

**ANALISIS SENTIMEN PADA ULASAN APLIKASI MAMIKOS
MENGUNAKAN METODE *SUPPORT VECTOR MACHINE***

SKRIPSI

**Oleh:
NUR SYAMSU PRIAMBUDI
NIM. 200605110150**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

**ANALISIS SENTIMEN PADA ULASAN APLIKASI MAMIKOS
MENGUNAKAN METODE *SUPPORT VECTOR MACHINE***

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
NUR SYAMSU PRIAMBUDI
NIM. 200605110150

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

HALAMAN PERSETUJUAN

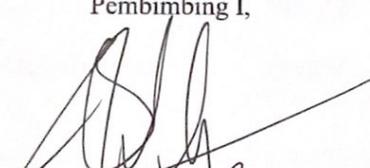
**ANALISIS SENTIMEN PADA ULASAN APLIKASI MAMIKOS
MENGUNAKAN METODE *SUPPORT VECTOR MACHINE***

SKRIPSI

Oleh:
NUR SYAMSU PRIAMBUDI
NIM. 200605110150

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 7 Mei 2024

Pembimbing I,


Fajar Rohman Hariri, M.Kom
NIP. 19890515 201801 1 001

Pembimbing II,


Dr. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrudin Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

**ANALISIS SENTIMEN PADA ULASAN APLIKASI MAMIKOS
MENGUNAKAN METODE *SUPPORT VECTOR MACHINE***

SKRIPSI

Oleh:
NUR SYAMSU PRIAMBUDI
NIM. 200605110150

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 16 Mei 2024

Susunan Dewan Penguji

Ketua Penguji : Dr. Fachrul Kurniawan, M.MT., IPM
NIP. 19771020 200912 1 001

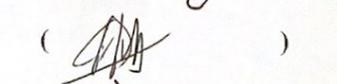
Anggota Penguji I : Syahiduz Zaman, M.Kom
NIP. 19700502 200501 1 005

Anggota Penguji II : Fajar Rohman Hariri, M.Kom
NIP. 19890515 201801 1 001

Anggota Penguji III : Dr. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007

()

()

()

()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrul Kurniawan, M.MT., IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Nur Syamsu Priambudi
NIM : 200605110150
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Analisis Sentimen Pada Ulasan Aplikasi Mamikos
Menggunakan Metode *Support Vector Machine*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 21 Mei 2024
Yang membuat pernyataan,



Nur Syamsu Priambudi
NIM. 200605110150

MOTTO

“Respect the Others”

HALAMAN PERSEMBAHAN

Puji syukur kehadiran Allah SWT, atas limpahan rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi ini dengan lancar. Skripsi ini saya persembahkan untuk kedua orang tua saya yang melalu memberikan penuh kasih sayang, kesabaran, dan pengorbanan membesarkan saya hingga mencapai titik ini. Ibu Herlawati dan Bapak Wagiman yang selalu memberikan dukungan dalam mencapai cita-cita saya dan mendoakan tiada henti yang menjadi sumber kekuatan saya dalam menyelesaikan studi ini. Kakak Awal Nur Qomar Agus dan Kakak Fajar Alamsyah yang sudah memberikan semangat, nasihat, dan kesabaran dalam memberikan arah. Seluruh keluarga yang tiada henti memberikan doa agar terselesaikan skripsi ini. Semoga Allah SWT selalu memberikan hal-hal baik kepada mereka.

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Dengan penuh rasa syukur, penulis panjatkan puji syukur kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya, serta shalawat dan salam semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW. Berkat berkah-Nya, penulis dapat menyelesaikan penulisan skripsi ini dengan judul “Analisis Sentimen Pada Ulasan Aplikasi Mamikos Menggunakan Metode *Support Vector Machine*”. Skripsi ini disusun sebagai salah satu syarat untuk mencapai gelar sarjana di Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Ucapan rasa syukur dan terima kasih penulis sampaikan kepada seluruh pihak yang telah membantu berupa kritik dan saran agar terlesainya skripsi ini. Dengan rasa hormat penulis mengucapkan terima kasih kepada :

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT., IPM, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Fajar Rohman Hariri, M.Kom selaku dosen pembimbing I dan Dr. Muhammad Faisal, M.T selaku dosen pembimbing II yang telah

membimbing serta memberikan bantuan dan arahan, sehingga penulis dapat menyelesaikan skripsi ini.

5. Dr. Fachrul Kurniawan, M.MT, IPM selaku dosen penguji I dan Syahiduz Zaman, M.Kom selaku dosen penguji II yang telah memberikan kritik dan saran, sehingga dapat menyelesaikan skripsi ini.
6. Segenap dosen dan jajaran staff Program Studi Teknik Informatika yang telah memberikan dukungan selama pengerjaan skripsi ini.
7. Kedua orang tua penulis, serta saudara kandung saya yang tiada henti memberikan motivasi dan doa untuk menyelesaikan skripsi ini dengan baik.
8. Teman - teman saya yaitu Izzul, Muzaki, Afrian, Khalid, Zaky, Imam, Jihan, Willia, Alya yang memberikan dukungan dan motivasi untuk lulus skripsi tepat waktu serta menjadi teman yang baik selama saya di tanah perantauan.
9. Teman – Teman Integer Teknik Informatika 2020 yang selalu memberikan semangat dan doa kepada penulis.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan, dari keilmuan maupun penulisan. Maka dari itu, kritik dan saran yang membangun sangat diharapkan agar lebih baik lagi kedepannya. Semoga dengan penyusunan skripsi ini bisa memberikan manfaat bagi banyak pihak.

Malang, 3 Mei 2024

Penulis

DAFTAR ISI

HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
ABSTRAK	xiv
ABSTRACT	xv
مستخلص البحث	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah.....	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
BAB II KAJIAN PUSTAKA	6
2.1 Penelitian Terkait	6
2.2 Analisis Sentimen.....	10
2.3 <i>Preprocessing Text</i>	10
2.4 <i>Term Frequency-Inverse Document Frequency</i>	11
2.5 <i>Support Vector Machine</i>	12
2.6 <i>K-Fold Cross Validation</i>	14
BAB III METODE PENELITIAN	16
3.1 Pengumpulan Data	16
3.2 Pelabelan Data.....	17
3.3 Rancangan Sistem	18
3.4 <i>Preprocessing Text</i>	21
3.5 Pembobotan Kata	23
3.5.1 Nilai <i>Term Frequency</i>	24
3.5.2 Nilai <i>Inverse Document Frequency</i>	25
3.5.3 Nilai <i>Term Frequency - Inverse Document Frequency</i>	27
3.6 <i>Support Vector Machine</i>	28
3.7 Skenario Pengujian.....	32
3.8 Evaluasi Performa	34
BAB IV UJI COBA DAN PEMBAHASAN	37

4.1 Data Penelitian	37
4.2 Skenario Uji Coba	38
4.2.1 Preprocessing Text	40
4.2.2 Pembobotan Kata TF-IDF	43
4.3 Pemodelan Algoritma <i>Support Vector Machine</i>	44
4.3.1 <i>Split</i> Data	38
4.3.2 <i>K-Fold Cross Validation</i>	49
4.4 Hasil Evaluasi	53
4.4.1 Evaluasi <i>Split</i> Data	54
4.4.2 Evaluasi <i>K-Fold Cross Validation</i>	55
4.5 Pembahasan	56
4.6 Integrasi Islam	59
4.6.1 Hubungan dengan Allah	59
4.6.2 Hubungan dengan Manusia	60
BAB V KESIMPULAN DAN SARAN	62
5.1 Kesimpulan	62
5.2 Saran	63
DAFTAR PUSTAKA	

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	8
Tabel 3.1 Data Terlabel.....	17
Tabel 3.2 Contoh Data Ulasan.....	24
Tabel 3.3 Tabel Perhitungan TF	25
Tabel 3.4 Tabel Perhitungan IDF	26
Tabel 3.5 Tabel Perhitungan TF-IDF.....	28
Tabel 4.1 Sampel Data Ulasan	38
Tabel 4.2 Hasil <i>Preprocessing Cleaning</i>	41
Tabel 4.3 Hasil <i>Preprocessing Case Folding</i>	41
Tabel 4.4 Hasil <i>Preprocessing Tokenizing</i>	42
Tabel 4.5 Hasil <i>Preprocessing Stopwords Removal</i>	43
Tabel 4.6 Hasil <i>Preprocessing Stemming</i>	43
Tabel 4.8 Hasil Klasifikasi Rasio 60:40.....	46
Tabel 4.9 Hasil Prediksi Sentimen Rasio 60:40	47
Tabel 4.10 Hasil Klasifikasi Rasio 70:30.....	47
Tabel 4.11 Hasil Prediksi Sentimen Rasio 70:30	47
Tabel 4.12 Hasil Klasifikasi Rasio 80:20.....	48
Tabel 4.13 Hasil Prediksi Sentimen Rasio 80:20	48
Tabel 4.14 Hasil Klasifikasi Rasio 90:10.....	48
Tabel 4.15 Hasil Prediksi Sentimen Rasio 90:10	49
Tabel 4.16 Hasil Klasifikasi K-5	50
Tabel 4.17 Hasil Prediksi Sentimen K-5.....	50
Tabel 4.18 Hasil Klasifikasi K-10	51
Tabel 4.19 Hasil Prediksi Sentimen K-10.....	51
Tabel 4.20 Hasil Klasifikasi K-15	52
Tabel 4.21 Hasil Prediksi Sentimen K-15.....	52
Tabel 4.22 Hasil Klasifikasi K-20	53
Tabel 4.23 Hasil Prediksi Sentimen K-20.....	53
Tabel 4.24 Perbandingan <i>Confusion Matrix Split Data</i>	54
Tabel 4.25 Nilai <i>Confusion Matrix</i> 90:10	55
Tabel 4.26 Perbandingan <i>Confusion Matrix K-fold Cross Validation</i>	55
Tabel 4.27 Nilai <i>Confusion Matrix</i> K-10	56

DAFTAR GAMBAR

Gambar 2.1 SVM Memisahkan Dua Data dengan <i>Hyperlane</i>	12
Gambar 2.2 Dua <i>Hyperlane</i> dalam Satu Data	13
Gambar 3.1 Perancangan Sistem	19
Gambar 3.2 Tahapan <i>Preprocessing Text</i>	21
Gambar 3.3 Visualisasi <i>5-fold Cross Validation</i>	33
Gambar 3.4 Visualisasi <i>Confusion Matrix</i>	35
Gambar 4.1 <i>Pie Chart</i> Perbandingan Kelas Sentimen	37
Gambar 4.2 <i>Pseudocode</i> Tahap <i>Cleaning</i>	41
Gambar 4.3 <i>Pseudocode</i> Tahap <i>Case Folding</i>	41
Gambar 4.4 <i>Pseudocode</i> Tahap <i>Tokenizing</i>	42
Gambar 4.5 <i>Pseudocode</i> Tahap <i>Stopwords Removal</i>	42
Gambar 4.6 <i>Pseudocode</i> Tahap <i>Stemming</i>	43

ABSTRAK

Priambudi, Nur Syamsu. 2024. **Analisis Sentimen Pada Ulasan Aplikasi Mamikos Menggunakan Metode *Support Vector Machine***. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Fajar Rohman Hariri, M.Kom. (II) Dr. Muhammad Faisal, M.T.

Kata kunci: Analisis Sentimen, *K-Fold Cross Validation*, Mamikos, *Support Vector Machine*

Merantau merupakan salah satu kegiatan yang dilakukan oleh masyarakat usia produktif di Indonesia dengan harapan mencapai kesuksesan dan menjadi individu yang lebih baik. Pertumbuhan jumlah perantau turut berkontribusi terhadap kebutuhan akan rumah sewa, namun mereka masih mengalami kesulitan dalam mendapatkan informasi yang lengkap. Untuk mengatasi hal tersebut terdapat aplikasi mamikos yang menyediakan data informasi sewa perumahan secara lengkap. Ulasan pengguna dalam aplikasi merupakan indikator penting dalam menilai kualitas aplikasi. Dalam ulasan tersebut tentunya terdapat berbagai komentar yang diberikan pengguna yang terbagi dalam tiga kategori, yaitu sentimen positif, sentimen negatif, dan sentimen netral. Penelitian ini bertujuan untuk mengimplementasikan algoritma *Support Vector Machine* dalam analisis sentimen pada ulasan aplikasi mamikos. Data yang digunakan sebanyak 1049 data ulasan yang diambil pada periode 4 Februari 2022 sampai 14 September 2023. Proses analisis sentimen menggunakan algoritma *Support Vector Machine* karena algoritma ini dapat menangani data berdimensi tinggi sehingga sesuai untuk memprediksi sentimen dari teks. Model diuji dengan dua metode yaitu *split data* dan *k-fold cross validation*, yang kemudian dievaluasi kinerjanya menggunakan *confusion matrix*. Hasil evaluasi yang diperoleh dari metode *split data* diperoleh nilai *accuracy* tertinggi sebesar 78,10% dengan rasio pembagian data *training* dan data *testing* sebesar 90:10. Pada uji *k-fold cross validation* nilai *accuracy* tertinggi diperoleh pada k-10 dengan rata-rata *accuracy* sebesar 73,21%. Hal ini menunjukkan bahwa pengujian *split data* mendapatkan nilai *accuracy* yang lebih baik dibandingkan dengan *k-fold cross validation*. Hasil evaluasi model dengan *k-fold cross validation* tidak berpengaruh terhadap peningkatan nilai *accuracy* sistem.

ABSTRACT

Priambudi, Nur Syamsu. 2024. **Sentiment Analysis in Mamikos Application Reviews Using the Support Vector Machine Method** Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Promotor: (I) Fajar Rohman Hariri, M.Kom (II) Dr. Muhammad Faisal, M.T.

Migrants are one of the activities carried out by people of productive age in Indonesia in the hope of achieving success and becoming better individuals. The growth in the number of migrants has contributed to the need for rental housing, but they still experience difficulties in obtaining complete information. To overcome this, there is a mamikos application which provides complete housing rental information data. In-app user reviews are an important indicator in assessing the quality of an app. In this review, of course, there are various comments given by users which are divided into three categories, namely positive sentiment, negative sentiment and neutral sentiment. This research aims to implement the Support Vector Machine algorithm in sentiment analysis in mamikos application reviews. The data used is 1049 review data taken in the period 4 February 2022 to 14 September 2023. The sentiment analysis process uses the Support Vector Machine algorithm because this algorithm can handle high-dimensional data so it is suitable for predicting sentiment from text. The model was tested using two methods, namely split data and k-fold cross validation, which then evaluated its performance using a confusion matrix. The evaluation results obtained from the split data method obtained the highest accuracy value of 78.10% with a training data and test data division ratio of 90:10. In the k-fold cross validation test, the highest accuracy value was obtained at k-10 with an average accuracy of 73.21%. This shows that split data testing gets better accuracy values compared to k-fold cross validation. The results of model evaluation with k-fold cross validation have no effect on increasing the system accuracy value.

Keywords: Mamikos, *K-Fold Cross Validation*, *Sentimen Analysis*, *Support Vector Machine*

مستخلص البحث

بريامبودي، نور سيامسو. ٢٠٢٤. تحليل المشاعر في مراجعات تطبيق Mamikos باستخدام طريقة آلة ناقل الدعم. أطروحة. برنامج دراسة الهندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية، مالانج. المشرف: (ط) فجر الرحمن الحريري، م.كوم. (الثاني) د. محمد فيصل، م.ت

، آلة ناقل الدعم Mamikos، K-Fold الكلمات الرئيسية: تحليل المشاعر، التحقق من صحة

المهاجر هو أحد الأنشطة التي يقوم بها الأشخاص في سن الإنتاج في إندونيسيا على أمل تحقيق النجاح وأن يصبحوا أفرادًا أفضل. وقد ساهم نمو عدد المهاجرين في زيادة الحاجة إلى مساكن للإيجار، لكنهم ما زالوا يواجهون صعوبات في الحصول على معلومات كاملة للتغلب على ذلك، يوجد تطبيق mamikos الذي يوفر بيانات كاملة عن معلومات تأجير المساكن تعد تقييمات المستخدمين داخل التطبيق مؤشراً مهماً في تقييم جودة التطبيق في هذه المراجعة، بالطبع، هناك العديد من التعليقات المقدمة من المستخدمين والتي تم تقسيمها إلى ثلاث فئات، وهي المشاعر الإيجابية والمشاعر السلبية والمشاعر المحايدة يهدف هذا البحث إلى تطبيق خوارزمية Support Vector Machine في تحليل المشاعر في مراجعات تطبيق البيانات المستخدمة هي ١٠٤٩ بيانات مراجعة مأخوذة في الفترة من ٤ فبراير ٢٠٢٢ إلى ٤ سبتمبر ٢٠٢٣. تستخدم عملية تحليل المشاعر خوارزمية Support Vector Machine لأن هذه الخوارزمية يمكنها التعامل مع البيانات عالية الأبعاد لذا فهي مناسبة للتنبؤ بالمشاعر من النصم اختبار النموذج باستخدام طريقتين، وهما البيانات المقسمة والتحقق المتقاطع k-fold، والذي تم بعد ذلك تقييم أدائه باستخدام مصفوفة الارتباك حصلت نتائج التقييم التي تم الحصول عليها من طريقة تقسيم البيانات على أعلى قيمة دقة قدرها ٧٨,١٠ مع نسبة تقسيم بيانات التدريب وبيانات الاختبار إلى ١٠:٩٠ في اختبار التحقق من صحة k-fold، تم الحصول على أعلى قيمة دقة عند k-١٠ بمتوسط دقة ٧٣,٢١ يوضح هذا أن اختبار البيانات المقسمة يحصل على قيم دقة أفضل مقارنةً بالتحقق المتقاطع من ليس لنتائج تقييم النموذج باستخدام التحقق المتقاطع k-fold أي تأثير على زيادة قيمة دقة النظام

BAB I

PENDAHULUAN

1.1 Latar Belakang

Merantau adalah tradisi yang umum di Indonesia di mana masyarakat pindah dari kampung halaman untuk mencari kesuksesan dan membantu kemajuan daerah asal mereka (Syamsuddin, 2018). Peningkatan jumlah perantau ini menyebabkan meningkatnya kebutuhan akan tempat tinggal sewa seperti apartemen, asrama, kos, dan kontrakan. Namun, mencari hunian sewa sering kali terkendala oleh kurangnya informasi yang lengkap dan komprehensif. Oleh karena itu, informasi yang menyeluruh sangat diperlukan untuk menemukan hunian sewa yang sesuai dengan kebutuhan (Wibowo, 2020).

