

BAB 3 METODOLOGI PENELITIAN

3.1 Studi Literatur

Penelitian sebelumnya dilakukan untuk mendapatkan perbandingan dan referensi yang relevan. Selain itu, tujuan lain menggunakan penelitian sebelumnya adalah untuk menghindari kesalahan asumsi dan memastikan obyektivitas tetap terjaga. Oleh karena itu, berbagai topik terkait analisis sentimen dan metode yang digunakan dalam penelitian sebelumnya telah dipelajari untuk memahami pendekatan yang paling efektif dan memperkaya kerangka teoretis penelitian ini. Beberapa jurnal dan penelitian yang relevan telah ditemukan dan dijadikan acuan, antara lain:

Tabel 3.1. Daftar penelitian terkait

No	Penulis	Judul	Hasil
1	J. L. Rizky, W. Gata (2024)	Analisis Sentimen Media Sosial Youtube Kereta Cepat (Whoosh) Menggunakan Algoritma Bidirectional-LSTM	Bidirectional-LSTM mencapai akurasi tertinggi sebesar 0,86, mengungguli Multinomial Naïve Bayes, USE-Transfer Learning, dan Text Vectorization and Embedding (TensorFlow), yang masing-masing mencapai akurasi 0,80.
2	J. Sanjaya, Tukino, B. Priyanta, S. S. Hilabi (2024)	Analisis Sentimen Terhadap Opini Proyek Kereta Cepat Menggunakan Metode Naïve Bayes Classifier	Naïve bayes: akurasi sebesar 81%, presisi 81%, recall 81%, dan f1-score 81%. %.

Lanjut pada halaman berikutnya

Tabel 3.1 Daftar penelitian terkait (lanjutan)

No	Penulis	Judul	Hasil
3	S. Wahyu (2023)	Perbandingan Model Algoritma Klasifikasi Pada Analisis Sentimen Opini Masyarakat Terhadap Layanan Kereta Cepat Jakarta Bandung (The Whoosh)	SVM memiliki akurasi tertinggi sebesar 86%, diikuti oleh KNN dengan 85%, dan Naïve Bayes dengan 84%.
4	M. D. Islamanda, T. Sibaroni (2024)	Whoosh User Sentiment Analysis on Social Media Using Word2Vec and the Best Naïve Bayes Probability Model	Kombinasi Gaussian Naïve Bayes dan Word2Vec Skip-Gram mencapai performa terbaik, dengan skor akurasi sebesar 77,18%, presisi 70,35%, recall 76,09%, dan f1-score 73,10%, dibandingkan dengan model Word2Vec CBOW.
5	T. Agustiranti, A. K. I. Kurdiana, B. A. Ghiffari, E. D. Juniar, D. G. Purnama (2024)	Penerapan Naive Bayes Terhadap Sentimen Analisis Media Sosial Twitter Pengguna Kereta Cepat Jakarta-Bandung (Whoosh)	Naive Bayes memperoleh akurasi 88%.
6	T. Hartati, R. T. Sohadi, E. Tohidi, E. Wahyudin (2024)	Penerapan Algoritma Naive Bayes pada Analisis Sentimen Ulasan Aplikasi Whoosh – Kereta Cepat Di Google Play Store	Model <i>Naive Bayes</i> memperoleh akurasi 81.25%, presisi untuk kelas positif 90%, recall positif 94%, dan recall negatif 64%.

Dari tinjauan literatur ini, ditemukan bahwa berbagai algoritma klasifikasi telah diterapkan untuk analisis sentimen terkait layanan kereta cepat Whoosh. Model Bidirectional-LSTM menunjukkan performa tertinggi dalam penelitian oleh Rizky dan Gata [40] dengan akurasi 86%, mengungguli algoritma lain

seperti Multinomial Naïve Bayes, USE-Transfer Learning, dan Text Vectorization and Embedding (TensorFlow) yang masing-masing mencapai akurasi 80%. Selanjutnya, studi oleh Sanjaya et al. [41] yang menggunakan Naïve Bayes Classifier untuk menganalisis opini publik terkait proyek kereta cepat menunjukkan akurasi sebesar 81% dengan metrik evaluasi yang konsisten pada presisi, recall, dan f1-score sebesar 81%.

