BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Dalam melakukan sebuah penelitian, penting untuk mencari referensi dari penelitian terdahulu untuk dapat membantu mengidentifikasi masalah, kekurangan, peluang, dan solusi yang dapat dikembangkan lebih lanjut. Melalui hasil penelitian terdahulu, peneliti dapat memperoleh wawasan baru yang menjadi dasar dalam solusi pada penelitian yang sedang dilakukan. Beberapa penelitian terdahulu dengan topik yang relevan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

Penelitian 1	
Judul Penelitian	*University Recommendation System for Abroad Studies using Machine Learning* [14]
Metode	Decision Tree (DT), Random Forest (RF), Extreme Gradient Boosting (XGBoost).
Objek Penelitian	Data informasi berbagai Universitas dari *college finder platform* bernama Yocket.
Hasil dan Kesimpulan	Menerapkan Random Forest, Decision Tree, dan XGBoost pada atribut seperti CGPA, GRE, dan TOEFL. XGBoost memiliki akurasi terbaik dalam sistem rekomendasi ini, yaitu sebesar 81%, dibandingkan dengan Random Forest dan Decision Tree yang hanya memiliki akurasi sebesar 76%.
Penelitian 2	
Judul Penelitian	*A Recommendation System for Selecting the Appropriate Undergraduate Program at Higher Education Institutions Using Graduate Student Data* [15]
Metode	Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), dengan *hyperparameter tuning* & menghilangkan *low-importance feature*
Objek Penelitian	Data historis akademik pelajar MBA di CMS Business School
Hasil dan Kesimpulan	Dengan melakukan *hyperparameter tuning,* menghilangkan fitur-fitur yang tidak penting, dan dengan rasio training:testing sebesar 80:20, algoritma RF mampu meraih akurasi tertinggi dengan perolehan akurasi 97.70%, sementara itu dengan metode yang sama pada algoritma SVM meraih 86% dan DT hanya meraih 83.70%.
Penelitian 3	

Judul Penelitian	*E-Commerce Product Recommendations using XGBoost with User Clusters and Clickstream* [16]
Metode	Extreme Gradient Boosting (XGBoost), Gradient Boosting (GB), K-Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF)
Objek Penelitian	Data aktivitas pengguna pada *E-Commerce*
Hasil dan Kesimpulan	Dari hasil perbandingan antar algoritma yang dipakai, XGBoost mampu menunjukkan hasil akurasi tertinggi disbanding algoritma lainnya, yakni sebesar 89.44%. Namun, hasil ini belum optimal sehingga XGBoost ditambahkan dengan metode K-Means dan clickstream, hingga mampu meningkatkan akurasi hingga 95.72%.
Penelitian 4	
Judul Penelitian	*XGBoost To Enhance Learner Performance Prediction* [17]
Metode	Logistic Regression, Extreme Gradient Boosting (XGBoost) dengan metode evaluasi *5-fold cross-validation*, dan *hyperparameter tuning*.
Objek Penelitian	Data interaksi siswa dengan sistem pembelajaran online
Hasil dan Kesimpulan	XGBoost mampu mengungguli Logistic Regression pada pendekatan model pembelajaran PFA dengan perolehan akurasi sebesar 79.9%, dengan Area Under Curve (AUC) yang dicapai sebesar 0.88. XGBoost juga meningkatkan nilai AUC pada model pembelajaran DAS3H. Namun, pada model pembelajaran IRT, XGBoost memberikan hasil prediksi yang hampir sama dengan Logistic Regression.
Penelitian 5	
Judul Penelitian	*Enhanced mastitis severity classification in dairy cows using DNN and RF: A study on PCA and correlation-based feature selection* [18]
Metode	Deep Neural Networks (DNN), Random Forest (RF), dengan menerapkan *Principal Component Analysis* (PCA), *correlation-based feature selection*, *Synthetic Minority Over-sampling Technique* (SMOTE), metode evaluasi *5-fold cross-validation*, dan *hyperparameter tuning*.
Objek Penelitian	Data medis sapi perah Holstein-Friesian yang berusia 3 hingga 4 tahun.
Hasil dan Kesimpulan	Penelitian ini membandingkan kinerja model Deep Neural Network (DNN) dan Random Forest (RF) dalam mengklasifikasikan tingkat keparahan mastitis pada sapi perah menggunakan teknik PCA, *correlation-based feature selection*, dan SMOTE. Hasilnya menunjukkan bahwa model RF secara konsisten mengungguli DNN, dengan akurasi rata-rata 97.46% dibandingkan dengan 87.09% untuk DNN.

