BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Referensi dari penelitian sebelumnya membantu dalam memahami penggunaan metode dan algoritma pada kasus prediksi *customer churn*. Berbagai penelitian terdahulu memanfaatkan model *machine learning* seperti *Logistic Regression* (LR), Support Vector Machine (SVM), *Random Forest* (RF), *XGBoost*, Artificial Neural Network (ANN), *Decision tree* (DT), K-Nearest Neighbor (KNN), serta beberapa metode *ensemble* dan *boosting* seperti AdaBoost, Gradient *Boosting*, dan LightGBM. Tahapan *preprocessing* yang digunakan pada penelitian terdahulu meliputi *data cleaning*, encoding, normalisasi, *feature selection*, serta penanganan *class imbalance*.

Selain itu, beberapa penelitian juga menerapkan teknik optimasi model dan metode *ensemble* untuk meningkatkan performa prediksi *customer churn*. Berbagai pendekatan tersebut menunjukkan bahwa *machine learning* memiliki kemampuan yang cukup baik dalam membantu perusahaan mengidentifikasi pelanggan yang berpotensi melakukan *churn* berdasarkan pola historis data pelanggan.

Tabel 2.1 menyajikan penelitian terdahulu secara lebih rinci, meliputi *dataset*, metode penelitian, hasil performa model, serta research gap yang masih dapat dikembangkan pada penelitian selanjutnya.

Tabel 2. 1 Penelitian Terdahulu

[12]	
Jurnal	Frontiers in *Artificial intelligence*
Judul	A Predictive Analytics Approach to Improve Telecom's *Customer retention*
Penulis dan Tahun	Asem Omari, Omaia Al-Omari, Tariq Al-Omari, Suliman Mohamed Fati (2025)
Dataset	IBM Telco *Customer churn Dataset*, 7043 data, 21 fitur
Metode	Tahap *preprocessing* pada penelitian ini meliputi penanganan *missing value*, encoding pada data kategorikal, normalisasi menggunakan metode Z-score, serta deteksi *outlier*. Pada penelitian ini, proses seleksi fitur dilakukan menggunakan metode Correlation Analysis, Mutual Information, dan Recursive Feature Elimination (RFE). Selain itu, penelitian turut

	mengimplementasikan beberapa algoritma klasifikasi, yaitu SVM, *Logistic Regression*, KNN, dan *Naïve Bayes*. Selanjutnya, *dataset* dibagi menjadi data pelatihan sebesar 80% dan data pengujian sebesar 20% menggunakan *stratified sampling* serta validasi silang 5-fold.
Hasil	*Model SVM* menghasilkan performa terbaik dengan *Accuracy* 97%, *Precision* 0.97, *Recall* 0.94, *F1-score* 0.95, dan *ROC-AUC* 0.98. *Model* lain menunjukkan performa lebih rendah yaitu *Logistic Regression* dan *KNN* dengan *accuracy* 89%, serta Naive Bayes 88%.
Kelemahan / Gap	Penelitian belum melakukan optimasi hyperparameter secara mendalam pada model yang digunakan sehingga potensi peningkatan performa dan stabilitas model masih memiliki ruang untuk ditingkatkan. Selain itu, meskipun penelitian telah menyoroti pentingnya interpretabilitas model dalam prediksi *customer churn*, analisis terhadap kontribusi fitur terhadap hasil prediksi masih belum dibahas secara komprehensif. Dengan demikian, diperlukan penelitian berikutnya yang tidak hanya mengutamakan akurasi model, tetapi juga menggabungkan pendekatan interpretabilitas untuk meningkatkan transparansi dan pemahaman terhadap hasil prediksi *customer churn*.
[13]	
Jurnal	Journal of Soft Computing and *Data mining*
Judul	*Customer churn prediction* of Telecom Company Using *Machine learning* Algorithms
Penulis dan Tahun	Angela Yi Wen Chong, Khai Wah Khaw, Wai Chung Yeong, Wen Xu Chuah (2023)
Dataset	Kaggle Telco *Customer churn Dataset*, 7043 data, 19 fitur
Metode	Penelitian menggunakan 6 algoritma supervised *machine learning*, yaitu *SVM, XGBoost, RF, KNN, AdaBoost,* dan *LR*. Tahapan metode meliputi *preprocessing data* (*data cleaning*, handling *missing value*, dropping column ID), *feature selection* menggunakan *Random Forest feature importance*, *One-hot encoding* untuk variabel kategorikal, *data splitting* 70% *training* dan 30% *testing*, serta *feature scaling* menggunakan *StandardScaler*.
Hasil	*Model XGBoost* memberikan performa terbaik dengan *accuracy* 79.67%, *precision* 64.67%, *recall* 51.87%, dan *F1-score* 57.57%. *Model* lain memiliki performa sedikit lebih rendah seperti *AdaBoost* (79.53%), *Logistic Regression* (79.19%), *Random Forest* (79.05%), *SVM* (78.06%), dan *KNN* (77.01%).
Kelemahan / Gap	Studi ini masih memiliki kendala terkait penanganan data dengan distribusi kelas yang tidak seimbang yang dapat mempengaruhi stabilitas dan performa model prediksi *customer churn*. Selain itu, proses optimasi model belum dilakukan secara menyeluruh pada seluruh parameter yang digunakan sehingga potensi peningkatan performa dan kemampuan generalisasi model masih dapat

