem dapat berjalan sesuai dengan yang dirancang, serta untuk mengidentifikasi adanya kesalahan atau kegagalan dalam proses klasifikasi sentimen. Tahapan ini juga bertujuan untuk mengevaluasi apakah sistem telah memenuhi spesifikasi yang telah dianalisis, dirancang, dan diimplementasikan sesuai dengan metode Naive Bayes.

4.3.1 Cara Kerja Sistem

Penggunaan sistem analisis sentimen ini diawali dengan beberapa tahapan persiapan agar aplikasi dapat dijalankan sesuai dengan fungsinya. Langkah-langkah yang perlu dipenuhi sebelum menjalankan sistem ini adalah sebagai berikut:

- a. Komputer instansi yang digunakan harus sudah terinstal seluruh perangkat lunak pendukung, seperti *Java Development Kit (JDK)*, python dan NetBeans IDE, sebagaimana telah dijelaskan pada bagian instalasi sistem. Hal ini penting untuk memastikan bahwa aplikasi dapat dijalankan dan berfungsi dengan baik.
- b. Yang pertama Pengguna harus membuka aplikasi Visual Studio Code atau semacamnya untuk menjalankan aplikasi *data scrapping* yang menggunakan Bahasa pemrograman python. Berikut adalah tampilan visual studio code yang telah berisi kode yang dibutuhkan seperti gambar 4.7 dibawah ini :

```
C:\Kuliah> testpy > ...
1 from google_play_scraper import reviews, Sort
2 import pandas as pd
3
4 app_id = 'com.utap_app' # ganti ke id app
5
6 result, _ = reviews(
7     app_id,
8     lang='id',          # Bahasa Indonesia
9     country='id',       # Play Store Indonesia
10    sort=Sort.NEWEST,   # Urutkan berdasarkan terbaru
11    count=900           # Ambil 100 ulasan
12 )
13
14 # catatan : konversi ke DataFrame
15 df = pd.DataFrame(result)
16
17 # Simpan ke CSV
18 df.to_csv(r"C:\Kuliah\ulasan_playstoretest.csv", index=False, encoding="utf-8-sig")
19
20 print("Scraping selesai! Data tersimpan dalam ulasan_playstore.csv")
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\ASUS\Documents\SIDANG SKRIP\hosting> & C:/Users/ASUS/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Kuliah/test.py
Scraping selesai! Data tersimpan dalam ulasan_playstore.csv
PS C:\Users\ASUS\Documents\SIDANG SKRIP\hosting>

Gambar 4. 7 Tampilan Awal VS Code

Untuk *coding* nya sendiri dapat di *copy* pada tabel 4.2 dibawah ini :

Tabel 4. 2 koding Data Scrapping Python

1	from google_play_scraper import reviews, Sort
2	import pandas as pd
3	app_id = 'com.utap_app' # ganti ke id app yang akan di ambil ulasannya
4	result, _ = reviews(
5	app_id,
6	lang='id', # Bahasa Indonesia
7	country='id', # Play Store Indonesia
8	sort=Sort.NEWEST, # Urutkan berdasarkan terbaru
9	count=200 # untuk mengambil berapa data ulasan
10	)
11	# catatan : Konversi ke DataFrame
12	df = pd.DataFrame(result)
13	# Simpan ke CSV
14	df.to_csv(r"C:\Kuliah\ulasan_playstoretest.csv", index=False,
15	encoding="utf-8-sig")
16	print("Scraping selesai! Data tersimpan dalam ulasan_playstore.csv")

Untuk mendapat app id yang kita butuhkan kita tinggal membuka play store di peramban browser di Handphone atau laptop lalu buka aplikasi yang kita ingin ambil ulasan nya, lalu isikan pada kolom app_id. (lihat tabel 4.2 diatas tanpa menghapus tanda ‘.). Lalu pada sort ganti *NEWEST* ke *Oldest* untuk

mengurutkan berdasarkan ulasan yang lama, atau tetap *NEWEST* untuk mengurutkan berdasarkan ulasan yang terbaru. Lalu untuk *count*, isi dengan data ulasan yang akan diambil seperti contoh pada tabel 4.2 diatas yaitu mengambil data ulasan sebanyak 200 data ulasan. Bisa lebih tetapi tidak dapat mengambil data lebih dari 500 sekaligus dan terkadang ada *delay* perharinya. Setelah itu klik run pada visual studio code, tunggu hingga muncul “Scraping selesai! Data tersimpan dalam ulasan_playstore.csv” dan nanti akan terdapat file bernama ulasan_playstore.csv yang berisi data ulasan, nama, rating dan lain lain.

