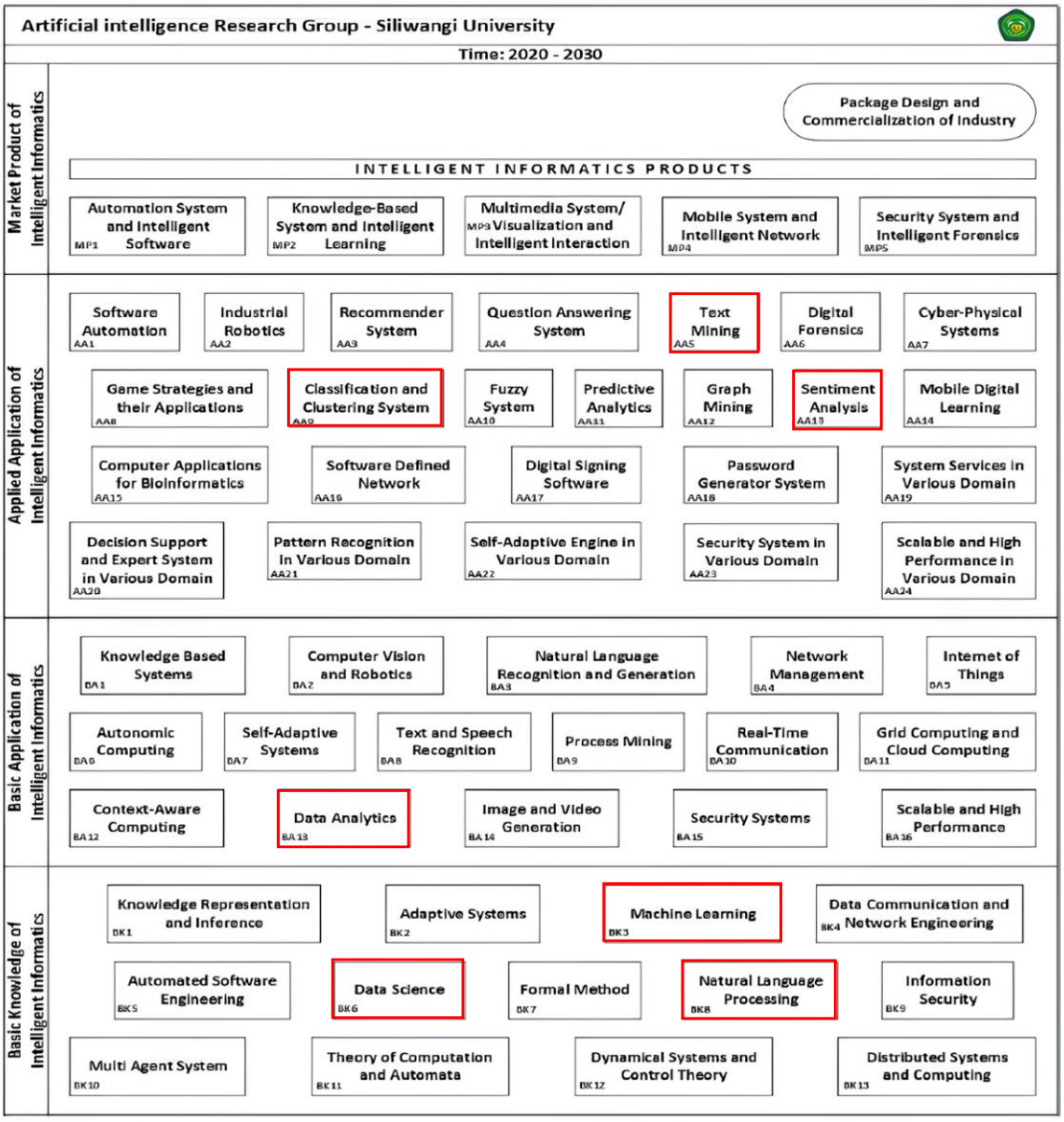
BAB III

METODOLOGI PENELITIAN

3.1 Peta Jalan (*Roadmap*) Penelitian

Roadmap dari penelitian ini masih sejalan dengan peta jalan penelitian Universitas Siliwangi dengan sub bidang *Artificial Intelligennce Research Group – Siliwangi University*. *Roadmap* penelitian tersebut dapat dilihat pada Gambar 3.1.