	dikembangkan lebih lanjut. Dari sisi interpretabilitas, penelitian juga belum membahas kontribusi masing-masing fitur terhadap hasil prediksi secara mendalam, sehingga faktor-faktor yang mempengaruhi *customer churn* belum dijelaskan secara komprehensif. Oleh karena itu, penelitian selanjutnya dapat mengintegrasikan pendekatan XAI untuk meningkatkan transparansi dan pemahaman terhadap hasil prediksi model.
[14]	
Jurnal	Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Judul	*Customer churn prediction* in Telecommunication Industry Using *Classification* and *Regression* Trees and Artificial Neural Network Algorithms
Penulis dan Tahun	Sulaiman Olaniyi Abdulsalam, Micheal Olaolu Arowolo, Yakub Kayode Saheed, Jesutofunmi Onaope Afolayan (2022)
Dataset	Telecom *churn dataset* (BigML), 3333 data, 20 atribut
Metode	Penelitian menerapkan metode *Relief-F* untuk melakukan *feature selection* dalam menentukan fitur-fitur penting pada *dataset*. *Model* klasifikasi yang digunakan meliputi *ANN* dan *Classification and Regression Tree (CART). Dataset* kemudian dipisahkan menjadi 75:25. Proses implementasi dilakukan menggunakan MATLAB melalui tahapan *preprocessing*, seleksi fitur, pelatihan *model*, serta evaluasi performa.
Hasil	*Model ANN* menunjukkan performa terbaik dengan *Accuracy* 93,88%, *Precision* 83,02%, *Sensitivity* 72,7%, dan F-Score 77,53%, sedangkan CART memperoleh *Accuracy* 91,6%. Hasil menunjukkan bahwa *ANN* lebih efektif dibanding CART dalam memprediksi *churn* pelanggan.
Kelemahan / Gap	Penelitian ini membandingkan algoritma ANN dan CART dalam proses prediksi *customer churn* dengan memanfaatkan *feature selection* untuk meningkatkan performa klasifikasi model. Meskipun demikian, optimasi hyperparameter pada model yang digunakan masih belum dilakukan secara mendalam sehingga potensi peningkatan performa dan kemampuan generalisasi model masih dapat dikembangkan lebih lanjut. Selain itu, interpretasi terhadap faktor-faktor yang mempengaruhi *customer churn* juga belum dibahas secara komprehensif karena analisis model masih terbatas pada hasil prediksi klasifikasi. Dengan demikian, penelitian selanjutnya dapat mengintegrasikan pendekatan XAI untuk meningkatkan transparansi dan pemahaman terhadap kontribusi masing-masing fitur pada hasil prediksi *churn*.
[15]	
Jurnal	International Journal of Computers
Judul	Big Data and Predictive Analytics for *Customer retention*: Exploring the Role of *Machine learning* in E-Commerce

Penulis dan Tahun	Mitra Penmetsa, Jayakeshav Reddy Bhumireddy, Rajiv Chalasani, Mukund Sai Vikram Tyagadurgam, Venkataswamy Naidu Gangineni, Sriram Pabbineedi (2025)
Dataset	E-commerce *customer churn dataset* (Kaggle), jumlah data tidak disebutkan
Metode	Penelitian menerapkan RF sebagai *model* utama dalam prediksi *churn*. Tahapan penelitian mencakup *preprocessing* data seperti penanganan *missing value* dan penghapusan *outlier*, dilanjutkan dengan *One-hot encoding* pada fitur kategorikal, standardisasi menggunakan *Z-score*, serta penyeimbangan data menggunakan SMOTE. *Dataset* dipisahkan menjadi 80% data *training* dan 20% data *testing*. Evaluasi performa *model* dilakukan menggunakan metrik *Accuracy, Precision, Recall, F1-score*, dan *ROC-AUC*.
Hasil	*Model RF* menghasilkan performa yang cukup tinggi dengan nilai *Accuracy* sebesar 95%, *Precision* 98%, *Recall* 83%, *F1-score* 86%, serta *ROC-AUC* 98.51%. Hasil tersebut mengindikasikan bahwa metode *ensemble* seperti *Random Forest* mampu memberikan prediksi *churn* yang efektif pada platform e-commerce.
Kelemahan / Gap	Penelitian ini masih menggunakan *dataset* dengan domain yang terbatas sehingga kemampuan generalisasi model pada karakteristik data yang berbeda belum dapat dievaluasi secara optimal. Selain itu, penelitian lebih berfokus pada performa prediksi tanpa melakukan optimasi model secara mendalam, sehingga potensi peningkatan stabilitas dan performa model masih dapat dikembangkan lebih lanjut. Dari sisi interpretabilitas, analisis terhadap faktor-faktor yang mempengaruhi *customer churn* juga belum dibahas secara komprehensif. Oleh karena itu, penelitian selanjutnya dapat mengintegrasikan optimasi model dan pendekatan *Explainable Artificial Intelligence (XAI)* untuk meningkatkan transparansi serta pemahaman terhadap hasil prediksi *customer churn*.
[16]	
Jurnal	Proceedings of the 3rd International Conference on Financial Technology and Business Analysis
Judul	Predicting *Customer churn* Rate in the Telecommunication Industry Using *Machine learning*
Penulis dan Tahun	Zidong Zhang (2025)
Dataset	Kaggle Telco *Customer churn Dataset*, 7043 data, 21 kolom
Metode	Penelitian menerapkan beberapa algoritma *machine learning*, yaitu *RF, LR, GBT,* dan *NN*. Tahapan penelitian mencakup Exploratory Data Analysis (EDA), *feature engineering*, *target encoding*, *One-hot encoding*, serta *feature selection* menggunakan filter dan *wrapper method*. Selain itu, penelitian juga menerapkan *model tuning* dan *stacking ensemble*. *Dataset* dipisahkan menjadi 5634 untuk *training* dan 1409 untuk *testing*.

Hasil	*Logistic Regression* bersama metode *upsampling* menghasilkan performa terbaik dibandingkan *model* lain, dengan performa antar *model* yang relatif mirip dan tidak menunjukkan *overfitting*.
Kelemahan / Gap	Penelitian sebelumnya masih berfokus pada peningkatan performa model klasifikasi *customer churn*, sementara proses optimasi hyperparameter dan evaluasi model belum dilakukan secara mendalam pada seluruh pendekatan yang digunakan. Selain itu, interpretasi hasil prediksi masih terbatas sehingga faktor-faktor yang mempengaruhi *customer churn* belum dijelaskan secara komprehensif. Dengan demikian, penelitian selanjutnya dapat mengintegrasikan pendekatan XAI untuk meningkatkan transparansi model sekaligus membantu memahami kontribusi fitur terhadap hasil prediksi *customer churn*.
[17]	
Jurnal	Journal of Innovations in Computer Science and Trends in IT
Judul	*Customer churn prediction* using *Machine learning*
Penulis dan Tahun	Eshan Manro, Dheeraj Malhotra, Deepali Kamthania (2025)
Dataset	Kaggle Telco *Customer churn Dataset*, 7043 data, jumlah fitur tidak disebutkan
Metode	Penelitian menerapkan beberapa algoritma *machine learning*, meliputi *DT, RF, XGBoost, LR, LightGBM,* serta *Soft Voting Ensemble*. Tahapan penelitian mencakup *preprocessing* data, *data cleaning*, encoding pada variabel kategorikal, dan *feature selection* menggunakan Correlation Analysis. *Dataset* kemudian dipisahkan menjadi 80:20. Evaluasi performa *model* dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, dan *ROC-AUC*.
Hasil	Hasil eksperimen memperlihatkan bahwa *Logistic Regression* menghasilkan performa paling optimal dengan *accuracy* sekitar 80,38%, *precision* 61%, *recall* 55%, serta *F1-score* sebesar 58%. Selain itu, *model* lain seperti *Random Forest* dan LightGBM juga memiliki performa yang cukup kompetitif, sedangkan pendekatan *ensemble* mampu menghasilkan nilai *precision* dan *F1-score* yang lebih tinggi.
Kelemahan / Gap	Penelitian ini telah menerapkan beberapa algoritma *machine learning* serta optimasi hyperparameter dalam proses pengembangan model prediksi *customer churn*. Meskipun demikian, proses optimasi yang dilakukan masih belum dievaluasi secara menyeluruh pada seluruh model sehingga efektivitas peningkatan performa antar algoritma belum dapat dibandingkan secara lebih mendalam. Selain itu, interpretasi model masih berfokus pada *feature importance* sehingga hubungan dan kontribusi masing-masing fitur terhadap hasil prediksi *churn* belum dianalisis secara komprehensif. Oleh karena itu, penelitian selanjutnya dapat mengintegrasikan pendekatan XAI untuk