Di Indonesia, ada aplikasi yang dapat mengatasi masalah ini, yaitu Mamikos. Aplikasi ini memberikan informasi lengkap tentang hunian sewa, termasuk alamat, peta lokasi, detail fasilitas dengan foto, informasi harga, dan ketersediaan kamar yang selalu diperbarui. Dengan menggunakan Mamikos, pencari kos dapat mendapatkan informasi yang akurat dan efisien (Fara et al. 2023).

Aplikasi Mamikos telah digunakan oleh banyak orang, sekitar 6-8 juta orang mencari kos setiap bulan, dan lebih dari 110.000 pemilik kos. Lebih dari 140 kota Indonesia dan lebih dari 2 juta kamar kos tersedia melalui aplikasi ini (Mamikos, 2023). Banyaknya pengguna ini menghasilkan banyak ulasan mengenai aplikasi tersebut. Ulasan pengguna sangat penting untuk menilai kualitas aplikasi, di mana pengguna memberikan umpan balik berdasarkan kepuasan dan pengalaman

mereka. Ulasan positif muncul ketika pengguna puas dengan layanan, sementara ulasan negatif diberikan ketika mereka tidak puas. Ulasan ini sangat mempengaruhi keputusan pengguna lain yang ingin menggunakan Mamikos dan membantu perusahaan dalam meningkatkan kualitas aplikasi serta memahami keinginan pengguna.

Analisis sentimen adalah salah satu dari banyak metode untuk memeriksa ulasan pengguna pada aplikasi Mamikos. Perusahaan dapat lebih mudah mengevaluasi keterbatasan aplikasi mereka dengan menggunakan analisis sentimen untuk mengkategorikan ulasan pengguna ke dalam kategori yang positif, negatif, atau netral (Rokhman et al., 2021). Tujuan dari analisis sentimen untuk mengkategorikan tulisan pendapat sebagai netral, negatif, atau positif. Data ulasan dari *Google Play Store* digunakan untuk memprediksi gambaran opini pengguna. Oleh karena itu, analisis opini pengguna diperlukan sebagai tolok ukur kualitas aplikasi Mamikos. Tujuan utama analisis sentimen adalah untuk memahami dan menilai pendapat yang diungkapkan dalam teks dan untuk menentukan apakah teks itu netral, positif, atau negatif. (Wahyudi & Kusumawardana, 2021).

Mengevaluasi pendapat dalam teks sangat penting untuk memahami informasi yang disampaikan dalam ulasan. Saat membaca teks ulasan, kita harus mampu mengidentifikasi pendapat yang disampaikan untuk membantu membedakan apakah informasi yang disampaikan valid atau tidak. Dengan demikian, kita dapat mengambil keputusan berdasarkan informasi yang valid. Namun, jika kita mendapatkan informasi dari sumber yang tidak valid, maka kita

harus memastikan kebenarannya terlebih dahulu, seperti yang dijelaskan pada Al-Qur'an dalam surah An-Nur ayat 15.

إِذْ تَلَقَّوْنَهُ بِأَلْسِنَتِكُمْ وَتَقُولُونَ بِأَفْوَاهِكُمْ مَا لَيْسَ لَكُمْ بِهِ عِلْمٌ وَتَحْسَبُونَهُ هَيِّنًا وَهُوَ عِنْدَ اللَّهِ عَظِيمٌ

“(Ingatlah) ketika kamu menerima (berita bohong) itu dari mulut ke mulut dan kamu katakan dengan mulutmu apa yang tidak kamu ketahui sedikitpun, dan kamu menganggapnya, remeh padahal dalam pandangan Allah itu soal besar.” (Q.S An-Nur : 15)

Menurut Al-Mukhtashar, menyebarkan berita tanpa mengetahui kebenarannya dianggap sangat berbahaya oleh Allah, meskipun bagi kita mungkin terlihat sepele. Kita sering kali terjebak dalam prasangka dan perasaan negatif tanpa mengetahui faktanya. Karena hanya Allah yang mengetahui seluruh kebenaran, kita harus berhati-hati dalam menilai dan membuat keputusan.

Dalam analisis sentimen, ada beberapa algoritma yang bisa digunakan, seperti *Naïve Bayes Classifier*, *K-NN*, *Random Forest*, dan *Support Vector Machine*. Algoritma SVM bekerja dengan cara membangun hyperplane yang memisahkan data ke dalam kelas-kelas yang berbeda dengan margin optimal. SVM dianggap sebagai metode terbaik untuk analisis sentimen karena mampu menangani data berdimensi tinggi dan menghasilkan tingkat akurasi yang tinggi. Dalam klasifikasi teks, SVM membagi data ke dalam tiga kelas: positif, negatif, dan netral. Diharapkan dengan menggunakan metode SVM, hasil akurasi yang diperoleh akan baik (Fikri et al., 2020). *Support Vector Machine* metode yang terbaik dibandingkan metode lainnya dalam melakukan analisis sentimen karena dapat mengolah data dengan dimensi tinggi untuk menghasilkan tingkat *accuracy* yang lebih baik. Klasifikasi teks dilakukan dengan membagikan 3 kelas yaitu

kelas positif, kelas negatif, dan kelas netral. Dengan menerapkan metode *Support Vector Machine (SVM)* diharapkan mampu mendapatkan hasil *accuracy* yang baik dalam penelitian ini (Arsi & Waluyo, 2021).

Salah satu tantangan dalam penggunaan SVM adalah mengoptimalkan parameter untuk mencapai hasil yang akurat. Dalam analisis sentimen, metrik penting seperti *accuracy*, *precision*, *recall*, dan *f1-score* digunakan untuk mengukur kinerja model. Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi efektivitas SVM dalam analisis sentimen dengan menggunakan metrik-metrik tersebut (Pramesti & Pratiwi, 2023).

Pada penelitian sebelumnya yang dilakukan oleh (Arsi & Waluyo, 2021) mengenai analisis sentimen terhadap wacana pemindahan ibu kota Indonesia menggunakan SVM menunjukkan bahwa dataset yang terdiri dari 1.320 tweet yang dikelompokkan menjadi kelas positif dan negatif menghasilkan akurasi sebesar 96,68%. Penelitian juga dilakukan oleh (Wahyudi & Kusumawardana, 2021) tentang sentimen ulasan aplikasi Grab menggunakan SVM menunjukkan hasil akurasi sebesar 85,54% dengan dataset sebanyak 1.000 ulasan pengguna.

Berdasarkan latar belakang dan permasalahan tersebut, penelitian ini menggunakan algoritma SVM untuk analisis sentimen ulasan aplikasi Mamikos. Penelitian ini bertujuan untuk mengukur sentimen positif, negatif, dan netral pada aplikasi Mamikos serta mengevaluasi seberapa baik klasifikasi yang dilakukan dan seberapa akurat hasilnya.

1.2 Pernyataan Masalah

Bagaimana performa model *Support Vector Machine* (SVM) dalam melakukan Analisis Sentimen pada Ulasan Aplikasi Mamikos?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Data yang digunakan merupakan data primer berupa teks yang diambil pada ulasan aplikasi mamikos di *google play store*.
2. Data ulasan menggunakan bahasa Indonesia dengan rentang tanggal 4 Februari 2022 hingga 14 September 2023 sebanyak 1049 ulasan.

1.4 Tujuan Penelitian

Mengetahui performa dengan menggunakan *Support Vector Machine* (SVM) dalam Analisis Sentimen pada Ulasan Aplikasi Mamikos.

1.5 Manfaat Penelitian

Dengan adanya penelitian diharapkan dapat memberikan manfaat ke beberapa pihak, diantaranya :

1. Membantu perusahaan dalam kualitas layanan mamikos untuk mengembangkan produk yang lebih baik sesuai kebutuhan dan keinginan pengguna.
2. Menambah pengetahuan para peneliti dalam menguasai analisis sentimen dengan menggunakan metode *Support Vector Machine*.
3. Membantu para pengguna dalam mengambil keputusan yang lebih baik berdasarkan data yang informatif dan relevan.

BAB II

KAJIAN PUSTAKA

2.1 Penelitian Terkait

Penelitian sebelumnya dapat dijadikan referensi untuk penelitian ini, memberikan panduan yang membantu penulis dalam menyelesaikan penelitian.

Pada tahun 2021, Ananda dan Pristyanto (Ananda & Pristyante, 2021) meneliti opini publik tentang program vaksinasi COVID-19 di Indonesia menggunakan analisis media sosial. Secara khusus, mereka menggunakan teknik yang *Support Vector Machine* untuk mengkategorikan sentimen yang diungkapkan di Twitter mengenai kebijakan vaksin. Tujuannya adalah memahami respons masyarakat dengan menganalisis bagaimana kata-kata didistribusikan dan dengan membangun model klasifikasi SVM. Mereka mengklasifikasikan opini menjadi tiga kelompok: positif, negatif, dan netral. Hasilnya menjanjikan, mencapai tingkat *accuracy* 90% dengan *precision*, *recall*, dan *f1-score*.

Penelitian oleh (Kulsum Ummi & Jajuli Mohamad, 2022) tentang sentimen aplikasi WETV Google Play Store menggunakan algoritma *Support Vector Machine*. Dengan menggunakan teknik *crawling*, data diperoleh dari *Google Play Store*. 5.378 data yang dikumpulkan dengan 11 Desember 2020 sampai 15 Januari 2021. Empat skenario 70:30, 80:20, dan 90:10 dengan 60% data *training* dan 40% data *testing* digunakan dalam penelitian ini. Hasil untuk skenario 60:40 adalah 83% untuk *accuracy*, 83% bagi *precision*, 89% untuk *recall*, dan 86% untuk *f1-score*. Hasil untuk skenario kedua adalah 70:30,: *accuracy* 85%, *precision* 86%, *recall* 89%, dan *f1-score* 87%. Kemudian pada skenario ketiga yaitu 80: 20

menghasilkan nilai *accuracy* sebesar 85%, *precision* 85%, *recall* 90%, *f1-score* 88%. Skenario terakhir yaitu 90:10 menghasilkan nilai *accuracy* sebesar 85%, *precision* 86%, *recall* 90%, dan *f1-score* 88%. Hasil nilai *accuracy* yang didapatkan sebesar 85% yang diperoleh dari skenario 70:30, 80:20, dan 90:10. Skenario 60:40 mendapatkan nilai akurasi terendah dengan nilai 83%.

Hasil dari penelitian sebelumnya yang dilakukan oleh (Nurmalasari et al., 2023) membandingkan algoritma SVM, KNN, dan NBC dalam analisis sentimen aplikasi layanan pinjaman. Penelitian ini bertujuan membandingkan tingkat akurasi ketiga algoritma tersebut pada aplikasi Kredivo, Akulaku, dan Indodana dengan total data 42.714. Data dibagi dengan skenario 70:30 untuk data *training* dan *testing*. Hasil *accuracy* pada aplikasi Kredivo adalah 84% untuk K-NN, 88% untuk NBC, dan 89% untuk SVM. Pada aplikasi Akulaku, hasil akurasi adalah 84% untuk K-NN, 88% untuk NBC, dan 89% untuk SVM. Sedangkan pada aplikasi Indodana, K-NN menghasilkan *accuracy* 81%, NBC 88%, dan SVM 88%. Hasil ini menunjukkan bahwa metode SVM memiliki *accuracy* yang lebih baik dibandingkan K-NN dan NBC.

Penelitian lainnya dilakukan juga oleh (Wijaya et al., 2021) menganalisis sentimen tentang UU ITE menggunakan algoritma SVM. Penelitian ini mengenai opini publik terhadap UU ITE di Indonesia dengan menggunakan data Twitter. Kumpulan data 3.000 tweet yang dikumpulkan melalui *crawling* dari 1 Maret hingga 7 April dianalisis. Pembagian *training* 90% dan *testing* 10% menghasilkan skor *accuracy* terbaik sebesar 84%, *precision* sebesar 84%, *recall* sebesar 65%, dan *f1-score* sebesar 71%.

Pada bagian ini, memaparkan penelitian yang ada dalam analisis sentimen untuk memberikan konteks pada penelitian tentang analisis sentimen ulasan aplikasi Mamikos dengan metode *Support Vector Machine*.

Tabel 2.1 Penelitian Terkait

No.	Peneliti (tahun)	Judul	Metode	Hasil	Perbedaan
1	(Ananda & Pristyanto, 2021)	Analisis Sentimen Pengguna Twitter Terhadap Program Vaksinasi Covid-19 di Indonesia Menggunakan Algoritma <i>Support Vector Machine</i>	<i>Support Vector Machine</i>	Hasil pengukuran cukup optimal dengan <i>accuracy</i> , <i>precision</i> , <i>recall</i> , dan <i>f1-score</i> .	Pengujian dengan menggunakan <i>k-fold cross validation</i> dengan nilai k 4. Peneliti menggunakan <i>split</i> data dan <i>k-fold cross validation</i> dengan nilai k 5, 10, 15, dan 20.
2	(Kulsum Umami & Jajuli Mohamad, 2022)	Analisis Sentimen Aplikasi WETV <i>Google Play Store</i> Menggunakan Algoritma <i>Support Vector Machine</i>	<i>Support Vector Machine</i>	Hasil nilai <i>accuracy</i> yang didapatkan sebesar 85% yang diperoleh dari skenario 70:30, 80:20, dan 90:10. Skenario 60:40 mendapatkan nilai <i>accuracy</i> terendah dengan nilai 83%.	Melakukan pengujian dengan <i>split</i> data tanpa <i>k-fold cross validation</i> , yaitu menggunakan 60:40, 70:30, 80:20, dan 90:10. Peneliti melakukan pengujian dengan <i>split</i> data dan <i>k-fold cross validation</i> .
3	(Nurmalasari et al., 2023)	Perbandingan Algoritma SVM, KNN, dan NBC Terhadap Analisis Sentimen Aplikasi Loan Service	<i>Support Vector Machine</i> , <i>K-Nearest Neighbor</i> , dan <i>Naïve Bayes Classifier</i> akurasi sebesar 86.81%.	Hasil <i>accuracy</i> yang didapatkan pada aplikasi kredivo dengan menggunakan K-NN sebesar 84, NBC, 88%, dan SVM 89%. Pada aplikasi akulaku mendapatkan hasil <i>accuracy</i> dengan	Metode yang digunakan melakukan perbandingan antara SVM, KNN, dan NBC. Pengujian dengan pembagian data <i>training</i> dan data <i>testing</i> dengan skenario 70:30.

				<p>menggunakan K-NN memperoleh nilai 79%, NBC 86%, dan SVM mendapat 87%.</p> <p>Aplikasi Indodana metode K-NN mendapat accuracy 81%, NBC 88%, dan SVM 88%.</p> <p>Dapat disimpulkan bahwa metode SVM memiliki hasil nilai <i>accuracy</i> yang lebih baik dibandingkan dengan metode K-NN dan NBC.</p>	<p>Peneliti hanya menggunakan metode SVM dan pengujian menggunakan <i>split</i> data dengan rasio pembagian 60:40, 70:30, 80:20, dan 90:10 serta juga menggunakan pengujian <i>k-fold cross validation</i></p>
4	(Wijaya et al., 2021)	<p>Analisis Sentimen Suputar UU ITE Menggunakan Algoritma Support Vector Machine</p>	<p>Support Vector Machine</p>	<p>Nilai <i>accuracy</i> tertinggi didapatkan pada skenario pembagian 90% data <i>training</i> dan 10% data <i>testing</i> dengan nilai <i>accuracy</i> 84%, <i>precision</i> 84%, <i>recall</i> 65%, dan <i>f1-score</i> 71%.</p>	<p>Melakukan pengujian dengan pembagian data <i>training</i> dan data <i>testing</i> tanpa <i>k-fold cross validation</i>, yaitu menggunakan 90:10, 80:20, 70:30, 60:40, 50:50, 40:60, 30:70, 20:80, dan 10:90.</p> <p>Peneliti melakukan pengujian dengan <i>split</i> data dengan rasio pembagian 60:40, 70:30, 80:20, dan 90:10 serta juga menggunakan pengujian <i>k-fold cross validation</i></p>

2.2 Analisis Sentimen

Analisis sentimen adalah bagian dari pemrosesan bahasa alami yang bertujuan untuk mengidentifikasi ekspresi sikap berbasis opini penulis, seperti baik atau buruk, suka atau tidak suka, terhadap suatu entitas tertentu (misalnya produk, topik, atau isu) atau aspek tertentu dari entitas tersebut (misalnya harga atau kualitas). Analisis sentimen sering digunakan untuk menentukan polaritas teks, tetapi juga bisa digunakan lebih umum untuk menilai sikap seseorang terhadap suatu sasaran atau topik tertentu (Cambria et al., 2019).