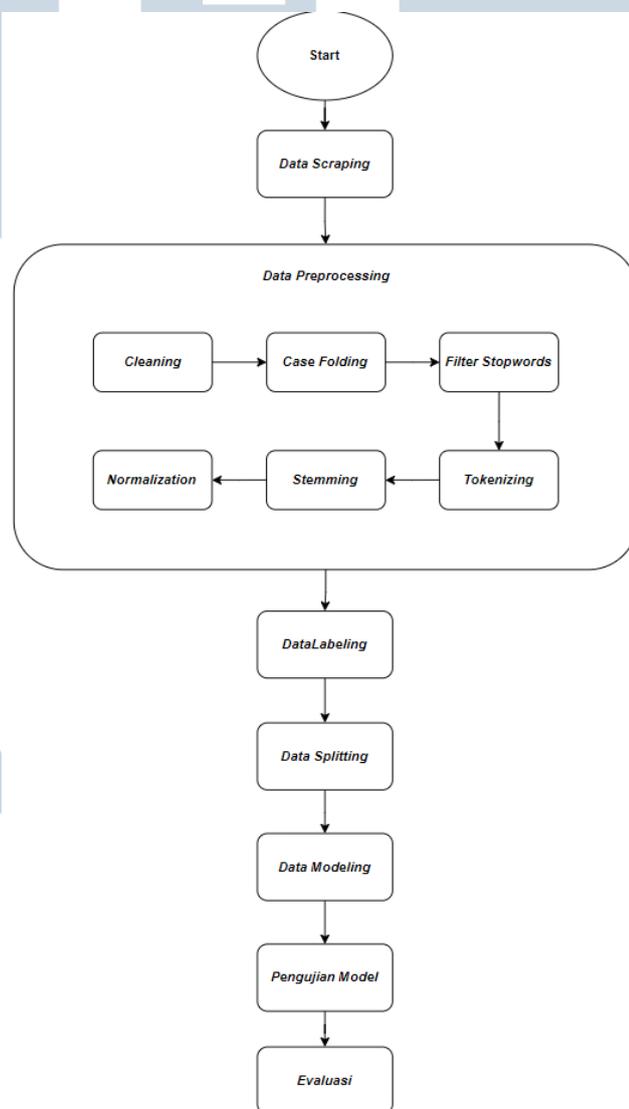
Sementara itu, Wahyu [42] membandingkan tiga model algoritma klasifikasi, menemukan bahwa Support Vector Machine (SVM) memiliki akurasi tertinggi sebesar 86%, diikuti oleh KNN dengan 85%, dan Naïve Bayes dengan 84%. Islamanda dan Sibaroni [43] menunjukkan bahwa kombinasi Gaussian Naïve Bayes dengan Word2Vec Skip-Gram mencapai performa optimal, menghasilkan akurasi 77,18% yang lebih baik daripada model Word2Vec CBOW. Di sisi lain, penelitian Agustiranti et al. [44] memperoleh akurasi tertinggi untuk analisis sentimen di Twitter menggunakan Naïve Bayes dengan akurasi 88%. Terakhir, studi oleh Hartati et al. [45] yang menganalisis ulasan aplikasi Whoosh di Google Play Store menunjukkan bahwa Naïve Bayes mencapai akurasi 81,25%, dengan presisi 90% dan recall positif 94%.

Selain itu, gap penelitian yang dapat diidentifikasi dari literatur tersebut adalah proses pelabelan dataset yang biasanya dilakukan secara manual. Metode manual ini sering kali memakan waktu dan rentan terhadap kesalahan. Sebagai pembeda, penelitian ini menggunakan *pretrained* model BERT khusus untuk membantu pelabelan ulasan secara otomatis.