Sejumlah penelitian terdahulu menunjukkan perkembangan signifikan dalam penerapan algoritma *machine learning* untuk sistem prediksi dan rekomendasi, khususnya dalam bidang pendidikan dan klasifikasi data. Berbagai studi tersebut memanfaatkan algoritma seperti Random Forest, Decision Tree, dan XGBoost, serta beberapa algoritma lain seperti Support Vector Machine (SVM) dan Deep Neural Network (DNN), dengan tujuan membandingkan performa dan akurasi dalam menyelesaikan permasalahan klasifikasi atau rekomendasi.

Beberapa penelitian, seperti yang dilakukan dalam prediksi penerimaan mahasiswa di perguruan tinggi dan rekomendasi program studi, memperlihatkan bahwa XGBoost dan Random Forest secara konsisten menunjukkan performa yang unggul dibandingkan dengan algoritma lain [14], [15]. Misalnya, dalam sistem prediksi perguruan tinggi berdasarkan nilai CGPA, GRE, dan TOEFL, XGBoost mampu mencapai akurasi sebesar 81%, mengungguli Random Forest dan Decision Tree yang masing-masing hanya mencapai 76% [14]. Sementara itu, penelitian lain yang membandingkan Random Forest, SVM, dan Decision Tree untuk rekomendasi program sarjana menunjukkan bahwa Random Forest berhasil meraih akurasi tertinggi hingga 97,70%, jauh di atas SVM (86%) dan Decision Tree (83,70%) [15]. Keunggulan Random Forest ini didukung oleh proses *hyperparameter tuning* dan seleksi fitur, yang mampu meningkatkan performa model secara signifikan.

Di sisi lain, penelitian yang dilakukan dengan objek penelitian non-pendidikan, seperti sistem rekomendasi *e-commerce*, kembali menegaskan dominasi XGBoost dengan akurasi mencapai 89,44%, dibandingkan dengan Gradient Boosting, K-Nearest Neighbor, dan Random Forest [16]. Keunggulan ini didasari pada kemampuan XGBoost dalam mengelola data yang kompleks dan melakukan optimasi *loss function* yang lebih efisien melalui teknik *boosting* yang terstruktur dan regularisasi yang baik. Penelitian lain bahkan menunjukkan bahwa XGBoost mampu mengungguli model *baseline* seperti Logistic Regression ketika dilakukan pemilihan *hyperparameter* dan penerapan *cross-validation* secara sistematis dengan akurasi 79.9%, dan *Area Under Curve* (AUC) sebesar 0.88 [17]. Ini menegaskan validitas pemilihan algoritma XGBoost berdasarkan hasil yang stabil dan unggul dalam berbagai studi sebelumnya, memperkuat posisi XGBoost sebagai algoritma yang sangat andal untuk sistem prediksi dan rekomendasi.

Namun, pada penelitian lain yang dilakukan di dunia medis khususnya untuk melakukan klasifikasi tingkat keparahan mastitis pada sapi, Random Forest justru lebih unggul dibandingkan Deep Neural Network (DNN) [18]. Hasil dari penelitian ini menunjukkan bahwa model Random Forest mampu mengungguli DNN secara konsisten, dengan akurasi rata-rata sebesar 97.46% dibandingkan dengan akurasi 87.09% pada model DNN [18]. Hal ini menunjukkan bahwa meskipun XGBoost sering kali memberikan performa terbaik pada banyak kasus, Random Forest memiliki stabilitas yang sangat baik, terutama pada data dengan kelas yang tidak seimbang dan perlu diolah menggunakan teknik seperti SMOTE dan *correlation-based feature selection*. Random Forest menunjukkan akurasi

tinggi yang konsisten, bahkan pada data yang tidak seimbang dan memiliki fitur yang kompleks.

Dari perbandingan ini dapat disimpulkan bahwa baik XGBoost maupun Random Forest memiliki keunggulan masing-masing tergantung pada konteks penggunaan, karakteristik data, dan teknik pemrosesan data yang diterapkan. XGBoost umumnya unggul pada nilai akurasi yang bisa mencapai nilai yang tinggi, terutama ketika digunakan pada data yang bersifat numerik dan memiliki banyak fitur yang saling berkaitan. Sementara itu, Random Forest menunjukkan keunggulan dari sisi stabilitas dan performa pada data yang kompleks atau tidak seimbang. Di sisi lain, Decision Tree umumnya menunjukkan performa yang lebih rendah dibanding kedua algoritma tersebut, meskipun tetap digunakan sebagai *baseline* atau model yang lebih mudah diinterpretasikan.

Berdasarkan berbagai penelitian terdahulu, penelitian ini bertujuan untuk membandingkan performa algoritma Decision Tree, Random Forest, dan XGBoost dalam memprediksi peminatan mahasiswa program studi Sistem Informasi di Universitas Multimedia Nusantara (UMN) berdasarkan performa akademik mereka. Pemilihan ketiga algoritma ini didasarkan pada bukti-bukti empiris dari penelitian terdahulu yang menunjukkan efektivitas dan keunggulan masing-masing algoritma dalam melakukan prediksi.

