BAB II
LANDASAN TEORI

2.1 Penelitian Terdahulu

Tabel 2.1 mendeksripsikan penelitian-penelitian terdahulu. Berdasarkan tabel penelitian terdahulu, penelitian ini menggunakan algoritma klasifikasi yaitu Naïve Bayes, Support Vector Machine, dan Neural Network untuk melakukan prediksi masa studi mahasiswa di Universitas Multimedia Nusantara. Hasil dan informasi dari penelitian terdahulu digunakan sebagai pembanding dan dipertimbangkan dalam pengerjaan penelitian ini.

Tabel 2.1 Penelitian Terdahulu

Penulis	Judul Artikel	Nama Jurnal	Metode	Hasil
Anwar, M (2021)	*Prediction of graduation rate of engineering education students using Artificial Neural Network Algorithms*	*International Journal of Research in Counseling and Education*, 5 (1): pp. 15-23 [10]	Feature Selection, Artificial Neural Network, Particle Swarm Optimization (PSO)	Akurasi model klasifikasi Waktu kelulusan mahasiswa menggunakan algoritma Artificial Neural Network sebesar 82.61% dan mendapat peningkatan sebesar 8.69% dengan menggunakan optimasi PSO
N. M. Suhaimi, S. Abdul-Rahman, S. Mutalib, N. H. A. Hamid, and A. M. A. Malik (2019)	*Review on Predicting Students' Graduation Time Using Machine Learning Algorithms*	*International Journal of Modern Education and Computer Science* 11(7):1-13 [11]	Naïve Bayes, Decision Tree, Neural Network, Support Vector Machine	Algoritma Neural Network dan Support Vector Machine memiliki kemungkinan dan potensi yang lebih tinggi dalam melakukan prediksi waktu kelulusan mahasiswa
L. Setiyani, M. Wahidin, D. Awaludin, and S.	Analisis Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode *Data Mining* Naïve Bayes :	Faktor Exacta 13 (1): 35-43, 2020 [1]	Data Mining, Naïve Bayes	Akurasi yang dihasilkan dengan algortima Naïve Bayes mencapai diatas 90% meskipun menggunakan

Penulis	Judul Artikel	Nama Jurnal	Metode	Hasil
Purwani (2020)	*Systematic Review*			variabel yang berbeda-beda.
E. Haryatmi and S. Pramita Hervianti (2021)	Penerapan Algoritma Support Vector Machine untuk Model Prediksi Kelulusan Mahasiswa Tepat Waktu	Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi) Vol.5 No. 2 (2021) 386–392 [12]	CRISP-DM, Support Vector Machine	Hasil prediksi dengan algoritma Support Vector Machine menghasilkan tingkat akurasi sebesar 94.4%
V. Riyanto, A. Hamid, and R. Ridwansyah (2019)	*Prediction of Student Graduation Time Using the Best Algorithm*	*Indonesian Journal of Artificial Intelligence and Data Mining* (IJAIDM) 1 Vol 2, No.1, March 2019, pp. 1 – 9 [13]	Neural Network, Support Vector Machine, Decision Tree, 10 *fold cross validation*	Algoritma terbaik merupakan Support Vector Machine dengan tingkat akurasi, presisi, dan sensitivitas berturut-turut sebesar 85.15%, 79.58%, dan 40.22%.
A. N. Noercholis (2020)	*Comparative Analysis of 5 Algorithm Based Particle Swarm Optimization (PSO) for Prediction of Graduate Time Graduation*	MATICS : Jurnal Ilmu Komputer dan Teknologi Informasi Volume 12, No. 1(2020), pp 1-9 [16]	Naïve Bayes, k-NN, Neural Network, Decision Tree, Support Vector Machine, PSO	Algoritma PSO melakukan optimasi algoritma Naïve Bayes, k-NN, Neural Network, Decision Tree, dan Support Vector Machine dengan peningkatan akurasi berturut-turut sebesar 2.13%, 4.26%, 5.11%, 5.21%, dan 1.79%.
W. E. Pangesti, I. Ariyati, P. Priyono, S. Sugiono, and R. Suryadithia (2024)	*Utilizing Genetic Algorithms To Enhance Student Graduation Prediction With Neural Networks*	*Sinkron*, vol. 9, no. 1, 2024 [18]	Neural Network, Genetic Algorithms	Terdapat peningkatan hasil akurasi algoritma Neural Network dari sebesar 84.55% menjadi 87.33% dengan setelah menggunakan Genetic Algorithm sebagai optimasi.
A. F. Subarkah, R. Kusumawati, and M.	*Comparison of Different Classification Techniques to Predict Student Graduation*	MATICS : J. Ilmu Komput. dan Teknol. Inf. (*Journal Comput. Sci. Inf. Technol.*,	Naïve Bayes, Support Vector Machine, Random Forest,	Algoritma Support Vector Machine mendapatkan hasil akurasi tertinggi yaitu sebesar 87%,

