

Analisis Sentimen Opini Mahasiswa Terhadap Aplikasi Portal Mahasiswa UTY Menggunakan Metode Naïve Bayes Classifier

Agus Ardiyanto ¹, Enny Itje Sela ^{2*}

^{1,2*} Program Studi Informatika, Fakultas Sains & Teknologi, Universitas Teknologi Yogyakarta, Kabupaten Sleman, Provinsi Daerah Istimewa Yogyakarta.

Email: agus.5200411156@student.uty.ac.id ¹, ennysela@uty.ac.id ^{2*}

Histori Artikel:

Dikirim 12 Oktober 2023; *Diterima dalam bentuk revisi* 15 November 2023; *Diterima* 1 Desember 2023; *Diterbitkan* 10 Januari 2024. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STMIK Indonesia Banda Aceh.

Abstrak

Universitas Teknologi Yogyakarta (UTY) telah merilis Aplikasi Portal Mahasiswa UTY sejak tahun 2020 yang merupakan peningkatan dalam pelayanan untuk menunjang kegiatan akademik mahasiswa. Aplikasi Portal Mahasiswa UTY dikembangkan oleh puskom dari Universitas Teknologi Yogyakarta. Aplikasi Portal Mahasiswa UTY merupakan sistem yang dirancang dan dibangun untuk mengelola data-data yang berhubungan dengan informasi akademik yang meliputi data mahasiswa, data dosen, rekaman hasil kuliah, jadwal perkuliahan dan lain sebagainya. Kehadiran Aplikasi Portal Mahasiswa ini memunculkan berbagai komentar dari penggunaannya yaitu para mahasiswa UTY. Melihat masalah tersebut, maka peneliti melakukan penelitian terhadap opini mahasiswa terhadap Aplikasi Portal Mahasiswa UTY menggunakan Naïve Bayes Classifier. Penelitian ini menggunakan bahasa pemrograman Python. Berdasarkan hasil pembahasan didapatkan tingkat akurasi sebesar 93% pada proses training dan akurasi testing sekitar 65,2% dengan pembagian data latih dan test sebesar 70%:30% dari 150 data text opini. Pembuatan model ini mengalami overfitting, karena akurasi testing yang dihasilkan jauh lebih kecil daripada akurasi training.

Kata Kunci: Naive Bayes; Aplikasi Portal Mahasiswa UTY; Sentimen Analisis.

Abstract

University Technology of Yogyakarta (UTY) has released the UTY Student Portal Application since 2020 which is an improvement in services to support student academic activities. The UTY Student Portal application was developed by Puskom from Yogyakarta Technology University. The UTY Student Portal application is a system designed and built to manage data related to academic information which includes student data, lecturer data, lecture results records, lecture schedules and so on. The presence of this Student Portal Application has given rise to various comments from its users, namely UTY students. Seeing this problem, the researchers conducted research on student opinions regarding the UTY Student Portal Application using the Naïve Bayes Classifier. This research uses the Python programming language. Based on the results of the discussion, it was found that the accuracy level was 93% in the training process and the testing accuracy was around 65.2% with a distribution of training and test data of 70%:30% from 150 opinion text data. This model creation experienced overfitting, because the resulting testing accuracy was much smaller than the training accuracy.

Keyword: Naive Bayes; UTY Student Portal Application; Sentiment Analysis.

1. Pendahuluan

Pada era saat ini, perkembangan teknologi semakin pesat sehingga untuk melakukan sesuatu berkaitan dengan kegiatan sehari-hari mudah. Salah satunya adalah dalam bidang pendidikan, sekarang mahasiswa dipermudah untuk melakukan kegiatan dalam bidang pendidikan. Penerapan teknologi dalam pendidikan biasanya dilakukan untuk memudahkan mahasiswa melakukan absensi, mengecek krs, melihat khs hingga menerima informasi penting dari universitas. Saat ini terdapat aplikasi yang dapat membantu kita sebagai mahasiswa khususnya mahasiswa UTY dalam mendapatkan informasi melakukan absensi didalam universitas yaitu aplikasi Portal Mahasiswa UTY. Aplikasi ini digunakan untuk mempermudah mahasiswa khususnya mahasiswa UTY dalam melakukan kegiatan dalam universitas, misal melakukan krs hingga mengecek khs dari mahasiswa yang bersangkutan [1].

Universitas Teknologi Yogyakarta (UTY) telah merilis Aplikasi Portal Mahasiswa UTY sejak tahun 2020 yang merupakan peningkatan dalam pelayanan untuk menunjang kegiatan akademik mahasiswa. Aplikasi Portal Mahasiswa UTY dikembangkan oleh puskom dari pihak Universitas Teknologi Yogyakarta. Suatu aplikasi selalu memiliki kekurangan dan kelebihan masing-masing tak terkecuali aplikasi Portal Mahasiswa UTY, dimana hal tersebut dapat menimbulkan berbagai respon dari pengguna aplikasi khususnya mahasiswa seperti kepuasan dan kekecewaan terhadap aplikasi tersebut. Forum di internet seperti media sosial menjadi salah satu tempat untuk melontarkan kepuasan dan kekecewaan pengguna atau opini terhadap aplikasi tersebut. Hal tersebut dapat dijadikan bahan analisis sentimen terhadap aplikasi Portal Mahasiswa UTY [2].

Permasalahan-permasalahan diatas muncul seiring dengan berjalannya waktu dimana kebutuhan mahasiswa yang semakin meningkat terkait dengan bidang pendidikan di dalam universitas. Dampak permasalahan yang datang bagi developer aplikasi Portal Mahasiswa yaitu bagian PUSKOM dari Universitas Teknologi Yogyakarta beragam, mulai dari kekecewaan mahasiswa yang menggunakan aplikasi tersebut hingga penggunaan aplikasi tersebut tidak dipakai lagi oleh mahasiswa. Oleh karena itu, diperlukan analisis sentimen untuk mendapatkan data dari respon mahasiswa mengenai opini tentang aplikasi Portal Mahasiswa UTY yang nantinya digunakan untuk evaluasi bagi pengembangan aplikasi selanjutnya.

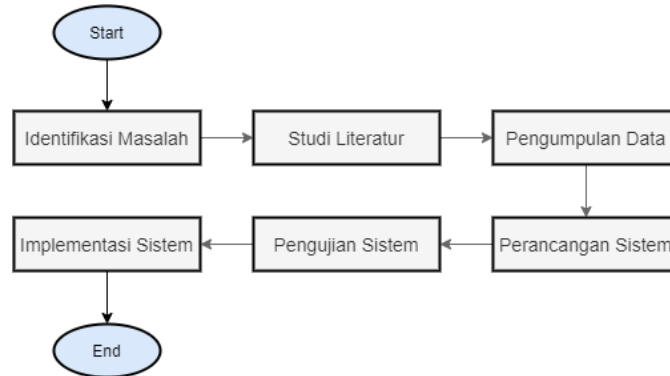
Penelitian yang dilakukan oleh [3], menunjukkan bahwa mereka mendapatkan Data yaitu data training yang terdiri dari 1179 kata yang telah diketahui kategorinya masing masing terdiri dari 522 kata negatif, 513 kata positif, dan 144 kata netral. Kemudian mereka mendapatkan akurasi klasifikasi yang dilakukan oleh aplikasi menghasilkan nilai akurasi tertinggi sebesar 83,3% dan accuracy, precision dan recall 83,3%, 65%, dan 63% dengan jumlah data latih 825 dan data uji sebesar 354. Penelitian ini kemudian menjadi acuan yang digunakan bahwa kita perlu melakukan analisis sentimen untuk mengetahui sejauh mana respon pengguna tentang aplikasi Portal Mahasiswa UTY. Penelitian ini kemudian menjadi acuan yang digunakan bahwa kita perlu melakukan analisis sentimen untuk mengetahui sejauh mana respon pengguna tentang aplikasi Portal Mahasiswa UTY.