Berdasarkan tinjauan literatur sebelumnya, penggunaan SVM dalam penelitian terkait aplikasi Whoosh masih terbatas. Hal ini menunjukkan adanya gap dalam eksplorasi algoritma ini dalam konteks analisis sentimen untuk aplikasi tersebut. SVM dikenal memiliki performa yang baik, terutama pada dataset kecil, sehingga menjadikannya pilihan yang relevan untuk menganalisis ulasan pengguna yang tidak selalu berjumlah besar. Selain itu, penelitian oleh S. Wahyu (2023) menunjukkan bahwa SVM memperoleh akurasi tertinggi dibandingkan dengan dua model algoritma klasifikasi lainnya dalam analisis sentimen. Oleh karena itu, penelitian ini berupaya untuk menggali lebih dalam efektivitas SVM dalam analisis sentimen aplikasi Whoosh.

3.2 Rancangan Sistem

Perancangan sistem pada penelitian ini mencakup langkah-langkah yang didasarkan pada studi literatur sebelumnya yang digunakan sebagai kerangka kerja. Sistem dirancang setelah data yang dibutuhkan telah terkumpul. Data yang telah dimining dan diolah kemudian melewati tahap *preprocessing* untuk dibersihkan dan dikonsolidasikan ke dalam bentuk yang sesuai. Penelitian analisis sentimen pada aplikasi Whoosh digambarkan dalam diagram tahapan pada Gambar 3.1, yang mencakup berbagai tahapan proses sebagai berikut:



Gambar 3.1. Alur penelitian

[21]

Diagram penelitian ini memberikan gambaran jelas mengenai tahapan-tahapan yang dilakukan dalam penelitian analisis sentimen aplikasi Whoosh.

3.2.1 Data Scrapping

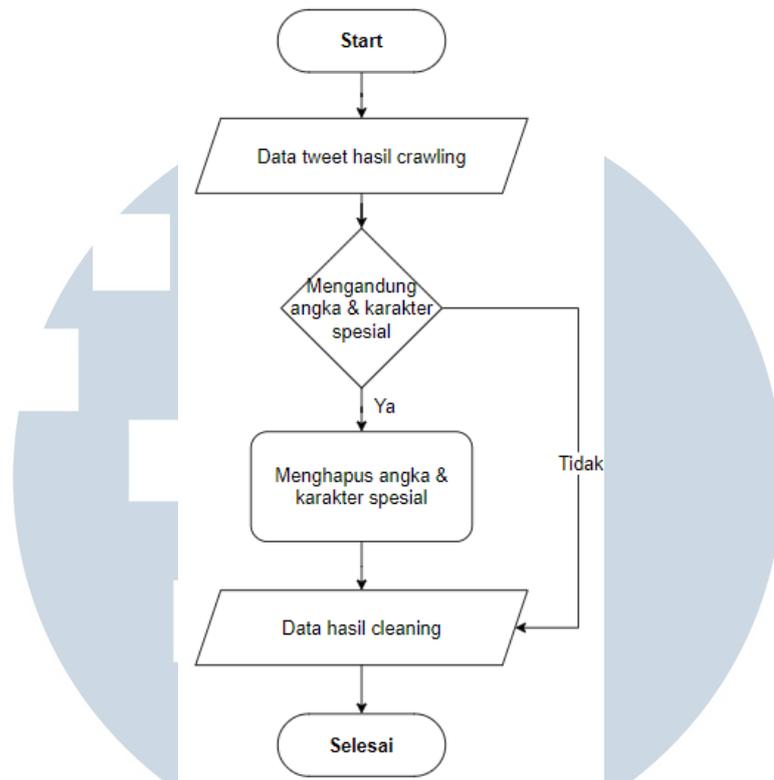
Data merupakan bagian yang besar dalam analisis sentimen, pada penelitian ini pengambilan data dilakukan menggunakan beberapa *library* untuk mengumpulkan ulasan dari aplikasi Goole Play Store dan App Store. *Google Play Scrapper* adalah *library* yang digunakan untuk mengumpulkan ulasan aplikasi dari Google Play Store dan *App Store Scrapper* merupakan *library* yang digunakan untuk mengumpulkan data ulasan dari App Store. Data yang di ekstrak akan disimpan dalam bentuk file *Comma Separated Values (CSV)* yang kemudian akan diproses lebih lanjut.

