BAB III

METODOLOGI PENELITIAN

3.1 Gambaran Umum Objek Penelitian

Sektor perbankan memainkan peran penting dalam mengawasi stabilitas keuangan sebuah negara. Tantangan utama yang dihadapi oleh sektor perbankan adalah tingginya tingkat *Non-Performing Loan* (NPL) yang dapat merugikan bank melalui kehilangan pendapatan bunga dan mempengaruhi profitabilitas serta likuiditas. Kasus krisis moneter pada tahun 1997 hingga 1998 memberikan dampak besar dari NPL yang tinggi sehingga menyebabkan kebangkrutan pada beberapa bank. Walaupun, NPL di Indonesia sudah mengalami penurunan namun masih terdapat beberapa bank yang memiliki tingkat NPL di atas 3% sehingga dapat mengancam stabilitas keuangan. Oleh karena itu, analisis terhadap NPL dengan menggunakan *machine learning* dapat membantu sektor perbankan, khususnya industri perbankan dalam menganalisis dan mengidentifikasi potensi NPL lebih awal, mitigasi risiko, dan meningkatkan manajemen pengelolaan kredit. Penelitian ini bertujuan untuk membandingkan algoritma *machine learning*, seperti *Random Forest, Decision Tree,* dan *XGBoost* untuk memperoleh model terbaik dalam menganalisis NPL pada industri perbankan. Penelitian ini menggunakan data debitur dari industri perbankan dengan periode waktu dari tahun 2023 hingga 2024. Data tersebut akan diproses untuk melatih ketiga algoritma *machine learning* tersebut dan mengevaluasi performa dari setiap algoritma menggunakan metrik, seperti akurasi, presisi, *recall*, dan *F1-Score*. Hasil evaluasi tersebut digunakan untuk menentukan algoritma mana yang paling terbaik untuk mengidentifikasi dan menganalisis NPL, serta memberikan rekomendasi kepada industri perbankan dalam manajemen risiko kredit menggunakan algoritma yang terpilih.

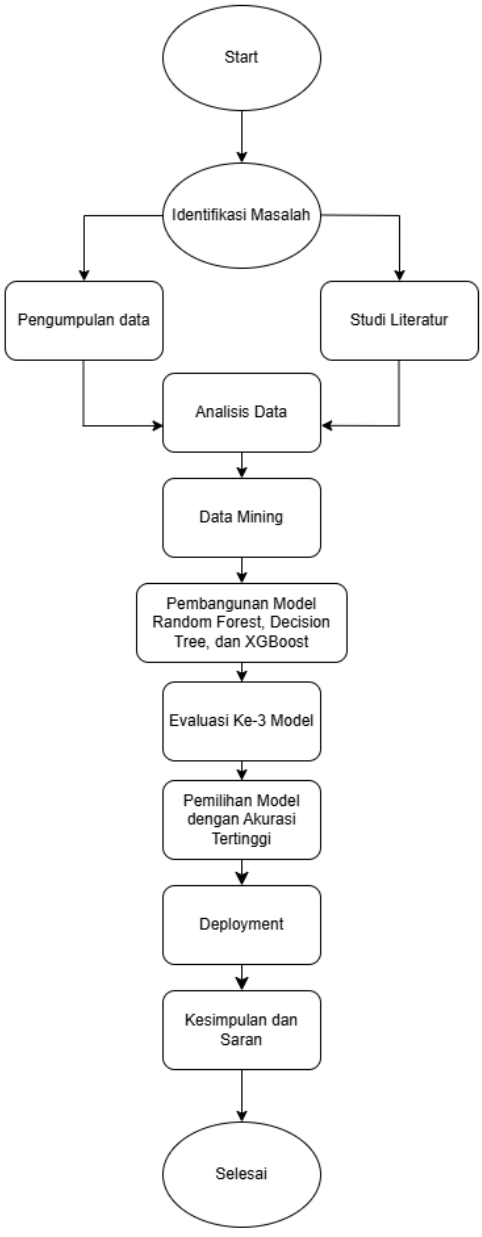
3.2 Metode Penelitian

Gambar 3.1 menunjukkan gambaran kerangka pemikiran atau alur pada penelitian ini. Berikut merupakan penjelasan untuk tiap tahap yang dilakukan:

1. Identifikasi Masalah: Tahapan ini merupakan tahap pertama untuk mengidentifikasi apa yang menjadi permasalahan pada penelitian ini, sehingga menjadi kunci utama adanya penelitian ini.
2. Pengumpulan Data: Pada tahapan ini dibutuhkan data yang berkaitan dengan permasalahan yang telah teridentifikasi pada penelitian. Pada penelitian ini, data yang digunakan berasal berasal dari industri perbankan yang berupa data debitur dengan periode waktu dua tahun, dari tahun 2023 hingga 2024. Data yang diperoleh adalah data dari bulan Januari hingga Desember yang akan diproses dan digabungkan menjadi satu *dataset* penelitian.
3. Studi literatur: Selain dari data yang telah dikumpulkan dibutuhkan juga wawasan yang mendukung penelitian ini dari studi literatur. Studi literatur dibutuhkan agar dapat memperkuat penelitian dengan adanya berbagai wawasan, seperti berbagai teori dan penelitian terdahulu yang mendukung penelitian.
4. Analisis data: Data yang telah terkumpul akan dianalisis agar dapat mengetahui keterkaitan antara variabel dependen dan independen. Proses analisis data dapat dimanfaatkan sebagai dasar perencanaan dan pengambilan keputusan dalam penelitian ini.
5. *Data mining*: Pada tahap ini, proses *data mining* dilakukan dengan memanfaatkan *tools data mining* berupa *Visual Studio Code* dan bahasa pemrograman *Python*. Selain itu, proses *data mining* akan menggunakan metode CRISP-DM.
6. Pembangunan model *Random Forest*, *Decision Tree*, dan *XGBoost*: Setelah melakukan analisis data, maka akan dilakukan pembuatan model klasifikasi menggunakan algoritma *Random Forest, Decision Tree*, dan *XGBoost* untuk menganalisis NPL.
7. Evaluasi ke-3 model: Setelah model *Random Forest*, *Decision Tree*, dan *XGBoost* dibangun maka diperlukan evaluasi dengan menggunakan metrik evaluasi yang disesuaikan dengan karakteristik data dan tujuan analisis.
8. Pemilihan model dengan akurasi tertinggi: Tahap evaluasi sebelumnya menggunakan metrik evaluasi untuk melihat model mana

yang memiliki tingkat akurasi paling tinggi. Setelah memperoleh model yang terbaik maka akan dilakukan pelaporan.

9. *Deployment*: Tahap *deployment* merupakan tahapan terakhir sebelum dilakukan pengambilan kesimpulan dan saran, dengan mengimplementasikan model yang terpilih ke dalam aplikasi berbasis *website*, yaitu *Streamlit*.
10. Kesimpulan dan saran: Berdasarkan analisis dan penelitian yang telah dilakukan, kesimpulan akhir dari penelitian ini akan disajikan, beserta sejumlah rekomendasi yang berkaitan dengan topik yang diteliti.



Gambar 3. 1 Kerangka Pemikiran

3.3 Teknik Pengumpulan Data

3.3.1 Populasi dan Sampel

Populasi dalam penelitian ini mencakup seluruh data pinjaman yang diberikan oleh industri perbankan selama periode yang telah ditentukan, yaitu tahun 2023 hingga 2024, yang mencakup beberapa informasi terkait debitur, pinjaman, serta status pembayaran dengan total 12 atribut. Sampel penelitian ini adalah sebagian data pinjaman dari populasi yang akan diambil yaitu kolom *aging_tunggakan"*. Sampel ini digunakan untuk analisis model *Non-Performing Loan* (NPL) dengan algoritma *Random Forest*, *Decision Tree*, dan *XGBoost*. Dengan menganalisis data NPL dalam sampel ini, maka penelitian dapat memperoleh temuan dan kesimpulan yang dapat diimplementasikan pada populasi penelitian.

3.3.2 Periode Pengambilan Data

Penelitian ini, melibatkan periode pengambilan data dua tahun terakhir dari tahun 2023 hingga tahun 2024 terkini untuk memperoleh gambaran yang jelas terkait data NPL pada industri perbankan di tahun mendatang.