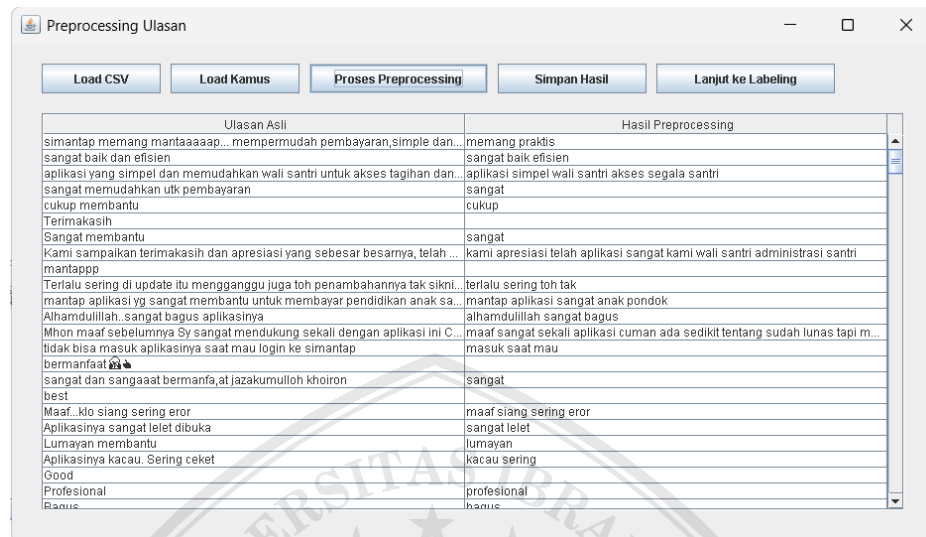
- c. Selanjutnya, Pengguna dapat dapat membuka aplikasi melalui NetBeans IDE dengan cara menjalankan file utama (misalnya di project ini adalah *PreprocessingFrame.java*) untuk memulai sistem analisis sentimen. Aplikasi ini tidak memerlukan web browser, karena berjalan secara langsung dalam lingkungan desktop.

Berikut beberapa tampilan dari aplikasi berbasis GUI dengan menggunakan Bahasa pemrograman *Java* yang sudah dirancang :

1. *PreprocessingFrame*

- a) Langkah Diawali dengan Memasukkan File berformat CSV dengan memiliki kolom bertuliskan ‘*content*’ atau ulasan. Setelah berhasil, masukkan kamus kata dasar yang berformat .txt. setelah itu klik proses *Preprocessing* maka akan muncul ulasan asli dengan ulasan yang telah di *Preprocessing*. Setelah itu klik simpan hasil untuk digunakan datanya pada *labelling* nanti. Setelah itu klik lanjut untuk memunculkan jendela

Labeling. Berikut adalah gambar halaman *Preprocessing* seperti gambar 4.8 dibawah ini.



Gambar 4. 8 Halaman *Preprocessing*

- b) Ambil contoh ulasan: “simantap memang mantaaaaap... mempermudah pembayaran,simple dan praktis...” yang pertama melakukan *Case folding*, atau mengubah semua huruf menjadi kecil.

"simantap memang mantaaaaap... mempermudah pembayaran,simple dan praktis..."
↓
"simantap memang mantaaaaap... mempermudah pembayaran,simple dan praktis..."

- c) Lanjut Ke *Cleansing* Menghapus karakter non-alfabet (angka, tanda baca, simbol) → diganti spasi. Seperti dibawah ini :

"simantap memang mantaaaaap... mempermudah pembayaran,simple dan praktis..."
↓
"simantap memang mantaaaaap mempermudah pembayaran simple dan praktis "

- d) lanjut ke *tokenizing* yaitu memecah teks menjadi potongan kata (tokens) berdasarkan spasi.

```
["simantap", "memang", "mantaaaaap", "mempermudah", "pembayaran",  
"simple", "dan", "praktis"]
```

- e) lanjut ke *Stopword Removal* yaitu Kata umum yang tidak memberi pengaruh pada makna sentimen dihapus. *Stopword* list kamu misalnya: "dan", "yang", "di", "ke", "dari", dan lain lain. Dari token tadi, "dan" termasuk *Stopword* → dibuang:

```
["simantap", "memang", "mantaaaaap", "mempermudah", "pembayaran",  
"simple", "praktis"]
```

- f) lanjut ke *Stemming* (percocokan Kamus Kata Dasar) Mengecek apakah kata ada di kamus kata dasar (kataDasar). Kalau data ada berarti tetap digunakan, kalau tidak ada maka dibuang. Misal di kamus hanya ada kata dasar : ["memang", "praktis"] Maka: "mantaaaaap" dan lainnya tidak ada di kamus maka dihilangkan, sementara Kata lain yang ada sesuai di kamus maka dipertahankan.

```
["memang", "praktis"]
```

- g) Lalu Kata kata tersebut digabung kembali menjadi satu kalimat. Dan itulah hasil *Preprocessing* nya.

```
Memang Praktis
```

2. LabelingFrame

- a) Pada halaman ini, masukkan data ulasan yang telah di *Preprocessing* sebelumnya. Dan otomatis akan muncul ulasan asli, Ulasan Bersih dan

label positif atau negatifnya. Label tersebut didapat dari daftar kata negatif yang ada misal.