3.2.2 Text-Preprocessing

Preprocessing merupakan Langkah awal dalam melakukan text mining untuk mengubah data yang diperoleh menjadi data yang bersih dan berkualitas untuk dianalisis. Berikut Langkah yang perlu dilakukan dalam proses preprocessing:

- a. *Cleaning* Pembersihan data merupakan proses menghilangkan elemen-elemen yang tidak relevan atau mengganggu, sehingga data menjadi lebih bersih dan terstruktur. Contohnya termasuk menghapus konten ulasan yang bukan alfabet, seperti emoji, karakter alfanumerik, dan simbol lainnya.

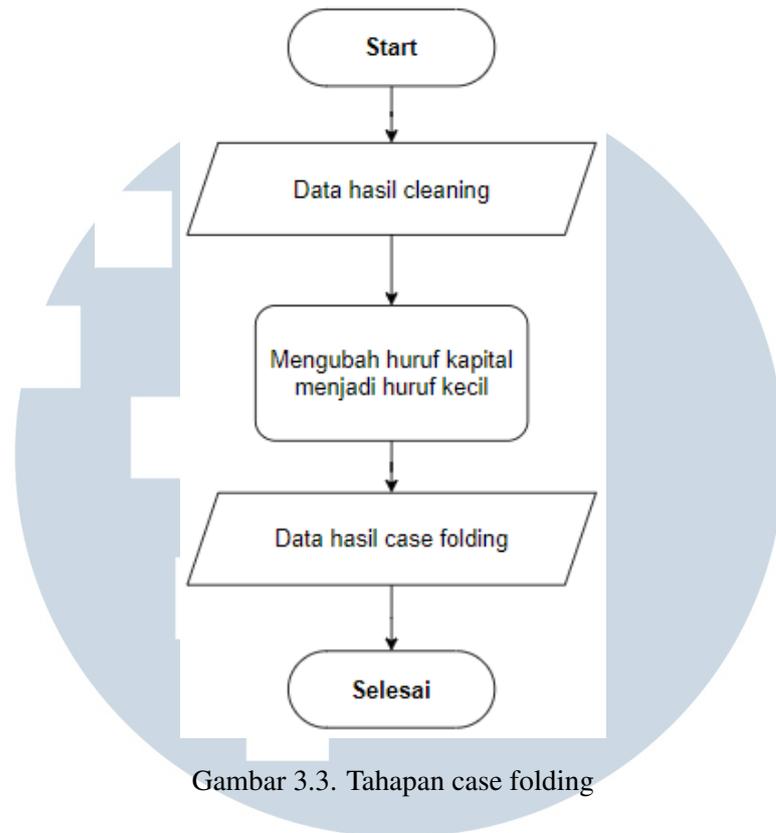
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.2. Tahapan data cleaning

- b. *Case Folding* adalah proses mengubah semua huruf dalam teks menjadi huruf kecil dari huruf besar untuk menghindari perbedaan pengolahan teks berdasarkan kapitalisasi. Misalnya, "Whoosh" dan "whoosh" akan dianggap identik setelah case folding.