Dalam prosesnya, analisis sentimen mengidentifikasi opini mengenai topik tertentu dari teks-teks tidak terstruktur yang berasal dari berbagai sumber. Tujuannya adalah untuk memahami dan mengukur opini masyarakat dalam kategori positif, negatif, atau netral. Hasil ini kemudian dapat digunakan sebagai dasar untuk pengambilan keputusan (Adhi Putra, 2021).

2.3 *Preprocessing Text*

Preprocessing text adalah langkah penting dalam menormalkan istilah dalam kalimat. Tujuannya adalah memastikan bahwa data yang digunakan untuk pelatihan akurat dan fitur yang diekstraksi konsisten, sehingga mempermudah pemrosesan data. *Preprocessing text* diperlukan untuk menstandarisasi teks-teks menjadi bentuk alami dengan menghapus kata-kata yang tidak relevan, bertujuan untuk mengoptimalkan kalimat yang akan digunakan dalam proses klasifikasi (Gifari et al., 2022). Pada penelitian ini terdapat 5 tahapan *preprocessing text* dengan sebagai berikut :

1. *Cleaning* merupakan proses penghapusan data teks dari kalimat yang tidak penting, tidak baku, dan tidak relevan, seperti emoticon, angka, tanda baca, simbol, tautan, dan tagar.
2. *Case Folding* merupakan proses mengubah huruf kapital yang ada di seluruh kalimat menjadi huruf kecil.
3. *Tokenizing* merupakan proses pemecah kalimat pada data menjadi pecahan kata tunggal.
4. *Stopword Removal* merupakan proses penghapusan data teks yang berisi kata tidak memiliki makna dan tidak relevan pada topik.
5. *Stemming* merupakan proses pengubahan data teks pada kata imbuhan menjadi kata dasar.

2.4 *Term Frequency-Inverse Document Frequency*

Term Frequency-Inverse Document Frequency adalah teknik yang mengubah teks menjadi data numerik. Ini memberikan bobot pada kata-kata berdasarkan signifikansinya dalam dokumen tertentu dan di seluruh kumpulan dokumen. TF, atau *Term Frequency*, menangkap seberapa sering sebuah kata muncul dalam dokumen, menekankan pentingnya kata tersebut dalam konteks tersebut. Sebaliknya, *Document Frequency* (DF) menunjukkan seberapa umum suatu kata di seluruh kumpulan dokumen. Dengan mengalikan TF dan IDF, TF-IDF memberikan bobot yang mencerminkan relevansi sebuah kata terhadap dokumen tertentu sambil mempertimbangkan prevalensinya secara keseluruhan. Bobot kata dipengaruhi oleh frekuensi kemunculannya dalam dokumen; jika sebuah kata sering muncul, bobotnya akan lebih besar, tetapi jika jarang muncul,

bobotnya akan lebih kecil (Septian et al., 2019). Nilai pembobotan TF-IDF dapat dihitung menggunakan persamaan 2.1.

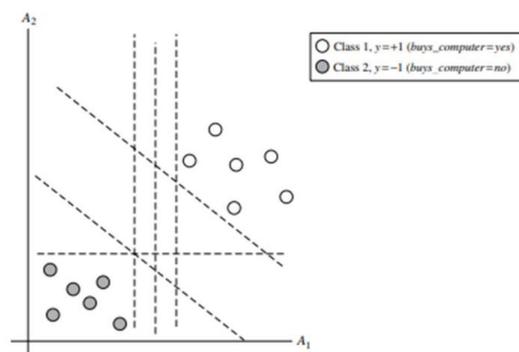
$$TF-IDF_{(t,d,D)} = TF_{(t,d)} \times IDF_{(t,D)} \quad (2.1)$$

Keterangan :

- TF-IDF_(t, d, D) = Pembobotan TF-IDF
- TF_(t,d) = Jumlah kemunculan *term* dalam dokumen
- IDF_(t,d) = Bobot *inverse* dalam nilai df

2.5 Support Vector Machine

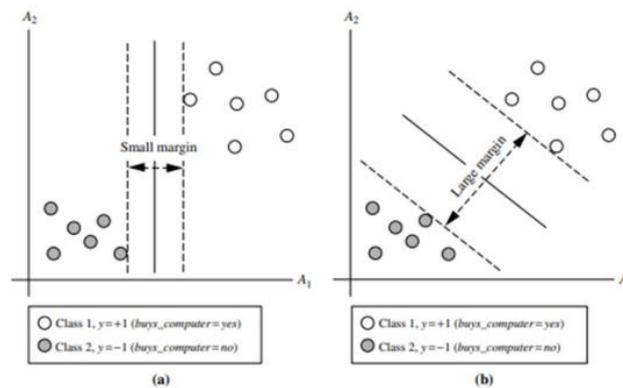
Support Vector Machine adalah metode pembelajaran terawasi yang digunakan untuk tugas klasifikasi. SVM sangat efektif dalam klasifikasi teks karena mampu menghasilkan tingkat *accuracy* yang tinggi. Dalam proses klasifikasinya, SVM berusaha menemukan *hyperplane* yang optimal dengan memaksimalkan jarak antara kelas-kelas yang berbeda (Pratiwi et al., 2021).



Gambar 2.1. SVM memisahkan dua data dengan *Hyperlane*

Pada gambar 2.1, terlihat bahwa data dua dimensi dapat dipisahkan secara linear karena kita dapat menarik garis lurus untuk memisahkan semua data dari kelas +1 dengan data dari kelas -1. Garis hitam di tengah, yang disebut *hyperplane*, bertindak sebagai pemisah antara kelas-kelas ini. *Hyperplane* adalah

fungsi yang digunakan untuk memisahkan kelas -1 dan $+1$, dan posisi optimalnya dicapai dengan memaksimalkan margin. Margin adalah jarak antara *hyperplane* dan data yang paling dekat dari setiap kelas (Gambar 2.2).



Gambar 2.2. Dua *hyperlane* dalam satu data

Untuk menemukan *hyperplane* terbaik, diperlukan berbagai garis pemisah. SVM menangani masalah ini dengan mencari *Maximum Marginal Hyperplane* (MMH), yang merupakan *hyperplane* yang memaksimalkan margin antara kelas-kelas yang berbeda. Dalam gambar 2.2, kedua *hyperplane* dapat mengklasifikasikan data dengan benar jika semua data dalam gambar tersebut dipisahkan oleh *hyperplane* tersebut. Namun, masalah muncul ketika data tidak dapat dipisahkan secara linear.

Keterbatasan *Support Vector Machine* adalah ketidakmampuannya menangani pemisahan data non-linier. Untuk mengatasi hal ini, SVM ditingkatkan dengan fungsi kernel. Fungsi-fungsi ini pada dasarnya memproyeksikan data ke dalam ruang berdimensi lebih tinggi, sehingga memungkinkan untuk mencapai pemisahan linier antar kelas. Konsep ini dikenal sebagai *Kernel Trick*. *Kernel Trick* memungkinkan SVM untuk menciptakan dimensi baru di mana *hyperplane*

dapat memisahkan data secara efektif jika data dimasukkan ke dalam dimensi tersebut (Fazar et al. 2021). Dengan menggunakan fungsi kernel, SVM dapat menangani data yang kompleks dan menghasilkan klasifikasi yang lebih akurat meskipun data tersebut tidak dapat dipisahkan secara linear dalam dimensi aslinya.

2.6 *K-Fold Cross Validation*

Untuk memastikan evaluasi yang tidak bias dan estimasi akurasi yang andal, *K-Fold Cross Validation* digunakan. Teknik ini melibatkan pemisahan kumpulan data awal secara acak menjadi k kali lipat yang sama (Larasati et al., 2022). Proses ini dilakukan untuk memastikan bahwa setiap bagian data digunakan baik untuk pelatihan maupun pengujian model.

K-Fold cross validation bekerja dengan membagi data menjadi kelompok-kelompok berukuran sama, yang disebut lipatan (k). Model tersebut kemudian dilatih dan dievaluasi sebanyak k kali. Selama setiap putaran, lipatan berbeda digunakan untuk pengujian, sedangkan lipatan lainnya digabungkan untuk pelatihan. Hal ini memastikan setiap titik data digunakan untuk pengujian satu kali dan untuk pelatihan $k-1$ kali.

K-fold cross validation memberikan keuntungan yang signifikan dengan menghasilkan penilaian kinerja model yang lebih andal. Hal ini dicapai dengan memanfaatkan seluruh kumpulan data untuk pelatihan dan pengujian secara bergantian. Pendekatan ini mengurangi risiko overfitting dan memberikan gambaran yang lebih jelas tentang bagaimana model akan melakukan generalisasi pada data yang benar-benar baru (Susianti et al., 2020).

Dengan mengulang proses ini sebanyak k kali, kita memastikan bahwa semua data digunakan baik untuk pelatihan maupun pengujian, yang membantu dalam memberikan evaluasi yang lebih robust dan terpercaya terhadap model yang sedang dikembangkan.

BAB III

METODE PENELITIAN

3.1 Pengumpulan Data

Penelitian ini mengumpulkan data dengan cara teknik *crawling* yang mengambil dari *google play store* pada ulasan aplikasi mamikos yang menggunakan bahasa pemrograman python. Untuk melakukan *crawling*, diperlukan informasi seperti ID aplikasi Mamikos, bahasa, dan negara agar ulasan yang diperoleh relevan. Proses pengumpulan data dilakukan selama periode 4 Februari 2022 hingga 14 September 2023, dan berhasil mengumpulkan sebanyak 1.049 ulasan. Data yang terkumpul disimpan dalam format Excel untuk memudahkan pemrosesan lebih lanjut.

Setelah pengumpulan data, data dibagi menjadi dua : data *training* dan *testing*. Data *training* berfungsi sebagai dasar model klasifikasi SVM, yang memungkinkannya mempelajari dan mengembangkan keterampilan klasifikasinya. Sebaliknya, data *testing* digunakan untuk menilai performa model pada data yang tidak terlihat. Pemisahan ini memastikan model dapat menggeneralisasi pengetahuannya secara efektif dan menawarkan ukuran yang dapat diandalkan mengenai kemampuannya untuk mengklasifikasikan ulasan baru. Dengan metode ini, peneliti dapat mengukur seberapa baik model SVM dalam mengklasifikasikan sentimen ulasan aplikasi Mamikos.

3.2 Pelabelan Data

Setelah data dikumpulkan, langkah selanjutnya adalah memberikan label pada ulasan untuk mengkategorikan setiap ulasan yang diterima. Ulasan tersebut dikategorikan menjadi tiga kelas: positif, negatif, dan netral. Proses pelabelan dilakukan secara manual untuk memastikan *accuracy* dan ketepatan kategori yang diberikan.

Setelah pelabelan awal dilakukan, hasilnya divalidasi oleh Ibu Iqlima Naqia, S.S., M.Pd., seorang dosen Bahasa Indonesia di Universitas Nahdlatul Ulama Indonesia (UNUSIA). Validasi ini penting untuk memastikan bahwa pelabelan yang dilakukan telah sesuai dengan konteks dan makna dari setiap ulasan.

Setelah proses validasi selesai, penulis mencantumkan sampel data yang telah dilabeli secara manual untuk memberikan gambaran mengenai hasil pelabelan. Hasil ulasan yang telah dilabeli yang ditunjukkan pada tabel 3.1.

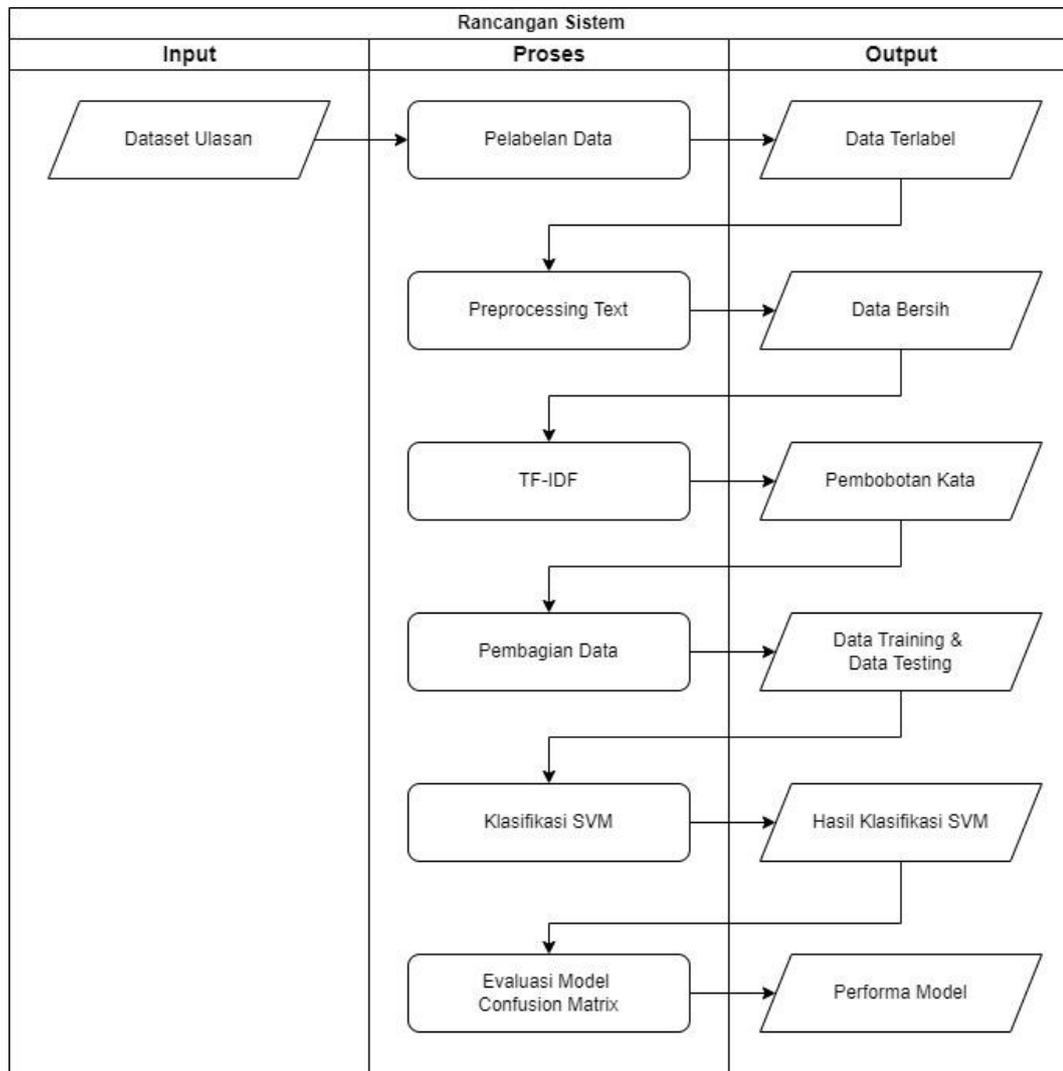
Tabel 3.1 Data Terlabel

Komentar	Kategori
Aplikasi sangat bagus untuk mencari kos bagi para perantau	Positif
Aplikasi gak jelas ..fitur search area apa yg muncul banyak area lalu filter harga dari sekian sampai sekian yg muncul banyak harga juga	Negatif
Aplikasi terjelek yg pernah saya temui, memaksa utk suruh harus update giliran diupdate tdk dapat mngunduhnya	Negatif
Banyak tempat pilihan , mahasiswa juga banyak cari dari aplikasi ini	Netral
Nggak perlu keluar jalan kaki panas panas muter	Netral
Aplikasi bagus buat pencarian sewa kos,kosnya nyaman,bersih Karyawannya ramah lokasinya strategis.	Positif
Mantap bgt nih aplikasinya, pas bgt buat aku mahasiswa yg lagi cari kos kosan dekat kampus aku. Sangat memmbantu sekali	Positif
Awal kerja jakarta nyari kos lewat mamikos	Netral
Aplikasi tipu tipu, kos cancel nggak ada pemberitahuan, cari lagi ternyata beda sama gambar	Negatif
Aplikasi Mamikos benar benar sangat memmbantu saya dalam mencari tempat kos-kosan daerah belum saya kenal	Positif

3.3 Rancangan Sistem

Sebelum membangun sistem dalam penelitian ini, diperlukan perancangan sistem untuk memudahkan jalannya proses penelitian. Tujuan utama dari klasifikasi data ulasan adalah untuk mengidentifikasi ulasan ke dalam tiga kategori: positif, negatif, dan netral. Selain itu, klasifikasi ini juga bertujuan untuk menilai tingkat akurasi ulasan aplikasi Mamikos dengan menggunakan algoritma *Support Vector Machine*.