"buruk", "jelek", "error", "gagal", "lemot", "parah", "susah", "lama", "tidak", "kecewa", "eror", "kacau", "update", "hang", "bug", "crash", "lelet", "bermasalah", "kurang", "tidak bisa", "tidak mau"

- b) kata tersebut dicek satu satu apabila ada salah satu dari kata negative tesebut, maka akan termasuk Label Negatif, dan jika sebaliknya maka akan masuk label Positif. Pengguna juga dapat mengganti label secara manual jika terdapat kesalahan pada pelabelan otomatisnya. Setelah selesai, klik simpan hasil *Labeling* untuk menyimpan hasil tersebut yang nanti akan digunakan di halaman *Naivebayesframe* Selanjutnya. Lalu terkahir klik lanjut ke *naivebayes* untuk memunculkan halaman *naivebayesframe* nya. Berikut tampilan *Labelingframe* seperti gambar 4.9 dibawah ini :

Ulasan Asli	Ulasan Bersih	Label
Tertalu sering di update itu mengganggu juga...	terlalu sering foh tak	Negatif
mantap aplikasi yg sangat membantu untuk m...	mantap aplikasi sangat anak pondok	Positif
Alhamdulillah..sangat bagus aplikasinya	alhamdulillah sangat bagus	Positif
Mhon maaf sebelumnya Sy sangat mendukung...	maaf sangat sekali aplikasi cuman ada sedikit...	Positif
tidak bisa masuk aplikasinya saat mau login k...	masuk saat mau	Positif
bermanfaat 🙏		Positif
sangat dan sangaaat bermanfa	at jazakumulloh khoiron	Positif
best		Positif
Maaf...klo siang sering error	maaf siang sering error	Negatif
Aplikasinya sangat lelet dibuka	sangat lelet	Negatif
Lumayan membantu	lumayan	Positif
Aplikasinya kacau. Sering ceket	kacau sering	Negatif
Good		Positif
Profesional	profesional	Positif

Gambar 4. 9 *Labelingframe*

3. *Naivebayesclassifier*

Class *NaiveBayesClassifier.Java* merupakan bagian penting dari sistem yang berfungsi untuk menjalankan proses klasifikasi sentimen menggunakan

metode Naive Bayes. Pada class ini tidak terdapat tampilan antarmuka (GUI), karena isinya hanya berupa kode logika program yang mengatur bagaimana data dilatih dan diprediksi.

Beberapa fungsi utama dalam class ini antara lain:

- train()*, digunakan untuk melatih model berdasarkan data ulasan yang sudah dilabeli.
- predict()*, digunakan untuk memprediksi apakah sebuah ulasan bersifat positif atau negatif.
- calculateProbability()*, digunakan untuk menghitung probabilitas suatu ulasan masuk ke kelas tertentu berdasarkan kata-kata yang muncul.

Berikut Penjelasan

- Class ini bekerja di belakang sistem, dengan cara kerja yang pertama yaitu membaca .csv dari *LabelingFrame*. Lalu setiap ulasan sudah bersih (hasil *Preprocessing*) dan sudah punya label (“Positif” / “Negatif”) lalu melakukan tokenisasi dan hitung frekuensi kata per kelas. dimana Semua ulasan Positif maka dihitung kata-katanya. sementara Semua ulasan Negatif juga dihitung kata-katanya. Contoh hitung :

```
Positif:
simantap → 1
mempermudah → 1
praktis → 1
menarik → 1
mudah → 1
digunakan → 1
Negatif:
error → 1
lemot → 1
susah → 1
Gagal → 1
```

- b) Menghitung probabilitas naïve bayes. Dengan rumus umum tiap kata dalam kelas c:

$$P(\text{Kata}|C) = \frac{\text{Jumlah kemunculan kata kelas di kelas } C + 1}{\text{Total kata di kelas } C + \text{jumlah semua kata unik}}$$

Dan menggunakan +1 = Laplace smoothing (supaya kata yang belum pernah muncul tidak nol probabilitasnya) lalu menyimpan hasil probabilitas kata per kelas.

- c) Contoh ada ulasan baru “ aplikasi praktis dan mudah” maka prosesnya sama seperti sebelumnya, hingga tokenisasi sampai *Stemming*. lalu hitung skor untuk positif dan negative nya:

$$\begin{aligned} &P(\text{positif}) \times P(\text{aplikasi}|\text{Positif}) \times P(\text{Praktis}|\text{positif}) \\ &\quad \times P(\text{mudah}|\text{Positif}) \\ &P(\text{negatif}) \times P(\text{aplikasi}|\text{Negatif}) \times P(\text{Praktis}|\text{Negatif}) \\ &\quad \times P(\text{mudah}|\text{Negatif}) \end{aligned}$$

- d) dari perhitungan tersebut dipilih dengan kelas skor terbesar. Setelah itu dilakukan lah evaluasi (*Confussion matrix*) dan hasil nya ditaruh di *confusion matrix*. berikut adalah table 4.2 confusion matrix :

Tabel 4. 3 tabel confusion matrix

Asli positif	Prediksi Positif	Prediksi Negatif
	True Positif	False Negatif
Asli negative	False Positif	True Negatif

Dan nantinya akan digunakan oleh class lain (seperti *NaiveBayesFrame.Java*) untuk menampilkan hasil klasifikasi ke pengguna.