Penulis	Judul Artikel	Nama Jurnal	Metode	Hasil
Imamudin (2023)		vol. 15, no. 2, 2023 [19]		diikuti dengan algoritma Random Forest sebesar 82% dan Naïve Bayes sebesar 76%.
A. Dineley, F. Natalia, and S. Sudirman (2024)	*Data Augmentation for Occlusion-Robust Traffic Sign Recognition Using Deep Learning*	ICIC Express Lett. Part B Appl., vol. 15, no. 4, 2024 [20]	Akurasi sebagai metrik evaluasi, *deep learning*, *transfer learning*	Metrik evaluasi akurasi digunakan sebagai parameter untuk menentukan hasil dari eksperimen.
N. B. Setiawan, F. Natalia, F. V. Ferdinand, S. Sudirman, and C. S. Ko (2021)	*Classification of skin diseases and disorders using convolutional neural network on a mobile application*	*ICIC Express Lett. Part B Appl.*, vol. 12, no. 8, 2021 [21]	Convolutional Neural Network, *k-fold cross-validation*	Hasil akurasi data *validation* meningkat dari 81.7% menjadi 90% dengan mengimplementasikan metode *k-fold cross-validation.*

Penelitian terdahulu dilakukan dengan tujuan melakukan prediksi tingkat kelulusan terhadap mahasiswa teknik dengan menggunakan algoritma Artificial Neural Network yang dioptimasi dengan algoritma Particle Swarm Optimization. Hasil akurasi algoritma Artificial Neural Network adalah sebesar 82.61% dan mendapat peningkatan sebesar 8.69% menjadi 91.30% setelah menggunakan optimasi Particle Swarm Optimization [10]. Penelitian terdahulu selanjutnya dilakukan pada tahun 2020 dengan tujuan melakukan perbandingan antara 5 algoritma klasifikasi yaitu Naïve Bayes, k-NN, Support Vector Machine, Neural Network, dan Decision Tree yang dioptimasi menggunakan algoritma Particle Swarm Optimization. Algoritma Naïve Bayes, k-NN, Neural Network, Decision Tree, dan Support Vector Machine mendapatkan peningkatan akurasi berturut-turut sebesar 2.13%, 4.26%, 5.11%, 5.21%, dan 1.79% [16]. Kedua penelitian terdahulu tersebut berkontribusi terkait penggunaan optimasi PSO dan algoritma Neural Network memiliki performa yang baik.

Penelitian terdahulu dilakukan pada tahun 2019 dengan tujuan melakukan pembahasan terkait beberapa penelitian yang telah dilakukan sebelumnya dan membandingkan algoritma terbaik dalam memprediksi waktu kelulusan mahasiswa. Hasil penelitian ini mengonfirmasi bahwa algoritma Neural Network

dan Support Vector Machine adalah pengklasifikasi paling kompetitif dibandingkan dengan Naïve Bayes dan Decision Tree [11]. Penelitian terdahulu dilakukan pada tahun 2024 dengan tujuan melakukan prediksi waktu kelulusan mahasiswa menggunakan algoritma Neural Network yang dioptimasi menggunakan Genetic Algorithm. Hasil menunjukkan bahwa terdapat peningkatan akurasi algoritma Neural Network dari sebesar 84.55% menjadi 87.33% dengan setelah menggunakan Genetic Algorithms sebagai optimasi [18]. Kedua penelitian terdahulu tersebut berkontribusi terkait penggunaan algoritma Neural Network dan Support Vector Machine memiliki performa yang baik dibandingkan algortima lainnya.

Penelitian terdahulu dilakukan untuk melakukan prediksi kelulusan mahasiswa tepat waktu. Algoritma yang digunakan pada penelitian adalah Support Vector Machine dengan *framework* CRISP-DM. Hasil prediksi dengan algoritma Support Vector Machine mendapatkan nilai akurasi sebesar 94.4% [12]. Penelitian terdahulu dilakukan untuk melakukan prediksi waktu kelulusan mahasiswa dengan 2 kelas, yaitu lulus tepat waktu dan lulus tidak tepat waktu. Algoritma yang digunakan pada penelitian ini adalah Neural Network, Support Vector Machine, dan Decision Tree. Penelitian juga menggunakan teknik *k-fold cross validation* dengan 10 lipatan. Hasil menunjukkan bahwa algoritma terbaik merupakan Support Vector Machine dengan tingkat akurasi, presisi, dan sensitivitas berturut-turut sebesar 85.15%, 79.58%, dan 40.22% [13]. Penelitian terdahulu dilakukan untuk melakukan komparasi terhadap 3 algoritma yaitu Naïve Bayes, Support Vector Machine, dan Random Forest dalam melakukan klasifikasi waktu kelulusan mahasiswa. Algoritma Support Vector Machine mendapatkan hasil akurasi tertinggi yaitu sebesar 87%, diikuti dengan algoritma Random Forest sebesar 82% dan Naïve Bayes sebesar 76% [19]. Ketiga penelitian terdahulu tersebut berkontribusi terkait penggunaan algoritma Support Vector Machine memiliki performa yang baik dibandingkan algortima lainnya, penggunaan framework CRISP-DM, dan teknik *k-fold cross validation* dengan 10 lipatan pada tahap persiapan data.