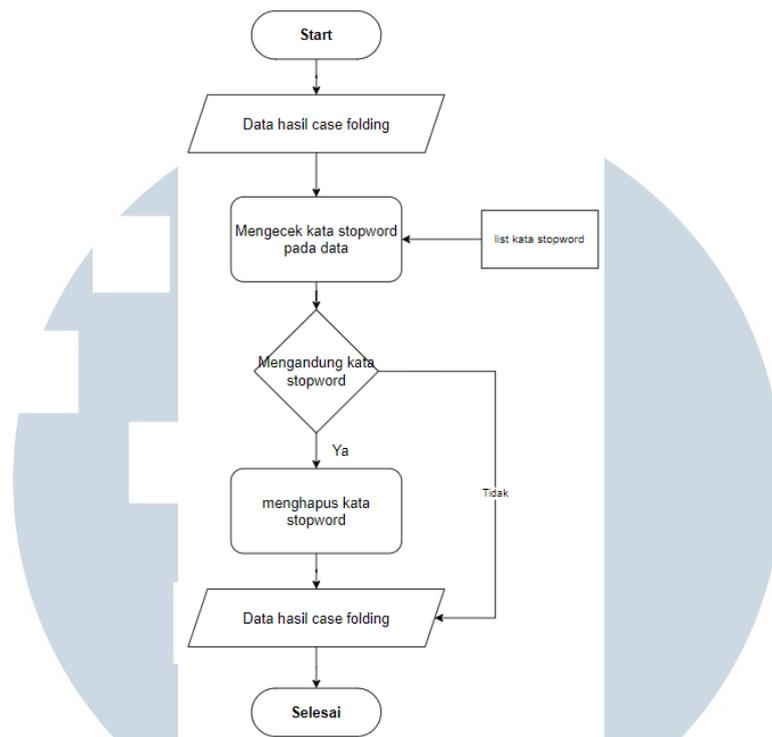
U M M N
 U N I V E R S I T A S
 M U L T I M E D I A
 N U S A N T A R A



Gambar 3.3. Tahapan case folding

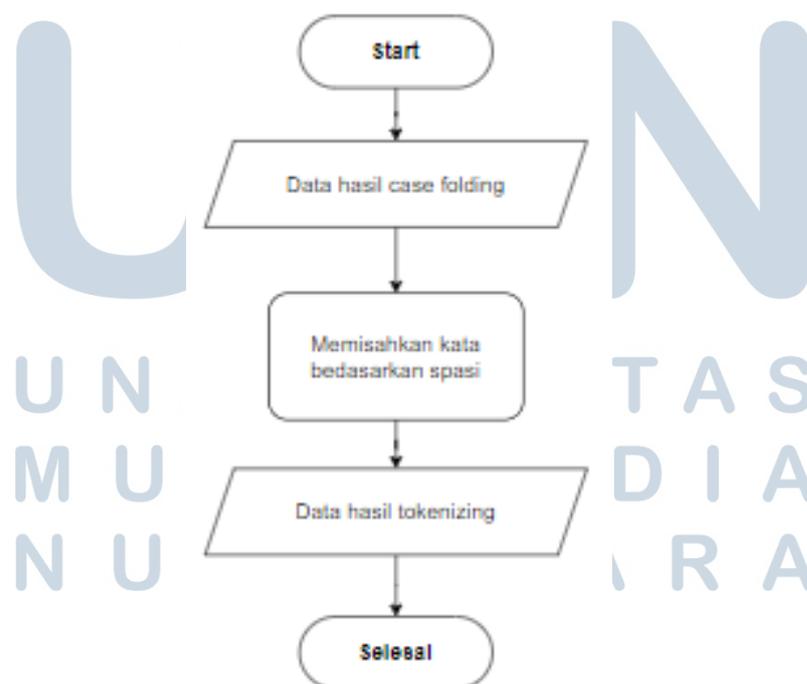
- c. *Stopwords Removal* merupakan proses menghilangkan kata-kata umum yang tidak memiliki makna penting, seperti "yang", "dan", "di". Hal tersebut bertujuan untuk mengurangi ukuran data dan mempercepat proses analisis.