4. *NaivebayesFrame*

Pada halaman ini masukkan Data hasil *Labeling* sebelumnya lalu klik *train & Evaluate* maka akan muncul ulasan, label asli dan label prediksi yang

dimana label prediksi tersebut digunakan untuk menghitung probabilitas ulasan tersebut masuk ke kelas positif atau negative berdasarkan Frekuensi kata pada data ulasan, Rumus probabilitas Naive Bayes serta Nilai log untuk mencegah *underflow*. halaman tersebut juga tampil *Confusion matrix* dari hasil ulasan tersebut seperti yang terlihat pada gambar 4.9

Ulasan	Label Asli	Prediksi
assalamualaikum mohon maaf cara agak sulit pa...	Positif	Positif
kecewa berat sangat lalai sudah lewat sudah beb...	Negatif	Negatif
kadang gagal meski benar	Negatif	Negatif
gak bisa login	Positif	Positif
buka	Positif	Positif
kadang ini aplikasi eror	Negatif	Negatif
gatau deh knp	Positif	Positif
	Positif	Positif
baik dalam relog maupun beranda data	Positif	Positif
segala hal	Positif	Positif
alokum sampai sekarang masih belum tampil ke...	Positif	Positif
sangat perlu kalau jangan pakai tapi langsung kal...	Negatif	Negatif
sering	Positif	Positif
simple untuk orang tua tapi untuk update biaya sa...	Negatif	Negatif
aplikasi bikin pusing wali santri semua wali santri...	Positif	Positif
akan tetapi setiap kami mau lihat tagihan karakny...	Positif	Positif
mohon maaf tolong uang uang jangan sampai ga...	Negatif	Negatif
aplikasi ter baik	Positif	Positif
terima sangat anak kami lebih baik	Positif	Positif

Confusion Matrix
TP = 303, TN = 17, FP = 7, FN = 0
Accuracy: 97.86%
Precision: 97.74%
Recall: 100.00%
F1 Score: 98.86%

Gambar 4. 10 Halaman NaivebayesFrame

a) contoh disini punya 4 dokumen *Training* (2 Positif, 2 Negatif):

Positif: “Mantap Mudah,” praktis Mantap” Tokens: mantap,mudah.
Praktis, mantap.

Negatif: “Sering error”, “gagal Login” token: sering ,error. Gagal, login.

Dari *Training* hitung: vocabulary $|V| = \{\text{mantap, mudah, praktis, sering, error, gagal, login}\} \rightarrow |V|=8$. Total kata di kelas Positif = 4 (mantap:2, mudah:1, praktis:1), total kata di kelas Negatif = 4 (sering:1, error:1, gagal:1, login:1). Prior tiap kelas:

$$P(Pos) = \frac{2}{4} = 0.5 \quad P(neg) = 0.5$$

Probabilitas kata dengan Laplace Smoothing

$$P(\omega|C) = \frac{\text{count}(\omega, C) + 1}{\text{totalWords } c + |V|}$$

Contoh angka

$$P(\text{mantap}|\text{Pos}) = \frac{2+1}{4+8} = \frac{3}{12} = 0.25$$

$$P(\text{mantap}|\text{Neg}) = \frac{0+1}{4+8} = \frac{1}{12} \approx 0.0833$$

$$P(\text{error}|\text{Neg}) = \frac{1+1}{4+8} = \frac{2}{12} \approx 0.1667$$

- b) Untuk memprediksi sebuah ulasan hitung score log tiap kelas (agar numerik stabil):

$$\text{Score}(C) = \text{Log } P(C) \sum_{\omega \in \text{ulasan}} \text{Log } P(\omega|C)$$

Contoh prediksi untuk ulasan baru "mantap praktis" (tokens: mantap, praktis):

$$\text{Score}(\text{Pos}) = \log 0.5 + \log 0.25 + \log ((1+1)/4+8) = -0.6931 + (-1.3863) + (-1.7918) \approx -3.8712$$

$$\text{Score}(\text{Neg}) = \log 0.5 + \log (1/12) + \log (1/12) = -0.6931 + (-2.4849) + (-2.4849) \approx -5.6629.$$

Karena $\text{Score}(\text{Pos}) > \text{Score}(\text{Neg})$, prediksi = Positif. Setelah memprediksi semua dokumen pada set uji susun *confusion matrix*:

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix}$$

Dengan Metrik :

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}, \text{Precision} = \frac{TP}{TP + FP}, \text{Recall} = \frac{TP}{TP + FN}$$

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

Jika misal test set berisi "mantap praktis" (Pos) dan "error login" (Neg)

dan model memprediksi Pos dan Neg masing-masing dengan benar, maka

TP=1, TN=1, FP=0, FN=0 → akurasi = 100%.

4.3.2 Hasil Pengujian

a. Hasil Pengujian secara Alfa

Hasil pengujian secara alfa menunjukkan bahwa seluruh fitur yang telah dirancang dan dikembangkan dapat berfungsi dengan baik dan berjalan sesuai dengan tujuan serta fungsinya yang telah ditentukan sebelumnya. Pengujian dilakukan secara menyeluruh dengan menggunakan berbagai skenario yang dirancang untuk mewakili kondisi penggunaan nyata dalam lingkungan pengembangan, yakni NetBeans. Setiap fungsi diuji secara mendetail, dimulai dari tahap input data yang diterima sistem, diikuti dengan proses pemrosesan atau eksekusi yang dilakukan oleh aplikasi, hingga tahap akhir yaitu proses ekspor model yang dihasilkan. Proses pengujian ini bertujuan untuk memastikan alur kerja dalam aplikasi berjalan lancar. Berikut tabel 4.2 dari hasil pengujian secara alfa.