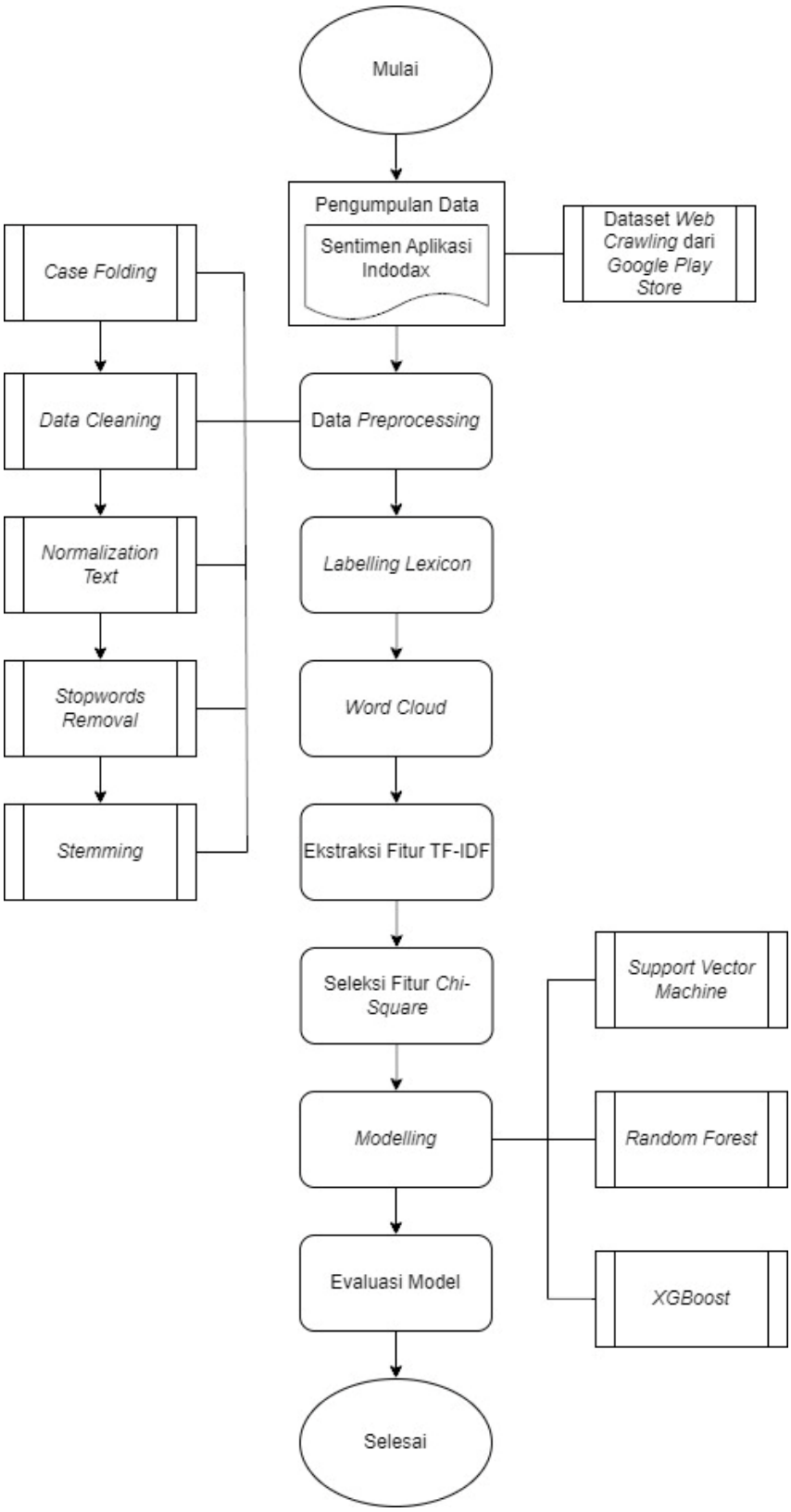


Gambar 3.1 *Artificial Intelligennce Research Group – Siliwangi University*

Pada Gambar 3.1 menunjukkan beberapa bidang utama dalam penelitian yang relevan sebagai panduan metodologi pada penelitian ini yang berjudul "Perbandingan Algoritma *Machine Learning* Untuk Analisis Sentimen *Secara Real-Time* Pada Aplikasi *Indodax* Menggunakan *Feature Selection Chi-Square*".