UMMN
 UNIVERSITAS
 MULTIMEDIA
 NUSANTARA



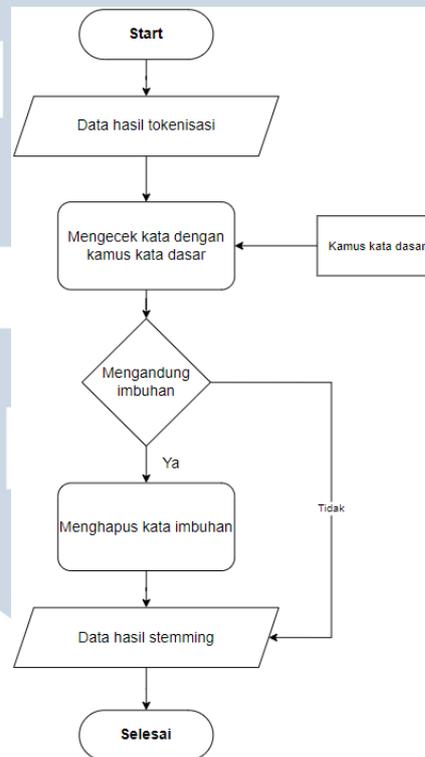
Gambar 3.4. Tahapan stopwords removal

d. *Tokenization* Adalah proses memecah teks menjadi kata-kata atau frasa-frasa tunggal, yang memudahkan analisis kata dasar melalui proses stemming.



Gambar 3.5. Tahapan tokenization

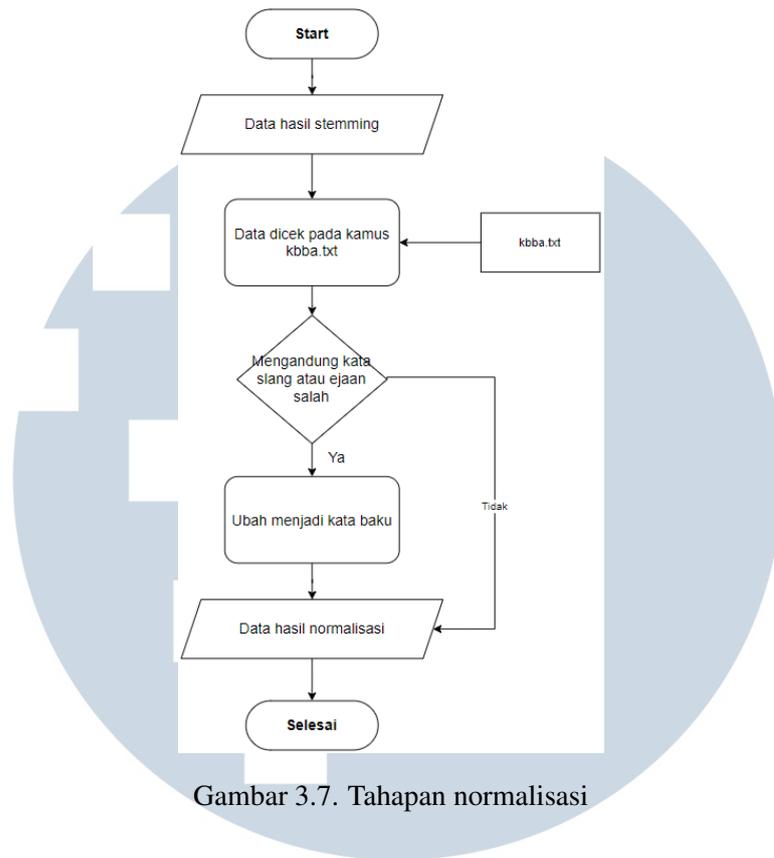
- e. *Stemming* adalah proses mengubah kata menjadi bentuk dasarnya atau menghilangkan imbuhan. Ini membantu menemukan hubungan antara kata-kata yang memiliki akar kata yang sama.



Gambar 3.6. Tahapan stemming

- f. *Normalization* adalah proses mengubah teks tidak baku menjadi bentuk baku. Misalnya, singkatan "bkn" akan diubah menjadi "bukan" selama proses normalisasi.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.7. Tahapan normalisasi