Alur perancangan sistem pada penelitian ini dimulai dalam pengambil data ulasan dengan cara teknik *crawling* untuk periode 4 Februari 2022 hingga 14 September 2023. Proses ini mencakup beberapa langkah penting, mulai dari pengumpulan data ulasan, *preprocessing text*, pelabelan ulasan, hingga penggunaan SVM untuk mengklasifikasikan ulasan tersebut. Setiap langkah dirancang dengan tujuan untuk memastikan bahwa data yang digunakan akurat dan bahwa hasil klasifikasinya dapat diandalkan. Dengan perancangan sistem yang baik, diharapkan proses penelitian dapat berjalan dengan lancar dan menghasilkan model klasifikasi yang efektif dan akurat dalam mengidentifikasi sentimen ulasan pengguna aplikasi Mamikos. Alur proses klasifikasi yang dilakukan dalam penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Perancangan Sistem

Alur perancangan sistem dalam penelitian ini dimulai dengan pengambilan dataset ulasan aplikasi Mamikos menggunakan teknik *crawling*, yang mencakup periode dari 4 Februari 2022 hingga 14 September 2023. Setelah data terkumpul, tahap berikutnya adalah pelabelan data secara manual ke dalam tiga kategori: positif, negatif, dan netral. Pelabelan manual ini memastikan bahwa setiap ulasan dikategorikan dengan tepat berdasarkan sentimennya.

Setelah pelabelan, data kemudian melalui tahap *preprocessing* teks untuk memudahkan pengolahan lebih lanjut oleh sistem. Proses *preprocessing* ini meliputi lima langkah utama:

1. *Cleaning*, langkah ini menghapus informasi asing seperti emoticon, angka, tanda baca, simbol, tautan, dan tagar untuk fokus pada konten tekstual inti.
2. *Case folding*, untuk memastikan konsistensi dalam analisis kata, semua huruf diubah menjadi huruf kecil.
3. *Tokenizing*, kalimat dipecah menjadi kata-kata individual untuk diproses lebih lanjut.
4. *Stopwords removal*, kata-kata dengan makna minimal, seperti preposisi umum ("dari", "ke"), dihilangkan untuk meningkatkan fokus pada konten yang relevan.
5. *Stemming*, kata-kata direduksi menjadi bentuk akarnya untuk menangkap makna intinya dan meningkatkan efisiensi secara keseluruhan.

Setelah *preprocessing*, dilakukan pembobotan kata menggunakan metode TF-IDF. TF-IDF menghitung signifikansi setiap kata dalam dokumen. Hal ini dicapai dengan mempertimbangkan dua faktor: seberapa sering kata tersebut muncul dalam dokumen tertentu (TF) dan seberapa umum kata tersebut di seluruh kumpulan data (IDF). Dengan mengalikan faktor-faktor ini, TF-IDF memberikan bobot yang mencerminkan relevansi sebuah kata terhadap dokumen tertentu sambil memperhitungkan prevalensinya secara keseluruhan.

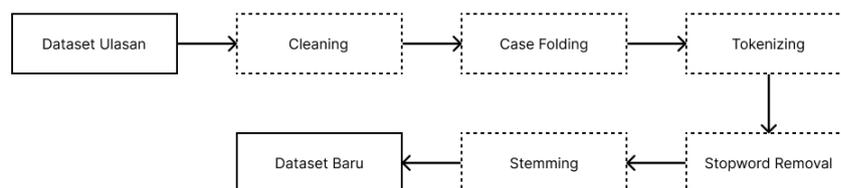
Langkah selanjutnya melibatkan klasifikasi sentimen menggunakan algoritma *Support Vector Machine*. Algoritma ini telah dilatih pada sekumpulan

ulasan berlabel, sehingga memungkinkannya mengkategorikan ulasan baru sebagai positif, negatif, atau netral berdasarkan sentimen yang diungkapkannya.

Tahap akhir sistem melibatkan evaluasi kinerjanya. Untuk mencapai hal ini, *confusion matrix* digunakan. Alat ini melakukan penghitungan *accuracy*, *precision*, *recall*, dan *f1-score*, yang memberikan wawasan tentang efektivitas model. Metode evaluasi ini membantu dalam menilai seberapa baik model SVM dalam mengklasifikasikan ulasan aplikasi Mamikos secara tepat dan efektif.

3.4 *Preprocessing Text*

Preprocessing text dilakukan untuk mengolah data mentah menjadi siap untuk dianalisis, dengan tujuan meningkatkan akurasi ulasan dalam sistem (Astari et al., 2020). Dalam penelitian ini, preprocessing teks melibatkan lima tahap yang dijelaskan dalam gambar 3.2.



Gambar 3.2 Tahapan *Preprocessing Text*

Tahap pertama adalah pembersihan (*cleaning*), yaitu proses menghapus karakter atau huruf yang tidak diperlukan dalam teks. Tujuan dari langkah ini adalah untuk membuat teks lebih bersih dan efisien dengan menghilangkan elemen seperti mention, hashtag, hyperlink, dan simbol-simbol lain yang tidak relevan.

Langkah kedua yaitu *case folding*, memastikan konsistensi dalam cara teks direpresentasikan. Hal ini dicapai dengan mengubah semua huruf besar menjadi huruf kecil. Ini membantu dalam menghindari duplikasi yang disebabkan oleh perbedaan antara huruf kapital dan huruf kecil (Astari et al., 2020).

Tahap ketiga adalah tokenisasi (*tokenizing*), yaitu proses memecah kalimat teks menjadi array token atau kata-kata tunggal. Setiap token dipisahkan oleh spasi, dan hanya alfabet yang digunakan sebagai tipe karakter (Muktafin et al., 2020).

Tahap keempat, *stopword removal* menghilangkan kata-kata yang sering muncul dalam teks namun kurang bermakna. Kata-kata ini, seringkali merupakan artikel atau preposisi, tidak memberikan kontribusi banyak pada konten inti. Contoh kata-kata ini termasuk "yang", "untuk", "pada", "ke", dan lain-lain. Menghapus stopwords membantu mengurangi kebisingan dalam data dan meningkatkan fokus pada kata-kata yang benar-benar penting.

Tahap kelima adalah *stemming*, berfokus pada pengurangan variasi bentuk kata. Hal ini dicapai dengan menghilangkan prefiks dan sufiks (imbuhan) dari kata-kata, yang bertujuan untuk merepresentasikan kata-kata dengan makna inti yang sama secara konsisten (Astari et al., 2020).

Proses-proses ini dirancang untuk memastikan bahwa data teks yang dihasilkan bersih, seragam, dan siap untuk dianalisis lebih lanjut menggunakan metode klasifikasi seperti *Support Vector Machine*. Dengan *preprocessing* yang efektif, sistem dapat lebih akurat dalam mengidentifikasi sentimen dalam ulasan aplikasi.

3.5 Pembobotan Kata

Setelah *preprocessing text*, data mengalami transformasi ke dalam format numerik melalui pembobotan kata. Proses ini memberikan bobot pada setiap kata, yang pada dasarnya mencerminkan signifikansinya untuk analisis. Umumnya, lebih banyak data berarti kumpulan fitur yang lebih besar (kata-kata berbobot), yang pada akhirnya meningkatkan kemampuan model untuk menangkap variasi dan nuansa halus dalam kumpulan data.

Tahap ini melibatkan pada dua konsep utama: TF dan IDF (Septian et al., 2019). TF menghitung seberapa sering sebuah kata muncul dalam dokumen tertentu. Sebaliknya, IDF menilai signifikansi sebuah kata di seluruh kumpulan data. Dengan menggabungkan nilai-nilai ini, TF-IDF memberikan bobot yang mencerminkan pentingnya sebuah kata dalam dokumen tertentu sambil mempertimbangkan prevalensinya di seluruh pengumpulan data.

Pembobotan kata hal yang penting karena memungkinkan model memprioritaskan kata yang paling informatif dan relevan dalam teks. Fokus ini pada akhirnya mengarah pada klasifikasi sentimen ulasan yang lebih akurat. Dengan mengubah teks menjadi data numerik yang berbobot, sistem dapat lebih efektif dalam menganalisis dan memahami pola dalam data ulasan aplikasi Mamikos.

Untuk contoh menghitung pembobotan kata, dilakukan perhitungan manual menggunakan tiga data sampel yang disajikan pada tabel 3.2. Perhitungan ini menunjukkan bagaimana setiap kata dalam sampel diberi bobot berdasarkan frekuensi kemunculannya dan relevansinya di seluruh dataset. Dengan cara ini,

dapat mengidentifikasi kata-kata yang paling signifikan dalam ulasan, yang kemudian digunakan sebagai fitur dalam model klasifikasi.

Tabel 3.2 Contoh Data Ulasan

Dokumen	Komentar
D1	“aplikasi rekom cari info kosan”
D2	“aplikasi ribet simpel cari kos grup fb”
D3	“makasih mamikos udah bantu iklan kosan kosan penuh”

3.5.1 Nilai *Term Frequency*

Langkah awal TF-IDF melibatkan penghitungan *Term Frequency* (TF). Matriks ini mencerminkan seberapa sering kata tertentu muncul dalam dokumen. Pada dasarnya, TF mengukur keunggulan kata tersebut dalam konteks tertentu.

TF (*Term frequency*) dihitung dengan membagi berapa kali kata tertentu muncul dalam dokumen dengan jumlah kata total dokumen tersebut. Proses normalisasi ini memastikan bahwa kata-kata yang sering muncul dalam dokumen menerima nilai TF yang lebih tinggi. Sederhananya, ini memberi bobot lebih pada kata-kata yang lebih sering muncul dalam ulasan tertentu.

Menghitung nilai TF memberikan pemahaman awal tentang distribusi frekuensi kata suatu dokumen. Ini berfungsi sebagai dasar untuk proses pembobotan kata, yang memberikan arti penting pada kata-kata untuk analisis selanjutnya. Langkah ini penting untuk memastikan bahwa setiap kata diberikan bobot yang sesuai berdasarkan frekuensi kemunculannya, yang kemudian digunakan dalam perhitungan IDF untuk mendapatkan nilai TF-IDF yang lengkap.

Persamaan 3.1 menampilkan rumus yang digunakan untuk menghitung TF . Selain itu, Tabel 3.3 menampilkan contoh penghitungan nilai TF untuk berbagai

kata dalam contoh dokumen. Tabel ini menunjukkan bagaimana TF dihitung untuk kata-kata tertentu, menyoroti frekuensi relatif setiap kata dibandingkan dengan keseluruhan teks

$$TF = \frac{\text{Jumlah kemunculan kata dalam dokumen}}{\text{Jumlah kata dalam satu dokumen}} \quad (3.1)$$

Tabel 3.3 Tabel Perhitungan TF

Term	D1	D2	D3	Nilai TF D1	Nilai TF D2	Nilai TF D3
Aplikasi	1	1	0	0,2	0,142857143	0
Rekom	1	0	0	0,2	0	0
Cari	1	1	0	0,2	0,142857143	0
Info	1	0	0	0,2	0	0
Kosan	1	0	2	0,2	0	0,222222222
Ribet	0	1	0	0	0,142857143	0
Simpel	0	1	0	0	0,142857143	0
Kos	0	1	0	0	0,142857143	0
Grup	0	1	0	0	0,142857143	0
Fb	0	1	0	0	0,142857143	0
Makasih	0	0	1	0	0	0,111111111
Mamikos	0	0	1	0	0	0,111111111
Udah	0	0	1	0	0	0,111111111
Bantu	0	0	1	0	0	0,111111111
Iklan	0	0	1	0	0	0,111111111
Penuh	0	0	1	0	0	0,111111111

3.5.2 Nilai *Inverse Document Frequency*

Langkah selanjutnya dalam komputasi TF-IDF berfokus pada IDF. IDF menilai signifikansi sebuah kata di seluruh koleksi dokumen. Kata-kata yang sering muncul di berbagai dokumen menerima nilai IDF yang lebih rendah, yang berarti kurang penting. Sebaliknya, kata-kata yang lebih jarang muncul diberi nilai IDF yang lebih tinggi, yang menandakan signifikansi yang lebih besar.

Nilai IDF membantu membedakan kata mana yang lebih spesifik dan unik dalam sebuah dokumen. Hal ini dicapai dengan membandingkan frekuensinya dalam satu dokumen dengan prevalensinya di seluruh koleksi. Kata-kata yang

umum muncul di banyak dokumen menerima nilai IDF yang lebih rendah, yang menunjukkan berkurangnya kekhasan. Sebaliknya, kata-kata khusus untuk dokumen yang ada akan memiliki nilai IDF yang lebih tinggi, sehingga menonjolkan keunikannya. Langkah ini melengkapi perhitungan TF dengan memberikan konteks yang lebih luas mengenai pentingnya setiap kata dalam kumpulan data. Nilai TF dan IDF kemudian digabungkan untuk menghasilkan skor akhir TF-IDF. Skor ini berfungsi sebagai bobot tepat yang ditetapkan untuk setiap kata, yang mencerminkan frekuensi kata tersebut dalam dokumen tertentu dan kepentingannya secara keseluruhan di seluruh kumpulan data.

Perhitungan IDF didasarkan pada rumus yang menggabungkan dua faktor utama: jumlah total dokumen dalam kumpulan data dan jumlah dokumen tempat kata tertentu muncul. Rumus ini membantu menentukan seberapa umum sebuah kata di seluruh koleksi, yang pada akhirnya memengaruhi bobotnya dalam analisis. Rumus ini dapat dilihat pada persamaan 3.2. Contoh perhitungan nilai IDF untuk beberapa kata disajikan pada tabel 3.4, yang menunjukkan bagaimana nilai ini dihitung berdasarkan distribusi kata-kata di seluruh dokumen dalam dataset.

$$IDF(t, D) = \text{Log}\left(\frac{N}{df(t)}\right) \quad (3.2)$$

Keterangan :

- $IDF_{(t,D)}$ = nilai IDF untuk kata kunci (*term*) t dalam dokumen D
- N = Jumlah total dokumen
- $df_{(t)}$ = Jumlah dokumen yang mengandung kata kunci (*term*) t

Tabel 3.4 Tabel Perhitungan IDF

Term	IDF
Aplikasi	0,176091
Rekom	0,477121

Cari	0,176091
Info	0,477121
Kosan	0,477121
Ribet	0,477121
Simpel	0,477121
Kos	0,477121
Grup	0,477121
Fb	0,477121
Makasih	0,477121
Mamikos	0,477121
Udah	0,477121
Bantu	0,477121
Iklan	0,477121
Penuh	0,477121

3.5.3 Nilai *Term Frequency - Inverse Document Frequency*

Pada langkah terakhir untuk mengetahui seberapa penting kata-kata, menghitung skor yang disebut TF-IDF. Skor ini mempertimbangkan dua hal: berapa kali sebuah kata muncul dalam dokumen tertentu dan seberapa jarang kata tersebut muncul di seluruh dokumen yang dilihat.

Dengan menggunakan nilai TF-IDF, kita dapat memastikan bahwa model analisis teks lebih akurat dalam mengidentifikasi kata-kata yang paling penting dan relevan dalam dokumen. Ini membantu dalam meningkatkan kualitas klasifikasi sentimen dan analisis teks secara keseluruhan.

Menemukan pentingnya kata-kata dalam sebuah dokumen melibatkan mengalikan dua skor: TF dan IDF. Perhitungan ini, seperti yang ditunjukkan pada persamaan 3.3, memberikan bobot lebih pada kata-kata yang umum dalam dokumen tertentu namun tidak umum di seluruh kumpulan dokumen. Kata-kata ini kemungkinan besar bermakna untuk dokumen tertentu.

Contoh perhitungan nilai TF-IDF untuk beberapa kata dapat dilihat pada tabel 3.5. Tabel ini menunjukkan bagaimana nilai TF dan IDF dikombinasikan

untuk menghasilkan bobot TF-IDF, memberikan gambaran tentang bagaimana setiap kata diberi bobot berdasarkan frekuensi kemunculannya dan relevansinya dalam keseluruhan dataset.

$$\text{TF-IDF}(t,d,D) = \text{TF}(t,d) \times \text{IDF}(t,D) \quad (3.3)$$

Keterangan :

- $\text{TF-IDF}_{(t,d,D)}$ = Pembobotan TF-IDF
- $\text{TF}_{(t,d)}$ = Jumlah kemunculan *term* dalam dokumen
- $\text{IDF}_{(t,D)}$ = Bobot inverse dalam nilai df

Tabel 3.5 Perhitungan TF-IDF

Term	TF-IDF		
	D1	D2	D3
Aplikasi	0,035218	0,025156	0
Rekom	0,095424	0	0
Cari	0,035218	0,025156	0
Info	0,095424	0	0,039131
Kosan	0,035218	0	0
Ribet	0	0,06816	0
Simpel	0	0,06816	0
Kos	0	0,06816	0
Grup	0	0,06816	0
Fb	0	0,06816	0,053013
Makasih	0	0	0,053013
Mamikos	0	0	0,053013
Udah	0	0	0,053013
Bantu	0	0	0,053013
Iklan	0	0	0,053013
Penuh	0	0	0,053013

3.6 Support Vector Machine

Penelitian ini menggunakan metode *Support Vector Machine* dengan model linear untuk mengklasifikasikan sentimen menjadi positif, negatif, dan netral menggunakan *hyperplane*. Prinsip dasar SVM dalam klasifikasi data adalah menentukan *hyperplane* terbaik yang mampu memisahkan kelas-kelas yang telah ditentukan dengan jelas.