Dengan adanya beberapa permasalahan diatas, mendorong penulis untuk melakukan penelitian dengan menganalisis sentimen opini mahasiswa terhadap aplikasi Portal Mahasiswa UTY. Analisis Sentimen digunakan untuk menemukan informasi berharga yang dibutuhkan dari data yang tidak terstruktur, sehingga diharapkan pada penelitian ini dapat diketahui sentimen opini mahasiswa terhadap aplikasi Portal Mahasiswa UTY. Didalam penelitian ini, akan dibahas tahapan yang dilalui untuk melakukan proses analisis sentimen terhadap komentar tentang aplikasi Portal Mahasiswa UTY. Dimulai dari tahap preprocessing sampai tahap analisis sentimen dengan metode *Naïve Bayes* serta menguji contoh data untuk mengetahui masuk kedalam kelas sentimen positif, netral atau negatif dan mengetahui tingkat akurasinya.

2. Metode Penelitian

Metode penelitian merupakan gambaran langkah – langkah yang dilakukan penelitian untuk mengumpulkan data serta menyelesaikan permasalahan dalam penelitian tersebut [4]. Penelitian ini

menggunakan algoritma Naïve Bayes Classifier dalam pengembangannya serta menggunakan dataset opini mahasiswa terhadap Aplikasi Portal Mahasiswa UTY. Tahap penelitian bisa dilihat pada gambar 1.

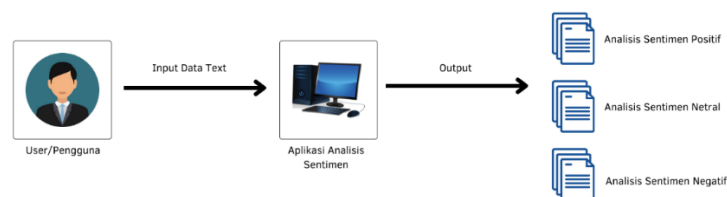


Gambar 1. Tahap Penelitian

Berdasarkan Gambar 1 Tahapan penelitian dapat dilihat penelitian ini dimulai dari identifikasi masalah yaitu peneliti merumuskan masalah didalam kehidupan sehari hari dan menentukan tujuan yang ingin dicapai agar penelitian ini memiliki tujuan akhir yang jelas. Kemudian studi literatur yaitu peneliti mencari referensi dari berbagai sumber seperti membaca artikel dan jurnal yang terkait dengan masalah yang didapat kemudian mencari kekurangan atau kelemahan yang ada pada artikel tersebut. Setelah itu adalah tahap pengumpulan data yaitu peneliti mengumpulkan data teks yang bersumber dari kuisioner yang diisi oleh responden. Tahap selanjutnya adalah proses perancangan sistem, perancangan sistem ini yaitu berupa perancangan sistem yang dikembangkan yaitu termasuk dengan pengolahan data yang sudah dikumpulkan pada tahap sebelumnya. Setelah itu adalah tahap pengujian sistem yang dirancang, pada tahap ini dilakukan pengujian dari sistem hingga mendapatkan hasil yang memuaskan. Kemudian implemtasi sistem, dalam tahap ini adalah pengimplemtasian sistem menjadi sebuah aplikasi yang bisa digunakan dalam penelitian ini atau menjadi hasil dari penelitian ini yaitu Aplikasi Analisis Sentimen Opini Mahasiswa Terhadap Aplikasi Portal Mahasiswa UTY.

2.1 Arsitektur Model

Dalam penjabaran sistem yang berjalan diatas Peneliti mengusulkan sebuah sistem untuk pengklasifikasian data teks opini mahasiswa terhadap Aplikasi Portal Mahasiswa menggunakan Algoritma Naïve Bayes Classifier. Arsitektur sistem dapa dilihat pada gambar 2 di bawah ini.



Gambar 2. Arsitektur Model

Dari Gambar 2. Arsitektur Model dapat dijelaskan untuk pengguna dari sistem ini nantinya sesuai permasalahan yang ada adalah para pengembang aplikasi Portal Mahasiswa UTY untuk lebih memudahkan dalam mengenali dalam menerima sebuah opini atau masukan dari pengguna yang nantinya bisa digunakan untuk memperbaiki aplikasi Portal Mahasiswa UTY agar lebih baik. Sistem ini Menggunakan teknologi sistem komputer dengan algoritma *naïve bayes*, inputannya adalah sebuah text dari opini mahasiswa. Setelah data text tersebut dimasukan ke dalam sistem kemudian data tersebut akan masuk ke dalam proses *Preprocessing* yaitu proses pembersihan data yang tidak perlu sebelum masuk ke dalam ke tahapan klasifikasi *naïve bayes*. Setelah melewati proses *Preprocessing* data

tersebut masuk ke dalam klasifikasi *naïve bayes*. Hasil akhir dari sistem adalah 3 kelas dimana nantinya terdiri dari kelas positif, netral dan negatif.

2.2 Data Penelitian

1) Data yang Diperoleh

Memudahkan dalam mengenali dalam menerima sebuah opini atau masukan dari pengguna yang nantinya bisa digunakan untuk memperbaiki aplikasi Portal Mahasiswa UTY agar lebih baik. Sistem ini Menggunakan teknologi sistem komputer dengan algoritma *naïve bayes*, inputannya adalah sebuah text dari opini mahasiswa. Setelah data text tersebut dimasukan ke dalam sistem kemudian data tersebut akan masuk ke dalam proses *Preprocessing* yaitu proses pembersihan data yang tidak perlu sebelum masuk ke dalam ke tahapan klasifikasi *naïve bayes*. Setelah melewati proses *Preprocessing* data tersebut masuk ke dalam klasifikasi *naïve bayes*. Hasil akhir dari sistem adalah 3 kelas dimana nantinya terdiri dari kelas positif, netral dan negatif.