3.2 Tahapan Penelitian

Penelitian ini terdiri dari beberapa tahapan yang dirancang untuk melakukan analisis sentimen terhadap ulasan *Indodax* di *Google Play Store* menggunakan algoritma *Support Vector Machine*, *Random Forest*, *XGBoost*. Tahapan penelitian ini meliputi proses pengumpulan data menggunakan metode *web crawling* dari *google play store*, data *preprocessing*, *labelling lexicon*, *word cloud*, ekstraksi fitur TF-IDF, seleksi fitur *Chi-Square*, *modelling (Support Vector Machine, Random Forest, XGBoost*, serta evaluasi model. Setiap tahapan memiliki tujuan yang berbeda untuk memastikan akurasi dan validitas hasil penelitian. Alur tahapan penelitian ditujukkan pada Gambar 3.2.



Gambar 3.2 Alur Tahapan Penelitian

Gambar 3.2 adalah alur tahapan penelitian yang diawali dengan proses pengumpulan data ulasan pengguna terhadap aplikasi *Indodax*, yang diperoleh secara langsung dari platform *Google Play Store* melalui metode *web crawling*. Data yang telah dikumpulkan selanjutnya melalui tahap praproses, yang meliputi *case folding, data cleaning, normalization text, stopwords removal,* dan *stemming* untuk menyederhanakan bentuk kata. Tahap berikutnya adalah proses *labelling lexicon* atau pelabelan data untuk mengidentifikasi pola sentimen, disertai visualisasi melalui *word cloud* untuk memperoleh gambaran umum distribusi kata dalam data. Setelah proses praproses dan pelabelan selesai, dilakukan ekstraksi fitur menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) yang bertujuan untuk merepresentasikan teks dalam bentuk numerik. Selanjutnya, diterapkan metode *Feature Selection Chi-Square* untuk memilih fitur yang paling relevan dan mengurangi redundansi data. Proses pemodelan dilakukan dengan menerapkan tiga algoritma klasifikasi, yaitu *Support Vector Machine* (SVM), *Random Forest*, dan *XGBoost*. Evaluasi performa model dilakukan melalui perhitungan *confusion matrix*, serta analisis komparatif terhadap nilai akurasi untuk menentukan algoritma yang memberikan hasil klasifikasi paling optimal.

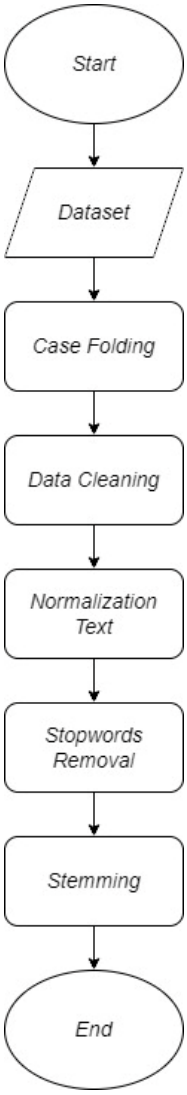
3.3 Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan menggunakan metode *web crawling* yang bertujuan untuk memperoleh data secara otomatis dari *Google Play Store*. Proses *crawling* ini dilakukan dengan menggunakan bahasa pemrograman *Python*. Data yang terkumpul nantinya akan membentuk sebuah *dataset* yang mencakup ribuan ulasan yang siap untuk diolah lebih lanjut melalui tahap-tahap

analisis tertentu, seperti pra-pemrosesan data, analisis sentimen, atau klasifikasi teks. *Dataset* ini akan menjadi bahan utama dalam pengolahan data dan analisis yang mendalam agar mencapai tujuan penelitian yang telah ditetapkan.