Cara kerja SVM adalah dengan mencari *hyperplane* yang optimal, yang memberikan pemisah maksimum antara kelas-kelas tersebut. *Hyperplane* ini memiliki margin maksimum, yang didefinisikan sebagai jarak antara titik data terdekat dari setiap kelas dengan *hyperplane* itu sendiri. SVM berupaya memaksimalkan margin ini untuk memastikan bahwa pemisahan antar kelas sebaik mungkin.

Rumus untuk menentukan *hyperplane* ditunjukkan pada persamaan 3.4. Dengan memaksimalkan margin, SVM memastikan bahwa model yang dihasilkan tidak hanya memisahkan data dengan benar tetapi juga memiliki generalisasi yang baik untuk data baru.

$$w_i x_i + b = 0 \quad (3.4)$$

Dalam persamaan 3.4 diketahui bahwa w_i merupakan bobot kata atau *term*, x_i merupakan parameter nilai dari masing-masing *term*, dan b merupakan bias. Berdasarkan dari persamaan 3.4, diketahui persamaan *hyperlane* untuk masing-masing kelas yang ditunjukkan pada persamaan 3.5.

$$w_i x_i + b \begin{cases} \geq 1 \text{ untuk } y = +1 \\ \leq -1 \text{ untuk } y = -1 \end{cases} \quad (3.5)$$

Model linear linear mengklasifikasikan dokumen dengan cara menghitung nilai parameter pada setiap dokumen yang ditunjukkan pada persamaan 3.6

$$f(y) = \sum_{i=1}^n w_i \cdot x_i + b \quad (3.6)$$

Dalam persamaan 3.6 diketahui bahwa w_i merupakan bobot pada kata, x_i merupakan nilai dari setiap fitur yang dilakukan pada tahap TF-IDF, dan b adalah bias. Dalam penelitian ini, dikatakan bahwa nilai $\sum_{i=1}^n w_i = 1$, sehingga dapat dikatakan bahwa perhitungan untuk mencari inisialisasi bobot w_i yang ditunjukkan pada persamaan 3.7.

$$w_i = \frac{1}{n} \quad (3.7)$$

Dalam persamaan 3.7 diketahui bahwa w_i merupakan nilai bobot inisial yang akan digunakan pada proses selanjutnya, sedangkan n adalah jumlah *term* yang diperoleh dari hasil seleksi fitur sebelumnya. Selain nilai w_i nilai bias juga dibutuhkan dalam melakukan pemodelan yang ditunjukkan pada persamaan 3.8.

$$b = \frac{1}{n+1} \quad (3.8)$$

Dari persamaan 3.7 dan 3.8 didapatkan nilai w_i serta nilai bias yang ditunjukkan dibawah ini.

$$w_i^0 = \frac{1}{16} = 0,0625$$

$$b^0 = \frac{1}{16 + 1} = 0,058$$

Dari rumus $f(y)$ yang dihasilkan dari persamaan 3.6, telah ditetapkan jika $f(y) < 0$ maka $y_i = -1$, sebaliknya, jika $f(y) > 0$ maka $y_i = +1$.

$$\begin{aligned}
 f(y) &= ((0,0625 \times 0,035218) + (0,0625 \times 0,095424) \\
 &\quad + (0,0625 \times 0,035218) + (0,0625 \times 0,095424) \\
 &\quad + (0,0625 \times 0,035218) + (0,0625 \times 0) + \dots + (0,0625 \times xn)) \\
 &\quad + 0,058
 \end{aligned}$$

$$f(y) = (0,0022 + 0,0059 + 0,0022 + 0,0059 + 0,0022 + 0 + \dots + 0) + 0,058$$

$$f(y) = 0,0764$$

$$f(y) > 0, \text{ maka } y = +1$$

Hasil dari perhitungan yang telah dilakukan mendapatkan nilai sebesar 0,0764. Nilai tersebut menunjukkan bahwa $y = +1$, maka hasil prediksi yang didapatkan pada D1 yang artinya masuk ke dalam kategori kelas positif. Proses iterasi terus berjalan dan nilai bobot diperbarui yang ditunjukkan pada persamaan 3.9.

$$w_i = w_0 + y_i \cdot x_{ij} \quad (3.9)$$

Data pelatihan digunakan untuk melatih sistem dan menghasilkan model klasifikasi SVM. Model ini kemudian diuji menggunakan data pengujian, di mana nilai parameter fitur dari data *training* digunakan sebagai input. Sistem menentukan kelas data pengujian berdasarkan nilai yang dihasilkan dari persamaan 3.7. Teks akan dikategorikan sebagai kelas negatif jika nilainya $f(y) \leq 0$, dan sebagai kelas positif jika nilainya $f(y) \geq 0$.

Setelah model SVM dilatih dengan data pelatihan, model ini diuji dengan menggunakan data pengujian untuk mengevaluasi kinerjanya. Data pengujian ini terdiri dari ulasan yang belum pernah dilihat oleh model selama proses pelatihan,

sehingga memberikan gambaran yang lebih akurat tentang kemampuan model dalam mengklasifikasikan data baru.

Nilai parameter fitur yang diperoleh dari data pelatihan digunakan sebagai input untuk model SVM selama pengujian. Berdasarkan persamaan 3.7, model SVM menghasilkan nilai output $f(y)$ yang digunakan untuk menentukan kategori sentimen dari teks yang diuji. Jika nilai $f(y)$ lebih kecil atau sama dengan nol, teks tersebut dikategorikan sebagai kelas negatif. Sebaliknya, jika nilai $f(y)$ lebih besar atau sama dengan nol, teks tersebut dikategorikan sebagai kelas positif.

Dengan metode ini, SVM mampu mengklasifikasikan teks secara efektif berdasarkan sentimen, memungkinkan sistem untuk secara akurat mengidentifikasi ulasan sebagai positif, negatif, atau netral. Menganalisis hasil klasifikasi ini memberikan informasi penting mengenai efektivitas model dan membuka jalan untuk menyempurnakan algoritma guna mencapai akurasi yang lebih baik.

3.7 Skenario Pengujian

Uji coba sistem dilakukan dengan melatih model menggunakan data pelatihan melalui algoritma *Support Vector Machine*, kemudian menguji kinerjanya dengan data pengujian. Untuk mengidentifikasi pemisahan optimal data pelatihan dan pengujian yang menghasilkan hasil klasifikasi teks paling akurat, data dibagi ke dalam berbagai rasio pada tahap pengujian awal. Rasio ini mencakup 90:10 (90% untuk *training* dan 10% untuk *testing*), 80:20, 70:30, dan 60:40. Dengan menggunakan serangkaian rasio terpisah dengan bertujuan untuk menentukan kombinasi yang menghasilkan model paling akurat.

Setelah pengujian awal, selanjutnya menggunakan *K-fold cross validation* untuk menilai lebih lanjut akurasi model. *K-fold cross validation* adalah teknik untuk mengevaluasi model, yang melibatkan pemisahan data dan menggunakan setiap lipatan untuk pengujian, sedangkan lipatan sisanya digunakan untuk pelatihan (Larasati et al., 2022). Proses ini diulangi untuk setiap lipatan. Dalam hal ini pengujian dilakukan sebanyak empat kali dengan nilai k yang bervariasi (5, 10, 15, dan 20) untuk menganalisis bagaimana pengaruh jumlah lipatan terhadap akurasi yang diperoleh.

K-fold cross validation, yang divisualisasikan pada gambar 3.3, adalah teknik yang digunakan untuk menilai akurasi klasifikasi model. Dengan metode ini, seluruh 1049 data digunakan untuk pelatihan dan pengujian menggunakan pengklasifikasi *Support Vector Machine*. Prosedur pelatihan dan pengujian diulangi sebanyak jumlah lipatan (k). Selama setiap iterasi, satu lipatan bertindak sebagai data pengujian sedangkan lipatan lainnya digunakan untuk pelatihan. Proses ini memastikan evaluasi kinerja model yang lebih kuat.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Iteration 1	Testing	Training	Training	Training	Training
Iteration 2	Training	Testing	Training	Training	Training
Iteration 3	Training	Training	Testing	Training	Training
Iteration 4	Training	Training	Training	Testing	Training
Iteration 5	Training	Training	Training	Training	Testing

Gambar 3.3 Visualisasi *5-fold cross validation*

Gambar 3.3 mengilustrasikan teknik dengan k-5. Metode *k-fold cross validation* melibatkan pemisahan kumpulan data dari 1049 data menjadi 5 lipatan,

masing-masing berisi sekitar 210 data. Pembagian ini diulangi sebanyak 5 kali untuk evaluasi yang lebih kuat.

Dengan menggunakan metode ini, setiap subset data secara bergantian digunakan untuk pengujian dan pelatihan, memastikan bahwa setiap data point diuji dan dilatih, yang membantu dalam memberikan estimasi akurasi yang lebih andal. Hal ini memungkinkan evaluasi menyeluruh terhadap performa model dan membantu dalam menentukan pengaturan optimal untuk mencapai akurasi terbaik dalam klasifikasi sentimen ulasan aplikasi Mamikos.

3.8 Evaluasi Performa

Untuk tahap terakhir penelitian ini, akan menilai efektivitas model menggunakan *Confusion Matrix*. *Confusion matrix* sangat penting untuk mengevaluasi kinerja model dengan membandingkan hasil prediksi dengan klasifikasi sebenarnya (Husada & Paramita, 2021). *Confusion Matrix* memberikan nilai untuk *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. Nilai-nilai ini kemudian digunakan untuk menghitung *matrix accuracy*, *precision*, *recall*, dan *f1-score*. Evaluasi ini menawarkan pemahaman menyeluruh tentang kemampuan klasifikasi model, menunjukkan dengan tepat kekuatan, kelemahan, dan area yang perlu ditingkatkan. Pada gambar 3.4 menunjukkan konsep dari penggunaan *confusion matrix*.

		True Class		
		Class 1	Class 2	Class 3
Predicted Class	Class 1	TP	FP	FP
	Class 2	FN	TN	TN
	Class 3	FN	TN	TN

Gambar 3.3 Visualisasi *Confusion Matrix*

Dimana nilai,

- TP : Data memiliki nilai positif dan model memprediksi sebagai data positif
- FP : Data memiliki nilai negatif dan model memprediksi sebagai data positif
- FN : Data memiliki nilai positif dan model memprediksi sebagai data negatif
- TN : Data memiliki nilai negatif dan model memprediksi sebagai data negatif

Rumus yang digunakan dalam menghitung performa klasifikasi dengan *confusion matrix* dapat dilihat pada persamaan 3.10, 3.11, 3.12, dan 3.13.

1. *Accuracy* merupakan pengukuran jumlah prediksi yang benar sesuai data secara keseluruhan.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3.10)$$

2. *Precision* merupakan persentase dokumen dengan teks yang relevan di antara semua dokumen yang telah dipilih oleh sistem.

$$Precision = \frac{TP}{TP+FP} \quad (3.11)$$

3. *Recall* merupakan perkiraan jumlah dokumen teks yang relevan terkendali diantara semua dokumen teks relevan yang terdapat pada data (Husada & Paramita, 2021)

$$Recall = \frac{TP}{TP+FN} \quad (3.12)$$

4. *F1-Score* merupakan proses penggabungan *precision* dan *recall* pada matriks evaluasi.

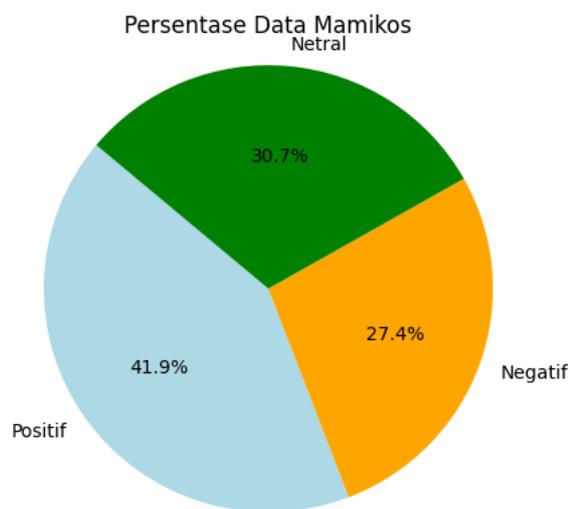
$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (3.13)$$

BAB IV

UJI COBA DAN PEMBAHASAN

4.1 Data Penelitian

Penelitian ini menggunakan 1049 ulasan yang dikumpulkan dari *Google Play Store* untuk aplikasi Mamikos dalam rentang waktu 4 Februari 2022 hingga 14 September 2023. Setelah mengumpulkan data ulasan, langkah berikutnya adalah melabeli data yang kemudian divalidasi oleh seorang dosen Bahasa Indonesia. Dari hasil pelabelan, didapatkan 440 ulasan positif, 287 ulasan negatif, dan 322 ulasan netral. Visualisasi perbandingan antara ulasan positif, negatif, dan netral ditunjukkan pada gambar 4.1 dalam bentuk *pie chart*.



Gambar 4.1 *Pie Chart* Perbandingan Kelas Sentimen

Dari gambar 4.1 hasil ulasan positif sebesar 41,9%, ulasan negatif 27,4%, dan ulasan netral 30,7% dari jumlah data yang digunakan. Hasil pelabelan data yang sudah divalidasi oleh dosen bahasa Indonesia Universitas Nahdlatul Ulama

Indonesia (UNUSIA) bernama Ibu Iqlima Naqia, S.S., M.Pd dapat dilihat pada tabel 4.1.

Tabel 4.1 Sampel Data Ulasan

No	Komentar	Kategori
1	Aplikasi ini bagus memmbantu banget untuk mencari rekomendasi kos yg mau sewa. 👍👍👍	Positif
2	Sangat kecewa dengan pihak mamikos, hari ini saya batalkan oleh pihak mamikos saya harus menanggung kerugian cukup besar.	Negatif
3	Sebagai mahasiswa baru lulus lalu langsung diterima kerja yg jauh dari rumah cocok buat cari kos terdekat dengan kerjaan.	Netral
4	Mamikos adalah tempat mencari kos-kosan.	Netral
5	Aplikasi ini sangat berguna memudahkan bagi orang mau cari kos tanpa ragu ² .	Netral
6	Aplikasi gak jelas!	Negatif
7	Terimakasih mami kost sudah mempermudah difabel seperti saya dalam mencari kost. ☺☺	Positif
8	Aplikasinya buruk bgt,malah lebih cepetan kl buka dibrowser.	Negatif
9	Jujur kecewa sama mamikos krn tambah ribet aturannya.	Negatif
10	Kecewa sama aplikasi ini karena sering kasih alamat yg fake.	Negatif
11	Aplikasi ini rekomendasi banget buat yg nyari kos.	Positif
12	Aplikasi sangat keren, sangat membantu cara pakai aplikasi juga sangat mudah.	Positif
13	Aplikasi sangat bagus untuk mencari kos bagi para perantau.	Positif
14	Lumayan ga susah nyari kontrakan	Netral
15	Saya mahasiswa perantauan, kebetulan saya tuh suka pindah ² kos, kaga kesulitan cari tempat pas nyaman.	Netral
16	Aplikasi terjelek yg pernah saya temui, memaksa utk suruh harus update giliran diupdate tdk dapat mngunduhnya.	Negatif
17	Aplikasinya ribet, lebih simpel cari tempat kos di grup FB	Negatif
18	Sangat membantu dalam mencari kos	Positif
19	Sangat bagus sekali aplikasi ini, membantu kita mencari rumah kost yang cocok dan nyaman untuk kita. Sangat Rekomendasi sekali	Positif
20	Kenapa harus foto selfie dengan KTP?	Netral

4.2 Skenario Uji Coba

Tahapan uji coba dalam penelitian ini mencakup berbagai skenario yang dirancang untuk menguji sistem secara komprehensif. Langkah pertama adalah pengambilan data menggunakan teknik *crawling* untuk mengumpulkan ulasan aplikasi Mamikos dari *Google Play Store*. Setelah data terkumpul, dilakukan preprocessing untuk memastikan data dalam kondisi optimal untuk analisis lebih lanjut.

Berdasarkan pembahasan metode pengujian pada bab sebelumnya, penelitian ini menggunakan dua pendekatan: *split* data dan *k-fold cross validation*. Dalam metode *split* data, data dipisahkan menjadi set pelatihan dan pengujian. Dataset menggunakan skenario pembagian 60:40, 70:30, 80:20, dan 90:10, dimana 60% untuk *training* dan 40% untuk *testing*.

K-fold cross validation memberikan cara untuk menilai kinerja model menggunakan semua data yang tersedia tanpa menimbulkan bias. Dalam metode ini, data dibagi menjadi 'k' lipatan yang sama, k=5, 10, 15, 20 (Tuntun et al., 2022). Selama setiap iterasi satu lipatan digunakan untuk menguji model, sedangkan lipatan sisanya digunakan untuk pelatihan. Hal ini memastikan semua titik data digunakan untuk pelatihan dan pengujian, meminimalkan bias dan memberikan evaluasi yang lebih komprehensif.