Tabel 1. Data Opini Mahasiswa

Nama	NPM	Opini Mahasiswa
Muhammad Ilham Abdu Sholeh	5200411164	Overall bagus, tapi di ponsel saya aplikasi SIA masih ada beberapa bug yaitu Force Close saat mengklik fitur Kunci Biometrik dan visual nilai yang terlihat kurang memuaskan
Abid Taufiqur Rohman	5200411064	Tampilan UI dari aplikasi sudah sangat menarik dan sangat user friendly. Untuk menunya sudah lengkap dan dapat membantu mahasiswanya.. mungkin kedepannya dapat ditambah fitur login dengan face lock.
Iqbal	136	aplikasi yang sangat membantu untuk presensi dan melihat jadwal mata kuliah
Saepul Reza Alparizi	5200411157	interface aplikasi bagus dan simpel tidak terlalu ramai
Muhammad Ifdhal Rumalutur	5200411176	Aplikasi ini mempunyai design interface menarik sehingga memudahkan pengguna (user friendly), fitur fitur di dalamnya juga sudah lengkap memuat semua informasi terkait mahasiswa dan juga universitas. namun kekurangannya kunci privasi saat membuka aplikasi hanya berupa pin, belum ada fitur berupa fingerprint untuk lebih mempermudah masuk ke dalam aplikasi.
Muhammad Naufal Alfa'iz Musthofa	5200411135	Untuk UI/UX sudah cukup memuaskan dan untuk keseluruhan sudah cukup baik. Namun sayangnya aplikasi ini masih belum dapat mengirimkan reminder mengenai jadwal ke device mahasiswa. Selain itu, hanya masih tersedia untuk android.
Alfito Herdiansyah	5200411361	Pada jadwal kuliah tidak muncul jadwal pada hari itu

2) Prosedur Pengumpulan Data

Data pada penelitian ini berupa data text, Data ini diambil menggunakan kuisioner yang diisi oleh responden khususnya adalah pengguna aplikasi Portal Mahasiswa UTY. Data tersebut dikumpulkan dalam periode waktu 11 oktober 2022 sampai 1 november 2022.

3) Alat Khusus

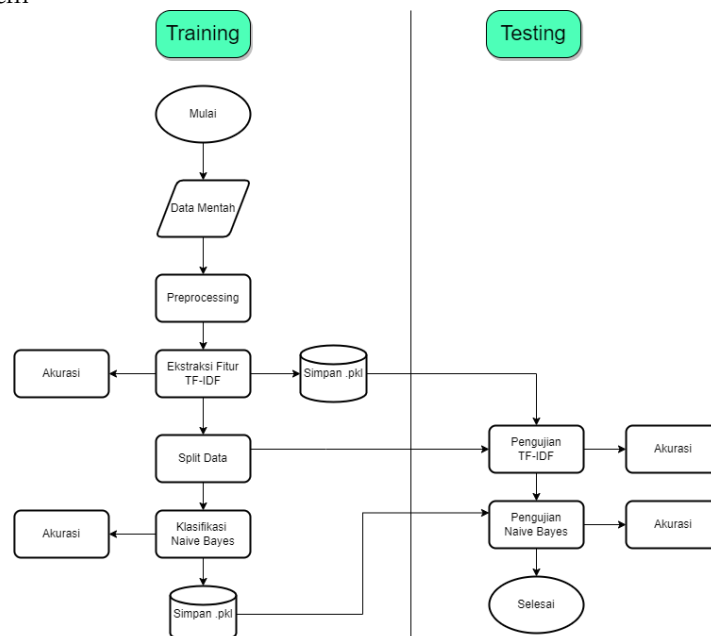
Alat khusus yang digunakan peneliti pada penelitian ini adalah sebuah situs yaitu *Google Form*. *Google Form* digunakan untuk mengumpulkan Data dalam penelitian ini yaitu data teks yang berupa kuisisioner. Kuisisioner ini dibagikan dan diisi oleh responden yaitu pengguna Aplikasi Portal Mahasiswa UTY. Link *Google Form* yang digunakan untuk mengumpulkan data pada penelitian ini adalah berikut <https://forms.gle/Q37Ah5KsoothiWd88>.

4) Labelling Manual

Data yang diperoleh merupakan data mentah yang didapatkan melalui kuisisioner opini mahasiswa terhadap aplikasi Portal Mahasiswa UTY. Langkah selanjutnya adalah memeberikan *class* sentimennya masing-masing secara manual yang nantinya digunakan untuk target class prediksi.

2.3 Desain system

a) *Flowchart* Sistem



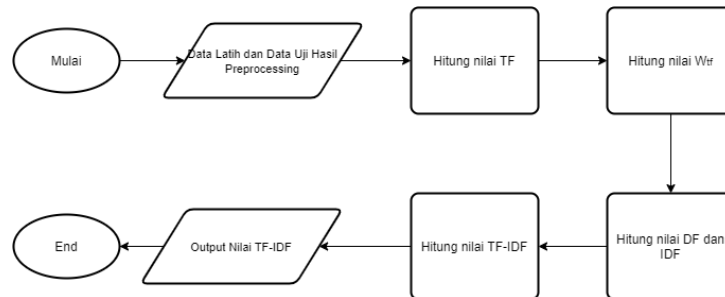
Gambar 3. *Flowchart* Sistem

Pada Gambar 3 *Flowchart* Sistem dijelaskan bahwa proses dari perancangan ini yaitu dimana proses yang dilakukan pertama adalah pengambilan data. Pengambilan data ini didapatkan dari kuisisioner yang diisi oleh para responden yaitu para pengguna aplikasi Portal Mahasiswa UTY. Data tersebut kemudian menjadi data mentah yaitu data yang sudah didapat sebelum dilakukan proses preprocessing. Data yang ada kemudian dilakukan preprocessing sehingga data yang didapat adalah data yang sudah bersih. Kemudian setelah melewati preprocessing dilakukan ekstraksi fitur menggunakan *TF-IDF* dan disimpan model *TF-IDF* yang didapat. Lalu data yang sudah dilakukan ekstraksi fitur *TF-IDF* kemudian dilakukan split data dengan rasio 70 persen untuk data uji dan 30 persen untuk data latih. Kemudian dilakukan klasifikasi menggunakan metode *naïve bayes classifier* yang modelnya juga disimpan. Kemudian tahap testing yaitu melakukan klasifikasi menggunakan model yang sudah dibuat menggunakan ekstraksi fitur *TF-IDF* dan menggunakan algoritma *Naïve Bayes Classifier*.

b) *Flowchart* TF-IDF

Data yang telah melalui tahap preprocessing harus berbentuk numerik. Untuk mengubah data tersebut menjadi numerik yaitu menggunakan metode pembobotan *TF-IDF*. Metode *TF-IDF* menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di

dalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen [6].