Adapun celah penelitian atau *research gap* yang menjadi dasar penelitian ini adalah belum ditemukannya penelitian yang secara khusus mengembangkan model prediksi untuk merekomendasikan peminatan mahasiswa berdasarkan performa akademik, khususnya pada Program Studi Sistem Informasi di UMN. Sebagian besar penelitian sebelumnya masih berfokus pada prediksi kelulusan, penerimaan mahasiswa baru, maupun rekomendasi program studi secara umum, tanpa menjadikan peminatan sebagai target prediksi yang spesifik. Selain itu, meskipun teknik seperti *hyperparameter tuning*, *feature selection*, analisis korelasi, dan SMOTE telah terbukti mampu meningkatkan performa model prediksi dalam berbagai studi terdahulu, belum ada penelitian yang secara komprehensif mengintegrasikan keempat teknik tersebut dalam satu eksperimen, terutama dalam kasus prediksi peminatan mahasiswa dengan algoritma berbasis *decision tree*.

Dengan demikian, penelitian ini akan menggabungkan beberapa teknik yang telah dilakukan pada penelitian terdahulu. Teknik-teknik seperti *hyperparameter tuning*, penghapusan fitur yang tidak penting, SMOTE, dan analisis korelasi antar fitur akan diimplementasikan pada penelitian ini, karena terbukti mampu meningkatkan akurasi dan performa model secara signifikan dalam penelitian sebelumnya. Hasil dari penerapan teknik-teknik ini akan digabungkan dan dibandingkan hingga mendapat hasil yang terbaik. Dengan begitu, kebaruan-kebaruan dalam penelitian ini terletak pada:

1) Objek penelitian, yaitu peminatan mahasiswa Program Studi Sistem Informasi Universitas Multimedia Nusantara berdasarkan performa akademik, yang belum pernah menjadi fokus dalam penelitian sebelumnya.
2) Kombinasi teknik-teknik pendukung seperti *hyperparameter tuning*, seleksi fitur, analisis korelasi, dan penerapan SMOTE, yang terbukti mampu meningkatkan performa model *machine learning* berbasis decision tree, yakni Decision Tree, Random Forest, dan XGBoost dalam studi terdahulu namun belum diterapkan secara bersamaan dalam satu penelitian.

Penggunaan dan penggabungan teknik-teknik ini yang menjadi kebaruan dalam penelitian ini, disertai dengan objek penelitian yang berbeda dibandingkan dengan penelitian-penelitian terdahulu. Dengan pendekatan ini, diharapkan model prediksi yang dikembangkan dapat memberikan hasil yang akurat dan membantu mahasiswa Sistem Informasi UMN dalam mengambil keputusan yang lebih tepat ketika hendak memilih peminatan yang ada di program studi Sistem Informasi UMN.

2.2 Tinjauan Teori

2.2.1 Sistem Rekomendasi

Sistem rekomendasi atau *recommender system* adalah jenis sistem yang digunakan dalam aplikasi untuk membantu memprediksi preferensi pengguna terhadap *item-item* tertentu dan memberikan rekomendasi berdasarkan riwayat preferensi serta batasan pengguna [19]. Adanya sistem rekomendasi dapat membantu mengatasi masalah informasi yang berlebihan dengan memberikan daftar rekomendasi yang disesuaikan secara personal bagi para pengguna. Sebagian besar organisasi dan perusahaan telah mengadopsi sistem rekomendasi ketika menjual produk secara *online*. Akan tetapi, sayangnya hampir semua situs web tidak memprioritaskan kepentingan pembeli, dan seringkali mereka lebih suka mendorong penjualan tambahan dengan merekomendasikan produk yang tidak relevan dan tidak diperlukan [20]. Hal ini menunjukkan pentingnya sistem rekomendasi yang dapat mementingkan kepentingan pembeli supaya dapat mendorong penjualan tambahan dengan merekomendasikan produk yang sesuai preferensi pengguna.

Melihat besarnya dampak yang diberikan sistem rekomendasi bagi suatu perusahaan atau organisasi, kini berbagai industri digital ikut menggunakan sistem rekomendasi, salah satunya adalah institusi akademik. Banyak institusi akademik mulai menerapkan sistem rekomendasi untuk beberapa hal, seperti sistem rekomendasi perguruan tinggi dan rekomendasi program studi [14]. Semua ini dilakukan untuk memudahkan mahasiswa dalam menemukan pilihan akademis

berdasarkan minat dan preferensi mereka. Semenjak adanya sistem rekomendasi ini, mahasiswa kini cenderung membuat keputusan lebih cepat dan cerdas dalam menemukan pilihan mereka [14]. Dalam mengembangkan model prediksi untuk memberi rekomendasi, terdapat beberapa pendekatan atau teknik yang dapat digunakan sistem rekomendasi, di antaranya seperti penerapan teknik klasifikasi *machine learning*.