Penelitian terdahulu dengan judul "Analisis Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Data Mining Naïve Bayes : Systematic Review" dilakukan dengan objektif melakukan identifikasi beberapa penelitian terdahulu yang menggunakan algoritma Naïve Bayes memprediksi waktu kelulusan mahasiswa. Hasil dari penelitian menunjukkan bahwa algoritma Naïve Bayes dapat melakukan klasifikasi dua kelas dengan baik. Hal ini dibuktikan dengan tingkat akurasi sebesar lebih dari 90% dari penelitian terdahulu yang diidentifikasi [1].

Penelitian terdahulu dilakukan dengan tujuan melakukan klasifikasi gambar terhadap multikelas tipe penyakit kulit menggunakan *deep convolutional neural network*. Penelitian terdahulu menggunakan teknik *k-fold cross validation* dalam melakukan pembagian kelompok data *training* dan *testing*. Hasil akurasi *data validation* meningkat dari 81.7% menjadi 90% dengan mengimplementasikan metode *k-fold cross-validation* [21]. Penelitian terdahulu dengan judul "Data Augmentation for Occlusion-Robust Traffic Sign Recognition Using Deep Learning" menggunakan metrik evaluasi akurasi dalam meilihat apakah adanya peningkatan dari eksperimen yang dilakukan [20]. Kedua penelitian terdahulu tersebut berkontribusi dalam tahap persiapan data dalam menggunakan teknik *k-fold cross validation* dan penggunaan akurasi sebagai metrik acuan dalam optimasi.

Mengacu pada penelitian terdahulu, penelitian ini menggunakan algoritma Naïve Bayes, Neural Network, dan Support Vector Machine untuk melakukan prediksi waktu kelulusan mahasiswa [1][11][19][16][18][12]. Penelitian ini memiliki pembeda hasil *output* berupa *multi-class classification* yaitu semester 7 hingga 14 dalam memprediksi waktu kelulusan secara lebih detail sesuai kebutuhan implementasi pada universitas, sedangkan penelitian-penelitian terdahulu menggunakan klasifikasi 2 kelas dengan luaran lulus tepat waktu dan lulus tidak tepat waktu. Penelitian ini menggunakan teknik *data mining* yang telah digunakan pada penelitian sebelumnya yaitu CRISP-DM [12], namun dengan objek yang berbeda yaitu data mahasiswa Program Studi Sistem Informasi Universitas Multimedia Nusantara. Penelitian ini juga melakukan pembaruan terhadap augmentasi data dengan mengusung teknik *expanding window* yang digunakan agar

model klasifikasi relevan untuk digunakan dalam melakukan prediksi dengan data baru (mahasiswa dari semester 1 hingga 14). Berdasarkan penelitian terdahulu, metrik evaluasi yang akan digunakan adalah akurasi, presisi, dan sensitivitas dengan tambahan spesifisitas, *f1-score* dan *confusion matrix* untuk melihat keseluruhan performa model [13]. Teknik validasi 10 fold yang telah digunakan pada penelitian terdahulu juga digunakan dengan sedikit modifikasi yaitu dengan penggunaan fungsi *stratified k-fold cross validation* [13][18]. Ketiga model klasifikasi akan dioptimasi menggunakan algoritma PSO seperti yang dilakukan pada penelitian terdahulu agar setiap algoritma dibandingkan berdasarkan performa optimal algoritma tersebut [10][16]. Optimasi yang dilakukan merupakan pengubahan parameter pada algoritma. Model prediksi akan diterapkan dan disimulasikan dalam *framework website* sebagai tahap implementasi dari hasil *output* penelitian ini.

2.2 Tinjauan Teori

2.2.1 Perguruan tinggi

Perguruan tinggi merupakan suatu pihak yang menyelenggarakan pendidikan akademik secara formal bagi para pengenyam pendidikan. Perguruan tinggi di Indonesia terbagi menjadi 5 berdasarkan sistem pendidikan yaitu universitas, politeknik, akademik, sekolah tinggi dan institut [1]. Kualitas dari sebuah perguruan tinggi di Indonesia diukur dengan suatu parameter yaitu akreditasi yang dikelola oleh Badan Akreditasi Nasional Perguruan Tinggi. Kualitas dari perguruan tinggi tersebut dinilai berdasarkan tujuh pilar utama, dimana dua diantaranya adalah Lulusan atau alumni dan Mahasiswa dengan salah satu komponen yang dinilai yaitu rata-rata masa studi dan IPK dari mahasiswa [23].