3.4 Teknik Analisis Data

Sesuai dengan metodologi penelitian yang telah ditetapkan sebelumnya, yaitu *framework* CRISP-DM dan algoritma klasifikasi, dibutuhkan alat dan bahasa pemrograman yang tepat untuk melaksanakan proses *data mining*. Berikut merupakan perbandingan dari bahasa pemrograman dan *tools* untuk memproses *data mining* serta *tools* visualisasi pada penelitian:

Tabel 3. 1 Perbandingan Bahasa Pemrograman

Indikator	Python	Bahasa R
Bidang Kegunaan	Untuk *data science* dan *software development*.	Untuk analisis statistik.
Kapasitas	Bahasa pemrograman yang berfokus pada objek data yang dipakai dalam analisis *big data*.	Bahasa pemrograman statistika yang mampu untuk menganalisis dan memanipulasi data pemodelan statistika.
User Interface	Lebih variatif karena sudah terintegrasi dengan aplikasi lain.	Terbatas karena tidak terintegrasi dengan aplikasi lain.

Indikator	Python	Bahasa R
Tingkat Kesulitan	Lebih mudah dipahami dan dipelajari karena memiliki struktur yang rapi.	Sulit dipelajari sehingga biasanya digunakan oleh ahli statistik.

Berdasarkan Tabel 3.1, peneliti menggunakan bahasa pemrograman Python dalam mengolah dan menganalisis data untuk menganalisis NPL pada industri perbankan. Hal ini dikarenakan, lewat tabel yang ada di atas bahasa pemrograman Python lebih cocok untuk digunakan karena penelitian ini dilakukan pada bidang *data science* dan berfokus pada objek data yang dipakai dalam analisis *big data*. Python sendiri juga fleksibel untuk digunakan pada bidang *machine learning* dan analisis data yang memiliki *framework* dan *library* lengkap. Kemudian, Python digunakan pada penelitian ini karena dapat mengakses dan memodifikasi sumber kodenya yang dapat membantu peneliti untuk menyesuaikan algoritma atau metode sesuai kebutuhan. Selain itu, Python juga memiliki sintaks yang mudah dibaca yang dapat kolaborasi atau mengkomunikasikan temuan dan metodologi kepada orang lain dengan lebih mudah [40].

Tabel 3. 2 Perbandingan Tools Data Mining

Indikator	R Studio	Visual Studio Code
Tujuan	Dirancang khusus bekerja dengan bahasa pemrograman R.	Dirancang sebagai editor kode yang mendukung berbagai bahasa pemrograman.
Penggunaan Umum	Visualisasi data dan analisis statistik. Serta digunakan untuk mengembangkan dan menjalankan R.	Mengembangkan aplikasi, *website* dan bekerja dengan banyak bahasa pemrograman, seperti JavaScript, Python, dan C++.
Fitur Utama	Fitur penyusun kode, *debugger*, dan penyorot sintaks.	Integrasi *Git*, penyusun kode, *debugger*, *highlight syntax*.
Kecepatan Eksekusi	Lebih lambat untuk beberapa tugas atau operasi yang memerlukan kecepatan.	Lebih cepat untuk eksekusi dibanding RStudio, dengan optimasi untuk banyak tugas pemrograman.
Integrasi dengan *Tools* Eksternal	Dapat diintegrasikan dengan *tools* eksternal, tapi membutuhkan konfigurasi tambahan.	Dapat diintegrasikan dengan *tools* eksternal, seperti *Git, Docker,* dan *Database Management Systems*.

Berdasarkan Tabel 3.2, peneliti menggunakan *tools data mining* berupa *Visual Studio Code* karena untuk bahasa pemrograman yang digunakan adalah Python. *Visual Studio Code* merupakan *tools* yang mendukung bahasa

pemrograman Python, sedangkan R *Studio* hanya berfokus pada bahasa pemrograman R saja. Selain itu, *Visual Studio Code* cocok untuk analisis prediksi dan klasifikasi karena fleksibilitasnya yang mendukung berbagai bahasa pemrograman, ekstensi untuk analisis data seperti Python, dan kemampuan *debugging* serta integrasi dengan alat eksternal, memungkinkan pengembangan model prediksi yang efisien [41].