3.4 *Preprocessing*

Preprocessing data merupakan langkah awal yang sangat penting dalam proses eksplorasi data. Langkah ini bertujuan untuk mengoptimalkan kualitas data agar lebih mudah dipahami, dianalisis dan siap digunakan dalam tahapan penelitian selanjutnya (Rivanie dkk. 2021). Data yang diperoleh dari berbagai sumber seringkali tidak terstruktur, mengandung *noise* dan tidak konsisten sehingga diperlukan proses *preprocessing* agar data dapat diubah menjadi format yang lebih bersih dan relevan. Pada tahap ini, data yang telah dikumpulkan akan melalui beberapa proses utama dalam *preprocessing*, tahapan *preprocessing* dapat dilihat pada ilustrasi Gambar 3.3.



Gambar 3.3 Tahapan *Preprocessing*

Pada ilustrasi Gambar 3.3 adalah tahapan *preprocessing* yang diawali dengan *Dataset*, *Case Folding*, *Data Cleaning, Normalized Text*, *Stopwords Removal*, *Stemming*.

1. *Case Folding*

Case Folding adalah proses mengubah seluruh teks menjadi huruf kecil (lowercase) untuk memastikan konsistensi dalam data. Langkah ini penting karena dalam teks, kata-kata dengan huruf besar atau kecil sering kali dianggap berbeda

oleh sistem padahal secara makna sama. Misalnya, kata "Data", dan "DATA" akan diubah menjadi "data" untuk menjaga konsistensi serta menghilangkan karakter yang bukan merupakan huruf.

2. *Data Cleaning*

Data Cleaning adalah proses penghapusan karakter-karakter yang tidak sesuai ketentuan yang dibuat seperti huruf atau karakter diluar alphabet a-z (termasuk tanda baca), menghapus *link* atau *URL*, *hashtag*, *username* (Nurtikasari dkk. 2022)

3. *Normalization Text*

Normalization Text adalah proses penyederhanaan teks dengan mengubah kata atau istilah yang tidak baku menjadi bentuk yang standar. Proses ini dapat mencakup pengubahan singkatan, penulisan ulang istilah tidak baku. Misalnya, "tdk" menjadi "tidak", "dr." menjadi "dokter".

4. *Stopwords Removal*

Stopwords Removal adalah kata-kata yang sering muncul dalam teks namun memiliki nilai makna yang rendah dalam analisis seperti "yang", "dan", "di", atau "dari". Proses ini bertujuan untuk menghapus kata-kata tersebut agar analisis lebih terfokus pada kata-kata penting yang memiliki makna signifikan.

Library yang digunakan dalam proses *stopword*s *removal* adalah NLTK *(Natural Language Toolkit)* adalah *Library* yang disediakan oleh oleh *Python* untuk membangun program analisis teks (Pasek dkk. 2022).

5. *Stemming*

Stemming adalah proses mengubah kata-kata ke bentuk dasar atau akar katanya. Misalnya, kata-kata seperti "berjalan", "berjalanlah", atau "berjalannya"

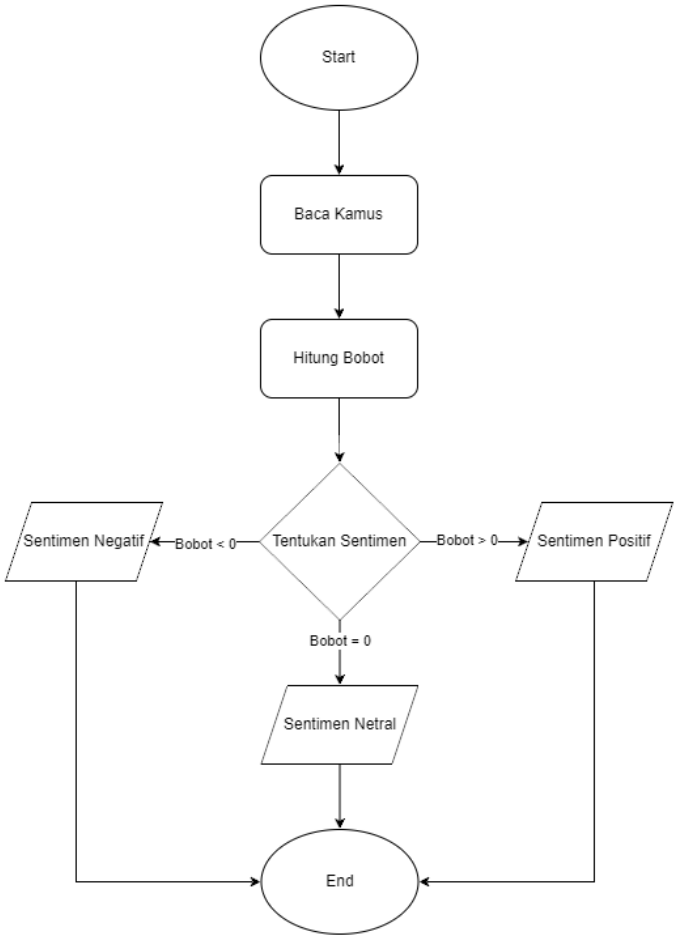
akan diubah menjadi "jalan". Proses ini bertujuan untuk menyederhanakan variasi kata yang berasal dari akar kata yang sama sehingga dapat memudahkan analisis dan mengurangi redundansi dalam data.