2.2.2 Kelulusan

Salah satu hal yang tak terpisahkan dari lingkungan perguruan tinggi adalah kelulusan para mahasiswanya. Kelulusan merupakan berbagai tahapan dari rangkaian proses pendidikan yang harus dilalui atau dihadapi oleh setiap mahasiswa. Beberapa hal yang harus dilalui oleh mahasiswa untuk

mendapatkan kelulusan yaitu harus menyelesaikan jumlah minimum mata kuliah yang telah ditetapkan, melakukan seminar proposal penelitian, seminar tugas akhir, dan beberapa hal lainnya yang telah ditentukan oleh perguruan tinggi [24].

2.2.3 *Data Mining*

Data mining adalah proses penanganan informasi dari berbagai data untuk menemukan pola atau hubungan untuk membuat prediksi yang valid. Oleh karena itu, *data mining* dapat digunakan dan bermanfaat dalam berbagai tujuan, sektor, dan bidang. Salah satunya adalah memajukan sains di bidang kecerdasan buatan dan statistik, karena data mining diprediksi akan menjadi cabang sains yang sangat dibutuhkan dan revolusioner selama dekade berikutnya, menurut MIT Technology Review. *Data mining* dapat membantu banyak bidang, termasuk manufaktur, perbankan, asuransi, pemasaran, kedirgantaraan, pendidikan, dan kesehatan [25]. Secara lebih khusus, teknik *data mining* yang akan digunakan untuk topik penelitian ini yang seputar pendidikan adalah *educational data mining*.

2.2.4 *Educational Data Mining*

Educational data mining adalah bidang yang difokuskan pada pembuatan alat untuk memeriksa jenis data khusus yang dihasilkan dalam pengaturan pendidikan. Ketika digunakan secara langsung, EDM atau *educational data mining* berbeda dari teknik *data mining* karena memperhitungkan hierarki bertingkat (dan memberikan peluang untuk menggunakannya) tetapi tidak memiliki akses ke data pendidikan independen. Secara umum, ada dua jenis pekerjaan EDM, yakni *web mining* dan visualisasi & statistik. EDM diklasifikasikan sebagai berikut [26]:

1) *Prediction*

   Dengan menggunakan data historis untuk variabel yang sama, prediksi mencoba meramalkan variabel yang tidak diketahui. Variabel input (variabel prediktor), bagaimanapun, dapat dikategorikan atau disimpan

sebagai variabel. Jenis variabel input menentukan seberapa baik model prediksi bekerja. Terdapat 3 tipe prediksi yaitu:

a) *Classification*

Dalam klasifikasi, informasi sebelumnya digunakan untuk membuat model pembelajaran, yang kemudian diterapkan pada data baru sebagai variabel biner atau kategoris. Banyak model, termasuk mesin vektor pendukung dan regresi logistik, telah dibuat dan digunakan sebagai pengklasifikasi.

b) *Regression*

Regression merupakan salah satu model untuk memprediksi variabel. Model regresi meramalkan variabel kontinu, tidak seperti klasifikasi. Di bidang EDM, banyak teknik regresi, termasuk regresi linier dan *neural network*, telah dimanfaatkan secara luas untuk meramalkan siswa mana yang harus diberi label sebagai berisiko.

c) *Density estimation*

Fungsi Gaussian adalah salah satu fungsi kernel yang digunakan untuk estimasi densitas.

2) *Clustering*

Analisis *clustering* banyak diterapkan di berbagai bidang, termasuk riset pasar, identifikasi pola, analisis data, dan pengolahan citra. *Clustering* dapat membantu pemasar dalam menentukan minat konsumen mereka berdasarkan kebiasaan pembelian mereka dan mengkarakterisasi kelompok klien.

3) *Relationship mining*

Menemukan korelasi antara berbagai faktor dalam kumpulan data dengan banyak variabel adalah tujuan dari *relationship mining*. Ini melibatkan penentuan variabel mana yang memiliki korelasi paling kuat dengan variabel tertentu yang diminati. Sejauh mana variabel

yang berbeda terkait satu sama lain juga diukur dengan penambangan hubungan. Relasi yang ditemukan oleh relationship mining harus memenuhi dua persyaratan. Relevansi dan minat dalam statistik

4) Penemuan dengan model

Dalam penemuan, model biasanya dibangun menggunakan penalaran manusia daripada teknik otomatis, seperti pengelompokan, prediksi, atau rekayasa pengetahuan. Model yang dibuat kemudian dimasukkan ke dalam model ekstensif tambahan, seperti penambangan hubungan.