3.5 Teknik Pengujian

Teknik pengujian yang digunakan pada model *Random Forest*, *Decision Tree*, dan *XGBoost* adalah *Cross-Validation* (CV), *GridSearchCV*, *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *Confusion Matrix*. *Cross-Validation* (CV) yang digunakan sebagai teknik pengujian model pada penelitian ini adalah *K-Fold Cross-Validation* yang merupakan teknik untuk mengurangi bias karena model diuji pada keseluruhan bagian data. Pada jenis *Cross-Validation* (CV) ini, data dibagi menjadi bagian misalnya cv=5 dan cv=10, namun pada penelitian ini menggunakan cv=5. Pada *Cross-Validation* (CV), model dilatih pada *fold* 1 dan diuji pada *fold* yang tersisa. Kemudian, proses ini diulang sehingga setiap *fold* digunakan sebagai data uji. Hasil dari seluruh *fold* akan digabungkan untuk memperoleh hasil akhir dari kinerja model.

Setelah melakukan validasi model, maka pada penelitian ini dilakukan *hyperparameter tuning* menggunakan *GridSearchCV*. Jenis *hyperparameter tuning* ini akan mencari seluruh kombinasi *hyperparameter* yang memungkinkan untuk menemukan yang terbaik, sehingga dapat meningkatkan performa model secara lebih signifikan. Setelah model diuji, maka diterapkan beberapa metrik evaluasi yang umum digunakan untuk mengukur performa atau kinerja model menggunakan *accuracy* yang merupakan persentase prediksi benar, *precision* yang merupakan keakuratan prediksi positif, *recall* yang merupakan kemampuan model dalam menangkap prediksi positif, dan *f1-score* yang merupakan gabungan dari *precision* dan *recall*. Setelah menerapkan metrik evaluasi tersebut maka *confusion matrix* dibangun untuk mengukur kinerja model klasifikasi dengan menampilkan perbandingan antara hasil prediksi model dengan label yang sebenarnya, dimana akan menunjukkan jumlah prediksi yang benar dan salah pada berbagai kategori. Dengan menerapkan *confusion matrix* maka dapat memberikan gambaran yang

jelas mengenai kesalahan yang dibuat model dan dapat menghitung metrik *precision, recall,* dan *f1-score* dengan lebih spesifik.

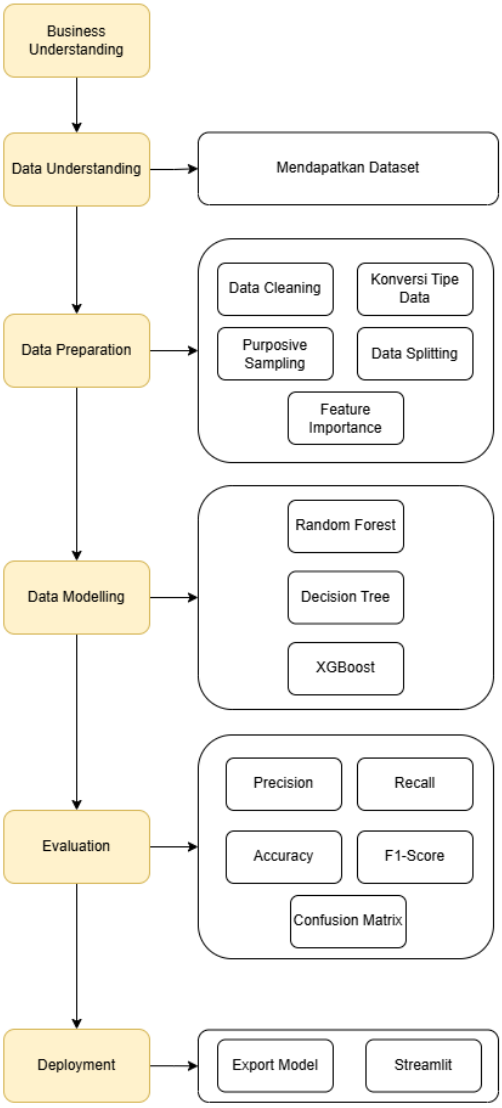
3.6 Metode Pengolahan Data

Metode pengolahan data yang diterapkan dalam penelitian ini meliputi teknik *data mining* dan teknik klasifikasi. Penggunaan metode didasarkan pada perbandingan *framework data mining*. Berikut merupakan perbandingan dari ketiga framework, yaitu CRISP-DM, KDD Process, dan SEMMA:

Tabel 3. 3 Perbandingan Teknik *Data Mining*

Indikator	CRISP-DM	KDD *Process*	SEMMA
Jumlah Tahapan	6	7	5
Fase	1. *Business Understanding* 2. *Data Understanding* 3. *Data Preparation* 4. *Modeling* 5. *Evaluation* 6. *Deployment*	1. *Pre-KDD* 2. *Selection* 3. *Pre-processing* 4. *Transformation* 5. *Data mining* 6. *Interpretation/ evaluation* 7. *Post-KDD*	1. *Sample* 2. *Explore* 3. *Modify* 4. *Model* 5. *Assessment*
Tujuan Utama	Menggunakan *data mining* untuk menyelesaikan masalah bisnis.	Memperoleh pengetahuan yang berguna dari data secara otomatis.	Membangun model prediktif yang dapat diinterpretasikan.
Fleksibilitas	Dapat diadaptasi dan fleksibel untuk berbagai jenis proyek.	Tergantung pada *domain* spesifik.	Dapat diadaptasi dan fleksibel untuk tugas prediktif.

Tabel 3.3 menunjukkan penelitian ini menggunakan *framework data mining* berupa CRISP-DM karena memiliki tahapan yang lebih ringkas dan dari *framework* KDD. Selain itu, beberapa penelitian sebelumnya menunjukkan bahwa teknik CRISP-DM memiliki tahapan yang lebih komprehensif dan fleksibel sesuai dengan kebutuhan penelitian sehingga menjadi alasan utama terhadap pemilihan *framework* ini. Adapun CRISP-DM menekankan pemahaman bisnis sebagai langkah awal yang dapat membantu memastikan solusi yang dihasilkan relevan dengan tujuan bisnis dan masalah yang ingin dipecahkan, serta memiliki tahap evaluasi yang kuat untuk mengukur sejauh mana model dan solusi yang dihasilkan efektif [38].



Gambar 3. 2 Alur Penelitian CRISP-DM

Gambar 3.2 menunjukkan terdapat enam tahapan dalam proses CRISP-DM yang akan diimplementasikan pada penelitian ini dengan penjelasan sebagai berikut:

1. *Business Understanding*

   Pada tahap ini, dilakukan pemahaman yang mendalam tentang tujuan, kebutuhan, dan permasalahan dalam penelitian proyek *data mining* yang berkaitan dengan analisis *Non-Performing Loan* (NPL) pada industri perbankan. Penelitian ini bertujuan untuk membandingkan algoritma *machine learning*, yaitu *Random Forest, Decision Tree*, dan *XGBoost* untuk menemukan model terbaik dalam menganalisis NPL pada industri perbankan.

2. *Data Understanding*

Pada tahap ini, akan dilakukan analisis lebih lanjut mengenai data tentang NPL dari industri perbankan dengan melakukan eksplorasi dan pemahaman data untuk mencari tahu kualitas, karakteristik, dan hubungan antar variabel pada data.

3. *Data Preparation*

Penelitian ini, dilakukan proses pembersihan data, transformasi data, dan penggabungan data agar data siap digunakan untuk pembuatan model *data mining*. Tahap persiapan data dimulai dengan melakukan *data cleaning* dari *dataset* perbulan dari bulan Januari hingga Desember dengan menghapus kolom yang tidak relevan, mengubah tipe data kolom, melakukan *label encoding* pada kolom kanwil, serta mengatasi data yang terduplikasi dan nilai yang hilang. Setelah *dataset* perbulan selesai dibersihkan maka dilakukan proses penggabungan hingga menjadi satu *dataset* yang akan digunakan pada pemrosesan data selanjutnya. Dari *dataset* tersebut dilakukan konversi tipe data kolom ke numerik, penghapusan kolom yang tidak relevan dengan penelitian, melakukan proses *encoding* menggunakan *labelEncoder*, penanganan terhadap *outlier* dengan menggunakan transformasi *log*, penanganan terhadap *missing values* dan data yang terduplikasi. Pada tahap ini juga, data penelitian dibagi menjadi dua bagian, yaitu 80% untuk data pelatihan dan 20% untuk data pengujian.

4. *Data Modeling*

Penelitian ini akan membangun model *data mining* dengan menerapkan metode dan algoritma yang telah dipilih pada data pelatihan yang telah dibagi pada tahap sebelumnya. Penelitian ini juga akan membandingkan algoritma *Random Forest, Decision Tree*, dan *XGBoost* dengan melakukan klasifikasi pada data NPL. Selanjutnya, analisis data NPL akan dilakukan dan model akan dibuat berdasarkan hasil klasifikasi yang telah diperoleh.