	meningkatkan transparansi dan interpretabilitas model *machine learning*.
[18]	
Jurnal	International Journal of Contemporary Research in Multidisciplinary
Judul	*Prediction*s of Customer's *Churn* in Telecommunication Industry Network Area using *Machine learning* (ML) Algorithm
Penulis dan Tahun	Anand Kumar Vishwakarma, Pankaj K. Goswami, Keshav Sinha (2025)
Dataset	Kaggle Telco *Customer churn Dataset*, 7043 data, 21 kolom
Metode	Penelitian menerapkan berbagai algoritma *machine learning*, meliputi RF, *XGBoost*, Gradient *Boosting*, AdaBoost, *ANN*, LR, *SVM*, *KNN*, *Decision tree*, serta Gaussian *Naïve Bayes*. Tahapan penelitian mencakup *preprocessing* data, Exploratory Data Analysis (EDA), *feature selection*, dan encoding pada variabel kategorikal. *Dataset* kemudian dipisahkan menjadi 80:20. Evaluasi performa *model* dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, serta AUC.
Hasil	Hasil eksperimen memperlihatkan bahwa *XGBoost* menghasilkan performa paling optimal dengan *accuracy* sekitar 84,20% dan AUC sebesar 86%, kemudian diikuti oleh *ANN* (82,98%) serta Gradient Boost (82,41%). Selain itu, RF dan LR juga mampu memberikan performa yang cukup baik dalam proses prediksi *churn* pelanggan.
Kelemahan / Gap	Penelitian ini membandingkan beberapa algoritma *machine learning* untuk memprediksi *customer churn* dengan menitikberatkan pada performa klasifikasi *model*. Namun, optimasi hyperparameter pada model yang digunakan belum dilakukan secara mendalam, sehingga performa serta stabilitas model masih memiliki peluang untuk ditingkatkan. Selain itu, interpretasi model masih berfokus pada analisis *feature importance* sehingga pengaruh masing-masing fitur terhadap hasil prediksi *churn* belum dapat dijelaskan secara menyeluruh. Oleh sebab itu, penelitian selanjutnya dapat mempertimbangkan penerapan XAI seperti SHAP, guna mendukung transparansi serta interpretabilitas pada model prediksi *customer churn*.
[19]	
Jurnal	Engineering, Technology & Applied Science Research (ETASR)
Judul	Comparative Analysis of *Oversampling* and *Undersampling* Techniques in Predicting *Customer churn* for Dqlab Telco
Penulis dan Tahun	Bima Pramudya Asaddulloh, Kusrini, Dhani Ariatmanto (2025)
Dataset	Kaggle Telco *Customer churn Dataset*, 7049 data, jumlah fitur tidak disebutkan
Metode	Penelitian menerapkan beberapa algoritma *machine learning*, yaitu RF, *KNN*, GB, dan DT. Penanganan data imbalance dilakukan melalui teknik resampling menggunakan Random

	Undersampling (RUS) serta SMOTE. Tahapan penelitian mencakup EDA, *preprocessing* data seperti *Label encoding*, handling *missing value*s, dan removing duplicates, dilanjutkan dengan *feature selection*, normalisasi, serta pemisahan data menjadi 80% *training* dan 20% *testing*. Evaluasi performa *model* dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, dan AUC.
Hasil	Hasil eksperimen memperlihatkan bahwa kombinasi RF dan SMOTE (RF+SMOTE) menghasilkan performa paling optimal dengan *accuracy* sebesar 79,23%, *precision* 79,32%, *recall* 80,15%, *F1-score* 79,73%, serta AUC 87,25%. Kombinasi tersebut mampu mengungguli metode lain seperti Gradient *Boosting* + RUS maupun *model* yang tidak menggunakan resampling.
Kelemahan / Gap	Penelitian ini berfokus pada penanganan ketidakseimbangan data menggunakan teknik resampling seperti SMOTE dan Random Undersampling (RUS) untuk meningkatkan performa klasifikasi *customer churn*. Meskipun demikian, evaluasi terhadap optimasi model dan stabilitas performa setelah proses resampling masih belum dianalisis secara mendalam. Selain itu, penelitian juga belum membahas interpretasi hasil prediksi untuk menjelaskan faktor-faktor yang mempengaruhi *churn* pelanggan secara lebih komprehensif. Dengan demikian, penelitian selanjutnya dapat mengintegrasikan optimasi model serta pendekatan XAI guna meningkatkan interpretabilitas dan pemahaman terhadap hasil prediksi *customer churn*.
[20]	
Jurnal	Cogent Engineering
Judul	An Analysis on *Classification model*s for *Customer churn prediction*
Penulis dan Tahun	Kathi Chandra Mouli, Ch. V. Raghavendran, V. Y. Bharadwaj, G. Y. Vybhavi, C. Sravani, Khristina Maksudovna Vafaeva, Rajesh Deorari, Laith Hussein (2024)
Dataset	Kaggle Telco *Customer churn Dataset*, 7043 data, 21 fitur
Metode	Penelitian ini membandingkan 10 algoritma klasifikasi, yaitu LR, SVC, Kernel *SVM*, *KNN*, GNB, DT, RF, AdaBoost, *XGBoost*, dan Gradient *Boosting*. Tahapan penelitian meliputi *preprocessing* data, encoding fitur kategorikal, eliminasi fitur menggunakan analisis korelasi dan VIF, penanganan data imbalance menggunakan SMOTE, serta *feature scaling*. Selanjutnya, *dataset* dibagi menjadi data *training* sebesar 80% dan data *testing* sebesar 20%. Evaluasi performa *model* dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, F2-score, dan *ROC-AUC* dengan penerapan 10-fold *cross-validation*.
Hasil	Hasil eksperimen menunjukkan bahwa *Random Forest* memperoleh performa terbaik dengan nilai *ROC-AUC* sekitar 85% serta akurasi sebesar 85,04% pada data pengujian. Selain itu,