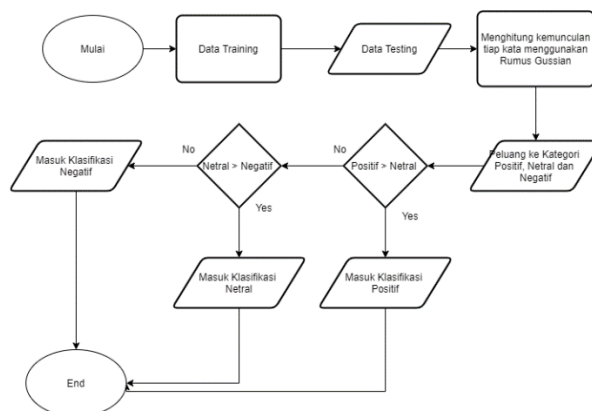


Gambar 4. Ekstraksi Fitur TF-IDF

Pada Gambar 3 Ekstraksi Fitur *TF-IDF* dijelaskan bahwa proses dari pembobotan ini dilakukan menggunakan data yang sudah bersih baik itu data training maupun data testing. Proses dari *TF-IDF* adalah hitung nilai TF, hitung nilai *Wtf*, hitung nilai DF IDF dan menghitung *TF-IDF*. Kemudian output dari proses pembobotan ini adalah nilai dari *TF-IDF* yang nantinya akan digunakan pada proses klasifikasi naïve bayes.

c) *Flowchart* Klasifikasi Naïve Bayes

Setelah didapatkan nilai dari proses *TF-IDF* selanjutnya adalah dilakukan klasifikasi. Naïve Bayes Classifier sendiri salah satu algoritma machine learning yang banyak digunakan untuk klasifikasi. Naive Bayes adalah sekumpulan algoritma yang didasarkan pada Teorema Bayes. Naïve Bayes adalah asumsi yang sangat kuat (naif) akan independensi dari masing-masing kondisi atau kejadian. Naive Bayes Classifier merupakan salah satu algoritma yang sederhana namun memiliki kemampuan dan akurasi yang tinggi dan termasuk dalam metode *machine learning*. Berdasarkan kompleksitasnya Naive Bayes merupakan salah satu algoritma paling sederhana dalam menerapkan aturan Bayes disertai beberapa keunggulan yaitu sangat efisien, membutuhkan data uji yang sedikit, mudah untuk diimplementasikan dan memiliki akurasi relatif tinggi [7]. Selain itu, *Naive Bayes* juga memiliki kekurangan diantaranya akurasi akan berkurang jika atribut yang menjadi parameter dalam klasifikasi tidak independen atau nonparametric-continuous [8].



Gambar 5. *Flowchart* Klasifikasi Naive Bayes

Pada Gambar 5 *Flowchart* Klasifikasi Naive Bayes dijelaskan bahwa proses dari klasifikasi ini

bermula ketika sebuah sistem klasifikasi menerima data training dan data testing yang sudah melalui preprocessing data dan pembobotan *TF-IDF*. Proses dari Klasifikasi Naïve Bayes adalah yang pertama adalah menghitung kemunculan kata menggunakan rumus Gaussian dan mengambil bobot kata dari pembobotan *TF-IDF*. Kemudian klasifikasi menghitung peluang kalimat tersebut masuk ke dalam kelas sentimen positif, netral atau negatif.

3. Hasil dan Pembahasan

3.1 Alat Penelitian

1) Spesifikasi Hardware

Perangkat keras yang digunakan untuk mengoperasikan Sistem Analisis Sentimen Opini Mahasiswa Terhadap Aplikasi Portal Mahasiswa UTY Menggunakan Metode Naïve Bayes Classifier adalah:

- a) Laptop Asus X550VX
- b) Processor Processor Intel Core i7 6700hq
- c) RAM 8192 MB
- d) Hardisk 1 TB dan SSD 120 GB
- e) NVIDIA GeForce 950M

2) Spesifikasi Software

Perangkat keras yang digunakan untuk mengoperasikan Sistem Analisis Sentimen Opini Mahasiswa Terhadap Aplikasi Portal Mahasiswa UTY Menggunakan Metode Naïve Bayes Classifier adalah: Perangkat lunak yang digunakan dalam membangun Sistem Analisis Sentimen Opini Mahasiswa Terhadap Aplikasi Portal Mahasiswa Uty Menggunakan Metode Naïve Bayes Classifier ini adalah:

- a) *Visual Studio Code*: digunakan untuk membuat kode program.
- b) *Anaconda*: sebuah aplikasi yang sudah tergabung dengan bahasa pemrograman python yang nantinya digunakan untuk download library yang digunakan.
- c) *Ms office*: digunakan dalam pembuatan laporan dan artikel.
- d) *Draw io*: digunakan dalam pembuatan *desain prototype* dan *desain flowchart*.

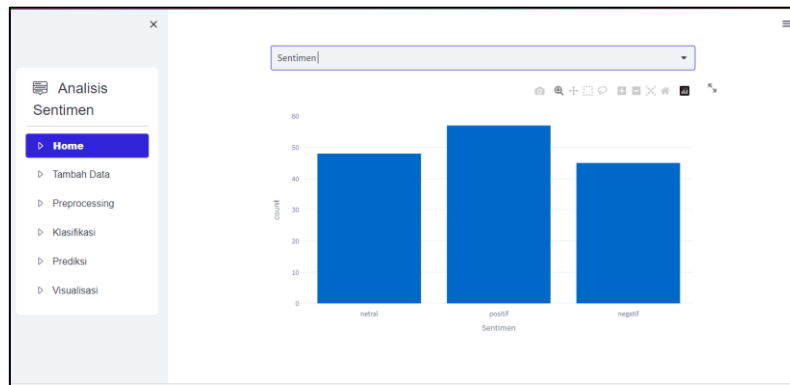
3.2 Data Exploration

Pada tahap ini peneliti melakukan *data exploration* atau eksplorasi data menggunakan library pandas, numpy, matplotlib, dan seaborn untuk melakukan eksplorasi data dan menggunakan streamlit untuk GUI (*Graphical User Interface*). Data yang diperoleh melalui kuisisioner tersebut dilakukan *labelling* manual terlebih dahulu sebelum dilakukan klasifikasi. Dimana dari proses tersebut menambahkan kolom baru yaitu "Sentimen". Hasil eksplorasi data yang telah dilakukan bahwa data yang didapatkan melalui kuisisioner adalah sejumlah 150 data dan 5 atribut kolom dengan terbagi menjadi beberapa kelas yaitu kelas Negatif, Netral dan Positif. Data tersebut dilakukan preprocessing awal yaitu dengan menghapus kolom yang tidak dibutuhkan dalam hal ini terdapat beberapa kolom yang tidak dibutuhkan seperti "Cap Waktu", "Nama", dan "NPM". Pada penelitian ini berfokus pada klasifikasi text dan hanya mengambil data pada kolom "Opini" dan "Sentimen" sebagai targetnya.

Proses pada data exploration ini adalah mengeksplorasi data yang ada pada tahap ini dilakukan pengecekan data yang kosong dan data yang duplikat bahwa dinyatakan tidak adanya indikasi dari data yang kosong dan data yang duplikat [9]. Target klasifikasi model ini terdapat pada kolom "Sentimen" yang terdiri dari beberapa label yaitu label negatif, label netral dan label positif. Pada Tabel 2 jumlah distribusi data berdasarkan label yaitu sebanyak 45 data untuk label negatif, 48 data untuk label netral, dan 57 data untuk label positif. Implementasi visualisasi Banyaknya data tersebut menggunakan library streamlit bisa dilihat pada gambar 5.