5) Distilasi data untuk penilaian manusia

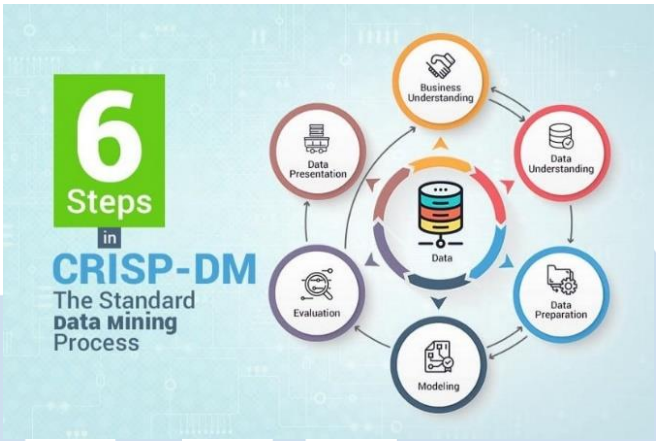
Membuat data dapat dipahami adalah tujuan penyulingan data untuk penilaian manusia. Otak manusia dapat mempelajari informasi baru dengan disajikan materi dalam berbagai cara. Penyulingan data digunakan di sektor pendidikan untuk dua tujuan, yaitu klasifikasi dan/atau identifikasi.

2.3 *Framework*, Algoritma, Persiapan Data dan Metode Evaluasi

2.3.1 *Framework*

2.3.1.1 *Framework* CRISP-DM

CRISP-DM (*Cross Industry Standard Process – Data Mining)* merupakan bentuk konsep *framework* yang kini menjadi salah satu *framework* yang paling umum dan banyak digunakan untuk melakukan proses *data mining* yang mengimplementasikan proses yang telah terstandarisasi dan memiliki sifat yang terbuka dimana langkah-langkah yang digunakan memiliki sifat yang terstruktur dan terdefinisi dengan efisien. *Framework* CRISP-DM telah menjadi salah satu metode yang paling umum digunakan untuk proses pengolahan data hingga tahap implementasi suatu proyek *data mining*. Metode CRISP-DM mendefinisikan pedekatan *general* yang umumnya digunakan oleh para ahli pengolah data atau *data mining* [27].



Gambar 2.1 *Framework* CRISP-DM

Sumber: [28]

Gambar 2.1 menunjukkan langkah-langkah dari proses pengolahan data menggunakan *framework CRISP*-DM. *Framework* CRISP-DM terbagi menjadi 6 langkah, antara lain [29]:

1) Pemahaman bisnis

Dalam fase ini, kita harus mengenal dan memahami proses bisnis dari data – data dan pihak yang akan di data mining, agar kita dapat mengetahui permasalahan dari bisnisnya dan menemukan solusinya. Dalam fase ini kita menentukan tujuan dari kegiatan *data mining* dan tujuan bisnis, menganalisa situasi, dan membuat rencana proyek *data mining* ini. Tujuan yang telah ditetapkan tersebut yang akan menjadi acuan seberapa berhasil dari proyek yang dilakukan.

2) Pemahaman data

Dalam fase ini, kita mengumpulkan data. Kita juga harus mengetahui dan mengenal data tersebut, seperti karakteristik, jenis, latar belakang dan sebagainya. Objektif dari fase ini adalah kita dapat mengetahui bahwa apakah data tersebut bisa digunakan untuk mengatasi permasalahan di fase 1. Umumnya dalam proses ini dilakukan metode statistika deskriptif.

3) Pengolahan/persiapan data

Dalam fase ini, kita dapat melakukan *pre-processing* data agar data tersebut dipastikan bahwa data tersebut baik dan layak digunakan supaya menghasilkan hasil yang optimal. Dalam fase ini umumnya, pengolah data dapat melakukan *re-format* dan *cleansing data* seperti penanganan nilai *null*, augmentasi data, dan lainnya. Dalam fase ini juga umumnya terjadi pembagian data *training* dan *testing*, seperti pembagian 80 dan 20 atau penggunaan k-fold *cross validation*.

4) Pemodelan

Dalam fase ini, kita memilih teknik *modeling* dan algoritma, terdapat beberapa model yang dapat digunakan. Dalam tahap ini yang akan membuat model dengan memanfaatkan data yang sudah dipersiapkan di tahap sebelumnya untuk memperoleh informasi tersembunyi/belum ditemukan. Dalam proses pembentukan model di fase ini juga tidak menutup kemungkinan akan adanya pencobaan beberapa algoritma.

5) Evaluasi

Fase evaluasi ini diperlukan karena walaupun kita telah melakukan pemodelan pada data dan memperoleh hasil, namun ada kemungkinan jika hasil yang diperoleh tidak memberikan solusi dan insight untuk mengatasi permasalahan di fase 1. Berdasarkan hal tersebut, diperlukannya tahap evaluasi ini untuk mengevaluasi hasil, mengulas proses, dan menentukan apakah akan melanjutkan ke step berikutnya atau mengulang proses pemodelan. Fase evaluasi ini menggunakan teknik pembagian data yang telah dilakukan pada fase persiapan data.