5. *Evaluation*

Penelitian ini, evaluasi akan dilakukan terhadap model yang telah dibangun menggunakan data uji. Pada tahap ini, kinerja model *Random Forest, Decision Tree*, dan *XGBoost* akan diukur dalam menganalisis NPL dengan menggunakan berbagai metrik evaluasi seperti presisi, *recall*, akurasi, *F1-score*, serta *confusion matrix*. Selain itu, evaluasi yang dilakukan juga menggunakan *cross validation* dengan menggunakan nilai 5-*fold*, dimana data akan dibagi ke dalam lima bagian dan akan dilakukan pengujian sebanyak lima kali. Penggunaan 5-*fold* didasarkan pada penelitian terdahulu dimana nilai tersebut banyak digunakan pada berbagai studi dan cukup representatif untuk menggambarkan performa model serta mengurangi *overfitting* [26]. Setelah model yang dibangun memenuhi tujuan *business understanding* dan performa model mencapai tingkat optimal, maka tahap selanjutnya, yaitu *deployment*.

6. *Deployment*

Pada tahap ini, dilakukan implementasi model *data mining* menggunakan model yang telah dievaluasi dan disetujui sebelumnya untuk pengambilan keputusan terkait permasalahan penelitian ini. Pada tahap ini, jika model dengan akurasi tertinggi terpilih maka akan dilakukan pembangunan model yang akan dikembangkan ke dalam aplikasi berbasis *website*, yaitu *Streamlit*.

Penelitian ini juga menggunakan teknik klasifikasi, yaitu algoritma dari *data mining* berupa *Random Forest*, *Decision Tree*, dan *XGBoost*. Berikut ini adalah beberapa perbandingan algoritma klasifikasi yang diterapkan dalam penelitian ini:

Tabel 3. 4 Perbandingan Algoritma Klasifikasi

Indikator	Random Forest	Decision Tree (C4.5)	XGBoost
Prinsip Dasar	*Ensemble* dari banyak *Decision Trees* untuk meningkatkan akurasi dengan randomisasi fitur dan data.	Membagi data berdasarkan keputusan berbentuk pohon biner.	Menggunakan *ensemble Decision Tree* untuk meningkatkan akurasi prediksi.
Parameter	Parameter jumlah pohon (estimators) dan kedalaman maksimum setiap pohon.	Kedalaman pohon, jumlah minimum sampel untuk *split*, dan kriteria pemisahan.	*Learning rate, max_depth, n_estimators,* dan *subsample*.

Indikator	Random Forest	Decision Tree (C4.5)	XGBoost
Overfitting	Lebih tahan terhadap *overfitting* karena menggabungkan hasil dari banyak pohon, yang mengurangi variansi.	Cenderung lebih rentan terhadap *overfitting* jika tidak diatur dengan baik.	Terjadi ketika model terlalu kompleks.
Bentuk *Cluster*	Beberapa pohon yang bekerja bersama untuk membentuk hasil yang lebih *robust*.	Menghasilkan satu pohon keputusan untuk membagi data.	Pohon keputusan yang berurutan dan setiap pohon akan memperbaiki kesalahan dari pohon sebelumnya.
Penggunaan	Lebih akurat dan tahan terhadap *overfitting* dengan kompleksitas yang lebih tinggi.	Lebih cepat dan mudah diinterpretasikan untuk model sederhana.	Lebih akurat pada *dataset* berukuran besar dan kompleks, tapi memerlukan penyesuaian *hyperparameter* agar tidak terjadi *overfitting*.

Tabel 3.4 menjelaskan perbandingan algoritma klasifikasi antara *Random Forest*, *Decision Tree* menunjukkan perbedaan dari prinsip dasar, parameter, *overfitting*, bentuk *cluster*, dan penggunaan dari setiap algoritma tersebut. Perbandingan algoritma pada penelitian ini bertujuan untuk menentukan algoritma yang paling sesuai untuk permasalahan sedang diteliti, mengukur performa dan untuk memperoleh hasil yang akurat dalam memecahkan masalah penelitian, membantu menilai efisiensi dan skalabilitas kedua algoritma tersebut, serta melakukan pengambilan keputusan yang jelas [39].