2.2.2 Machine Learning

Machine learning adalah bagian dari kecerdasan buatan di mana sebuah teknologi diberikan pelatihan algoritma untuk belajar dari data dan membuat prediksi atau keputusan tanpa harus diprogram secara eksplisit [21]. Ada berbagai kategori machine learning, termasuk *supervised*, *unsupervised*, *semi-supervised*, dan *reinforcement learning*. *Supervised learning* digunakan ketika data sudah berlabel, sedangkan *unsupervised learning* digunakan ketika data tidak berlabel. *Semi-supervised learning* adalah kombinasi dari *supervised* dan *unsupervised learning*, sedangkan *reinforcement learning* digunakan ketika algoritma belajar melalui uji coba dan kesalahan [21]. Penggunaan *machine learning* sangatlah populer dan telah diterapkan pada banyak hal, salah satunya adalah sistem rekomendasi. Sistem rekomendasi merupakan aplikasi dari *machine learning* yang berurusan dengan peringkat atau penilaian produk atau pengguna [19]. *Machine learning* digunakan untuk memprediksi apa yang mungkin disukai oleh seorang pengguna berdasarkan data historis.

Sistem Rekomendasi pada umumnya menggunakan algoritma klasifikasi *machine learning* yang bertujuan untuk membantu pengguna dalam melakukan *decision-making* dengan memberikan sebuah saran/rekomendasi atas suatu pilihan yang lebih akurat melalui prediksi dari masing-masing algoritma. Algoritma klasifikasi dalam machine learning merupakan salah satu teknik yang banyak digunakan dalam pengembangan sistem rekomendasi, termasuk untuk pemilihan mata kuliah perminatan. Algoritma ini dapat digunakan untuk mengelompokkan data sesuai dengan pola tertentu yang ditemukan dalam dataset sebelumnya. *Decision Tree, Logistic Regression, Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, dan XGBoost* adalah beberapa algoritma klasifikasi yang cocok digunakan untuk membuat sistem rekomendasi [14][16][17].

Penggunaan algoritma klasifikasi dalam sistem rekomendasi telah banyak diterapkan pada berbagai bidang, salah satunya dalam pemilihan program studi mahasiswa. Salah satu penelitian terdahulu melakukan perbandingan antara algoritma Decision Tree (DT), Random Forest (RF), dan Support Vector Machine (SVM) [15]. Penelitian ini

memanfaatkan data historis akademik pelajar untuk membangun sistem yang mampu memberikan rekomendasi program studi terbaik untuk para pelajar. Hasil eksperimen penelitian ini menunjukkan bahwa Random Forest memberikan performa terbaik dengan akurasi mencapai 97%, mengungguli Decision Tree yang hanya memperoleh akurasi 83%. Random Forest dinilai lebih efektif karena kemampuannya mengurangi *overfitting* yang umum terjadi pada Decision Tree melalui mekanisme *ensemble* yang menggabungkan prediksi dari banyak pohon keputusan, sehingga menghasilkan model yang lebih stabil dan akurat. Penelitian lain juga menunjukkan penggunaan algoritma klasifikasi *machine learning* pada sistem rekomendasi produk *e-commerce* menggunakan algoritma Random Forest, Gradient Boosting, K-Nearest Neighbor, dan Decision Tree, di mana XGBoost mampu meraih akurasi tertinggi sebesar 89,44% dibanding algoritma lainnya [16]. Sementara itu, penelitian lain yang membandingkan XGBoost dengan Logistic Regression menggunakan 5-fold cross-validation menunjukkan bahwa XGBoost secara konsisten unggul [17]. Hal ini memperkuat posisi XGBoost sebagai algoritma yang andal untuk prediksi, berkat kemampuannya menangani data kompleks dan menghasilkan akurasi tinggi.

Temuan-temuan dari berbagai penelitian terdahulu memperkuat argumen bahwa algoritma klasifikasi berbasis *decision tree* seperti Random Forest dan XGBoost merupakan dua algoritma klasifikasi yang unggul dan layak dipertimbangkan dalam pengembangan sistem rekomendasi. Dalam pengembangan sistem rekomendasi mata kuliah perminatan untuk mahasiswa Sistem Informasi, pemilihan algoritma machine learning yang tepat sangat krusial untuk memastikan akurasi dan relevansi rekomendasi. Maka, dalam penelitian ini yang bertujuan memprediksi peminatan mahasiswa berdasarkan performa akademik, adopsi algoritma-algoritma tersebut menjadi pilihan yang logis karena didukung oleh bukti empiris dari berbagai penelitian terdahulu. Maka pada penelitian ini akan menggunakan tiga algoritma yang sering digunakan dalam sistem rekomendasi seperti Decision Tree, Random Forest, dan XGBoost. Masing-masing algoritma memiliki karakteristik, kelebihan, dan kekurangan yang perlu dipertimbangkan.