6) Implementasi/penyebaran

Jika hasil yang telah dievaluasi menunjukan outcome yang baik maka fase akan dilanjutkan ke fase penyebaran / implementasi ke berbagai *platform* dalam operasi bisnis atau lainnya. Dalam tahap ini kita dapat melakukan pembuatan perencanaan fase implementasi, pembuatan perencanaan pengontrolan dan pemeliharaan dari

implementasi, dan membuat laporan akhir yang mencakup semua yang telah dikerjakan sebelumnya

2.3.1.2 *Framework* KDD

KDD atau *Knowledge Discovery in Database* menawarkan model proses konseptual untuk teori dan teknik dalam ilmu komputasi yang memfasilitasi penemuan pengetahuan (ekstraksi informasi) dari data. *Data mining* adalah tahap khusus yang merupakan bagian dari proses KDD untuk penemuan pengetahuan. KDD memiliki manfaat dalam mempertimbangkan akses dan penyimpanan data, penskalaan algoritma, interpretasi dan visualisasi hasil, serta interaksi komputer manusia berkat sembilan fase utamanya [30].

2.3.1.3 *Framework* SEMMA

Framework SEMMA yang merupakan singkatan dari *Sample, Emplore, Modify, Model,* dan *Assess* merupakan salah satu *framework* yang juga digunakan sebagai pembanding pada penelitian ini. Perbedaan utama SEMMA dengan kedua *framework* KDD dan CRISP-DM adalah pra-syarat pemahaman dan kebutuhan dari bisnis dan *database* secara keseluruhan [31]. *Framework* ini juga dapat dilihat sebagai tahap-tahap *data mining* yang lebih praktis dan sederhana dibandingkan KDD [32].

2.3.2 Algoritma

2.3.2.1 Neural-Network (NN)

Sebagai salah satu algoritma yang dipertimbangkan sebagai salah satu teknik paling *advanced* di lingkungan *data mining*, Neural Network merupakan salah satu algoritma *machine learning* yang dapat digunakan untuk tugas klasifikasi dimana cara kerja algoritma ini terinspirasi dari sistem saraf manusia, NN terdiri dari jaringan neuron terstruktur dengan layer dan bobot [31]. Bobot ini dipelajari melalui proses pelatihan untuk memetakan input data ke output prediksi. NN memiliki kemampuan belajar pola kompleks, menangani data *nonlinear*, dan mencapai akurasi prediksi

tinggi. Dari sisi kekurangannya, NN membutuhkan data pelatihan yang besar, waktu pelatihan yang lama, dan komputasi yang besar [33]. Cara kerja algoritma Neural Network ditunjukkan pada Rumus 2.1 dan Tabel 2.2.

$$y = f(w \cdot x + b) \tag{2.1}$$

Rumus 2.1 Rumus Algoritma Neural Network [33]

Berikut penjelasan dari Rumus 2.1:

1) y adalah *output* dari neural network.
2) f adalah fungsi aktivasi yang diterapkan pada hasil dari $w \cdot x + b$
3) w adalah vektor bobot yang menghubungkan *neuron-neuron* di layer sebelumnya dengan *neuron* di layer berikutnya.
4) x adalah vektor input.
5) b adalah bias.

Pseudocode untuk algoritma Neural Network dapat dituliskan seperti Tabel 2.2.

Tabel 2.2 *Pseudocode* Algoritma Neural Network (NN)

(1) Mendefinisikan jenis jaringan, jumlah *layer*, jumlah neuron pada setiap layer, fungsi aktivasi, dan fungsi *loss*
(2) Melakukan inisialisasi bobot atau *w*
(3) Mulai iterasi latihan
(4) Mengambil *batch* data dari set training.
(5) Menghitung *output* jaringan untuk setiap contoh data dalam *batch*.
(6) Menghitung *error* (loss) antara *output* jaringan dan label data.
(7) Menghitung gradien *error* terhadap bobot jaringan.
(8) Memperbarui bobot jaringan menggunakan algoritma optimasi
(9) Evaluasi performa jaringan pada set validasi secara berkala.
(10) Hentikan iterasi jika mencapai batas iterasi maksimum.