3.2.3 Data Labeling

Pelabelan dataset merupakan tahap penting dalam analisis sentimen yang melibatkan penentuan kelas atau sentimen yang sesuai untuk setiap tanggapan ulasan dalam dataset. Dalam penelitian ini, terdapat dua kategori kelas yang digunakan, yaitu data positif yang mencakup tanggapan positif terhadap aplikasi Whoosh dan data negatif yang mencakup tanggapan negatif. Proses pelabelan akan dilakukan menggunakan *library BERT* dengan model *bert-base-multilingual-uncased-sentiment*. *Library* ini memungkinkan pengklasifikasian teks menjadi sentimen positif, negatif, dan netral. Hasil yang teridentifikasi sebagai sentimen netral akan diperiksa secara manual dan dipindahkan ke kategori sentimen positif atau negatif sesuai dengan konteksnya. Dengan menggunakan BERT, penelitian ini dapat mengidentifikasi dan menganalisis sentimen yang terkandung dalam data ulasan aplikasi Whoosh dengan lebih akurat.

3.2.4 Oversampling SMOTE

Dalam mengatasi masalah ketidakseimbangan data pada data latih, metode SMOTE (*Synthetic Minority Over-sampling Technique*) akan digunakan. SMOTE menghasilkan sampel baru dari kelas minoritas untuk menyeimbangkan dataset. Metode ini meningkatkan representasi dan akurasi klasifikasi data dari kelas minoritas. Dengan begitu, model klasifikasi tidak akan bias terhadap kelas mayoritas dan dapat memberikan hasil yang lebih akurat dalam mengklasifikasikan sentimen.

3.2.5 Pembobotan TF - IDF

Setelah tahap *preprocessing*, data ulasan akan dibobotkan menggunakan *TF-IDF*. *TF-IDF* membantu menyoroti kata-kata yang lebih informatif dan relevan dengan menghitung skor setiap kata berdasarkan frekuensinya dalam ulasan dan seberapa jarang kata tersebut muncul.

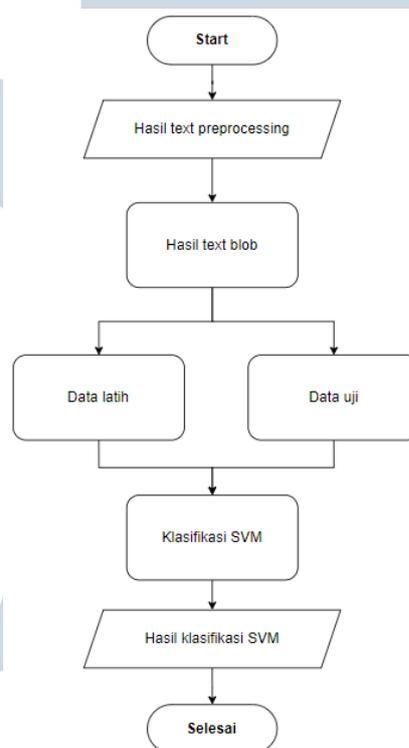
3.2.6 Data Splitting

Data Splitting adalah proses membagi data menjadi dua atau lebih bagian, yang memungkinkan model untuk dilatih, diuji, dan dievaluasi secara efektif [46]. Proses ini tidak hanya penting untuk menilai kinerja model machine learning, tetapi juga untuk mencegah terjadinya *overfitting* dan *underfitting*. *Overfitting* terjadi ketika model terlalu rumit dan cenderung mengandalkan data pelatihan secara berlebihan, sehingga kesulitan dalam menerapkan pola pada data pengujian. Di sisi lain, *underfitting* terjadi saat model terlalu sederhana dan tidak dapat mengenali pola yang ada dalam data dengan baik [47]. Dalam model klasifikasi, *data splitting* mempengaruhi hasil; oleh karena itu, data *train* harus lebih besar dari data *test* karena hasil dipengaruhi oleh jumlah data *train* yang diberikan kepada model.

Penelitian oleh Armanda, M., dan Tobing, F., [48] menerapkan tiga rasio pembagian data pelatihan dan pengujian, yaitu 60:40, 70:30, dan 80:20, dalam menguji performa model analisis sentimen dengan metode SVM pada kernel linear, polinomial, dan RBF. Penelitian ini bertujuan untuk mengetahui pengaruh rasio data pelatihan terhadap akurasi model, serta menentukan kernel yang menghasilkan kinerja terbaik. Penelitian ini akan menerapkan rasio pembagian data latih dan uji yang sama untuk menentukan kinerja model yang optimal berdasarkan pembagian data yang berbeda.