Tabel 2. Jumlah Dataset berdasarkan kelas

Kelas	Jumlah
Negatif	45 Data
Netral	48 Data
Positif	57 Data



Gambar 6. Visualisasi Data

Selanjutnya setelah dilakukan pengecekan dan pengamatan dari penulis pada dataset yang diperoleh terdapat beberapa data teks yang masih belum bersih, seperti data teks yang mengandung *emoticon*, hashtag (@), tag (#), kosakata yang disingkat, kalimat dengan huruf besar semua dan kosakata yang berawalan huruf besar, tanda baca, kata singkatan, dan kata yang mengandung angka. Data tersebut diperlukan proses *Text Preprocessing* untuk dapat digunakan dalam klasifikasi.

3.3 Preprocessing Text

Preprocessing Text merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Sebuah teks yang harus dipisahkan, hal ini dapat dilakukan dalam beberapa tingkatan yang berbeda. Selain itu pada tahapan ini keberadaan digit angka, huruf kapital atau karakter-karakter yang lainnya dihilangkan dan dirubah [10]. Pada tahap ini peneliti melakukan beberapa langkah *Preprocessing text* dimana dalam tahap ini dilakukan *preprocessing* data text opini yang sudah dimasukan untuk menghasilkan data text yang bersih sebelum masuk kedalam pembuatan model dan evaluasi. Beberapa tahap *Preprocessing text* adalah sebagai berikut.

1) Casefolding

Proses yang pertama adalah casefolding yaitu mengubah seluruh teks menjadi lower case/huruf kecil [11]. Data yang dipreprocessing adalah hanya data opini maka dalam kode program yang diaplikasikan adalah preprocessing hanya untuk kolom opini.

2) Data Cleansing

Setelah mendapatkan hasil dari case folding, proses selanjutnya yang dilalui adalah data *cleansing*. Pada tahapan ini teks yang diperoleh dari hasil *case folding* akan dibersihkan dengan menghapus seluruh link url, angka, tanda baca, dan karakter huruf [12].

3) Remove Emoticon

Proses selanjutnya dalam preprocessing adalah remove emoticon, yaitu menghapus semua emoticon baik itu emoticon menggunakan tanda baca atau emoticon berupa gambar. Pada bagian ini menggunakan library emoji untuk menghapus emoticon.

4) Remove Stopwords

Proses selanjutnya dalam tahapan *preprocessing* adalah *remove stopwords*. Tahapan *remove stopwords* merupakan tahapan filterisasi dari data yang telah melalui proses data *cleansing* dan *remove emoticon*. Kata yang akan dihapus merupakan kata yang tidak memiliki arti, sehingga dapat mengurangi

fitur term dalam proses pembobotan [13]. Tahapan stopword removal pada *pre-processing* penelitian ini menggunakan *stopwords* dari *library sastrawi*.

5) Stemming Data

Proses selanjutnya dalam tahapan *preprocessing* adalah *stemming data*. Pada tahapan stemming adalah mengubah data *text* menjadi *root* atau kata dasar [14]. Pada tahapan ini menggunakan *library stemming sastrawi* dalam mengubah kata yang memiliki imbuhan *prefix*, *infix*, maupun *suffix* menjadi kata dasar.

6) Tokenisasi Data

Tahapan selanjutnya dalam *preprocessing* adalah tokenisasi. Pada tahapan ini dilakukan pemecahan kalimat menjadi *term* atau kata atau tahap pemotongan string input berdasarkan tiap kata yang menyusunnya [15].

7) Remove Punctuation

Tahapan selanjutnya dalam *preprocessing* adalah *remove punctuation* yang merupakan tahapan akhir dari *Pre-processing*. *Remove punctuation* adalah proses dimana sistem akan menghilangkan tanda baca atau simbol yang ada dalam dataset [16]. Tanda baca atau simbol ini dihapus karena tidak berpengaruh pada hasil sentimen analisis. Pada tahap *remove punctuation* ini menggunakan *library string*.

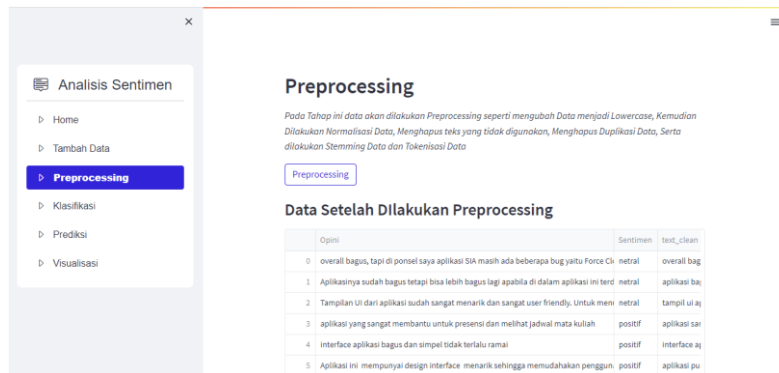
Setelah beberapa tahap tersebut dilakukan menghasilkan sebuah data text yang sudah bersih dan siap digunakan dalam proses klasifikasi. Kemudian tabel dibawah ini adalah data yang sudah mengalami proses *preprocessing*. Implementasi *GUI (Graphical User Interface)* streamlit pada proses *preprocessing text* bisa dilihat pada gambar 6.

Tabel 3. Hasil *Preprocessing*

Sebelum Preprocessing	Setelah Preprocessing
overall bagus, tapi di ponsel saya aplikasi SIA masih ada beberapa bug yaitu Force Close saat mengklik fitur Kunci Biometrik dan visual nilai yang terlihat kurang memuaskan	overall bagus di ponsel aplikasi sia ada beberapa bug force close klik fitur kunci biometrik visual nilai lihat kurang muas
Tampilan UI dari aplikasi sudah sangat menarik dan sangat user friendly. Untuk menunya sudah lengkap dan dapat membantu mahasiswanya.. mungkin kedepannya dapat ditambah fitur login dengan face lock.	tampil ui aplikasi sangat tarik sangat user friendly menu lengkap bantu mahasiswa mungkin depan dapat tambah fitur login face lock
aplikasi yang sangat membantu untuk presensi dan melihat jadwal mata kuliah	aplikasi sangat bantu presensi lihat jadwal mata kuliah
interface aplikasi bagus dan simpel tidak terlalu ramai	interface aplikasi bagus simpel terlalu ramai
Aplikasi ini mempunyai design interface menarik sehingga memudahkan pengguna (user friendly), fitur fitur di dalamnya juga sudah lengkap memuat semua informasi terkait mahasiswa dan juga universitas. namun kekurangannya kunci privasi saat membuka aplikasi hanya berupa pin, belum ada fitur berupa fingerprint untuk lebih mempermudah masuk ke dalam aplikasi.	aplikasi punya design interface tarik memudahkan guna user friendly fitur fitur dalam sudah lengkap muat semua informasi kait mahasiswa juga universitas kurang kunci privasi buka aplikasi rupa pin ada fitur rupa fingerprint lebih mudah masuk dalam aplikasi
Untuk UI/UX sudah cukup memuaskan dan untuk keseluruhan sudah cukup baik. Namun sayangnya aplikasi ini masih belum dapat mengirimkan reminder mengenai jadwal ke device mahasiswa. Selain itu, hanya masih	ui ux cukup muas seluruh cukup baik sayang aplikasi masih dapat kirim reminder kena jadwal device mahasiswa itu masih sedia untuk android