2.3.2.2 Naïve Bayes (NB)

Algoritma Naïve Bayes bekerja berdasarkan teorema Bayes, NB mengasumsikan kemandirian fitur dan menghitung probabilitas kelas berdasarkan fitur data [34]. NB memiliki kelebihan yaitu mudah diimplementasikan, cepat dalam pelatihan, dan membutuhkan komputasi yang kecil. Namun, NB memiliki kekurangan yaitu menghasilkan akurasi prediksi yang cenderung lebih rendah dari algoritma lain, tidak dapat

menangani data nonlinear, dan sensitif terhadap fitur yang tidak independen [35]. Cara kerja algoritma Naïve Bayes ditunjukkan pada Rumus 2.2 dan Tabel 2.3.

$$P(A \mid B) = \frac{(P(B \mid A) * P(A))}{P(B)} \quad (2.2)$$

Rumus 2.2 Rumus Algoritma Naïve Bayes [35]

Berikut penjelasan dari Rumus 2.2:

1) P(A | B): Probabilitas kelas A terjadi dengan data B
2) P(B | A): Probabilitas data B terjadi dengan kelas A
3) P(A): Peluang kelas A
4) P(B): Peluang data B

Pseudocode untuk algoritma Naïve Bayes dapat dituliskan seperti Tabel 2.3.

Tabel 2.3 *Pseudocode* Algoritma Naïve Bayes (NB)

(1) Hitung probabilitas P(B | A) untuk setiap kelas A dan fitur B.
(2) Hitung probabilitas P(A) untuk setiap kelas A
(3) Untuk setiap data baru x yang ingin diklasifikasikan, hitung P(A | B) untuk setiap kelas A
(4) Pilih kelas A dengan probabilitas tertinggi P(A | B) sebagai prediksi untuk data baru B

2.3.2.3 Support Vector Machine (SVM)

Algoritma SVM adalah algoritma *supervised learning* yang dapat digunakan untuk kebutuhan klasifikasi maupun regresi [34]. Algoritma Support Vector Machine bekerja dengan mencari *hyperplane* pemisah optimal yang memaksimalkan margin antara data dari kelas yang berbeda. SVM memiliki kemampuan menangani data *nonlinear*, mencapai akurasi prediksi tinggi, dan *robust* terhadap *overfitting*. Di sisi lain, algoritma Support Vector Machine membutuhkan waktu pelatihan yang cenderung lama, komputasi yang cukup besar, dan memiliki sifat yang sensitif terhadap pemilihan *hyperparameter* [36]. Cara kerja algoritma Support Vector Machine ditunjukkan pada Rumus 2.3 dan Tabel 2.4.

$$f(x) = sign(w \cdot x + b) \quad (2.3)$$

Rumus 2.3 Rumus Algoritma Support Vector Machine [36]

Berikut penjelasan dari Rumus 2.3:

1) f(x) adalah fungsi keputusan yang memprediksi kelas dari data input x.
2) w adalah vektor bobot.
3) x adalah vektor fitur input.
4) b adalah bias.
5) *sign* adalah fungsi tanda positif atau negatif.

Pseudocode untuk algoritma Support Vector Machine dapat dituliskan seperti Tabel 2.4.

Tabel 2.4 *Pseudocode* Algoritma Support Vector Machine (SVM)

(1) Pilih fungsi kernel yang sesuai
(2) Menentukan parameter C dan gamma
(3) Latih model menggunakan algoritma optimasi
(4) Menentukan *hyperplane* atau batas keputusan yang memaksimalkan margin antara kelas luaran

2.3.2.4 Particle Swarm Optimization (PSO)

Algoritma PSO merupakan algoritma yang terdiri dari suatu kelompok partikel yang setiap partikelnya mewakili suatu solusi potensial dimana algoritma PSO dapat digunakan untuk menunjang dalam mempertahankan sifat konvergensi yang lebih stabil dan meminimalkan perhitungan [37]. Pada tahun 1995 Kennedy and Eberhart menginisialisasi penemuan dan perkenalan algoritma PSO. Ide serta formula dari algoritma PSO diinisialisasi dari pengamatan atau observati pada perilaku sosial kawanan burung dan kawanan ikan [38]. Terdapat 3 hal yang membuat algoritma PSO menjadikan algortima optimasi yang menarik yaitu pengimplementasian yang mudah dan sederhana, algoritma PSO hanya memiliki 3 parameter pengontrol, dan algoritma PSO memiliki sifat yang fleksibel untuk dilakukan penggabungan dengan algoritma optimasi lainnya

[39]. Cara kerja algoritma Particle Swarm Optimization ditunjukkan pada Rumus 2.4 dan Tabel 2.5.

$$V_{ij}^{t+1} = w \cdot V_{ij}^{t} + C_1 \cdot R_1 \cdot (pbest_{ij}^{t} - X_{ij}^{t}) + C_2 \cdot R_2 \cdot (gbest_{j}^{t} - X_{ij}^{t}) \quad (2.4)$$

$$X_{ij}^{t+1} = X_{ij}^{t} + V_{ij}^{t+1} \quad (2.5)$$

Rumus 2.4 Rumus Algoritma Particle Swarm Optimization [36]

Berikut penjelasan dari persamaan (2.4) dan (2.5):

1) v_{ij}^{t+1} adalah kecepatan partikel ke-i dalam dimensi ke-j pada iterasi t+1.
2) x_{ij}^{t} adalah posisi partikel ke-i dalam dimensi ke-j pada iterasi t.
3) $pbest_{ij}^{t}$ adalah posisi terbaik partikel ke-i dalam dimensi ke-j hingga iterasi t.
4) $gbest_{j}^{t}$ adalah posisi terbaik secara global dalam dimensi ke-j hingga iterasi t.
5) w adalah faktor inersia.
6) c_1 dan c_2 adalah koefisien akselerasi kognitif dan sosial, masing-masing.
7) r_1 dan r_2 adalah nilai acak dari distribusi uniform pada interval [0,1].