Setelah model dilatih dan diuji dengan kedua metode tersebut, tahap selanjutnya adalah evaluasi performa model untuk analisis sentimen. Untuk menilai performa model, menggunakan *confusion matriks*. Evaluasi ini membandingkan prediksi model positif, negatif, netral dengan label ulasan sebenarnya. Dari perbandingan ini, *confusion matriks* menghasilkan *accuracy*, *precision*, *recall*, dan *f1-score*, yang memberikan gambaran komprehensif tentang efektivitas model.

Confusion matriks memberikan gambaran tentang kemampuan model untuk mengkategorikan data secara akurat. *Accuracy*, matriks paling dasar, memberi tahu proporsi keseluruhan klasifikasi yang benar (positif, negatif, netral). *Precision* menggali lebih dalam, dengan fokus pada proporsi ulasan positif yang

diidentifikasi model dengan benar. Di sisi lain, *recall* menyoroti kemampuan model untuk menemukan ulasan positif aktual, dan menunjukkan berapa banyak ulasan yang tidak terlewatkan. Terakhir, *f1-score* menghasilkan keseimbangan, menggabungkan presisi dan perolehan menjadi satu matriks untuk pemahaman yang lebih mendalam tentang performa model.

Melalui evaluasi yang komprehensif ini, penelitian bertujuan untuk mengidentifikasi kekuatan dan kelemahan model klasifikasi, serta melakukan penyesuaian yang diperlukan untuk meningkatkan akurasi dan efisiensi dalam analisis sentimen ulasan aplikasi Mamikos.

4.2.1 Preprocessing Text

Preprocessing data adalah langkah pertama yang penting dalam menyiapkan data tinjauan untuk klasifikasi sentimen. Melibatkan beberapa teknik seperti *cleaning*, *case folding*, *tokenizing*, *stopwords removal*, dan *stemming*. Semua langkah ini bekerja sama untuk memastikan model berfokus pada bagian tinjauan yang paling relevan untuk analisis sentimen yang akurat.

Langkah pertama adalah *cleaning*, yaitu proses membersihkan teks dari karakter atau elemen yang tidak relevan seperti tanda baca, karakter khusus, atau tag HTML yang sering ditemukan dalam teks web. *Cleaning* juga mencakup penghapusan spasi ekstra atau karakter ganda yang dapat mengganggu analisis.

Proses *cleaning* bertujuan untuk memastikan teks yang digunakan dalam analisis adalah teks yang bersih dan siap untuk tahap-tahap berikutnya. Implementasi dari tahap *cleaning* dapat dilihat pada gambar 4.2, dan hasil dari proses *cleaning* ini ditampilkan pada tabel 4.2.

```

function cleaningkos(text)
    hasil = hasil.translate(str.maketrans("", "",
    string.punctuation))
    hasil = hasil.strip()
    return hasil
end function

```

Gambar 4.2 Pseudocode tahap *Cleaning*Tabel 4.2 Hasil *Preprocessing Cleaning*

Sebelum Cleaning	Sesudah Cleaning
Terimakasih mamikos, dengan adanya aplikasi ini sangat membantu memudahkan saya mencari tempat kos yang nyaman	Terimakasih mamikos dengan adanya aplikasi ini sangat membantu memudahkan saya mencari tempat kos yang nyaman

Pada analisis tahap kedua *case folding*, proses yang disebut pelipatan huruf mengubah semua huruf menjadi huruf kecil. Hal ini memastikan konsistensi dengan memperlakukan kata-kata seperti "Teks" dan "teks" sebagai hal yang sama, terlepas dari penggunaan huruf kapital. Hal ini menyederhanakan analisis dengan berfokus pada arti kata, bukan pada variasi kapitalisasinya. Implementasi *case folding* dapat dilihat pada gambar 4.3. Adapun hasil yang didapatkan ditunjukkan pada tabel 4.3.

```

function casefoldingkos(text)
    casefolding = text.lower()
    hasil = re.sub(r"\d+", "", casefolding)
    return hasil
end function

```

Gambar 4.3 Pseudocode tahap *Case Folding*Tabel 4.3 Hasil *Preprocessing Case Folding*

Sebelum Case Folding	Sesudah Case Folding
Terimakasih mamikos dengan adanya aplikasi ini sangat membantu memudahkan saya mencari tempat kos yang nyaman	terimakasih mamikos dengan adanya aplikasi ini sangat membantu memudahkan saya mencari tempat kos yang nyaman

Tahap ketiga melibatkan tokenizing. Di sini, teks dibagi menjadi unit-unit dasar yang disebut "token." Token ini dapat berupa kata individual, kombinasi

kata (frasa), atau karakter tunggal. *Tokenizing* biasanya dilakukan dengan membagi teks menjadi kata-kata individu, membuat teks lebih mudah untuk diproses dan dianalisis. Implementasi *tokenizing* dapat dilihat pada gambar 4.4. Adapun hasil yang didapatkan ditunjukkan pada tabel 4.4.

```
import nltk
nltk.download('all')
from nltk.tokenize import word_tokenize
def tokenizingkos(text):
    tokens = nltk.tokenize.word_tokenize(text)
    return tokens
```

Gambar 4.4 Pseudocode Tahap *Tokenizing*

Tabel 4.4 Hasil *Preprocessing Tokenizing*

Sebelum Tokenizing	Sesudah Tokenizing
terimakasih mamikos dengan adanya aplikasi ini sangat membantu memudahkan saya mencari tempat kos yang nyaman	“terimakasih”, “mamikos”, “dengan”, “adanya”, “aplikasi”, “ini”, “sangat”, “membantu”, “memudahkan”, “saya”, “mencari”, “tempat”, “kos”, “yang”, “nyaman”

Langkah keempat melakukan pada *stopwords removal*. Proses ini menghilangkan kata-kata umum, seperti preposisi dan artikel, yang tidak memiliki arti penting dan tidak berkontribusi besar dalam memahami konten secara keseluruhan. Implementasi *stopwords removal* dapat dilihat pada gambar 4.5. Adapun hasil yang didapatkan ditunjukkan pada tabel 4.5.

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()
custom_stopwords_file = 'skripsistopword.txt'
with open(custom_stopwords_file, 'r', encoding='utf-8') as
file:
    custom_stopwords = file.read().splitlines()
stopwords.extend(custom_stopwords)
def stopwordskos(text):
    return [word for word in text if word not in stopwords]
```

Gambar 4.5 Pseudocode tahap *Stopwords Removal*

Tabel 4.5 Hasil *Preprocessing Stopwords Removal*

Sebelum Stopwords Removal	Sesudah Stopwords Removal
terimakasih mamikos dengan adanya aplikasi ini sangat membantu memudahkan saya mencari tempat kos yang nyaman	terimakasih mamikos aplikasi membantu memudahkan mencari kos nyaman

Tahap kelima adalah *stemming*, yaitu proses menghapus akhiran atau awalan kata sehingga hanya kata dasarnya yang tersisa. *Stemming* memperpendek kata dengan menghilangkan awalan dan akhiran, sehingga menghasilkan bentuk kata yang lebih umum. Ini membantu mengidentifikasi kata-kata dengan akar kata yang sama sebagai padanannya, meskipun kata-kata tersebut sedikit berbeda dalam tense atau konjugasi. Implementasi *stemming* dapat dilihat pada gambar 4.6. Adapun hasil yang didapatkan ditunjukkan pada tabel 4.6