	model tersebut juga memberikan hasil evaluasi yang lebih baik dibandingkan *Decision tree* dan *XGBoost* berdasarkan metrik evaluasi yang digunakan.
Kelemahan / Gap	Penelitian ini menitikberatkan pada perbandingan performa beberapa algoritma klasifikasi serta penerapan optimasi hyperparameter dalam pengembangan model. Namun, hasil pengujian masih menunjukkan adanya perbedaan performa yang cukup besar antara data *training* dan *testing* sehingga mengindikasikan potensi overfitting dan kemampuan generalisasi model yang belum optimal. Selain itu, peningkatan performa setelah proses tuning belum memberikan hasil yang konsisten pada seluruh metrik evaluasi yang digunakan. Dari aspek interpretabilitas, penelitian juga belum mengkaji secara mendalam pengaruh masing-masing fitur terhadap prediksi *churn* yang dihasilkan model. Oleh sebab itu, penelitian selanjutnya dapat mempertimbangkan penggunaan evaluasi generalisasi model serta pendekatan XAI guna meningkatkan transparansi dalam prediksi *customer churn*.
[21]	
Jurnal	IAES International Journal of *Artificial intelligence* (IJ-AI)
Judul	Explainable *Machine learning Model*s Applied to Predicting *Customer churn* for E-Commerce
Penulis dan Tahun	Ikhlass Boukrouh, Abdellah Azmani (2025)
Dataset	*Dataset* E-commerce *Customer churn* dari Kaggle dengan 2.841 data pelanggan dan 16 fitur, seperti gender, marital status, *tenure*, preferred order category, order count, complaints, dan satisfaction score. *Dataset* terdiri dari 2.362 pelanggan tidak *churn* dan 479 *churn*.
Metode	Penelitian ini menerapkan tujuh algoritma *machine learning*, yaitu DT, RF, *SVM*, LR, NB, *KNN*, dan *ANN*. Tahapan penelitian meliputi *data cleaning*, encoding pada variabel kategorikal, serta pembagian data menjadi 80% *training* dan 20% *testing*. Evaluasi performa *model* dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix*. Selain itu, penelitian ini juga menerapkan pendekatan XAI menggunakan SHAP untuk mengevaluasi pengaruh setiap fitur terhadap hasil prediksi.
Hasil	Hasil eksperimen memperlihatkan bahwa *ANN* menghasilkan performa paling optimal dengan *accuracy* sebesar 92,09%. *Random Forest* berada pada posisi berikutnya dengan *accuracy* 91,21%, sedangkan *Naïve Bayes* memperoleh performa terendah dengan *accuracy* sekitar 75,04%.
Kelemahan / Gap	Penelitian ini telah mengimplementasikan pendekatan XAI seperti SHAP dan LIME untuk meningkatkan interpretabilitas pada model prediksi *churn*. Namun, fokus utama penelitian masih berada pada peningkatan performa model sehingga proses

	optimasi hyperparameter belum dilakukan evaluasi secara lebih lanjut dan kemampuan generalisasi model secara lebih mendalam. Selain itu, penggunaan *dataset* dengan karakteristik yang terbatas juga berpotensi mempengaruhi stabilitas performa model pada data yang berbeda. Dengan demikian, penelitian selanjutnya dapat mengintegrasikan optimasi model, evaluasi generalisasi, serta interpretasi model secara lebih komprehensif dalam proses prediksi *customer churn*.

Berdasarkan Tabel 2.1, penelitian mengenai prediksi *customer churn* pada industri telekomunikasi terus berkembang melalui penerapan berbagai algoritma *machine learning*. Sebagian besar penelitian memanfaatkan IBM Telco *Customer churn Dataset* maupun *dataset* telekomunikasi lainnya karena memiliki karakteristik yang sesuai untuk permasalahan klasifikasi biner. Algoritma yang digunakan juga semakin beragam, mulai dari *Logistic Regression*, *Decision tree*, *K-Nearest Neighbors (KNN)*, *Naïve Bayes*, *Random Forest*, Artificial Neural Network (ANN), hingga algoritma berbasis *ensemble* dan *boosting* seperti *XGBoost*, Gradient *Boosting*, AdaBoost, dan LightGBM. Selain membandingkan performa algoritma, penelitian terdahulu juga menggunakan berbagai metrik evaluasi, seperti Accuracy, Precision, Recall, F1-Score, ROC-AUC, dan *Confusion matrix* untuk menilai kemampuan model dalam memprediksi *customer churn*.

Di antara penelitian yang dikaji, penelitian oleh Omari et al. [12] dipilih sebagai penelitian pembanding utama karena memiliki tingkat kesesuaian yang paling tinggi dengan penelitian ini, baik dari sisi permasalahan maupun penggunaan IBM Telco *Customer churn Dataset*. Penelitian tersebut juga membandingkan beberapa algoritma *machine learning* sehingga hasilnya dapat dijadikan acuan dalam mengevaluasi pengembangan model pada penelitian ini. Selain menunjukkan bahwa penerapan *machine learning* mampu menghasilkan performa prediksi yang baik, penelitian tersebut juga memberikan peluang pengembangan pada aspek interpretabilitas model dan implementasi hasil prediksi. Oleh karena itu, penelitian ini mengadopsi penggunaan IBM Telco *Customer churn Dataset* serta pendekatan perbandingan beberapa algoritma sebagai dasar penelitian, kemudian mengembangkannya melalui integrasi tahapan optimasi model, evaluasi performa model pada data pengujian, interpretabilitas model, dan implementasi model ke dalam satu alur penelitian.

Berdasarkan hasil kajian penelitian terdahulu, penelitian ini menggunakan *Logistic Regression*, *K-Nearest Neighbors (KNN)*, *Naïve Bayes*, *Random Forest*, dan *XGBoost* sebagai algoritma yang akan dibandingkan. Kelima algoritma tersebut dipilih karena mewakili pendekatan klasifikasi yang berbeda dan banyak digunakan dalam penelitian prediksi *customer churn*. Dengan karakteristik tersebut, perbandingan kelima algoritma diharapkan dapat memberikan gambaran yang lebih komprehensif mengenai performa model pada permasalahan prediksi *customer churn*.