tersedia untuk android.
Pada jadwal kuliah tidak muncul jadwal pada
hari itu

jadwal kuliah muncul jadwal hari



Gambar 7. Implementasi Preprocessing data

3.4 Ekstraksi Fitur

Kemudian selanjutnya adalah halaman Klasifikasi, setelah Data yang diperoleh sudah bersih kemudian masuk ke dalam Halaman Klasifikasi. Dalam proses klasifikasi ada beberapa proses yang dilakukan yang pertama adalah menghitung TF-IDF yang akan digunakan nantinya dalam pengklasifikasian menggunakan Naive Bayes Classifier. Untuk kode program dalam menghitung TF-IDF bisa dilihat pada Gambar 9 Kode Program Hitung TF-IDF.

```
1 # Mengambil kolom "Opini" dan "Sentimen"
2 opinions = df['text_clean'].astype(str).tolist()
3 sentiments = df['Sentimen'].tolist()
4
5 # Membuat daftar kata unik dari seluruh dokumen
6 unique_words = list(set(" ".join(opinions).split()))
7
8 # Menghitung Term Frequency (TF)
9 tf_dict = []
10 for opinion in opinions:
11     word_count = {}
12     words = opinion.split()
13     total_words = len(words)
14     for word in words:
15         word_count[word] = word_count.get(word, 0) + 1 /
16 total_words
17     tf_dict.append(word_count)
18
19 # Menghitung Document Frequency (DF)
20 df_dict = {}
21 for word in unique_words:
22     for opinion in opinions:
23         if word in opinion:
24             df_dict[word] = df_dict.get(word, 0) + 1
25
26 # Menghitung Inverse Document Frequency (IDF)
27 idf_dict = {}
28 total_documents = len(opinions)
29 for word in unique_words:
30     df = df_dict.get(word, 0)
31     idf_dict[word] = math.log(total_documents / (df + 1))
32
33 # Menghitung TF-IDF
34 tfidf_dict = []
35 for tf in tf_dict:
36     tfidf = {}
37     for word, tf_score in tf.items():
38         tfidf[word] = tf_score * idf_dict[word]
39     tfidf_dict.append(tfidf)
40
41 for tfidf in tfidf_dict:
42     for word in unique_words:
43         if word not in tfidf:
44             tfidf[word] = 0.0
45
46 joblib.dump(tfidf_dict, 'tfidf_model_baru.pkl')
47 # Simpan nilai IDF dalam file Terpisah
48 joblib.dump(idf_dict, 'idf_dict.pkl')
```

Gambar 8. Kode Program Ekstraksi Fitur

Berdasarkan diatas, dalam melakukan perhitungan TF-IDF ada beberapa Tahap yaitu adalah yang pertama menghitung *Term Frequency*, menghitung *Document Frequency*, menghitung *Inverse Document Frequency* dan baru menghitung TF-IDF. Setelah proses menghitung TF-IDF selesai kemudian menyimpan Model yang digunakan dalam bentuk PKL. Kemudian Untuk Gambar Implementasi halaman Klasifikasi dan Hasil TF-IDF bisa dilihat pada Gambar 9 dibawah ini.



Gambar 9. Implementasi Ekstraksi Fitur TF-IDF

Setelah data melewati proses perhitungan TF-IDF dan menghasilkan data yang sudah mendapatkan nilai vektorisasi yang akan digunakan dalam klasifikasi menggunakan metode naïve bayes classifier. Langkah selanjutnya adalah membagi data atau split data kedalam data latih dan data uji. Pada proses kali ini bertujuan untuk membagi data latih dan data uji pada metode klasifikasi ini. Sebelum membagi data latih dan data uji bisa dilihat pada Gambar 10 Kode Program Split Data, juga dilakukan pembuatan feature dan target yang akan dilakukan split data atau X dan y. Feature itu sendiri merupakan data bersih yang sudah dilakukan *preprocessing* dan sudah dilakukan perhitungan TF-IDF dan Sentimen sebagai target yang nantinya akan menentukan target data tersebut termasuk dalam sentimen negatif, netral atau positif.

```
1 # Memisahkan fitur (TF-IDF) dan target (Sentimen)
2 X = df.drop('Sentimen', axis=1)
3 y = df['Sentimen']
4
5 # Memisahkan data menjadi subset pelatihan dan uji
6 X_train, X_test, y_train, y_test = train_test_split(X,
7 y, test_size=0.3, random_state=199)
```

Gambar 10. Kode Program Split Data

Berdasarkan Gambar 10 Kode Program Split Data, dalam pembagian data training dan data testing dalam penelitian ini adalah sebanyak 70 persen data training dan sebanyak 30 persen data testing. Pembagian data train dan data testing dalam penelitian ini menggunakan random state 199. Random state digunakan sebagai acuan untuk melakukan split data agar data yang dilakukan split data tidak selalu berubah yang masuk ke data training dan ke data testing.

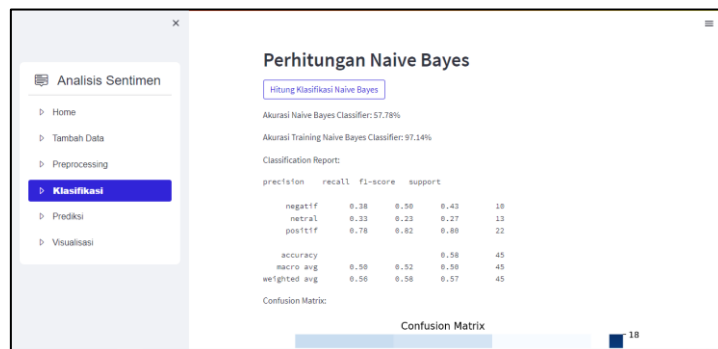
3.5 Klasifikasi

Proses selanjutnya adalah proses pembuatan model klasifikasi *naïve bayes* evaluasi model klasifikasi dan hasil dari akurasi naïve bayes. Berdasarkan Gambar 11 Kode Program Klasifikasi Naive Bayes, data yang sudah dibagi kedalam data uji dan data latih kemudian dilakukan pembuatan model klasifikasi naïve bayes.

```
1 # Inisialisasi model Naive Bayes Classifier
2 nb_classifier = MultinomialNB()
3
4 # Melatih model menggunakan data pelatihan
5 nb_classifier.fit(X_train, y_train)
6
7 # Melakukan prediksi pada data uji
8 y_pred = nb_classifier.predict(X_test)
9
10 # Menghitung akurasi
11 accuracy = accuracy_score(y_test, y_pred)
12 st.write("Akurasi Naive Bayes Classifier:
13 {:.2f}%".format(accuracy * 100))
14
15 # Menghitung akurasi training
16 y_train_pred = nb_classifier.predict(X_train)
17 train_accuracy = accuracy_score(y_train, y_train_pred)
18 st.write("Akurasi Training Naive Bayes Classifier:
19 {:.2f}%".format(train_accuracy * 100))
20 joblib.dump(nb_classifier, 'naive_bayes_model.pkl')
```

Gambar 11. Kode Program Klasifikasi Naive Bayes

Berdasarkan Gambar 11 Kode Program Klasifikasi Naive Bayes, pada bagian save model pada kode program ini menggunakan library joblib dari sklearn untuk save model dan berekstensi.pkl. Kemudian untuk implementasi hasil evaluasi model dan akurasi data testing dalam tahap ini bisa dilihat pada Gambar 11 Implementasi Klasifikasi.