Module Stemming yang digunakan dalam penelitian ini, mengambil dari corpus *Stemming* sastrawi. Sastrawi adalah sebuah *corpus* sederhana pada *python* yang merupakan sebuah project Sastrawi yang ditulis dalam *PHP* (Petronella Purba dan Transver Wijaya t.t.).

Setiap tahap *preprocessing* ini dirancang untuk memastikan bahwa data yang dihasilkan lebih bersih, relevan dan siap untuk dianalisis secara mendalam. Dengan data yang telah melalui tahapan *preprocessing*, diharapkan hasil analisis dapat menjadi lebih akurat dan representatif.

3.5 *Labelling Lexicon*

Penelitian ini menggunakan pendekatan *labelling lexicon* untuk mengidentifikasi sentimen teks. *Dataset* yang digunakan terdiri dari daftar kata positif dan negatif yang disimpan yaitu kamus positif dan kamus negatif. Setiap kamus mengandung dua kolom yaitu kata dan bobot sentimen yang mewakili kekuatan sentimen dari masing-masing kata. Penelitian ini menerapkan metode hasil *preprocessing* dari *dataset* yang sedang dianalisis, memberikan label sentimen yang akurat sesuai dengan bobot kata-kata yang ada dalam teks. Ilustrasi alur proses *labelling lexicon* ditujukkan pada Gambar 3.4.



Gambar 3.4 Alur Proses *Labelling Lexicon*

Pada Gambar 3.4 merupakan proses dari *labelling lexicon*, langkah pertama dalam analisis ini adalah memuat data dari kedua kamus tersebut dan mengonversinya menjadi kamus dengan kata sebagai kunci, bobot sebagai nilai. Kamus positif dan negatif diambil dari *GitHub* yang diunggah oleh fajri91 yang bernama *Labelling Lexicon InSet* dengan link *(github.com/fajri91/InSet)*, penelitian terdahulu yang menggunakan kamus ini dilakukan oleh (Fernanda dan Fathoni 2024) dan menghasilkan akurasi 87,83% lebih besar dibandingkan *Labelling Lexicon Vader* dengan akurasi 82,65%. Kamus positif berisi kata-kata dengan bobot positif sebanyak 3610 data, sementara kamus negatif berisi kata-kata dengan bobot

negatif sebanyak 6610 data. Selanjutnya, untuk setiap teks dalam *dataset* kata-kata dipecah dan dibandingkan dengan kata-kata dalam kamus. Jika kata tersebut ada dalam kamus positif, bobotnya ditambahkan ke total bobot. Jika kata tersebut ada dalam kamus negatif, bobotnya dikurangkan dari total bobot. Berdasarkan nilai total bobot, sentimen teks ditentukan:

a. Jika total bobot > 0, teks dilabeli sebagai "positif".
b. Jika total bobot < 0, teks dilabeli sebagai "negatif".
c. Jika total bobot = 0, teks dilabeli sebagai "netral".

3.6 *Splitting Data*

Tahapan selanjutnya yaitu melakukan splitting data agar dapat menjadi data pelatihan dan data pegujian. Pembagian ini berarti memberdayakan kerangka kerja untuk meninjau kumpulan data terlebih dahulu, kemudian, pada saat itu, pengujian informasi akan diselesaikan untuk memastikan tingkat akurasi (Husen dkk. 2023). Pada sistem ini spliting data dapat ditetapkan sesuai dengan kemauan penggunanya (Muafa M. D. Automata 2022). Data pengujian yaitu data yang sebelumnya tidak pernah digunakan dalam penelitian namun berguna untuk menilai berhasil atau tidaknya penelitian, sedangkan data pelatihan yaitu data yang akan digunakan dalam melakukan penelitian (Husen dkk. 2023).