Selain pemilihan algoritma, hasil kajian penelitian terdahulu juga menunjukkan bahwa tahapan pengembangan model memiliki pengaruh yang besar terhadap performa prediksi. Berbagai penelitian telah menerapkan *hyperparameter tuning*, teknik *oversampling* untuk menangani *class imbalance*, maupun pendekatan *Explainable Artificial Intelligence (XAI)* menggunakan SHAP atau LIME. Namun, penerapan teknik-teknik tersebut umumnya masih dilakukan secara terpisah sesuai dengan fokus masing-masing penelitian. Kondisi tersebut menunjukkan bahwa masih terdapat peluang untuk mengintegrasikan berbagai tahapan pengembangan model ke dalam satu alur penelitian yang lebih sistematis dan komprehensif.

Berdasarkan hasil analisis tersebut, penelitian ini mengembangkan proses pembangunan model menggunakan *framework* CRISP-DM dengan mengintegrasikan perbandingan performa beberapa algoritma *machine learning*, penanganan *class imbalance* menggunakan *Synthetic Minority Over-sampling Technique (SMOTE)*, *hyperparameter tuning* menggunakan *GridSearchCV*, evaluasi kemampuan model pada data pengujian model, interpretasi hasil prediksi menggunakan SHAP (Shapley Additive Explanations), serta implementasi model ke dalam aplikasi berbasis *Streamlit*. Pendekatan tersebut diharapkan dapat menghasilkan proses pembangunan model yang lebih komprehensif dibandingkan penelitian terdahulu.

Selain meningkatkan performa model, penelitian ini juga memberikan perhatian terhadap kemampuan model dalam mengidentifikasi pelanggan yang

benar-benar berpotensi melakukan *customer churn*. Pada kasus *customer churn*, kesalahan mengidentifikasi pelanggan yang sebenarnya akan berhenti menggunakan layanan (false negative) dapat menyebabkan perusahaan kehilangan kesempatan untuk melakukan strategi retensi pelanggan. Oleh karena itu, penelitian ini tidak hanya mempertimbangkan nilai accuracy, tetapi juga memberikan perhatian terhadap nilai recall sebagai salah satu indikator penting dalam mengevaluasi kemampuan model mendeteksi pelanggan yang berpotensi melakukan *customer churn*, tanpa mengabaikan metrik evaluasi lainnya.

Dengan demikian, penelitian ini tidak hanya melakukan perbandingan performa beberapa algoritma *machine learning*, tetapi juga mengembangkan proses pembangunan model melalui integrasi penanganan *class imbalance*, optimasi hyperparameter, evaluasi performa model pada data pengujian, interpretasi hasil prediksi menggunakan SHAP, serta implementasi model ke dalam aplikasi berbasis *Streamlit* dalam satu alur penelitian yang sistematis. Pendekatan tersebut diharapkan dapat menghasilkan model prediksi *customer churn* yang memiliki performa prediksi yang baik, mudah diinterpretasikan, dan lebih siap diterapkan untuk mendukung pengambilan keputusan pada industri telekomunikasi.

2.2 Teori yang berkaitan

2.2.1 Customer churn

Customer churn terjadi ketika pelanggan memutuskan untuk berhenti berlangganan dan beralih ke layanan dari perusahaan lain. Dalam konteks manajemen hubungan pelanggan, *churn* mencerminkan kegagalan perusahaan dalam mempertahankan loyalitas pelanggan akibat menurunnya persepsi nilai layanan, baik dari aspek kualitas, harga, maupun pengalaman pengguna secara keseluruhan. Fenomena ini menjadi indikator penting dalam mengevaluasi keberhasilan strategi retensi pelanggan, karena hilangnya pelanggan secara berkelanjutan dapat mengurangi stabilitas basis pelanggan dan melemahkan posisi kompetitif perusahaan [10].

Pada industri telekomunikasi, *customer churn* menjadi permasalahan yang sangat krusial mengingat karakteristik pasar yang kompetitif, tingkat

kemiripan layanan antar penyedia, serta rendahnya hambatan bagi pelanggan untuk berpindah operator. Tingginya tingkat *churn* dapat memberikan dampak terhadap kinerja bisnis perusahaan karena pengeluaran untuk memperoleh pelanggan baru umumnya lebih banyak daripada mempertahankan pelanggan lama. Selain berdampak pada penurunan pendapatan, kondisi *churn* yang tidak terkontrol juga dapat memengaruhi reputasi perusahaan serta menghambat keuntungan jangka panjang. Oleh sebab itu, untuk merencanakan retensi pelanggan dengan lebih baik, bisnis memerlukan metode berbasis data untuk mengidentifikasi pelanggan yang mungkin melakukan *churn* lebih awal [4].

2.2.2 Machine learning

Machine learning merupakan salah satu bidang dalam Artificial Intelligence (AI) yang memungkinkan sistem komputer mempelajari pola dari data secara otomatis tanpa perlu diprogram secara khusus untuk setiap kondisi [22]. Pendekatan ini menitikberatkan pada pengembangan algoritma yang dapat mempelajari informasi dari data melalui proses pelatihan sehingga performa *model* dapat meningkat secara bertahap [23]. Selain itu, *machine learning* dimanfaatkan untuk membangun *model* yang mampu melakukan prediksi, klasifikasi, maupun pengambilan keputusan berdasarkan data historis yang tersedia [24].

Pada umumnya, proses *machine learning* memanfaatkan *dataset* sebagai data *training* untuk membantu model memahami hubungan antara variabel *input* dan *output*. *Model* yang sudah melalui proses pelatihan selanjutnya digunakan untuk melakukan prediksi dan juga klasifikasi terhadap data baru. Performa *model machine learning* dipengaruhi oleh beberapa faktor, seperti kualitas data, pemilihan algoritma, serta parameter yang digunakan selama proses pelatihan [25].

Machine learning bekerja dengan algoritma tujuan untuk mengenali hubungan, pola, dan karakteristik tertentu dari data sehingga sistem dapat meningkatkan performanya seiring bertambahnya data yang diproses [24].

Kemampuan *machine learning* dalam memproses data berukuran besar secara efisien membuat teknologi ini banyak digunakan pada bidang prediksi pelanggan, deteksi fraud, sistem rekomendasi, analisis sentimen, kesehatan, serta pengolahan citra [22], [26].

2.2.3 Supervised learning

Dalam *Machine learning*, *Supervised learning* merupakan metode pembelajaran yang memanfaatkan data ber*Label* dengan target atau output yang telah diketahui sebelumnya [27]. Metode ini memungkinkan *model* mempelajari keterkaitan antara variabel input dan output sehingga mampu mengenali pola data untuk melakukan prediksi maupun klasifikasi pada data baru [28]. Selain digunakan untuk meningkatkan kemampuan prediksi, *Supervised learning* juga berperan dalam mengurangi kesalahan selama proses pelatihan serta meningkatkan kemampuan generalisasi *model* terhadap data yang belum pernah dipakai [29].