```

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()
def stem_word(word):
    return stemmer.stem(word)
def stem_text(text):
    # Convert the list elements to strings
    text = ' '.join(map(str, text))
    words = text.split() # Split the string into words
    stemmed_words = [stem_word(word) for word in words]
    return ' '.join(stemmed_words)

```

Gambar 4.6 *Pseudocode Tahap Stemming*Tabel 4.6 Hasil *Preprocessing Stemming*

Sebelum Stemming	Sesudah Stemming
“terimakasih”, “mamikos”, “aplikasi”, “membantu”, “memudahkan”, “mencari”, “kos”, “nyaman”	“terimakasih”, “mamikos”, “aplikasi”, “bantu”, “mudah” “cari”, “kos”, “nyaman”

4.2.2 Pembobotan Kata TF-IDF

Setelah tahap *preprocessing text*, ditentukan bobot kata menggunakan metode TF-IDF. TF, atau term frekuensi, dihitung terlebih dahulu untuk mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen.

Selanjutnya, IDF, atau frekuensi dokumen terbalik, dihitung untuk mengukur pentingnya kata di seluruh kumpulan dokumen. Terakhir, bobot TF-IDF diperoleh dengan mengalikan nilai TF dan IDF.

Pembobotan kata dengan TF-IDF dilakukan untuk memberikan bobot yang mencerminkan pentingnya setiap kata dalam konteks dokumen tertentu dan dalam keseluruhan kumpulan data. Dalam penelitian ini, proses pembobotan dilakukan dengan menggunakan library *TfidfVectorizer* dari Python. Library ini mempermudah proses perhitungan dengan secara otomatis mengonversi teks menjadi representasi numerik berdasarkan nilai TF-IDF, sehingga setiap kata dalam dokumen mendapatkan bobot yang sesuai.

Proses ini memprioritaskan kata-kata yang sering muncul dalam satu dokumen tetapi jarang ditemukan di seluruh kumpulan data. Memberikan bobot yang lebih tinggi pada kata-kata ini akan membuat kata-kata tersebut lebih berpengaruh dalam analisis. Dengan menggunakan TF-IDF, model ini unggul dalam mengenali kata-kata dan pola penting dalam data. Hal ini pada akhirnya menghasilkan klasifikasi sentimen yang lebih tepat dalam analisis ulasan aplikasi Mamikos.

4.3 Pemodelan Algoritma *Support Vector Machine*

Support Vector Machine adalah model yang ampuh untuk menganalisis data karena dapat menangani kumpulan data kompleks dengan banyak fitur. SVM mencapai hal ini dengan menemukan garis pemisah yang jelas (hyperplane) antara berbagai kategori dalam data. Untuk masalah klasifikasi, SVM dapat

menggunakan fungsi kernel untuk mengubah data menjadi ruang berdimensi lebih tinggi, sehingga pemisahan antar kategori lebih mudah dilakukan.

Efektivitas model SVM bergantung pada pemilihan fungsi kernel yang sesuai. Hal ini karena kernel menentukan bagaimana titik data diubah menjadi ruang berdimensi lebih tinggi, yang pada akhirnya memengaruhi kemampuan model untuk memisahkan data secara efektif. Setelah eksplorasi awal, kernel linear dipilih karena menunjukkan bahwa ulasan teks cenderung dapat dipisahkan secara linier dengan performa yang baik dibandingkan dengan kernel lainnya.

Pembagian data adalah langkah krusial dalam proses pembelajaran mesin untuk memastikan bahwa model yang dilatih dapat digeneralisasikan dengan baik ke data baru. Pembagian yang tidak tepat dapat menyebabkan *overfitting* atau *underfitting*, yang akan mempengaruhi kinerja model. Untuk mengevaluasi performa model, data dibagi menjadi empat rasio training dan testing yaitu 60:40 (60% *training*, 40% *testing*), 70:30 (70% *training*, 30% *testing*), 80:20 (80% *training*, 20% *testing*), dan 90:10 (90% *training*, 10% *testing*).

Model dilatih menggunakan data pelatihan dengan *hyperparameter* yang telah dioptimalkan. Proses pelatihan melibatkan penyelesaian masalah optimisasi untuk menemukan *hyperplane* yang memisahkan kelas-kelas data dengan margin terbesar. Setelah model dilatih, kinerjanya akan dilakukan evaluasi menggunakan data testing. Proses evaluasi ini menyimulasikan kemampuan model dalam menangani data yang tidak terlihat. Dengan menguji data yang belum pernah ditemukan sebelumnya, dimana untuk memastikan bahwa seberapa baik model dapat mengklasifikasikan ulasan baru.

Dengan menggunakan pendekatan ini, penelitian ini bertujuan untuk mengembangkan model klasifikasi yang robust dan akurat dalam mengidentifikasi sentimen dalam ulasan aplikasi Mamikos, yang pada akhirnya dapat digunakan untuk memberikan wawasan berharga bagi pengembangan dan peningkatan layanan aplikasi tersebut.

4.3.1 *Split Data*

Split data terdapat empat kali percobaan dengan nilai skenario rasio pembagian 60:40, 70:30, 80:20, dan 90:10.

4.3.1.1 Uji Coba 60:40

Pada pengujian pertama awal, data dibagi untuk training dan testing dengan perbandingan 60:40. Hasil pengujian yang mengkategorikan ulasan ke dalam kategori positif, negatif, dan netral disajikan pada tabel 4.8.

Tabel 4.8. Hasil Klasifikasi Rasio 60:40

Kategori	Precision	Recall	F1-Score
Positif	74,27%	76,97%	75,60%
Netral	59,82%	51,15%	55,14%
Negatif	75,18%	83,06%	78,93%

Pada skenario 60:40, hasil klasifikasi menunjukkan bahwa untuk kelas positif, nilai *precision* adalah 74,27%, *recall* 76,97%, dan *f1-score* 75,60%. Untuk kelas negatif, nilai *precision* adalah 75,18%, *recall* 83,06%, dan *f1-score* 78,93%. Sedangkan untuk kelas netral, nilai *precision* adalah 59,82%, *recall* 51,15%, dan *f1-score* 55,14%. Hasil prediksi sentimen pada data *testing* ditampilkan dalam tabel 4.9.

Tabel 4.9. Hasil Prediksi Sentimen Rasio 60:40

Ulasan	Sentimen	Prediksi
aplikasi langganan aku dari zaman kuliah	Positif	Netral
aplikasi bagus sangat bantu semoga berkah manfaat	Positif	Positif
dasar aplikasi lemot	Negatif	Negatif
aplikasi bagus lebih gampang cari kos	Netral	Positif
admin respond dengan cepat	Netral	Negatif

4.3.1.2 Uji Coba 70:30

Pengujian kedua dilakukan dengan menggunakan skenario pembagian data dengan rasio 70:30. Dari hasil pengujian yang telah dilakukan didapatkan hasil klasifikasi positif, negatif, dan netral yang ditunjukkan pada tabel 4.10.

Tabel 4.10. Hasil Klasifikasi Rasio 70:30

Kategori	Precision	Recall	F1-Score
Positif	78,23%	79,51%	78,86%
Netral	59,74%	49,46%	54,12%
Negatif	72,81%	83%	77,57%

Pada skenario 70:30, hasil klasifikasi menunjukkan bahwa untuk kelas positif, nilai *precision* adalah 78,23%, *recall* 79,51%, dan *f1-score* 78,86%. Untuk kelas negatif, nilai *precision* adalah 72,81%, *recall* 83%, dan *f1-score* 77,57%. Sedangkan untuk kelas netral, nilai *precision* adalah 59,74%, *recall* 49,46%, dan *f1-score* 54,12%. Hasil prediksi sentimen pada data testing ditampilkan dalam tabel 4.11

Tabel 4.11. Hasil Prediksi Sentimen Rasio 70:30

Ulasan	Sentimen	Prediksi
sangat suka dengan aplikasi ini	Netral	Positif
maaf aku turinin karena sekarang fitur pesan harus bayar	Negatif	Negatif
aplikasi keren banget bantu mahasiswa cari kos	Positif	Positif
aplikasi bagus cukup bantu	Positif	Positif
error gimana tidak lihat foto kamar	Netral	Negatif

4.3.1.3 Uji Coba 80:20

Pengujian ketiga dilakukan dengan menggunakan skenario pembagian data dengan rasio 80:20. Dari hasil pengujian yang telah dilakukan didapatkan hasil klasifikasi positif, negatif, dan netral yang ditunjukkan pada tabel 4.12.

Tabel 4.12. Hasil Klasifikasi Rasio 80:20

Kategori	Precision	Recall	F1-Score
Positif	85,54%	80,68%	83,04%
Netral	61,11%	55,93%	58,41%
Negatif	73,97%	85,71%	79,41%

Pada skenario 80:20, hasil klasifikasi menunjukkan bahwa untuk kelas positif, nilai *precision* adalah 85,54%, *recall* 80,68%, dan *f1-score* 83,04%. Untuk kelas negatif, nilai *precision* adalah 73,97%, *recall* 85,71%, dan *f1-score* 79,41%. Sedangkan untuk kelas netral, nilai *precision* adalah 61,11%, *recall* 55,93%, dan *f1-score* 58,41%. Hasil prediksi sentimen pada data testing disajikan pada tabel 4.13.

Tabel 4.13. Hasil Prediksi Sentimen Rasio 80:20

Ulasan	Sentimen	Prediksi
aplikasi mamikos bantu mahasiswa cari kos	Positif	Positif
tolong perbaiki sistem	Negatif	Netral
sudah lama aku menggunakan aplikasi ini memang bagus	Positif	Positif
sangat bantu kos aku full karena mamikos	Positif	Positif
bismillah semoga dapat kos dekat	Netral	Netral

4.3.1.4 Uji Coba 90:10

Pengujian terakhir dilakukan dengan menggunakan skenario pembagian data dengan rasio 90:10. Dari hasil pengujian yang telah dilakukan didapatkan hasil klasifikasi positif, negatif, dan netral yang ditunjukkan pada tabel 4.14.

Tabel 4.14. Hasil Klasifikasi Rasio 90:10

Kategori	Precision	Recall	F1-Score
Positif	87,50%	83,33%	85,37%
Netral	66,67%	53,33%	59,26%

Negatif	75,61%	93,34%	83,78%
---------	--------	--------	--------

Pada skenario 80:20, hasil klasifikasi menunjukkan bahwa untuk kelas positif, nilai *precision* adalah 87,50%, *recall* 83,33%, dan *f1-score* 85,37%. Untuk kelas negatif, nilai *precision* adalah 75,61%, *recall* 93,34%, dan *f1-score* 83,78%. Sedangkan untuk kelas netral, nilai *precision* adalah 66,67%, *recall* 53,33%, dan *f1-score* 59,26%. Hasil prediksi sentimen pada data *testing* disajikan pada tabel 4.15.

Tabel 4.15. Hasil Prediksi Sentimen Rasio 90:10

Ulasan	Sentimen	Prediksi
mamikos sangat bantu kuliah luar kota	Positif	Positif
sistem lebih tingkatkan lagi	Negatif	Netral
aplikasi kurang simpel	Negatif	Negatif
aplikasi sangat bagus	Positif	Positif
bayar saldo mami ads gagal	Netral	Positif

4.3.2 K-Fold Cross Validation

Untuk mengidentifikasi tingkat akurasi yang optimal, *k-fold cross validation* digunakan dengan empat nilai *k* yang berbeda: 5, 10, 15, dan 20. Dalam metode ini, data dibagi menjadi *k* lipatan, dan setiap lipatan digunakan untuk menguji akurasi model. Lipatan yang tersisa digunakan untuk *training*. Dengan membandingkan *accuracy* pada nilai *k* yang berbeda ini, dapat menentukan konfigurasi mana yang mencapai performa terbaik secara keseluruhan.

4.3.2.1 Uji Coba K-5

Pengujian dengan *k-5* akan melibatkan 5 iterasi dari keseluruhan data. Pada setiap iterasi, dataset akan dibagi rata menjadi lima lipatan. Pada *k-fold cross validation*, data dibagi menjadi beberapa lipatan (*k*). Selama setiap pengujian, satu

iterasi digunakan untuk menguji keakuratan model dalam mengklasifikasikan ulasan sebagai positif, negatif, atau netral. Iterasi yang tersisa digunakan untuk melatih model. Proses ini diulangi untuk semua lipatan. Hasil evaluasi ini disajikan pada Tabel 4.16.

Tabel 4.16. Hasil Klasifikasi K-5

Kategori	Precision	Recall	F1-Score
Positif	77,01%	80,69%	78,66%
Netral	61,94%	51,69%	56,20%
Negatif	76,82%	85,35%	80,81%

Pada skenario k-5, hasil klasifikasi menunjukkan bahwa untuk kelas positif, nilai *precision* adalah 77,01%, *recall* 80,69%, dan *f1-score* 78,66%. Untuk kelas negatif, nilai *precision* adalah 76,82%, *recall* 85,35%, dan *f1-score* 80,81%. Sedangkan untuk kelas netral, nilai *precision* adalah 61,94%, *recall* 51,69%, dan *f1-score* 56,20%. Hasil prediksi sentimen pada data *testing* ditampilkan dalam tabel 4.17.

Tabel 4.17. Hasil Prediksi Sentimen K-5

Ulasan	Sentimen	Prediksi
aplikasi sangat manfaat	Positif	Positif
mamikos selalu jadi andalan sejak zaman kuliah	Positif	Netral
kurang puas	Negatif	Negatif
kenapa balasan pesan sekarang batasi	Negatif	Negatif
menu tampilan mudah pahami	Netral	Netral

4.3.2.2 Uji Coba K-10

Pada pengujian dengan k-10, dilakukan 10 iterasi dari keseluruhan data. Pada setiap iterasi, dataset dibagi menjadi 10 bagian yang sama besar. Pada k-fold cross validation, data dibagi menjadi beberapa lipatan (k). Selama setiap pengujian, satu iterasi digunakan untuk menguji keakuratan model dalam mengklasifikasikan ulasan sebagai positif, negatif, atau netral. Iterasi yang tersisa

digunakan untuk melatih model. Proses ini diulangi untuk semua lipatan. Hasil evaluasi ini disajikan pada Tabel 4.18.

Tabel 4.18. Hasil Klasifikasi K-10

Kategori	Precision	Recall	F1-Score
Positif	78,17%	80,47%	79,14%
Netral	62,33%	54,28%	57,62%
Negatif	75,78%	83,26%	78,97%

Pada skenario k-10, hasil klasifikasi menunjukkan bahwa untuk kelas positif, nilai *precision* adalah 78,17%, *recall* 80,47%, dan *f1-score* 79,14%. Untuk kelas negatif, nilai *precision* adalah 75,78%, *recall* 83,26%, dan *f1-score* 78,97%. Sedangkan untuk kelas netral, nilai *precision* adalah 62,33%, *recall* 54,28%, dan *f1-score* 57,62%. Hasil prediksi sentimen pada data *testing* ditampilkan dalam tabel 4.19.

Tabel 4.19. Hasil Prediksi Sentimen K-10

Ulasan	Sentimen	Prediksi
mamikos terdepan untuk cari kos	Netral	Netral
sekarang merugikan pemilik kos kontrakan	Negatif	Netral
menyesal install aplikasi tidak bantu	Negatif	Negatif
aplikasi tidak bantu	Negatif	Positif
sangat bantu jadi lebih mudah	Positif	Positif

4.3.2.3 Uji Coba K-15

Pengujian dengan k-15 melibatkan 15 iterasi dari keseluruhan data. Pada setiap iterasi, dataset dibagi menjadi 15 bagian yang sama besar. Pada k-fold cross validation, data dibagi menjadi beberapa lipatan (k). Selama setiap pengujian, satu iterasi digunakan untuk menguji keakuratan model dalam mengklasifikasikan ulasan sebagai positif, negatif, atau netral. Iterasi yang tersisa digunakan untuk melatih model. Proses ini diulangi untuk semua lipatan. Hasil evaluasi ini disajikan pada Tabel 4.20

Tabel 4.20. Hasil Klasifikasi K-15

Kategori	Precision	Recall	F1-Score
Positif	77,29%	79,66%	78,24%
Netral	61,40%	53,32%	56,59%
Negatif	74,69%	83,54%	78,43%

Pada skenario k-15, hasil klasifikasi menunjukkan bahwa untuk kelas positif, nilai *precision* adalah 77,29%, *recall* 79,66%, dan *f1-score* 78,24%. Untuk kelas negatif, nilai *precision* adalah 74,69%, *recall* 83,54%, dan *f1-score* 78,43%. Sedangkan untuk kelas netral, nilai *precision* adalah 61,40%, *recall* 53,32%, dan *f1-score* 56,59%. Hasil prediksi sentimen pada data *testing* ditampilkan dalam tabel 4.21.

Tabel 4.21. Hasil Prediksi Sentimen K-15

Ulasan	Sentimen	Prediksi
lebih baik manual saja cari kos terlalu rumit	Negatif	Negatif
belum install aplikasi mamikos cepat install	Netral	Netral
aplikasi bantu untuk pelajar mahasiswa	Positif	Positif
cari kos dengan mudah tanpa rumit	Netral	Negatif
lebih terpercaya menggunakan aplikasi mamikos	Positif	Netral

4.3.2.4 Uji Coba K-20

Pengujian dengan k-20 melibatkan 20 iterasi dari keseluruhan data. Pada setiap iterasi, dataset dibagi menjadi 20 bagian yang sama besar. Pada *k-fold cross validation*, data dibagi menjadi beberapa lipatan (k). Selama setiap pengujian, satu iterasi digunakan untuk menguji keakuratan model dalam mengklasifikasikan ulasan sebagai positif, negatif, atau netral. Iterasi yang tersisa digunakan untuk melatih model. Proses ini diulangi untuk semua lipatan. Hasil evaluasi ini disajikan pada Tabel 4.22.

Tabel 4.22. Hasil Klasifikasi K-20

Kategori	Precision	Recall	F1-Score
Positif	77,09%	79,18%	77,84%
Netral	61,48%	52,58%	55,98%
Negatif	73,53%	82,45%	77,14%

Hasil klasifikasi yang didapatkan pada skenario k-20 pada kelas positif mendapatkan nilai *precision* 77,09%, *recall* 79,18%, dan *f1-score* 77,84%. Untuk kelas negatif mendapatkan nilai *precision* 73,53%, *recall* 82,45%, dan *f1-score* 77,14%. Sedangkan pada kelas netral mendapatkan nilai *precision* 61,48%, *recall* 52,58%, dan *f1-score* 55,98%. Dari hasil tersebut didapatkan hasil prediksi sentimen pada data *testing* yang ditunjukkan pada tabel 4.23.

Tabel 4.23. Hasil Prediksi Sentimen K-20

Ulasan	Sentimen	Prediksi
aplikasi keren bantu cari kos	Positif	Positif
bantu banget cari kos sangat bantu cari kos	Netral	Positif
sangat bantu cari kos	Positif	Positif
aplikasi mamikos semakin lama semakin lemot	Negatif	Negatif
mamikos sangat rekomendasi untuk mahasiswa	Netral	Netral

4.4 Hasil Evaluasi

Untuk mengevaluasi kinerja model, penulis menggunakan dua metode: *split* data dan *k-fold cross validation*. Dalam metode *split* data dengan berbagai rasio pembagian data *training* dan *testing*. yaitu 60:40, 70:30, 80:20, dan 90:10. *K-fold cross validation* menggunakan iterari nilai k yang berbeda, yaitu 5, 10, 15, dan 20. Kedua metode ini diterapkan untuk menemukan hasil terbaik. Setelah model diklasifikasikan, langkah selanjutnya adalah evaluasi performa untuk memperoleh nilai *accuracy*, *precision*, *recall*, dan *f1-score*.

4.4.1 Evaluasi *Split Data*

Sistem menerapkan algoritma *Support Vector Machine* dan mengevaluasi kinerjanya menggunakan *confusion matriks* untuk mendapatkan *accuracy*, *precision*, *recall*, dan *f1-score*. Evaluasi pada metode yang digunakan *split data* di mana data dibagi menjadi data *training* dan *testing* dan pengujian dengan rasio berbeda, yaitu 60:40, 70:30, 80:20, dan 90:10. Hasil kinerja rinci untuk rasio pembagian yang berbeda disajikan pada Tabel 4.24.

Tabel 4.24. Perbandingan *Confusion Matrix Split Data*

Skenario	Accuracy	Precision	Recall	F1-Score
60:40	70,71%	69,76%	70,39%	69,89%
70:30	71,75%	70,26%	70,66%	70,18%
80:20	75,24%	73,54%	74,11%	73,62%
90:10	78,10%	76,59%	76,87%	76,14

Berdasarkan hasil perbandingan pada tabel 4.24 menunjukkan bahwa *accuracy* tertinggi didapatkan pada skenario 90:10 dengan nilai sebesar 78,10%. Sedangkan *accuracy* terendah didapatkan pada skenario 60:40 dengan nilai sebesar 70,71%. Dari hasil perbandingan keempat model tersebut menunjukkan bahwa semakin kecil rasio data *testing* yang digunakan maka hasil *accuracy* yang didapatkan semakin tinggi. Terbukti bahwa pengujian yang dilakukan dengan skenario 60:40, 70:30, 80:20, dan 90:10 nilai tertinggi *accuracy* didapatkan oleh skenario 90:10 dengan 90% data *training* dan 90% data *testing*. Nilai performa evaluasi *confusion matrix* mengacu pada nilai *True Positive (TP)*, *False Positive (FP)*, *False Positive (FP)* dan *True Negative (TN)*. Hasil dari nilai-nilai tersebut ditunjukkan pada tabel 4.25.

Tabel 4.25. Nilai *Confusion Matrix* 90:10

Aktual	Prediksi		
	Positif	Negatif	Netral
Positif	31	2	0
Negatif	9	16	5
Netral	1	6	35

Berdasarkan hasil dari nilai performa evaluasi, didapatkan nilai *True Positive* sebanyak 31, *False Positive* sebanyak 2, *False Negative* sebanyak 10, dan *True Negative* sebanyak 62.

4.4.2 Evaluasi *K-Fold Cross Validation*