2.2.2.1 Decision Tree

Decision Tree adalah algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi dan regresi [22]. Algoritma ini membagi dataset menjadi subset yang lebih kecil berdasarkan atribut tertentu, membentuk struktur pohon dengan node internal mewakili atribut, cabang sebagai hasil uji, dan daun sebagai keputusan atau klasifikasi akhir. Salah satu metode yang digunakan dalam Decision

Tree adalah algoritma C4.5, yang menggunakan konsep *Entropy* dan *Information Gain* untuk menentukan pemilihan atribut pada setiap node. *Entropy* mengukur tingkat ketidakpastian atau impuritas dalam sekumpulan data dan dirumuskan sebagai berikut:

$$\text{Entropy (S)} = \sum_{i=1}^{c} p_i \log 2^p i$$

Rumus 2.1 Formula Perhitungan Entropy

Information Gain digunakan untuk memilih atribut yang memberikan pengurangan terbesar dalam ketidakpastian, dengan rumus:

$$\text{Gain (S,A)} = \sum_{v \in v\,(A)} \frac{|S_v|}{|S|}\ Entropy(S_v)$$

Rumus 2.2 Formula Perhitungan Gain

Keunggulan Decision Tree meliputi kemudahan interpretasi, kemampuan menangani data numerik dan kategorikal, serta efisiensi dalam menangani dataset besar. Namun, algoritma ini rentan terhadap *overfitting*, terutama jika pohon tumbuh terlalu kompleks, dan dapat mengalami bias jika terdapat ketidakseimbangan kelas dalam data. Selain itu, *Decision Tree* dapat menjadi tidak stabil, di mana perubahan kecil pada data dapat menghasilkan pohon yang sangat berbeda [22].

2.2.2.2 Random Forest

Random Forest adalah algoritma *machine learning* berbasis *ensemble* yang menggabungkan beberapa pohon keputusan (*Decision Trees*) untuk meningkatkan akurasi prediksi [23]. Algoritma ini bekerja dengan membangun banyak pohon keputusan dari subset data yang dipilih secara acak, kemudian menggabungkan hasil prediksi masing-masing pohon menggunakan metode *voting* (untuk klasifikasi) atau rata-rata (untuk regresi). Random Forest menggunakan *Gini Index* sebagai metode utama untuk mengukur impuritas dalam pemisahan data pada setiap node pohon. Rumus *Gini Index* dinyatakan sebagai berikut:

$$\sum\sum j \neq i \left(\frac{f(CiT)}{|T|}\right)\left(\frac{f(CiT)}{|T|}\right)$$

Rumus 2.3 Formula Perhitungan Gini

Keunggulan Random Forest mencakup ketahanan terhadap *overfitting*, kemampuan menangani data yang hilang, serta

fleksibilitas dalam mengolah data numerik dan kategorikal. Namun, kekurangannya adalah waktu komputasi yang lebih lama dibandingkan Decision Tree, serta interpretasi model yang lebih kompleks karena kombinasi banyak pohon keputusan [23].

2.2.2.3 XGBoost

XGBoost (*Extreme Gradient Boosting*) merupakan algoritma *boosting* yang mengoptimalkan performa prediksi dengan membangun model secara bertahap, di mana setiap model baru berusaha memperbaiki kesalahan dari model sebelumnya [23]. XGBoost bekerja dengan menggunakan fungsi objektif yang terdiri dari dua komponen utama: fungsi *loss* yang mengukur kesalahan model dan regularisasi yang mengontrol kompleksitas model. Berikut merupakan rumusnya:

$$y_i^{(t)} = \sum_{k=1}^{t} \int k\ (\chi i) = y_i^{(t-1)} + \int t\ (\chi i)$$

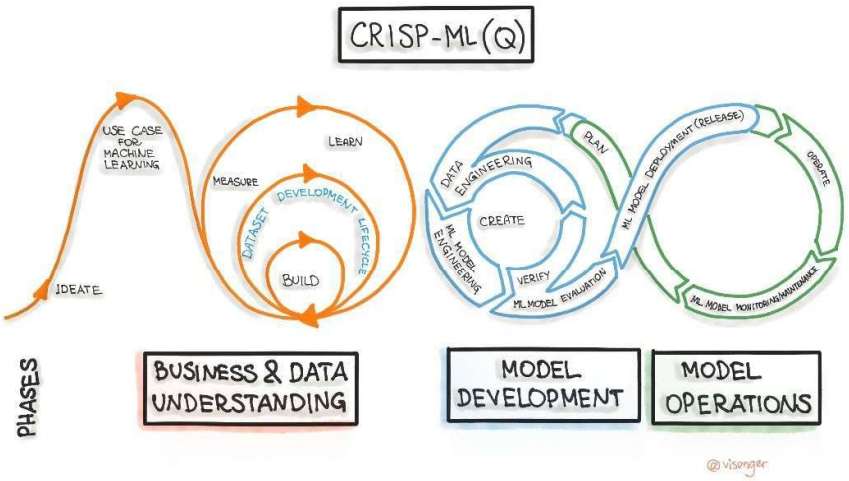
Rumus 2.4 Formula Perhitungan Algoritma XGBoost