3.2.7 Pemodelan Support Vector Machine

Model SVM akan diterapkan pada data latih yang telah diolah sebelumnya. Proses pelatihan SVM melibatkan pencarian parameter optimal seperti *kernel* dan parameter *cost* (C). Setelah model SVM terlatih, evaluasi akan dilakukan menggunakan data *test* yang telah dipisahkan untuk memeriksa kinerja model dalam memprediksi ulasan. SVM akan memisahkan ulasan positif dan negatif dengan menggambarkan *hyperlane*, yang merupakan garis pembatas antara kedua kelas. Setiap titik data dalam set *train* berperan dalam menentukan posisi hyperplane ini. Ketika ada ulasan baru, SVM akan menempatkannya pada sisi *hyperplane* yang sesuai. Dengan pendekatan ini, SVM dapat mempelajari pola dari data *test* untuk meningkatkan kemampuannya dalam mengklasifikasikan ulasan.



Gambar 3.8. Tahapan klasifikasi SVM

3.2.8 Visualisasi dan Analisis Data

Setelah selesai melakukan pemodelan data, tahap selanjutnya adalah visualisasi dan analisis hasil algoritma tersebut. Visualisasi data merupakan langkah penting untuk menyajikan hasil analisis dalam format yang mudah dipahami. Dalam penelitian ini, visualisasi hasil klasifikasi sentimen aplikasi Whoosh dapat

menggunakan *word cloud*. *Word cloud* merupakan teknik visualisasi yang berfungsi untuk menampilkan kata-kata yang paling sering muncul dalam sebuah dokumen atau kumpulan data. Dalam pendekatan ini, kata-kata ditampilkan dengan ukuran bervariasi; semakin sering suatu kata muncul, semakin besar dan mencolok ukurannya.

Dalam konteks analisis sentimen, *word cloud* berfungsi untuk menampilkan kata-kata yang paling sering diasosiasikan dengan sentimen positif dan negatif terhadap suatu objek atau pembahasan. Kata-kata yang lebih sering muncul dan memiliki ukuran teks yang lebih besar dalam *word cloud* akan menjadi representasi visual yang kuat tentang kata-kata kunci yang paling berperan dalam sentimen pengguna terhadap aplikasi Whoosh.

3.2.9 Pengujian Model

Setelah model terlatih, model SVM akan diuji menggunakan data *test*. Hasil pengujian tersebut menghasilkan laporan mengenai *precision*, *recall*, dan *f-1 score* untuk kedua kategori sentimen, baik positif maupun negatif. Selain itu, akurasi total juga dihitung dan disajikan dalam *classification report*.

3.2.10 Evaluasi

Tahapan akhir dalam perancangan sistem pada penelitian ini adalah mengevaluasi kinerja model SVM yang telah dikembangkan dan analisis kata-kata yang berpengaruh dalam pemodelan tersebut. Evaluasi kinerja model dilakukan untuk mengukur seberapa baik algoritma SVM dalam mengklasifikasikan ulasan aplikasi. Hasil dari *classification report* memberikan informasi mengenai metrik kinerja seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Informasi ini dapat digunakan untuk mengukur keunggulan dan kelemahan dari model yang diterapkan.

Selain itu, hasil sentimen dari ulasan pengguna dapat memberikan wawasan yang lebih mendalam mengenai masalah-masalah yang sering dihadapi pengguna dalam menggunakan aplikasi Whoosh. Visualisasi hasil pelabelan sentimen positif (1) dan negatif (-1) dengan menggunakan *word cloud* dapat membantu mengidentifikasi kata-kata atau frasa yang sering muncul dalam konteks sentimen tertentu. Wawasan ini dapat digunakan sebagai saran perbaikan dan pengembangan aplikasi secara keseluruhan.