Sistem menerapkan algoritma *Support Vector Machine* dan mengevaluasi kinerjanya menggunakan *confusion matrix* untuk mendapatkan *accuracy*, *precision*, *recall*, dan *f1-score*. Evaluasi ini sangat penting untuk metode *k-fold cross validation*, di mana data dibagi menjadi beberapa bagian nilai k seperti 5, 10, 15, dan 20 untuk *training* dan *testing*. Hasil kinerja untuk nilai k yang berbeda disajikan pada Tabel 4.26.

Tabel 4.26. Perbandingan *Confusion Matrics K-fold Cross Validation*

Nilai K	Accuracy	Precision	Recall	F1-Score
5	73,02%	72,62%	73,02%	72,47%
10	73,21%	73,16%	73,21%	72,76%
15	72,83%	72,95%	72,83%	72,40%
20	73,64%	72,94%	72,64%	72,18%

Berdasarkan hasil perbandingan pada tabel 4.26 menunjukkan bahwa *accuracy* tertinggi didapatkan pada nilai k-10 dengan nilai sebesar 73,21%. Sedangkan *accuracy* terendah didapatkan pada nilai k-15 dengan nilai sebesar 72,83%. Dari hasil perbandingan keempat nilai tersebut menunjukkan bahwa semakin tinggi nilai k tidak memberikan pengaruh dalam meningkatkan nilai *accuracy*. Terbukti dengan pengujian yang dilakukan dengan nilai k-5, k-10, k-15,

dan k-20 nilai tertinggi accuracy didapatkan oleh nilai k-10 dan nilai terendah didapatkan oleh nilai k-15. Nilai performa evaluasi *confusion matrix* mengacu pada nilai *True Positive (TP)*, *False Positive (FP)*, *False Positive (FP)* dan *True Negative (TN)*. Hasil dari nilai-nilai tersebut ditunjukkan pada tabel 4.27.

Tabel 4.27. Nilai *Confusion Matrix* K-10

Aktual	Prediksi		
	Positif	Negatif	Netral
Positif	24	4	1
Negatif	6	17	9
Netral	2	7	35

Berdasarkan hasil dari nilai performa evaluasi, didapatkan nilai *True Positive* sebanyak 24, *False Positive* sebanyak 5, *False Negative* sebanyak 8, dan *True Negative* sebanyak 68.

4.5 Pembahasan

Dari hasil uji coba yang dilakukan, 1049 ulasan yang telah dilabeli secara manual dan divalidasi oleh dosen Bahasa Indonesia dibagi menjadi tiga kategori: positif, negatif, dan netral. Setelah pelabelan, terdapat 440 ulasan positif, 287 ulasan negatif, dan 322 ulasan netral. Setelah ulasan dikategorikan positif, negatif, netral, penulis melakukan *preprocessing text*. Ini melibatkan pembersihan data teks dan memberikan bobot pada kata-kata berdasarkan kepentingannya. Kemudian, membuat dua model berbeda menggunakan algoritma Support Vector Machine (SVM) untuk menganalisis sentimen, yaitu dengan *split* data dan *k-fold cross validation*. Terakhir, memeriksa seberapa baik kinerja model dengan menggunakan *confusion matriks* dan menghitung matriks seperti *accuracy*, *precision*, *recall*, dan *f1-score*.

Pada metode *split* data menggunakan berbagai rasio data, seperti 90% data *training* dan 10% data *testing*, 80:20, 70:30, dan 60:40. Model ini memiliki performa terbaik dengan pembagian 90:10, mencapai *accuracy* 78,10%. Artinya model tersebut melakukan *training* dari 90% ulasan dan kemudian melakukan *testing* pada 10% sisanya yang belum pernah dilihat sebelumnya. Pembagian 60:40 menghasilkan akurasi terendah yaitu 70,71%. Data dipilih secara acak dengan *random state* 42 untuk memastikan model tidak bias terhadap tinjauan tertentu.

Selanjutnya menguji *accuracy* model menggunakan metode yang disebut *k-fold cross validation*. Di sini, data dibagi menjadi beberapa bagian, yaitu k-5, k-10, k-15, dan k-20. Dalam setiap iterasi, satu lipatan digunakan untuk *testing* sedangkan lipatan lainnya digunakan untuk *training*. Proses ini diulangi untuk semua lipatan. Keuntungannya adalah semua data digunakan untuk *training* dan *testing*, namun keakuratannya dapat sedikit berbeda antar putaran karena model dilatih pada subset data yang berbeda. Dengan *k-fold cross validation*, *accuracy* tertinggi yang dicapai adalah 73,21% dengan nilai k=10.

Pengujian menunjukkan bahwa metode *split* data dengan rasio *training* dan *testing* 90:10 mencapai performa terbaik secara keseluruhan, mencapai *accuracy* 78,10% dalam mengklasifikasikan sentimen ulasan aplikasi Mamikos menggunakan Support Vector Machine (SVM) model. *K-fold cross validation* tidak meningkatkan *accuracy* secara signifikan. Hal ini menunjukkan bahwa metode *split* data sangat cocok untuk melakukan analisis sentimen.

Penelitian ini memberikan informasi bagaimana algoritma Support Vector Machine dapat digunakan untuk mengklasifikasikan sentimen dalam ulasan aplikasi Mamikos. Keakuratan model dipengaruhi oleh beberapa faktor. Memilih fitur informatif dari ulasan dan pembagian data yang tepat antara data *training* dan *testing* adalah hal yang penting. Selain itu, *preprocessing text* ulasan sebelumnya secara signifikan meningkatkan kualitas data model. Algoritma tersebut mampu mengidentifikasi pola dan fitur yang terkait dengan sentimen positif, negatif, dan netral, sehingga memungkinkan model tersebut mengklasifikasikan ulasan baru secara akurat.

Sistem analisis sentimen ini bertujuan untuk mendapatkan *accuracy* tinggi dalam mengklasifikasikan ulasan aplikasi Mamikos sebagai positif, negatif, atau netral. Untuk mencapai hal ini, sistem memanfaatkan model Support Vector Machine dan mengevaluasi kinerjanya menggunakan confusion matriks untuk mendapatkan *accuracy*, *precision*, *recall*, dan *f1-score*. Evaluasi komprehensif ini membantu memastikan efektivitas sistem dan memvalidasi keakuratan analisis sentimen.

Dengan menggunakan pendekatan ini, penelitian ini bertujuan untuk mengembangkan model klasifikasi yang robust dan akurat dalam mengidentifikasi sentimen dalam ulasan aplikasi Mamikos, yang pada akhirnya dapat digunakan untuk memberikan wawasan berharga bagi pengembangan dan peningkatan layanan aplikasi tersebut.

4.6 Integrasi Islam

Penelitian ini bertujuan pada analisis sentimen yang diungkapkan dalam ulasan untuk mengevaluasi keandalan opini. Namun hal ini lebih dari sekedar sentimen, dengan juga mempertimbangkan konsep Islam tentang muamalah, yang mencakup interaksi kita dengan Tuhan (Allah Subhanahu Wa Ta'ala) dan orang lain. Bagian berikut akan merinci hasil pengintegrasian analisis sentimen.

4.6.1 Hubungan dengan Allah

Dalam melakukan sewa menyewa harus memberikan manfaat dan tidak merugikan antara kedua belah pihak. Islam menekankan perjanjian sewa yang adil dan menyenangkan antara penyewa dan yang memberikan sewa. Keputusan sepihak, dimana hanya satu pihak yang mempunyai suara, dianggap tidak sah. Hal ini melindungi kedua belah pihak dari potensi kerugian dan mengedepankan prinsip Islam yang saling menguntungkan di antara orang-orang beriman. Al-Qur'an surat Al-Maidah ayat 1 mengajarkan kita tentang pentingnya bekerjasama dalam amal shaleh dan menghindari perbuatan yang dapat merugikan orang lain:

يَا أَيُّهَا الَّذِينَ ءَامَنُوا أَوْفُوا بِالْعُقُودِ أُحِلَّتْ لَكُمْ بَهِيمَةُ ٱلْأَنْعَامِ ٱلْأَمَّا مَا يُنْتَلَىٰ عَلَيْكُمْ غَيْرَ مُجْلَىٰ ٱلصَّيْدِ وَأَنْتُمْ حُرْمٌ إِنَّ ٱللَّهَ يَحْكُمُ مَا يُرِيدُ

“Hai orang-orang yang beriman! Penuhilah janji-janji, janji disini adalah janji setiap hamba kepada Allah dan perjanjian yang dibuat oleh manusia dalam pergaulan sesamanya.”(Q.S Al-Maidah’5 : 1)

Al-Azhar berkata keimanan kepada Allah (Subhanahu Wa Ta'ala) yang tercermin dalam hidup sesuai syariat Islam (Syariah) dan menjauhi perilaku yang dilarang. Salah satu aspek penting adalah menyempurnakan akad atau perjanjian. Jika seseorang telah membuat perjanjian, ia harus memenuhi dan mematuhi

sepenuhnya. (Lesmono & Sudiarti, 2022). prinsip Islam berlaku dalam berbagai aspek kehidupan sehari-hari, termasuk transaksi keuangan seperti perjanjian, jual beli, sewa, dan kontrak. Prinsip-prinsip ini memastikan kegiatan ini sejalan dengan ajaran Allah (Subhanahu Wa Ta'ala). Islam menekankan pada menepati janji, baik kepada Tuhan maupun sesama manusia, sebagaimana disabdakan oleh Rasulullah Shallallaahu Alaihi Wasalam.

وَأَنْفُسِهِمْ أَمْوَالِهِمْ عَلَى النَّاسِ أَمْنَهُ مَنْ وَالْمُؤْمِنُ، وَيَدِهِ لِسَانِهِ مِنَ النَّاسِ سَلِمَ مَنْ الْمُسْلِمِ
عَنْهُ اللَّهُ نَهَى مَا هَجَرَ مَنْ وَالْمُهَاجِرُ

“Barang siapa yang memutuskan janji, maka bukanlah di antara kami.”
(HR.Bukhari)

4.6.2 Hubungan dengan Manusia

Setiap manusia diharapkan untuk mendapatkan rezeki dengan cara sesuai dengan syariat islam. Salah satunya dengan cara melakukan perdagangan atau penyewaan seperti yang ada dalam penjelasan Qur'an Surah An-Nisa ayat 29 :

يَا أَيُّهَا الَّذِينَ ءَامَنُوا لَا تَأْكُلُوا أَمْوَالَكُمْ بَيْنَكُمْ بِالْبَاطِلِ إِلَّا أَنْ تَكُونَ تِجَارَةً عَنْ تَرَاضٍ مِّنْكُمْ وَلَا تَقْتُلُوا أَنْفُسَكُمْ إِنَّ اللَّهَ كَانَ بِكُمْ رَحِيمًا

“Wahai orang-orang yang beriman! janganlah kamu saling memakan harta sesamamu dengan jalan yang batil (tidak benar), kecuali dalam perdagangan yang berlaku atas dasar suka sama suka diantara kamu. Dan janganlah kamu membunuh dirimu. Sungguh, Allah Maha Penyayang kepadamu.” (Q.S An-Nisa' 4 : 29)

Al-Wajiz berpendapat bahwa memperoleh harga melalui cara-cara yang dilarang oleh hukum Islam, seperti riba, perjudian, pemerasan, atau penipuan, adalah melanggar hukum. Sebaliknya, diperbolehkan mengambil harta orang lain melalui transaksi perdagangan yang didasarkan pada kerelaan kedua belah pihak

dan sesuai dengan syariat. Al-Ijarah adalah konsep sewa-menyewa dalam Islam, yang merupakan salah satu bentuk pembiayaan untuk mendapatkan aset atau jasa dengan membayar sejumlah uang sewa untuk penggunaan atau manfaat dari aset atau jasa tersebut (Munandar & Hasan Ridwan, 2023). pembunuhan dilarang kecuali dalam keadaan tertentu yang diperbolehkan oleh hukum Islam (syariah). Hal ini mencerminkan besarnya rahmat Allah terhadap umat manusia. Islam mengajarkan untuk melakukan penyewaan dengan kondisi yang baik dan sesuai syariat, sebagaimana disabdakan oleh Rasulullah Shallallaahu Alaihi Wasalam.

اسْتَأْجِرْتُمْ خَيْرًا وَإِذَا فِيهِ مَا فَاشْتَرُوا تُمْ اشْتَرِي إِذَا " : وَسَلَّم عَلَيْهِ اللَّهُ صَلَّى اللَّهُ رَسُولُ قَالَ
خَيْرٌ فِيهِ مَا فَأَجِرُوا

“Apabila kamu berbelanja (situ), belilah yang tidak buruk dan apabila kamu menyewakan (sesuatu), sewakanlah yang tidak buruk pula.” (HR. Ibnu Majah)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini mengumpulkan data dari *Google Play Store* melalui teknik *crawling* dan menerapkan algoritma *Support Vector Machine* pada analisis sentimen ulasan aplikasi Mamikos. Sebanyak 1049 ulasan dikumpulkan dan diproses melalui langkah *preprocessing text*. Setelah dilabeli secara pribadi dan diverifikasi oleh dosen bahasa Indonesia, ulasan ini menghasilkan 322 sentimen netral, 287 sentimen negatif, dan 440 sentimen positif. Dalam analisis sentimen, data berlabel ini digunakan sebagai data *training* dan *testing*.

Evaluasi performa sistem dengan metode *split data* menunjukkan bahwa nilai *accuracy* tertinggi sebesar 78,10% diperoleh pada rasio pembagian data *training* dan data *testing* 90:10. Sedangkan pada pengujian *k-fold cross validation*, nilai *accuracy* tertinggi sebesar 73,21% diperoleh pada nilai *k*-10. Faktor-faktor yang mempengaruhi nilai *accuracy* meliputi penggunaan fitur pada dataset, rasio pembagian data pada klasifikasi, dan *preprocessing* yang tepat. Penggunaan *k-fold cross validation* tidak memberikan peningkatan yang signifikan pada nilai *accuracy*. Penelitian ini menunjukkan bahwa nilai *accuracy* yang relatif baik dihasilkan oleh analisis sentimen yang memanfaatkan algoritma *Support Vector Machine* bersama dengan metode *split data* dan *k-fold cross validation*.

5.2 Saran

Penulis menyadari bahwa penelitian ini memiliki keterbatasan. Oleh karena itu, penulis memberikan saran untuk peneliti selanjutnya agar menghasilkan hasil performa yang lebih baik berdasarkan hasil yang sudah didapatkan seperti sebagai berikut.

1. Memperbanyak jumlah data ulasan yang digunakan.
2. Menambahkan jumlah validator data dalam proses pelabelan data.
3. Mengimplementasikan feature selection, seperti *chi square* untuk membantu proses klasifikasi pada sistem dan menghasilkan hasil yang lebih akurat.

DAFTAR PUSTAKA

- Adhi Putra, A. D. (2021). Sentiment Analysis on User Reviews of the Bibit and Bareksa Application with the KNN Algorithm. *Jurnal Teknik Informatika Dan Sistem Informasi*, 8(2), 636–646. <https://doi.org/10.35957/jatisi.v8i2.962>
- Ananda, F. D., & Pristyanto, Y. (2021). Sentiment Analysis of Twitter Users on Internet Service Providers Using Support Vector Machine Algorithm. *Jurnal Manajemen, Teknik Informatika, Dan Rekayasa Komputer*, 20(2), 407–416.
- Arsi, P., & Waluyo, R. (2021). Sentiment Analysis of Discourse on Moving the Indonesian Capital City Using the Support Vector Machine (SVM) Algorithm. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 8(1), 147. <https://doi.org/10.25126/jtiik.0813944>
- Astari, N. M. A. J., Dewa Gede Hendra Divayana, & Gede Indrawan. (2020). Sentiment Analysis of Twitter Documents Regarding the Impact of the Corona Virus Using the Naive Bayes Classifier Method. *Jurnal Sistem Dan Informatika*, 15(1), 27–29. <https://doi.org/10.30864/jsi.v15i1.332>
- Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (2019). *Socio-Affective Computing 5 A Practical Guide to Sentiment Analysis*. <https://doi.org/10.1007/978-3-319-55394-8>
- Fara Sofia Ashari, Aisyah Amini, Adlin Nahar Lubis, P. R. S. (2023). Effectiveness of the Mamikos Application in Increasing Visitors and Marketing for Boarding Houses in Medan City. *JURNAL MANAJEMEN AKUNTANSI*, 3(1,2), 72–81. <https://doi.org/doi.org/10.36987/jumsi.v3i1.3716>
- Fazar Tri Hidayat, T., Ridha, A. A., & Garno. (2021). Sentiment Analysis of Capital City Relocation on Twitter Using the Support Vector Machine Method. *Jurnal Ilmu Komputer*, 14(2), 49–59.
- Fikri, M. I., Sabrila, T. S., & Azhar, Y. (2020). Comparison of Naïve Bayes and Support Vector Machine Methods in Twitter Sentiment Analysis. *Smatika Jurnal*, 10(02), 71–76. <https://doi.org/10.32664/smatika.v10i02.455>
- Gifari, O. I., Adha, M., Freddy, F., & Durrand, F. F. S. (2022). Film Review Sentiment Analysis Using TF-IDF and Support Vector Machine. *Journal of Information Technology*, 2(1), 36–40. <https://doi.org/10.46229/jifotech.v2i1.330>
- Husada, H. C., & Paramita, A. S. (2021). Sentiment Analysis for Airlines on the Twitter Platform Using the Support Vector Machine (SVM) Algorithm. *Teknika*, 10(1), 18–26. <https://doi.org/10.34148/teknika.v10i1.311>
- Kulsum Umami, Jajuli Mohamad, S. N. (2022). Sentiment Analysis of Mola

- Application Reviews on the Google Play Store Using the Support Vector Machine Algorithm. *INTECOMS: Journal of Information Technology and Computer Science*, 5(1), 1–7. <https://doi.org/10.31539/intecom.v5i1.3708>
- Larasati, F. A., Ratnawati, D. E., & Hanggara, B. T. (2022). Sentiment Analysis of Dana Application Reviews Using the Random Forest Method. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(9), 4305–4313. <http://j-ptiik.ub.ac.id>
- Lesmono, B., & Sudiarti, S. (2022). Interpretation of the First Verse in Surah Al-Maidah. *Mubeza*, 11(1), 30–34. <https://doi.org/10.54604/mbz.v11i1.53>
- Mamikos. (2023). <https://mamikos.com/>
- Muktafin, E. H., Kusrini, K., & Luthfi, E. T. (2020). Sentiment Analysis in Product Purchase Reviews on the Shopee Marketplace Using a Natural Language Processing Approach. *Jurnal Eksplora Informatika*, 10(1), 32–42. <https://doi.org/10.30864/eksplora.v10i1.390>
- Munandar, A., & Hasan Ridwan, A. (2023). Interpretation of Surah An-Nisa Verse 29 as the Legal Basis for the Ba'i Assalam Contract in the Practice of Online Buying and Selling. *Rayah Al-Islam*, 7(1), 271–287. <https://doi.org/10.37274/rais.v7i1.659>
- Nurmalasari, D., Hermanto, T. I., & Ma, I. (2023). Comparison of SVM, KNN and NBC Algorithms for Loan Service Application Sentiment Analysis. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 7, 1521–1530. <https://doi.org/10.30865/mib.v7i3.6427>
- Pramesti, L. A., & Pratiwi, N. (2023). Twitter Sentiment Analysis of the MBKM Program Using Decision Tree and Support Vector Machine. *Journal of Information System Research*, 4(4), 1145–1154. <https://doi.org/10.47065/josh.v4i4.3807>
- Pratiwi, R. W., H, S. F., Dairoh, D., Af'idah, D. I., A, Q. R., & F, A. G. (2021). Sentiment Analysis in the Female Daily Skincare Review Using the Support Vector Machine (SVM) Method. *Journal of Informatics, Information System, Software Engineering and Applications*, 4(1), 40–46. <https://doi.org/10.20895/inista.v4i1.387>
- Rokhman, K. A., Berlilana, B., & Arsi, P. (2021). Comparison of Support Vector Machine and Decision Tree Methods for Sentiment Analysis Review Comments on Online Transportation Applications. *Journal of Information System Management*, 3(1), 1–7. <https://doi.org/10.24076/joism.2021v3i1.341>
- Septian, J. A., Fachrudin, T. M., & Nugroho, A. (2019). Analysis of Twitter User Sentiment towards the Indonesian Football Polemic Using TF-IDF and K-Nearest Neighbor Weighting. *Journal of Intelligent System and Computation*, 1(1), 43–49. <https://doi.org/10.52985/insyst.v1i1.36>
- Susianti, I., Ningsih, S. S., Haris, M. Al, & Utami, T. W. (2020). Sentiment

Analysis on Twitter Regarding the New Normal Using the Naïve Bayes Classifier Method. *Prosiding Seminar Edusainstech FMIPA UNIMUS*, 354–363.

<https://prosiding.unimus.ac.id/index.php/edusaintek/article/view/576/578>

Syamsuddin, M. (2018). Overseas Madurese in the Special Region of Yogyakarta. *Aplikasia: Jurnal Aplikasi Ilmu-Ilmu Agama*, 18(1), 1. <https://doi.org/10.14421/aplikasia.v18i1.1378>

Tuntun, R., Kusrini, K., & Kusnawi, K. (2022). Comparative Analysis of Classification Algorithm Performance Using the K-Fold Cross Validation Method. *Jurnal Media Informatika Budidarma*, 6(4), 2111. <https://doi.org/10.30865/mib.v6i4.4681>

Wahyudi, R., & Kusumawardana, G. (2021). Sentiment Analysis on the Grab Application on Google Play Store Using Support Vector Machine. *Jurnal Informatika*, 8(2), 200–207. <https://doi.org/10.31294/ji.v8i2.9681>

Wibowo, A. (2020). Android Based Boarding House Mobile Information System. *Jurnal Ilmu Teknik Dan Komputer*, 4(1), 17. <https://doi.org/10.22441/jitkom.2020.v4.i1.002>

Wijaya, Y. V., Erfina, A., & Warman, C. (2021). Sentiment Analysis Regarding the ITE Law Using the Support Vector Machine Algorithm. *Progresif: Jurnal Ilmiah Komputer*, 17(2), 1. <https://doi.org/10.35889/progresif.v17i2.644>