Proses pembagian *dataset* berlandaskan pada penelitian sebelumnya yang dilakukan oleh (N. B. Putri dan Wijayanto 2022) menggunakan rasio 80% untuk data latih dan 20% untuk data uji.

1. Data Latih

   Data latih yang digunakan pada penelitian ini diambil sebanyak 80% dari total *dataset*, dengan hasil akhir *dataset* uji sebanyak 49006 data.

2. Data Uji

   Data uji yang digunakan pada penelitian ini diambil sebanyak 20% dari total *dataset*, dengan hasil akhir *dataset* uji sebanyak 12251 data.

3.7 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) bertujuan untuk membobot kata-kata dalam teks sehingga bisa digunakan sebagai fitur dalam analisis sentimen. TF-IDF adalah teknik pengukuran yang memperhitungkan seberapa sering sebuah kata muncul dalam sebuah dokumen (*term frequency*) serta seberapa relevan kata tersebut diantara seluruh dokumen dalam *dataset* (*inverse document frequency*). Kombinasi kedua komponen ini memberikan bobot yang lebih tinggi pada kata-kata yang sering muncul dalam satu dokumen tetapi jarang ditemukan dalam dokumen lain, sehingga mendapatkan informasi yang lebih relevan untuk analisis.

Proses pembobotan dengan TF-IDF dilakukan untuk mengurangi pengaruh kata-kata umum yang mungkin tidak memiliki makna penting dalam membedakan sentimen, seperti kata penghubung atau kata yang sering muncul dalam berbagai dokumen. TF-IDF menghasilkan nilai numerik untuk setiap kata, yang kemudian digunakan sebagai representasi fitur dalam proses pelatihan model. Dengan demikian, model dapat lebih efektif mengidentifikasi pola sentimen berdasarkan kata-kata yang memiliki bobot penting dan relevan.

3.8 *Feature Selection Chi-Square*

Pada tahap ini, dilakukan seleksi fitur menggunakan *Chi-Square* untuk memilih fitur-fitur yang paling relevan dalam analisis sentimen. Seleksi fitur bertujuan untuk mengurangi jumlah fitur yang akan digunakan oleh model, sehingga dapat meningkatkan efisiensi pemrosesan dan akurasi. Dalam penelitian ini, akan dipilih 2.000 fitur utama berdasarkan nilai *Chi-Square* tertinggi yang menunjukkan bahwa fitur-fitur tersebut memiliki korelasi yang signifikan dengan label sentimen.

Metode *Chi-Square* bekerja dengan menghitung nilai statistik untuk setiap kata dalam teks, lalu menentukan seberapa besar pengaruh masing-masing kata terhadap pengelompokan kelas sentimen (positif, negatif, atau netral). Kata-kata yang memiliki hubungan kuat dengan kategori sentimen tertentu akan mendapatkan skor *Chi-Square* yang lebih tinggi, menunjukkan bahwa kata tersebut relevan dan memiliki pengaruh dalam menentukan sentimen.

Pemilihan 2.000 fitur terbaik dapat mengurangi kompleksitas model dan waktu komputasi yang diperlukan untuk melatih dan menguji model. Jumlah fitur yang lebih kecil memungkinkan model untuk bekerja lebih efisien, karena hanya fitur-fitur yang paling relevan yang disertakan.

Dengan demikian, metode *Chi-Square* dan pemilihan 2.000 fitur utama dapat memberikan keseimbangan antara efisiensi dan akurasi yang memungkinkan model untuk fokus pada kata-kata yang memiliki pengaruh kuat dalam menentukan sentimen teks.