Gambar 12. Kode Program Klasifikasi Naive Bayes

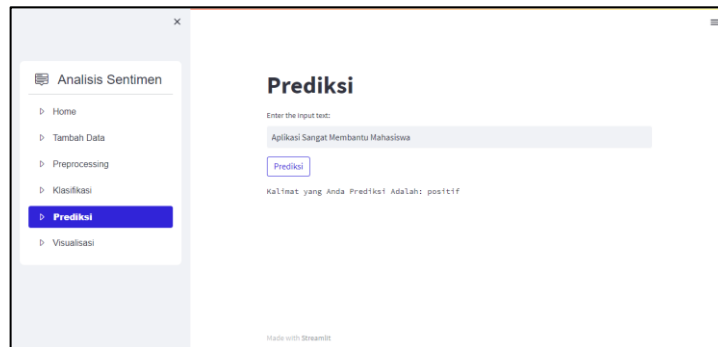
3.6 Prediksi

Tahap selanjutnya adalah prediksi yaitu proses yang digunakan oleh user atau pengguna untuk melakukan prediksi. Dalam prediksi ini user dapat memprediksi kata menggunakan model yang sudah dibuat pada proses-proses sebelumnya. Untuk melakukan prediksi dalam program ini bisa dilihat pada Gambar 13 Kode Program Prediksi.

```
1 # Muat model TF-IDF dari file terpisah
2 tfidf_model = joblib.load('tfidf_model.pkl')
3
4 # Muat nilai IDF dari file terpisah
5 idf_dict = joblib.load('idf_dict.pkl')
6
7 # Muat model Naive Bayes dari file terpisah
8 naive_bayes_model = joblib.load('naive_bayes_model.pkl')
9
10 # Preprocess the input text to match the TF-IDF features
11 input_tfidf = []
12 words = input_text.split()
13 total_words = len(words)
14 word_count = {}
15 for word in words:
16     word_count[word] = word_count.get(word, 0) + 1 /
17 total_words
18 input_tfidf.append(word_count)
19
20 # Transform the input text TF-IDF using the IDF values from
21 the training data
22 input_tfidf_transformed = []
23 for tf in input_tfidf:
24     tfidf = {}
25     for word, tf_score in tf.items():
26         if word in tfidf_model[1]:
27             tfidf[word] = tf_score * idf_dict[word]
28         else:
29             tfidf[word] = 0.0
30     input_tfidf_transformed.append(tfidf)
31
32 # Create a DataFrame from the transformed input text TF-IDF
33 input_df = pd.DataFrame(input_tfidf_transformed,
34 columns=tfidf_model[1])
35
36 # Fill missing values with 0
37 input_df = input_df.fillna(0)
38
39 # Make predictions using the loaded Naive Bayes model
40 predictions = naive_bayes_model.predict(input_df.values)
41
42 # Display the predictions
43 st.text(f"Kalimat yang Anda Prediksi Adalah:
44 {predictions[0]}")
```

Gambar 13. Kode Program Prediksi

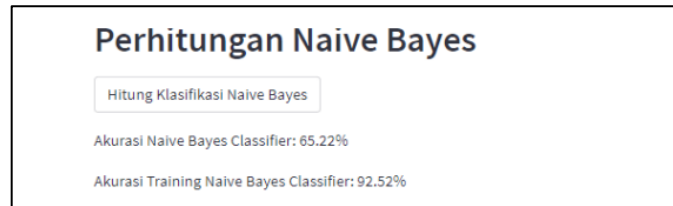
Berdasarkan Gambar 13 Kode Prediksi, pada program ini mengambil model yang tadi sudah dibuat pada proses klasifikasi. Kemudian sistem mengambil data yang dimasukkan untuk dilakukan klasifikasi naïve bayes menggunakan model yang sudah dibuat. Untuk tampilan hasil output dari kode program diatas bisa dilihat pada Gambar 14 Tampilan Hasil Prediksi. Dimana nantinya Ketika kata yang diprediksi dimasukkan dan dilakukan prediksi maka akan muncul hasil yang diprediksi dibawah button prediksi.



Gambar 14. Tampilan Hasil Prediksi

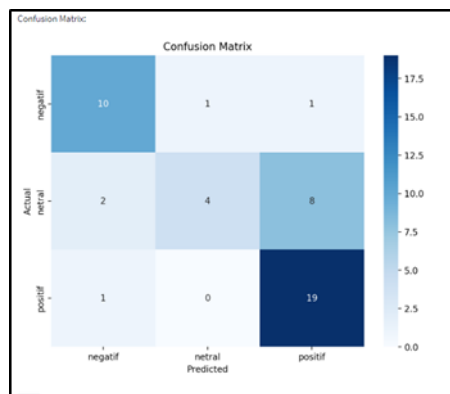
3.7 Hasil Klasifikasi

Pada bagian ini membahas hasil pelatihan model dari percobaan yang dilakukan pada data training dan testing menggunakan model yang sudah dibuat baik itu model TF-IDF maupun model Naïve Bayes Classifier. Untuk hasil dari Aplikasi Analisis Sentimen Opini Mahasiswa Terhadap Aplikasi Portal Mahasiswa Bisa Dilihat Pada Gambar 15 Hasil Klasifikasi. Berdasarkan Gambar 13 Hasil Klasifikasi, Aplikasi yang dibuat oleh penulis ini mendapatkan akurasi sebesar 65,22 % untuk data testing yang digunakan dan mendapatkan akurasi Training sebesar 92,62 %.



Gambar 15. Hasil Klasifikasi

Pembahasan Pada klasifikasi naive bayes classifier. Dataset yang ada di bagi menjadi 70% data training atau sebanyak 105 data training dan 30% data testing atau sebanyak 45 data testing. Untuk hasil klasifikasi dari data training 70% dan data testing 30% bisa dilihat pada Gambar 16 *Confusion Matrix*.



Gambar 16. *Confusion Matrix*

Pada Gambar 16 ditunjukkan nilai *true positive*, *true netral*, *true negative*, *false positive*, *false netral*, dan *false negative*. Untuk data testing, sentimen opini positif yang berhasil diklasifikasi adalah sebanyak 19 dari 20 data opini. Kemudian sentimen opini netral yang berhasil diklasifikasi adalah sebanyak 2 dari 14 data opini. Kemudian yang terakhir yaitu sentimen negatif yang berhasil diklasifikasi sebanyak 10 dari 12 data opini. Kemudian untuk classification report bisa dilihat pada Gambar 17 *Classification Report*.