Pada *supervised learning*, proses pembelajaran dilakukan penggunaan data pelatihan yang terdiri dari fitur dan *Label* target. *Model* selanjutnya mempelajari pola dari data tersebut untuk menghasilkan fungsi prediksi yang optimal [30]. Pendekatan ini banyak diterapkan pada berbagai bidang, seperti prediksi *customer churn*, deteksi penyakit, klasifikasi citra, analisis sentimen, dan prediksi perilaku pengguna karena mampu memberikan tingkat akurasi yang tinggi apabila didukung oleh data yang relevan serta berkualitas [31].

Secara umum, *supervised learning* dibagi menjadi dua jenis utama yaitu *classification* dan *regression* [32]. *Classification* digunakan untuk memprediksi kategori atau kelas tertentu, misalnya untuk menentukan apakah pelanggan berpotensi melakukan *churn* atau tidak. Sementara itu, regression digunakan dalam memprediksi nilai numerik kontinu, seperti estimasi harga maupun volume penjualan [33].

Karakteristik utama *supervised learning* terletak pada penggunaan data ber*Label* sebagai dasar proses pembelajaran *model* [27]. Selain itu,

metode ini mampu melakukan generalisasi terhadap data baru berdasarkan pola yang telah dipelajari sebelumnya [34]. *Supervised learning* juga mendukung berbagai algoritma, seperti *Decision tree*, *Random Forest*, *SVM*, *Logistic Regression*, dan *XGBoost*, yang memiliki keunggulan masing-masing dalam proses klasifikasi maupun prediksi [35].

Dalam penerapannya, *supervised learning* umumnya melalui beberapa tahapan, mulai dari *preprocessing* data, pemisahan data untuk *training* dan *testing*, pelatihan *model*, hingga evaluasi performa menggunakan metrik tertentu. Tahap *preprocessing* menjadi bagian penting karena kualitas data memiliki pengaruh besar terhadap performa *model* yang dihasilkan. Beberapa teknik *preprocessing* yang umum digunakan meliputi penanganan *missing value*, encoding pada data kategorikal, normalisasi data, serta *feature selection* [30].

Keunggulan utama *Supervised learning* terletak pada kemampuannya membangun *model* prediktif dengan tingkat akurasi yang tinggi apabila menggunakan data historis yang memadai [31]. Selain itu, metode ini cukup mudah untuk dievaluasi karena hasil prediksi dapat dibandingkan dengan *Label* aktual menggunakan metrik seperti *accuracy*, *precision*, *recall*, *F1-score*, dan AUC [32]. Oleh karena itu, *supervised learning* menjadi salah satu pendekatan yang banyak diterapkan dalam penelitian berbasis data maupun pengembangan sistem prediksi.

Pada penelitian ini, *supervised learning* diterapkan karena mampu melakukan klasifikasi pada data *customer churn* berdasarkan pola historis pelanggan [35]. Pada penelitian ini, *supervised learning* diterapkan karena mampu melakukan klasifikasi pada data *customer churn* berdasarkan pola historis pelanggan [30].

2.2.4 Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence (XAI) merupakan pendekatan pada bidang kecerdasan buatan yang digunakan untuk meningkatkan transparansi serta interpretabilitas *model machine learning* sehingga hasil

keputusan sistem dapat dipahami oleh manusia. Seiring berkembangnya algoritma *machine learning* yang semakin kompleks, banyak *model* prediktif bersifat *black-box*, yaitu *model* yang mampu menghasilkan prediksi dengan tingkat akurasi tinggi tetapi sulit dijelaskan proses pengambilan keputusannya. Oleh sebab itu, pendekatan XAI dikembangkan agar proses prediksi *model* dan faktor-faktor yang mempengaruhi hasil prediksi dapat dipahami dengan lebih jelas [36].

Interpretabilitas dalam *machine learning* mengacu pada kemampuan manusia untuk memahami alasan di balik keputusan yang dihasilkan oleh *model*. Dengan menggunakan metode interpretasi, pengguna dapat mengetahui kontribusi setiap fitur terhadap hasil prediksi sehingga proses pengambilan keputusan menjadi lebih transparan serta mudah dipahami. Selain itu, pendekatan *explainability* memiliki peran penting dalam meningkatkan kepercayaan pengguna terhadap sistem berbasis kecerdasan buatan. Penelitian menunjukkan bahwa sistem AI yang mampu memberikan penjelasan secara jelas mengenai hasil prediksi akan lebih mudah dipahami dan diterima oleh pengguna [37].

2.2.5 Shapley Additive Explanations (SHAP)

SHAP merupakan salah satu metode dalam XAI yang digunakan untuk menjelaskan pengaruh setiap fitur terhadap hasil prediksi *model machine learning*. Metode ini dikembangkan berdasarkan konsep SHAPley Value dalam teori permainan, di mana setiap fitur dianggap memberikan kontribusi terhadap hasil prediksi *model* [38], [39], [40].

Menurut [38] dan [39], SHAP bersifat *model*-agnostic sehingga dapat diterapkan pada berbagai *model* supervised *machine learning*, baik untuk klasifikasi maupun regresi. Selain itu, SHAP mampu memberikan interpretasi dalam skala global dan lokal. Interpretasi global digunakan untuk mengidentifikasi fitur yang memiliki pengaruh paling besar terhadap keseluruhan hasil prediksi *model*, sedangkan interpretasi lokal digunakan untuk menjelaskan kontribusi fitur pada data tertentu [39] [40].

Dalam penerapannya, SHAP dapat divisualisasikan melalui beberapa jenis plot, seperti *Summary Plot* dan *Waterfall Plot* [38]. *Summary Plot* digunakan untuk melihat pengaruh fitur terhadap *model* secara global, sedangkan *Waterfall Plot* berfungsi untuk menjelaskan pengaruh masing-masing fitur terhadap hasil prediksi pada setiap data secara individual.

2.3 *Framework* yang digunakan

2.3.1 Teori CRISP DM

Metodologi CRISP-DM (Cross Industry Standard Process for *Data mining*) digunakan sebagai kerangka kerja dalam proses *data mining* serta pengembangan *model machine learning*. Metodologi ini mendukung proses analisis data secara sistematis, mulai dari identifikasi permasalahan hingga tahap implementasi *model* sesuai tujuan penelitian [41]. Selain banyak diterapkan pada penelitian dan proyek analisis data, CRISP-DM juga memiliki alur yang fleksibel sehingga setiap tahap dapat disesuaikan dengan kebutuhan penelitian [42].