3.9 *Modelling*

Tahap berikutnya adalah membangun model analisis sentimen menggunakan algoritma *Support Vector Machine, Random Forest* dan *XGBoost*. Dalam konteks analisis sentimen, beberapa model tersebut akan memprediksi kategori sentimen berdasarkan probabilitas fitur-fitur yang dihasilkan dari data pelatihan. Beberapa algoritma ini sering dipilih dalam pemodelan teks karena kemampuannya untuk memberikan hasil yang baik bahkan dengan data yang memiliki ukuran fitur yang besar, seperti dalam analisis sentimen. Dengan menggunakan *Feature Selection Chi-Square* diharapkan menghasilkan akurasi yang optimal dan menentukan salah satu algoritma yang mendapatkan tingkat akurasi paling tinggi.

3.10 Uji Performa Terbaik

Pada tahap ini, dilakukan pengujian performa model untuk menentukan jumlah fitur yang paling optimal dalam memengaruhi akurasi model analisis sentimen. Tujuan dari proses ini adalah untuk menemukan jumlah fitur yang memberikan keseimbangan terbaik antara akurasi dan efisiensi model.

3.11 Evaluasi

Setelah model analisis sentimen selesai dibangun dan diuji, langkah selanjutnya adalah melakukan evaluasi kinerja model menggunakan *Confusion Matrix*. *Confusion matrix* merupakan alat yang sangat berguna dalam klasifikasi yang memungkinkan kita untuk memahami bagaimana performa model dalam memprediksi kelas-kelas sentimen yang berbeda. *Confusion Matrix* memberikan informasi rinci tentang jumlah prediksi yang benar dan salah untuk setiap kelas

yang ada sehingga kita dapat menganalisis kinerja model secara lebih mendalam. *Confusion matrix* biasanya terdiri dari empat komponen utama, yaitu:

1. *True Positive* (TP): Jumlah prediksi benar untuk kelas positif, dimana model dengan tepat mengidentifikasi sentimen positif.
2. *True Negative* (TN): Jumlah prediksi benar untuk kelas negatif, dimana model dengan tepat mengidentifikasi sentimen negatif.
3. *False Positive* (FP): Jumlah prediksi salah untuk kelas positif, dimana model mengklasifikasikan sentimen negatif sebagai positif.
4. *False Negative* (FN): Jumlah prediksi salah untuk kelas negatif, dimana model mengklasifikasikan sentimen positif sebagai negatif.

Dengan menggunakan *Confusion matrix*, kita dapat menghitung beberapa matriks evaluasi yang penting, diantaranya:

a) *Accuracy*: Proporsi prediksi yang benar dari seluruh prediksi yang dibuat. Persamaan perhitungan *Accuracy* dapat dilihat pada persamaan 3.1:

$$\frac{TP + TN}{TP + TN + FP + FN} \text{ x}100\% \tag{3.1}$$

b) *Precision*: Proporsi prediksi positif yang benar dari seluruh prediksi positif. Persamaan perhitungan *precision* dapat dilihat pada persamaan 3.2:

$$\frac{TP}{TP + FP} \text{ x } 100\% \tag{3.2}$$

c) *Recall*: Proporsi dari semua kasus positif yang terdeteksi dengan benar oleh model. Persamaan perhitungan *recall* dapat dilihat pada persamaan 3.3:

$$\frac{TP}{TP + FN} \text{ x } 100 \tag{3.3}$$

d) *F1-Score*: Harmonik rata-rata dari *precision* dan *recall*, memberikan gambaran yang seimbang antara keduanya. Persamaan perhitungan *F1-Score* dapat dilihat pada persamaan 3.4:

$$2 \text{ x } \frac{\text{precision x } recall}{\text{precision} + recall} \tag{3.4}$$

Dengan menganalisis *Confusion matrix* dan matriks-matriks evaluasi diatas, kita dapat memperoleh wawasan yang lebih jelas tentang seberapa baik model dapat membedakan antara kelas-kelas sentimen. Misalnya, jika model menunjukkan akurasi tinggi tetapi memiliki nilai *recall* yang rendah untuk kelas tertentu, ini menunjukkan bahwa model mungkin gagal dalam mengenali sentimen positif atau negatif tertentu, meskipun prediksi secara keseluruhan cukup baik.

Oleh karena itu, evaluasi menggunakan *Confusion matrix* sangat penting untuk memahami dan meningkatkan kinerja model analisis sentimen, serta memberikan informasi yang berharga untuk langkah-langkah selanjutnya.