Tabel 4. 4 pengujian secara alfa

No	Komponen yang Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Apakah sesuai yang diharapkan	Keterangan
1	ProcessingFrame	LoadCSV	White Box	Berhasil
		LoadKamus	White Box	Berhasil
		Proses Preprocessing	White Box	Berhasil

		Simpan Hasil <i>Preprocessing</i>	<i>White Box</i>	Berhasil
		Lanjut halaman Labelling	<i>White Box</i>	Berhasil
2	<i>LabelingFrame</i>	<i>Load Preprocessed CSV</i>	<i>White Box</i>	Berhasil
		Simpan Hasil Labelling	<i>White Box</i>	Berhasil
		Lanjut Halaman NaiveBayes	<i>White Box</i>	Berhasil
3	<i>NaiveBayesFrame</i>	<i>Load CSV yang diLabeling</i>	<i>White Box</i>	Berhasil
		<i>Train Dan Evaluate</i>	<i>White Box</i>	Berhasil
		<i>Ekspor Ke RapidMiner</i>	<i>White Box</i>	Berhasil

4.4 Maintenance

Sistem analisis sentimen ini tetap memerlukan perawatan dari pengguna agar dapat berjalan dengan optimal. Pengguna perlu memperhatikan kestabilan perangkat dan memastikan file pendukung tersimpan dengan baik. Sistem ini memiliki dua versi:

1. Versi Lokal – Aplikasi berjalan langsung di komputer pengguna, sehingga seluruh proses pengolahan data dilakukan secara lokal. Versi ini telah diekspor menjadi file berformat .EXE dan dapat diunduh melalui Google Drive, memungkinkan pengguna menjalankan aplikasi kapan saja tanpa koneksi internet. Kelebihannya adalah kecepatan eksekusi lebih tinggi karena pemrosesan dilakukan secara lokal dan data sensitif tetap berada di perangkat pengguna. Link download untuk aplikasinya dibawah ini :

https://drive.google.com/drive/folders/1zO2zsmI7c6VfNJyCeU3pUOVV1uU7Ngzf?usp=drive_link
2. Versi Web (Streamlit Hosting) – Aplikasi juga tersedia dalam versi berbasis web yang dihosting menggunakan Streamlit. Dengan versi ini, pengguna dapat mengakses sistem analisis sentimen dari mana saja melalui browser, tanpa perlu menginstal perangkat lunak tambahan.

Kelebihannya termasuk kemudahan akses lintas perangkat, pembaruan otomatis, dan fleksibilitas penggunaan di berbagai platform tanpa tergantung pada sistem operasi tertentu. Berikut adalah link untuk versi web :

<https://skripsibismillah.streamlit.app/>

dan Berikut adalah beberapa tampilan serta sedikit penjelasan tentang yang versi web pada gambar 4.11 dibawah ini:



Gambar 4.11 Tampilan awal dan data scrapping.

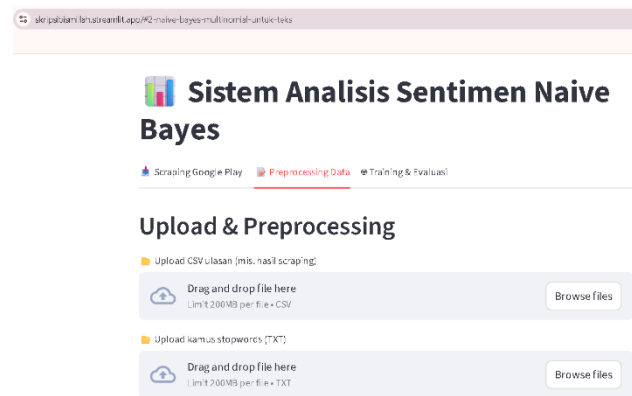
pada halaman tersebut terdapat tampilan bar scrapping google play store dimana kita bisa mendapatkan data ulasan dari aplikasi yang ada di play store. Sebagai contoh disini saya akan mengambil ulasan aplikasi elSantri (yang dulu SIMANTAP) dengan cara mengcopy code id app nya saja yang dapat dilihat di url browser ketika membuka playstore melalui aplikasi browser. Seperti contoh pada gambar 4.12 dibawah ini dan yang kita gunakan adalah com.utap_app :



Gambar 4. 12 app id Play store yang di copy

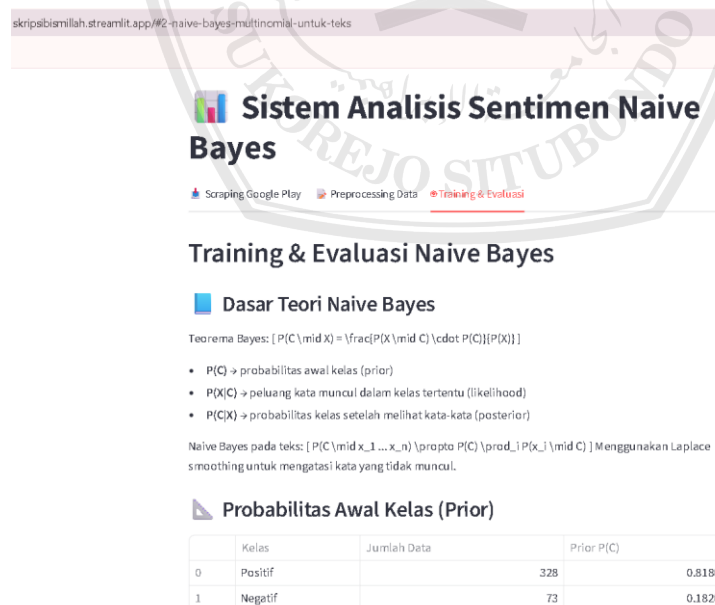
Setelah mengcopy dan menaruh app id kedalam kolom app id. Isi berapa ulasan yang akan diambil, data ulasan yang diambil memiliki batas sehingga tidak bisa diambil semua, setelah selesai. Klik mulai scrapping dan nanti akan terdownload file berupa excel berformat .CSV yang berisi ulasan dari aplikasi yang kita ambil. Setelah itu lanjut ke tampilan jendela preprocessing data.