Keunggulan utama *XGBoost* adalah efisiensi komputasi, kemampuannya menangani data besar, serta kontrol *overfitting* melalui regularisasi. Namun, kekurangannya adalah kompleksitas tuning parameter yang lebih tinggi dibandingkan algoritma lain serta interpretasi model yang lebih sulit dibandingkan Random Forest atau Decision Tree [23].

2.3 Framework yang digunakan

2.3.1 *Cross-Industry Standard Process for Machine Learning* (CRISP-ML)

CRISP-ML (*Cross-Industry Standard Process for Machine Learning*) merupakan sebuah kerangka kerja yang dirancang untuk mengarahkan praktisi machine learning dalam mengembangkan aplikasi machine learning secara sistematis dan terjamin kualitasnya [24]. *Framework* ini muncul sebagai respons terhadap masalah umum dalam proyek *machine learning* yang biasanya kurang terorganisir, serta sulit direproduksi hasilnya. Dengan diterapkannya CRISP-ML, proses pengembangan machine learning dapat berjalan lebih terstruktur, terdokumentasi dengan baik, serta memenuhi standar kualitas yang telah ditentukan.



Gambar 2.1 Siklus Tahapan Framework CRISP-ML [24]

Gambar 2.1. menunjukkan siklus tahapan pada CRISP-ML. Berikut ini penjelasan mengenai tahapan yang ada dalam *framework* CRISP-ML:

1) *Business and Data Understanding*

Tahap pertama dalam CRISP-ML adalah dengan memahami permasalahan yang terjadi, pemahaman mendalam terhadap masalah bisnis, sehingga dapat menentukan tujuan proyek, menentukan ruang lingkup proyek, menentukan kriteria keberhasilan, serta melakukan verifikasi kualitas data. Tujuan utama dari fase ini adalah memastikan kelayakan proyek sebelum memasuki tahap pengembangan lebih lanjut

2) *Data Engineering (Data Preparation)*

Tahap kedua dalam proses CRISP-ML berfokus pada persiapan data yang akan digunakan pada tahap pengembangan model. Hal utama dalam yang dilakukan pada tahap ini bisa beragam untuk mempersiapkan data, seperti melakukan pemilihan data, pembersihan data, *feature engineering*, atau standarisasi data. Pemilihan fitur dilakukan untuk memastikan hanya fitur yang penting dan diperlukan yang digunakan dalam pelatihan model. Selain itu, pemilihan data juga bisa melalui penghapusan data yang tidak memenuhi standar kualitas data. Jika ditemukan ketidakseimbangan kelas pada data, metode seperti *over-sampling* atau *under-sampling* dapat diterapkan untuk mengatasi masalah tersebut. Untuk proses pembersihan data dapat dilakukan dengan cara mendeteksi dan mengoreksi kesalahan yang terdapat dalam data. Salah satu langkah penting untuk meminimalisir risiko kesalahan yang terbawa ke tahap selanjutnya adalah dengan menambahkan unit testing pada data. Bergantung pada jenis tugas *machine learning*, tahap ini juga dapat terdiri dari beberapa cara seperti melakukan *feature*

engineering dan *data augmentation*, seperti penerapan *one-hot encoding*. Tidak hanya itu, hal yang dapat dilakukan selanjutnya adalah standarisasi data yang bertujuan untuk menyatukan format input yang digunakan dalam *machine learning* sehingga dapat mengurangi risiko kesalahan akibat ketidaksesuaian data. Normalisasi data juga terkadang diperlukan untuk mencegah adanya bias pada fitur-fitur yang memiliki skala nilai besar.

3) *Machine Learning Model Engineering*

Tahap ini menjadi inti dari proses pengembangan machine learning karena pada tahap inilah satu atau beberapa model ML disiapkan untuk nantinya akan diterapkan. Tahap ini merupakan tahap pengembangan model *machine learning* yang sesuai dengan masalah bisnis dan data yang ada. Parameter spesifik untuk model juga ditentukan pada langkah ini. Untuk merancang sebuah model, perlu membagi data menjadi dua bagian, yakni data pelatihan dan data pengujian untuk evaluasi model.