Metodologi CRISP-DM terdiri atas enam tahapan utama, yaitu *Business understanding*, *Data understanding*, *Data preparation*, *Modeling*, *Evaluation*, dan *Deployment* [41], [42]. Setiap tahapan pada CRISP-DM saling berhubungan dan dapat dilakukan secara iteratif sesuai dengan kebutuhan penelitian.

2.3.1.1 Business understanding

Tahap ini dilakukan untuk memahami kebutuhan bisnis maupun permasalahan penelitian ingin diselesaikan, sekaligus memutuskan target yang hendak dicapai melalui proses analisis data.

2.3.1.2 Data understanding

Tahap ini berfokus pada proses melakukan pengumpulan, eksplorasi, serta memahami *dataset* yang digunakan. Pada tahap *Data understanding*, dilakukan identifikasi karakteristik data, seperti *missing value*s, *outlier*, dan distribusi data.

2.3.1.3 Preprocessing data

Tahap ini adalah proses persiapan data sebelum dipakai pe*model*an *machine learning*. Proses tersebut mencakup *data cleaning*, transformasi data, encoding, *feature selection*, scaling, serta penanganan data imbalance. Dalam proses *data mining*, tahap *Data preparation* umumnya menjadi salah satu tahapan yang paling banyak memerlukan waktu [43].

2.3.1.4 Modeling

Tahap ini merupakan proses pengembangan *model machine learning* menggunakan algoritma tertentu. Pada *Modeling*, dilakukan pelatihan dan pengujian *model* serta penyesuaian parameter untuk menghasilkan performa *model*.

2.3.1.5 Evaluation

Dalam tahap ini, performa *model* dianalisis yang dipakai metrik *accuracy*, *precision*, *recall*, dan *F1-score* untuk memastikan *model* mampu memenuhi tujuan penelitian.

2.3.1.6 Deployment

Langkah ini adalah fase implementasi *model* untuk digunakan dalam system atau aplikasi tertentu. Dalam beberapa penelitian, *Deployment* dilaksanakan melalui penerapan *model* ke dalam aplikasi berbasis online untuk memfasilitasi penggunaan *model machine learning* [44], [45].

Pada penelitian ini, metodologi CRISP-DM digunakan sebab bisa memberikan alur penelitian yang sistematis, fleksibel, dan mendukung pengembangan *model machine learning* secara iteratif [42].

2.4 Tools/software yang digunakan

2.4.1 Teori Python

Sebagai bahasa pemrograman *open-source*, Python banyak digunakan dalam berbagai bidang, termasuk komputasi ilmiah, pengembangan perangkat lunak, serta analisis data. Sintaks Python yang sederhana dan mudah dipahami membuat bahasa ini diterapkan pada pengembangan perangkat lunak, analisis

data, serta kecerdasan buatan [46], [47]. Selain itu, Dukungan Python terhadap berbagai paradigma pemrograman, seperti prosedural, fungsional, dan berorientasi objek, memungkinkan pengembangan aplikasi dilakukan dengan lebih fleksibel [47], [48].

Dalam bidang data science dan *machine learning*, Python termasuk salah satu bahasa pemrograman yang paling banyak digunakan karena didukung oleh ekosistem library yang luas dan lengkap. Beberapa library yang sering digunakan meliputi *NumPy* untuk kebutuhan komputasi numerik, *Pandas* dalam pengolahan data, *Matplotlib* untuk visualisasi data, *Scikit-learn* pada implementasi *machine learning*, serta *TensorFlow* dan *PyTorch* yang umum digunakan dalam pengembangan *deep learning* [49], [50]. Keberadaan berbagai library tersebut membantu proses pengembangan *model machine learning* menjadi lebih cepat, efisien, dan terstruktur.

Python juga memiliki kompatibilitas yang baik dengan berbagai platform seperti Windows, Linux, dan macOS sehingga memudahkan proses implementasi sistem pada lingkungan yang berbeda [46]. Selain itu, Python mendukung integrasi dengan berbagai *framework* dan layanan cloud sehingga sering digunakan dalam pengembangan sistem berbasis web maupun aplikasi berbasis data [51].

Penggunaan Python dalam penelitian maupun pengembangan sistem dipilih karena mampu mempercepat proses pemrograman melalui sintaks yang sederhana dan dokumentasi yang luas [49]. Python juga memiliki komunitas pengembang yang besar sehingga memudahkan pengguna dalam menemukan referensi, dokumentasi, maupun solusi terhadap permasalahan yang muncul selama proses pengembangan [50].

Dalam penerapan *machine learning*, Python dinilai efisien karena mendukung proses *preprocessing* data, pelatihan *model*, evaluasi *model*, serta visualisasi hasil analisis dalam satu lingkungan pengembangan [50]. Kondisi tersebut membuat Python banyak diterapkan pada penelitian akademik maupun

industri untuk membangun sistem prediksi, klasifikasi, clustering, dan analisis data secara otomatis [49], [50].

Selain digunakan pada *machine learning*, Python juga banyak diterapkan dalam pengembangan computer vision, chatbot, automation system, serta pengolahan big data. Fleksibilitas tersebut membuat Python menjadi salah satu bahasa pemrograman utama dalam pengembangan sistem modern yang berorientasi pada data dan kecerdasan buatan [51].

2.4.2 Teori Jupyter Notebook

Jupyter Notebook ialah aplikasi berbasis web yang dikembangkan dalam ekosistem Project Jupyter untuk mendukung komputasi interaktif, analisis data, visualisasi, dan dokumentasi kode terintegrasi. Jupyter Notebook memungkinkan pengguna untuk mengintegrasikan kode pemrograman, teks naratif, persamaan matematika, visualisasi data, dan hasil eksekusi dalam satu dokumen notebook interaktif [52], [53]. Notebook yang dihasilkan bersifat shareable dan mendukung berbagai bahasa pemrograman melalui sistem kernel, seperti Python, R, dan Julia [54].