pada halaman tersebut masih sama dengan versi yang Java Atau aplikasinya, setelah link dibuka. Kita harus mengupload Data yang berisi ulasan di di kolom 'content' dan mengupload kamus yang berisi kata kata yang akan digunakan untuk stopword khusus. Setelah kedua file di upload akan muncul ulasan asli serta ulasan yang telah dilakukan proses Preprocessing di table preview Data labelling, dan juga kita dapat mengubah secara manual label tersebut menjadi positif atau negative apabila ada kesalahan atau kekeliruan dari labeling otomatis yang web proses. Setelah itu klik simpan hasil labelling. Berikut adalah tampilan halaman dari preprocessing data:



Gambar 4. 13 Halaman Preprocessing Data

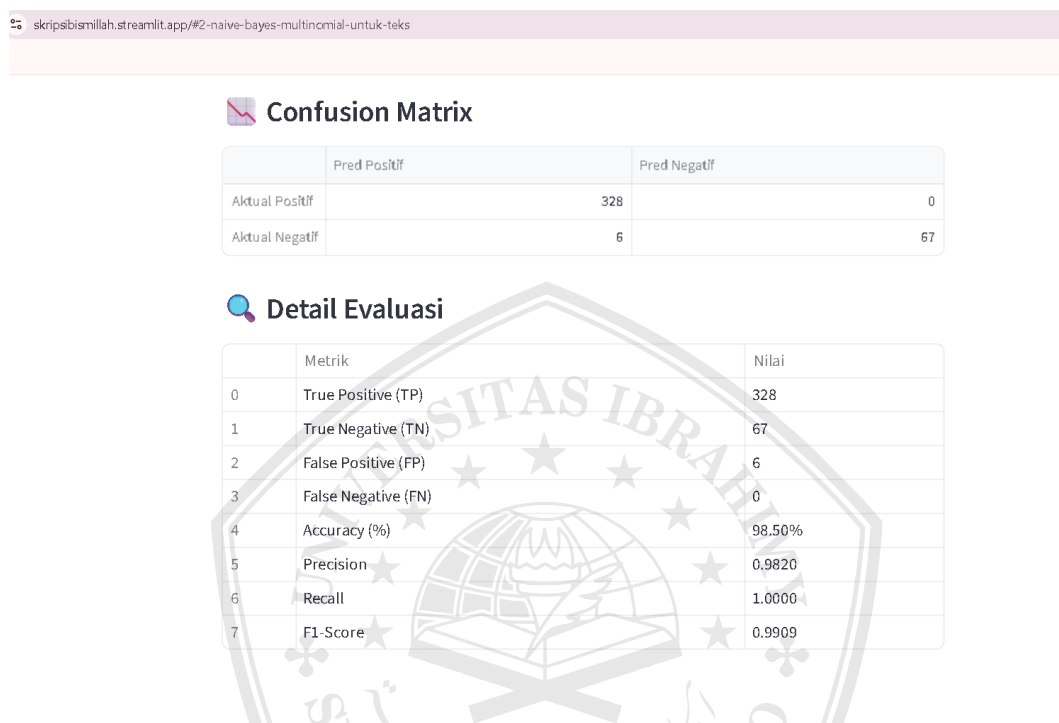
Lalu klik training dan evaluasi untuk melihat hasil prediksi Naïve bayes serta confusion Matrix lengkap dengan tp, tn, fp, fn, accuracy, precision, recall, f1 nya. Berikut adalah tampilan jendela tranning dan evaluasi dari website yang memunculkan hasil dasar teori naïve bayes lengkap dengan rumus dan penjelasan. Serta probabilitas awal kelas (prior) seperti di gambar 4.14 dibawah ini :



Gambar 4. 14 tampilan jendela tranning dan Evaluasi

Pada Halaman tersebut. Kita dapat melihat hasil prediksi serta confusion

matrix dari data yang telah diproses dengan *naïve bayes classifier* dan juga kita dapat melihat *Confusion matrix* serta detail evaluasinya, seperti gambar 4.15 dibawah ini:



Gambar 4.15 halaman CM dan Detail Evaluasi

Kedua versi ini dirancang untuk memberikan fleksibilitas bagi pengguna, baik yang ingin menjalankan aplikasi secara lokal maupun yang menginginkan kemudahan akses melalui web.