4) *Evaluating Machine Learning Models*

Setelah tahap pelatihan model selesai, langkah berikutnya adalah melakukan evaluasi model untuk memeriksa hasil model dan memastikan bahwa tujuan bisnis tercapai. Pada tahap ini, performa model yang telah dilatih dievaluasi menggunakan *test set* untuk menilai sejauh mana model mampu melakukan prediksi dengan baik dan mengukur keakuratan model yang dibuat.

5) *Deployment*

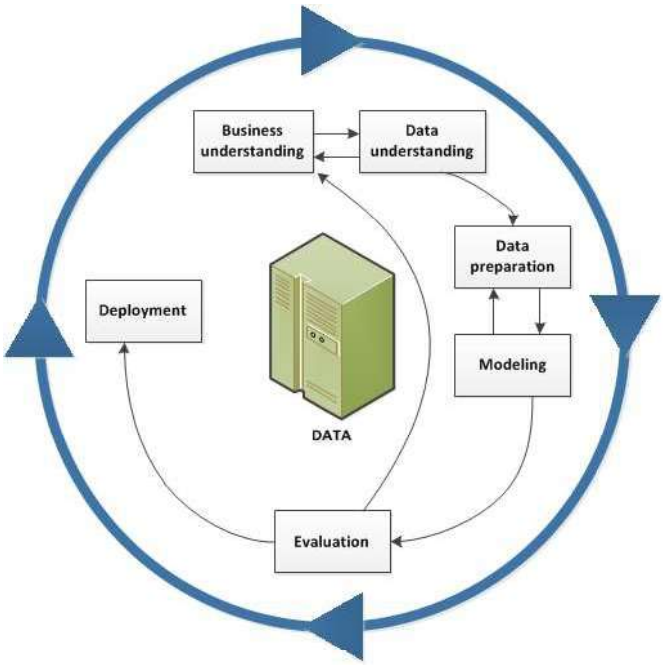
Setelah model berhasil melewati tahap evaluasi, model tersebut siap untuk dipindahkan ke lingkungan *production*. Pada tahap ini, yang akan dilakukan adalah mengimplementasikan model tersebut. Strategi deployment biasanya telah dirancang sejak fase awal pengembangan ML, dan metode implementasinya dapat bervariasi tergantung pada kasus penggunaan. Salah satu metode implementasi atau penerapan model dapat berupa integrasi model ML ke dalam suatu aplikasi, sehingga memungkinkan untuk digunakan oleh pengguna.

6) *Monitoring and Maintenance*

Setelah model *machine learning* diterapkan ke lingkungan *production*, pemantauan kinerja dan pemeliharaan model menjadi langkah yang sangat penting. Maka pada tahap ini akan dilakukan *maintenance* atau perawatan terhadap model yang telah diproduksi, dengan harapan model dapat terus digunakan dengan baik.

2.3.2 *Cross-Industry Standard Process for Data Mining* (CRISP-DM)

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) adalah sebuah *framework* atau kerangka kerja proses yang banyak digunakan dalam proyek data mining, yang terdiri dari enam fase yang diatur secara berulang untuk pengembangan model *data mining* [25]. *Framework* ini dianggap sebagai metodologi data mining yang paling lengkap dalam memenuhi kebutuhan proyek-proyek pengembangan data dan telah menjadi proses yang paling banyak digunakan untuk proyek-proyek *data mining*.



Gambar 2.2 Siklus Tahapan *Framework* CRISP-DM [25]

Gambar 2.2 menunjukkan siklus tahapan pada CRISP-DM. Pada framework ini terdapat 6 tahapan, yakni *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*. Berikut penjelasan rinci mengenai 6 tahap dalam CRISP-DM:

1) *Business Understanding*

Tahap pertama dalam CRISP-DM ialah pemahaman mendalam terhadap masalah bisnis, menentukan tujuan proyek, dan mengidentifikasi sasaran penambangan data. Selain itu juga dapat dilakukan evaluasi situasi bisnis untuk memahami sumber daya yang tersedia dan yang dibutuhkan. Pada langkah ini juga akan menetapkan jenis teknik *data mining* yang akan digunakan, seperti klasifikasi, prediksi, dan lainnya. Pada proses ini juga akan dilakukan pembuatan rencana proyek yang merupakan langkah wajib.

2) *Data Understanding*

Tahap kedua adalah pemahaman terhadap data yang akan digunakan dalam penelitian. Tahap ini fokus pada pengumpulan data dari berbagai sumber, eksplorasi data, dan persiapan data untuk pemodelan. Identifikasi *missing value*, *outlier*, dan masalah kualitas data lainnya adalah bagian penting dari proses ini. Pada tahap ini dilakukan identifikasi dan pemahaman data seperti struktur data, jenis data, dan potensi masalah dalam data tersebut melalui analisis statistik dan penentuan atribut serta hubungan antar atribut.