Jupyter Notebook mendukung proses eksplorasi data secara langsung sehingga memudahkan peneliti maupun pengembang sistem dalam melakukan pengujian algoritma, *preprocessing* data, visualisasi, hingga evaluasi *model* secara bertahap [55]. Fitur utama yang dimiliki oleh Jupyter Notebook meliputi code cell untuk mengeksekusi kode program secara langsung, markdown cell untuk dokumentasi dan penjelasan teori, dukungan visualisasi data, serta kompabilitas dengan berbagai library analisis data dan *machine learning* [52], [53]. Jupyter Notebook juga mendukung penyimpanan dokumen dalam format .ipynb yang memungkinkan hasil pengembangan dapat dibagikan dan direproduksi oleh pengguna lain [54]. Kemampuan ini menjadikan Jupyter Notebook sebagai salah satu tools yang mendukung reproducibility dalam penelitian dan pengembangan sistem berbasis data [56].

Dalam penelitian dan pengembangan sistem, penggunaan Jupyter Notebook memberikan beberapa keuntungan, seperti kemudahan eksperimen *model*,

proses debugging yang lebih cepat, serta kemampuan dokumentasi yang terintegrasi dengan kode program [55]. Selain itu, lingkungan interaktif pada Jupyter Notebook memungkinkan proses pengembangan dilakukan secara iteratif sehingga mempermudah analisis hasil eksperimen dan optimasi *model machine learning* [57]. Oleh karena itu, Jupyter Notebook banyak digunakan sebagai tools utama dalam proses pengembangan sistem analitik data dan penelitian berbasis *machine learning*.

2.4.3 Teori Streamlit

Sebagai *framework open-source* berbasis *Python*, *Streamlit* banyak dimanfaatkan dalam pengembangan aplikasi web interaktif pada bidang *data science* dan *machine learning*. Dengan menggunakan *Streamlit*, script Python dapat dikonversi menjadi aplikasi web tanpa memerlukan kemampuan front-end seperti HTML, CSS, maupun JavaScript [58]. *Framework* ini banyak dimanfaatkan oleh data scientist dan *machine learning* engineer karena mampu mempercepat proses pengembangan aplikasi berbasis data melalui sintaks yang sederhana dan mudah dipahami [59].

Dalam pengembangan sistem, *Streamlit* berperan sebagai platform untuk implementasi antarmuka pengguna yang memungkinkan interaksi langsung dengan *model machine learning* yang telah dikembangkan. Dengan *Streamlit*, proses input data, visualisasi data, pengujian *model*, dan penyajian hasil prediksi dapat dilaksanakan secara *real-time* dalam format aplikasi online interaktif [60]. *Streamlit* juga dapat diintegrasikan dengan berbagai library Python, seperti *Pandas*, *NumPy*, *Matplotlib*, *Scikit-learn*, dan *XGBoost*, untuk mendukung pengembangan sistem analitik serta prediksi berbasis *machine learning* secara lebih efisien [61].

Streamlit memiliki berbagai fitur utama yang mendukung pengembangan aplikasi data-driven. Beberapa fitur tersebut meliputi kemudahan pembuatan dashboard interaktif, visualisasi data secara dinamis, penyediaan widget input seperti slider, selectbox, dan form, serta kemampuan *Deployment* aplikasi secara cepat melalui *Streamlit* Community Cloud [58]. Selain itu, *Streamlit*

menyediakan API sederhana yang memungkinkan pengembang membuat aplikasi hanya dengan beberapa baris kode sehingga proses prototyping dapat dilakukan secara lebih efisien [62]. Fitur live update yang dimiliki *Streamlit* juga memungkinkan perubahan kode langsung ditampilkan pada aplikasi tanpa proses reload manual, sehingga meningkatkan produktivitas pengembang [63].

Penggunaan *Streamlit* dalam penelitian dan pengembangan sistem dipilih karena *framework* ini mampu mempercepat implementasi *model machine learning* menjadi aplikasi yang mudah digunakan oleh pengguna akhir. *Streamlit* juga mendukung pengembangan aplikasi berbasis penelitian dengan pendekatan rapid prototyping sehingga mempermudah proses evaluasi dan demonstrasi hasil penelitian [64]. Selain itu, sifat open-source dan kompatibilitasnya dengan ekosistem Python menjadikan *Streamlit* sebagai salah satu *framework* yang umum diterapkan pada aplikasi *machine learning* modern [59]. Dengan adanya tampilan antarmuka yang interaktif dan proses *Deployment* yang relatif sederhana, *Streamlit* dapat membantu meningkatkan aksesibilitas sistem prediksi maupun sistem analitik yang dikembangkan dalam penelitian [60].

2.4.4 Teori Visual Studio Code

Microsoft mengembangkan Visual Studio Code sebagai editor kode berbasis open-source yang digunakan untuk mendukung proses pengembangan perangkat lunak. Selain memiliki performa yang ringan dan fleksibel, VS Code juga mendukung berbagai bahasa pemrograman dalam satu lingkungan pengembangan [65]. Selain digunakan sebagai editor kode, VS Code juga menyediakan fitur Integrated Development Environment (IDE), seperti debugging, terminal, dan version control yang terintegrasi untuk membantu proses pengembangan aplikasi [66]. VS Code menyediakan dukungan untuk berbagai bahasa pemrograman seperti Python, Java, JavaScript, PHP, dan C++ melalui extension yang tersedia pada marketplace resminya. Selain itu, perangkat lunak ini dapat digunakan pada sistem operasi Windows, Linux, maupun macOS sehingga memberikan adaptabilitas dalam proses pengembangan perangkat lunak dan penelitian [65].

Beberapa fitur utama yang dimiliki oleh Visual Studio Code antara lain syntax highlighting, IntelliSense, integrated terminal, debugging tools, serta Git integration [66]. Fitur IntelliSense berfungsi untuk memberikan rekomendasi kode secara otomatis berdasarkan sintaks program sehingga dapat meningkatkan efisiensi dan meminimalkan kesalahan penulisan kode. Selain itu, fitur debugging pada VS Code membantu proses identifikasi dan perbaikan error selama tahap pengembangan sistem [67].

Visual Studio Code banyak digunakan dalam pembelajaran dan pengembangan sistem karena antarmuka yang sederhana dan ringan, sehingga mudah diakses baik oleh pemula maupun pengembang professional [68]. Penggunaan Visual Studio Code dinilai dapat meningkatkan produktivitas pengembangan perangkat lunak karena mendukung pengelolaan source code, pengujian program, dan integrasi berbagai tools pengembangan dalam satu lingkungan kerja [69].

Dalam penelitian ini, Visual Studio Code dipakai sebagai tools pembuatan sistem karena memiliki kompatibilitas yang baik dengan bahasa pemrograman dan library yang digunakan selama proses implementasi sistem. Selain itu, dukungan extension dan fitur debugging pada Visual Studio Code membantu kegiatan pengembangan, pemeriksaan, serta pengelolaan sistem agar lebih efisien dan terstruktur [70], [67].