BAB V

PENUTUP

5.1 Kesimpulan

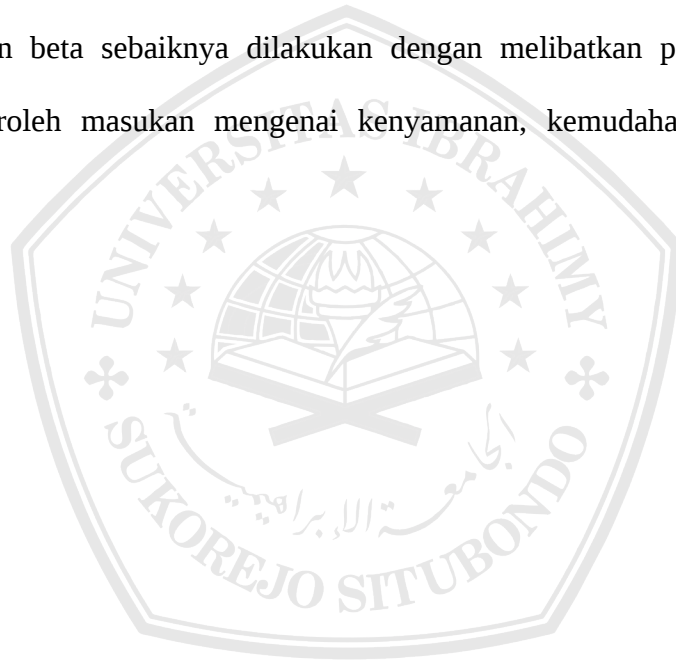
Berdasarkan Berdasarkan hasil penelitian, sistem analisis sentimen dengan metode Naïve Bayes mampu mengklasifikasikan ulasan pengguna aplikasi SIMANTAP di *Google Play Store* menjadi dua kategori, yaitu positif dan negatif. Model yang dibangun mencapai akurasi 97,86%, dengan nilai precision 97,74%, recall 100%, dan F1-score 98,86%. Tahap *Preprocessing* yang meliputi tokenisasi, case folding, *Stopword Removal*, dan *Stemming* terbukti berperan penting dalam meningkatkan kualitas data latih sehingga kinerja klasifikasi menjadi lebih optimal.

Seluruh proses, mulai dari pelatihan model, klasifikasi, hingga evaluasi dengan confusion matrix berhasil dijalankan sepenuhnya menggunakan bahasa pemrograman *Java* pada NetBeans tanpa melibatkan aplikasi pihak ketiga seperti RapidMiner. Hasil pengujian alfa juga menunjukkan bahwa fitur utama sistem berfungsi sesuai rancangan. Dengan demikian, sistem ini dapat digunakan sebagai alat bantu bagi pengembang aplikasi SIMANTAP untuk memahami sentimen pengguna sekaligus menjadi acuan dalam peningkatan kualitas layanan.

5.2 Saran

Untuk pengembangan selanjutnya, disarankan agar jumlah *Dataset* diperbanyak sehingga model memiliki cakupan yang lebih luas dan mampu mengenali variasi kata dengan lebih baik. Penelitian berikutnya juga bisa menambahkan kategori sentimen lain, misalnya positif, negatif, dan netral, agar

hasil analisis lebih detail. Selain itu, penggunaan algoritma lain atau gabungan beberapa algoritma (ensemble) dapat dipertimbangkan sebagai pembanding akurasi dan kinerja metode NaïveBayes. Dari sisi teknis, integrasi API *Google Play Store* sangat dianjurkan supaya proses pengambilan data dapat berjalan otomatis tanpa input manual. Sedangkan dari sisi antarmuka, tampilan GUI bisa dikembangkan menjadi lebih interaktif dan mudah dipahami, sehingga dapat digunakan juga oleh pengguna non-teknis. Terakhir, sebelum digunakan secara luas, pengujian beta sebaiknya dilakukan dengan melibatkan pengguna akhir untuk memperoleh masukan mengenai kenyamanan, kemudahan, dan fungsi sistem.



Daftar Pustaka

- [1] A. Dogtiev, "App Stores List," BusinessofApps. [Online]. Available: <https://www.businessofapps.com/guide/app-stores-list/#:~:text=The Google Play Store for,to a huge potential audience.>
- [2] I. S. K. Idris, Y. A. Mustofa, and I. A. Salihi, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM)," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 1, pp. 32–35, 2023, doi: 10.37905/jjee.v5i1.16830.
- [3] E. Martantoh and N. Yanih, "Implementasi Metode Naïve Bayes Untuk Klasifikasi Karakteristik Kepribadian Siswa Di Sekolah MTS Darussa'adah Menggunakan Php Mysql," *J. Teknol. Sist. Inf.*, vol. 3, no. 2, pp. 166–175, 2022, doi: 10.35957/jtsi.v3i2.2896.
- [4] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," *Processing*, vol., pp. 1–6, 2009.
- [5] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI/ICML-98 Work. Learn. Text Categ.*, pp. 41–48, 1998, doi: 10.1.1.46.1529.
- [6] Muhajirin, Risnita, and Asrulla, "11+Gm+82-92," *J. Genta Mulia*, vol. 15, no. 1, pp. 82–92, 2024.
- [7] A. M. Dawis *et al.*, *Pengantar Metodologi Penelitian*. 2023.
- [8] D. D. A. Yani, H. S. Pratiwi, and H. Muhardi, "Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace," *J. Sist. dan Teknol. Inf.*, vol. 7, no. 4, p. 257, 2019, doi: 10.26418/justin.v7i4.30930.