3) *Data Preparation*

Setelah pemahaman data, tahap selanjutnya adalah persiapan data. Hal yang akan dilakukan pada tahap ini adalah mengatasi masalah kualitas data dengan pemilihan data melalui pembersihan data, integrasi data dari berbagai sumber jika diperlukan, transformasi data, dan pemilihan subset data yang relevan. Pada tahap ini perlu dipastikan bahwa data telah siap digunakan untuk proses selanjutnya

4) *Modeling*

Tahap ini merupakan tahap perancangan model pemodelan yang sesuai dengan masalah bisnis dan data yang ada. Parameter spesifik untuk model juga ditentukan pada langkah ini. Evaluasi model terhadap kriteria evaluasi digunakan untuk memilih model terbaik. Untuk merancang sebuah model, perlu membagi data menjadi dua bagian, yakni data pelatihan dan data pengujian untuk evaluasi model.

5) *Evaluation*

Setelah model dibuat, tahap selanjutnya adalah tahap evaluasi untuk memeriksa hasil model dan memastikan bahwa tujuan bisnis tercapai. Hasil dari model diinterpretasikan untuk menentukan tindakan selanjutnya. Pada proses ini juga dilakukan evaluasi dan peninjauan umum terhadap seluruh proses untuk mengukur keakuratan model yang dibuat.

6) *Deployment*

Setelah model dievaluasi, tahap selanjutnya adalah dengan mengimplementasikan model tersebut. Tahap implementasi atau penerapan model dapat berupa penyusunan laporan akhir atau pengembangan aplikasi, yang tersusun secara terstuktur mulai dari perencanaan implementasi model, pemantauan kinerja model, dan langkah pemeliharaan model yang diperlukan.

2.4 Teori tentang tools/software yang digunakan

2.4.1 Visual Studio Code

Visual Studio Code adalah *open-source code editor* yang dikembangkan oleh Microsoft yang menawarkan berbagai fitur, seperti IntelliSense untuk penyelesaian kode otomatis dan *debugging* [26]. Dapat diinstal di Windows, macOS, dan Linux, serta mendukung berbagai bahasa pemrograman seperti Python, C#, C++, PHP, dan lainnya. Visual Studio Code tidak hanya berfungsi sebagai alat penulisan kode program, tetapi juga dilengkapi dengan integrasi Git, fungsi kolaborasi, dan fitur *debugging*. Sebagai editor teks open-source, pengguna dapat menggunakan ekstensi tambahan untuk meningkatkan fungsionalitasnya.

2.4.2 Python

Python merupakan bahasa pemrograman yang populer untuk aplikasi web, pengolahan data, dan pengembangan perangkat lunak, serta *Machine Learning* (ML) [27]. Bahasa ini terkenal karena kemudahan pembelajarannya, efisiensinya, dan kemampuannya untuk berjalan di berbagai *platform*. Python juga digunakan secara luas di berbagai industri seperti *Face Recognition, Artificial Intelligence, Machine Learning*, dan bidang lainnya. Ini adalah bahasa pemrograman "*Interpreter*" yang berarti kode dapat langsung dijalankan sesuai dengan perintah yang ditulis tanpa perlu dikompilasi terlebih dahulu. Python memiliki berbagai modul dan *library* yang mendukung berbagai kebutuhan, seperti TensorFlow, NumPy, SciPy, Pandas, Matplotlib, Keras, SciKit-Learn, PyTorch, dan Scrapy [28].

Diciptakan pada tahun 1990 oleh Guido van Rossum di Belanda, Python telah digunakan secara luas di industri dan akademis [29]. Python menawarkan struktur data tingkat tinggi seperti *array, dynamic binding, class, exceptions*, dan lainnya. Bahasa ini mudah dimengerti dan digunakan oleh komputer, serta tersedia secara gratis. Python merupakan bahasa pemrograman *open source* yang didukung oleh berbagai *platform* dan sistem operasi, dan menawarkan berbagai *library* dan modul untuk berbagai keperluan seperti pemrosesan teks, perhitungan matematika, pengembangan perangkat lunak, dan *machine learning* [30].

2.4.3 Streamlit

Streamlit adalah *framework* aplikasi sumber terbuka yang memungkinkan seseorang membuat aplikasi web interaktif untuk *data science* dan *machine learning* [31]. *Framework* ini menyediakan cara yang sederhana dan intuitif untuk membuat aplikasi berbasis data dengan kode minimal. Streamlit dapat diinstal menggunakan pip, dan dapat digunakan bersama dengan Visual Studio Code untuk pengembangan dan *debugging* [31]. Dengan Streamlit, seorang *data scientist* atau

machine learning engineer dapat melakukan *deploy* hasil kerja mereka ke dalam bentuk aplikasi web yang dapat digunakan secara *real-time* oleh pengguna.