Classification Report:				
	precision	recall	f1-score	support
negatif	0.70	0.58	0.64	12
netral	0.50	0.38	0.43	13
positif	0.69	0.86	0.77	21
accuracy				0.65
macro avg	0.63	0.61	0.61	46
weighted avg	0.64	0.65	0.64	46

Gambar 17. *Classification Report*

Berdasarkan Gambar 15 bisa dilihat yaitu *precision*, *recall* dan *f1-score* dari klasifikasi yang dilakukan. Untuk hasil *precision* akurasi tertinggi terdapat pada kategori negatif yaitu sebanyak 70% dan terendah pada kategori netral yaitu 50%. Kemudian untuk *recall* akurasi tertinggi terdapat pada kategori positif yaitu sebanyak 86% dan terendah pada netral yaitu 38%. Kemudian untuk *f1-score* akurasi tertinggi terdapat pada kategori positif yaitu 77% dan akurasi terendah pada netral yaitu 43%. Kemudian untuk hasil akurasi terhadap model keseluruhnya yaitu sebesar 65%.

4. Kesimpulan

Penelitian ini merupakan analisis sentimen opini mahasiswa terhadap Aplikasi Portal Mahasiswa menggunakan metode Naive Bayes Classifier dengan pemanfaatan library seperti sklearn untuk pembuatan model dan proses *preprocessing* yang melibatkan *library sastrawi*, *nltk*, *emoji*, dan *string*. Data juga divisualisasikan dengan bantuan *Wordcloud*. Proses deploy aplikasi analisis sentimen ini dilakukan dengan streamlit. Hasil pengujian menunjukkan bahwa tingkat akurasi data testing dengan rasio pembagian data 70%:30% mencapai 65,2% dalam melakukan klasifikasi analisis sentimen. Namun, model yang dihasilkan masih mengalami overfitting karena akurasi testing lebih rendah dibandingkan dengan akurasi training, sehingga memerlukan peningkatan dalam pengoptimalan model.

5. Daftar Pustaka

- [1] Kaka, D. L., Pati, G. K., & Rato, K. W. (2023). Analisis Sentimen Komentar SIAKAD Menggunakan Metode Naive Bayes Classifier. *JURNAL KRIDATAMA SAINS DAN TEKNOLOGI*, 5(02), 266-277. DOI: <https://doi.org/10.53863/kst.v5i02.933>.
- [2] Deni, D. K., & Ferida, F. Y. (2023). Usability Testing Penggunaan Menu Kartu Hasil Studi Di Website Sistem Informasi Akademik Universitas Teknologi Yogyakarta. *Jurnal Teknologi dan Manajemen Industri Terapan*, 2(1), 41-52. DOI: <https://doi.org/10.55826/tmit.v2i1.57>.
- [3] Locarso, G. K. (2022). Analisis Sentimen Review Aplikasi Pedulilindungi Pada Google Play Store Menggunakan NBC. *JTIK (Jurnal Teknik Informatika Kaputama)*, 6(2), 353-361.
- [4] Mujahid, T. A., & itje Sela, E. (2019). Analisis Perbandingan Rule Pakar dan Decision Tree J48 Dalam Menentukan Jumlah Produksi Kain Tenun Menggunakan Metode Fuzzy Tsukamoto. *JURIKOM (Jurnal Riset Komputer)*, 6(5), 501-505. DOI: <http://dx.doi.org/10.30865/jurikom.v6i5.1510>.
- [5] Sekaran, U. (2011). *Research Methods for Business: Metodologi Penelitian Untuk Bisnis*, Buku 1.
- [6] Deolika, A., Kusri, K., & Luthfi, E. T. (2019). Analisis pembobotan kata pada klasifikasi text mining. (*JurTI*) *Jurnal Teknologi Informasi*, 3(2), 179-184. DOI: <https://doi.org/10.36294/jurti.v3i2.1077>.
- [7] Zalukhu, P. S., Handhayani, T., & Sitorus, M. (2023). Analisis Sentimen terhadap Kenaikan BBM di Indonesia pada Media Sosial Twitter Menggunakan Metode Naive Bayes. *Simtek: jurnal sistem informasi dan teknik komputer*, 8(1), 65-69. DOI: <https://doi.org/10.51876/simtek.v8i1.177>.
- [8] Sari, K. (2022). *ANALISIS SENTIMEN REVIEW UNTUK POPULARITAS MARKETPLACE INDONESIA MENGGUNAKAN METODE WEBSCRAPING= SENTIMENT REVIEW ANALYSIS FOR INDONESIAN MARKETPLACE POPULARITY USING WEBSCRAPING METHOD* (Doctoral dissertation, Universitas Hasanuddin).
- [9] Hutagalung, J. S. (2023). ANALISIS SENTIMEN KEUANGAN (DATA FIQA AND FINANCIAL PHRASEBANK) MENGGUNAKAN ALGORITMA LOGISTIC REGRESSION DAN SUPPORT VECTOR MACHINE. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, 4(3), 1654-1669. DOI: <https://doi.org/10.35870/jimik.v4i3.404>.

- [10] Dewi, N. K. (2023). Identifikasi Berita Hoax dengan Menerapkan Algoritma Text Mining. *Journal of Informatics, Electrical and Electronics Engineering*, 2(3), 65-74. DOI: <https://doi.org/10.47065/jieee.v2i3.888>.
- [11] Luqyana, W. A., Cholissodin, I., & Perdana, R. S. (2018). Analisis Sentimen Cyberbullying Pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(11), 4704-4713.
- [12] Syarifuddinn, M. (2020). Analisis Sentimen Opini Publik Mengenai Covid-19 Pada Twitter Menggunakan Metode Naïve Bayes Dan Knn. *Inti Nusa Mandiri*, 15(1), 23-28. DOI: <https://doi.org/10.33480/inti.v15i1.1347>.
- [13] Parasati, W., Bachtiar, F. A., & Setiawan, N. Y. (2020). Analisis Sentimen Berbasis Aspek pada Ulasan Pelanggan Restoran Bakso President Malang dengan Metode Naïve Bayes Classifier. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(4), 1090-1099.
- [14] Simanjuntak, I. Z. (2022). Analisa Kombinasi Algoritma Stemming Dan Algoritma Soundex Dalam Pencarian Kata Bahasa Indonesia. *Informasi dan Teknologi Ilmiah (INTI)*, 10(1), 24-30.
- [15] Rozi, I. F., Hamdana, E. N., & Alfahmi, M. B. I. (2018). Pengembangan Aplikasi Analisis Sentimen Twitter Menggunakan Metode Naïve Bayes Classifier (Studi Kasus SAMSAT Kota Malang). *Jurnal Informatika Polinema*, 4(2), 149-149. DOI: <https://doi.org/10.33795/jip.v4i2.164>.
- [16] Kurniawan, D., & Yasir, M. (2022). Optimization Sentimen Analysis using CRISP-DM and Naive Bayes Methods Implemented on Social Media. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, 6(2), 74-85. DOI: <http://dx.doi.org/10.22373/cj.v6i2.12793>.