- [9] F. V. Sari and A. Wibowo, "Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi," *J. SIMETRIS*, vol. 10, no. 2, pp. 681–686, 2019.
- [10] A. Nurian, "Analisis Sentimen Ulasan Pengguna Aplikasi Google Play Menggunakan Naïve Bayes," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3s1, pp. 829–835, 2023, doi: 10.23960/jitet.v11i3s1.3348.
- [11] N. Nurzaman, N. Suarna, and W. Prihartono, "Analisis Sentimen Ulasan Aplikasi Threads Di Google Playstore Menggunakan Algoritma Naïve Bayes," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 1, pp. 967–974, 2024, doi: 10.36040/jati.v8i1.8708.
- [12] A. Asrumi, D. Suharijadi, A. D. Setiari, and D. P. Wulanda, *Analisis Sentimen dan Penggalan Opini*. Eureka Media Aksara, 2023.
- [13] A. R. Ismail, R. Bagus, and F. Hakim, "Implementasi Lexicon Based Untuk Analisis Sentimen Dalam Mengetahui Trend Wisata Pantai Di DI Yogyakarta Berdasarkan Data Twitter," vol. 1, no. 1, pp. 37–46, 2023.
- [14] S. Diwandari and A. P. Wibowo, "Modul Pemrosesan teks," *Univ. Teknol. Yogyakarta*, no. Query date: 2023-10-16 08:24:54, 2022, [Online]. Available: [http://eprints.uty.ac.id/12403/1/Modul PEMROSESAN TEKS_PRAKTIKUM_FINAL.pdf](http://eprints.uty.ac.id/12403/1/Modul_PEMROSESAN_TEKS_PRAKTIKUM_FINAL.pdf)
- [15] J. Enterprise, *Otodidak Perograman Python*, 1 st ed.,. Jakarta: PT. Elex Media Komputindo, 2017.
- [16] W. Gata and G. Gata, "Penerapan Bahasa Pemrograman Java Dalam," *Budi Luhur Inf. Technol.*, vol. 10, no. 1, pp. 80–87, 2013.

- [17] A. Sutanti, M. Komaruddin, P. Damayanti, and P. U. Studi Sistem Informasi Metro, “Rancang Bangun Aplikasi Perpustakaan Keliling Menggunakan Pendekatan Terstruktur,” *J. Ilm. Komput. dan Inform.*, vol. 9, no. 1, 2022.
- [18] M. Yanto, P. E. Bima, M. Bahron, and I. H. Ikasari, “Pemrograman Menggunakan Java NetBeans,” *BIIKMA Bul. Ilm. Ilmu Komput. dan Multimed.*, vol. 1, no. 3, pp. 367–377, 2023, [Online]. Available: <https://jurnalmahasiswa.com/index.php/biikma/article/view/555>
- [19] M. Rafi Nahjan, Nono Heryana, and Apriade Voutama, “Implementasi Rapidminer Dengan Metode Clustering K-Means Untuk Analisa Penjualan Pada Toko Oj Cell,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 101–104, 2023, doi: 10.36040/jati.v7i1.6094.
- [20] M. B. Nendya, B. Susanto, G. I. W. Tamtama, and T. J. Wijaya, “Desain Level Berbasis Storyboard Pada Perancangan Game Edukasi Augmented Reality Tap The Trash,” *Fountain Informatics J.*, vol. 8, no. 1, pp. 1–6, 2023, doi: 10.21111/fij.v8i1.8836.



PONDOK PESANTREN SALAFIYAH SYAFI'YAH SUKOREJO
UNIVERSITAS IBRAHIMY
PERPUSTAKAAN IBRAHIMY
NPP. 3512142F2006567
Jl. KHR. Syamsul Arifin No. 1-2 PO. Box. 2 Kode Pos. 68374 Phone (0338) 452666 Fax. (0338) 453068
SUMBEREJO BANYUPUTIH SITUBONDO JAWA TIMUR



SURAT KETERANGAN HASIL PEMERIKSAAN PLAGIASI

Yang bertanda tangan di bawah ini

Nama : Muhammad Ali Ridla, M.Kom.

Jabatan : Kepala Perpustakaan

Menyatakan dengan sebenarnya bahwa:

NIM : 2021503118

Nama : Naufal Arif Maulana

Fakultas : Sains dan Teknologi

Prodi : Teknologi Informasi

Kecamatan : Situbondo

Kabupaten : Situbondo

Provinsi : Jawa Timur

Judul Skripsi : Analisis sentimen ulasan pengguna terhadap aplikasi simantap di google play store menggunakan naive bayes.

Dengan dosen Pembimbing :

1. Firman Santoso, M.Kom.

2. Nur Azise, M.Kom.

Telah dilakukan cek plagiasi di Perpustakaan Universitas Ibrahimiyah dengan persentase plagiasi terakhir sebesar 26% .

Demikian Surat Keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Sukorejo, 30 Agustus 2025

Kepala Perpustakaan,



Muhammad Ali Ridla, M.Kom.



UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1
"Informasi Elektronik dan/atau Dokumen Elektronik
dan/atau hasil cetaknya merupakan alat bukti yang sah."