Pseudocode untuk algoritma Particle Swarm Optimization dapat dituliskan seperti Tabel 2.5.

Tabel 2.5 *Pseudocode* Algoritma Particle Swarm Optimization (PSO)

(1) Inisialisasi populasi awal partikel dengan posisi dan kecepatan acak
(2) Mulai iterasi
(3) Hitung dan evaluasi nilai *fitness* untuk setiap partikel dalam populasi, nilai *fitness* yang tinggi menunjukkan solusi yang lebih baik
(4) Melakukan pembaruan posisi pbset dari hasil nilai *fitness*
(5) Menentukan posisi gbest sebagai solusi terbaik
(6) Melakukan pembaruan kecepatan dan posisi menggunakan Rumus 2.4.
(7) Akhiri iterasi ketika syarat iterasi sudah terpenuhi

2.3.3 Persiapan Data

2.3.3.1 *Data Imputation*

Null atau nilai yang tidak ada merupakan salah satu kondisi yang umum dialami pada proses pengolahan data. Penanganan nilai null pada suatu proses pengolahan data dapat dilakukan dengan beberapa cara dan penelitian ini akan menggunakan cara imputasi data. Teknik imputasi data merupakan suatu metode atau prosedur untuk menggantikan nilai yang hilang atau *missing* values dengan suatu nilai yang ditentukan. Penentuan nilai sebagai alternatif nilai yang hilang secara umum diklasifikasikan menjadi 2 cara yaitu *single imputation* dan *multiple imputation*. Penanganan *missing values* pada penelitian ini akan menggunakan cara *single imputation* dengan metode imputasi data dengan nilai konstan, hal ini dilakukan untuk menjaga sifat dari nilai yang hilang pada kumpulan data [40].

2.3.3.2 K-*fold Cross Validation*

Teknik *k-fold cross validation* digunakan denga tujuan untuk membagi kelompok data menjadi kelompok *training* dan *testing*. Penggunaan teknik *k-fold cross validation* adalah suatu metode pendukung evaluasi yang popular untuk algoritma klasifikasi [41]. Metode *k-fold cross validation* membagi satu kelompok data menjadi berberapa lipatan sesuai dengan k yang ditentukan dan menjadikan salah satu lipatan menjadi kelompok *testing*, sedangkan sisa lipatannya menjadi kelompok *training*. Proses pelatihan model akan dilakukan berulang kali sesuai dengan total lipatan yang tersedia, iterasi akan terus diulang hingga semua lipatan telah menjadi kelompok *testing* [42]. Penggunaan teknik *k-fold cross validation* dilakukan sebagai salah satu alternatif solusi terhadap masalah *overfitting* [21].

2.3.4 Evaluasi

Evaluasi digunakan sebagai tahapan untuk melihat dan mengambarkan performa model. Metrik akurasi sebagai salah satu metrik yang umum dimanfaatkan dalam mengukur performa model [20], juga digunakan pada

penelitian ini. Selain akurasi, untuk konteks klasifikasi sebagai prediksi terdapat beberapa metriks evaluasi lainnya yang dapat digunakan untuk melihat atau membandingkan hasil kinerja model, yaitu:

1) Akurasi

Akurasi merupakan suatu metrik yang digunakan untuk mendapatkan suatu nilai dari pembagian jumlah total data yang diuji benar dengan jumlah data secara keseluruhan [43].

$$Akurasi = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (2.6)$$

Rumus 2.5 Rumus Metrik Akurasi [43]

Berikut penjelasan dari Rumus 2.5:

1) TP *(True Positive)*: Hasil klasifikasi model dengan benar memprediksi kelas positif
2) TN (*True Negative*): Hasil klasifikasi model dengan benar memprediksi kelas negatif
3) FP (*False Positive*): Hasil klasifikasi model salah memprediksi kelas negatif sebagai positif
4) FN (*False Negative*): Hasil klasifikasi model salah memprediksi kelas positif sebagai negatif

2) Presisi

Presisi merupakan suatu metrik atau ukuran yang dipakai untuk melihat dan mendapatkan perbandingan jumlah total nilai yang diprediksi positif dibandingkan jumlah nilai yang diprediksi positif secara keseluruhan [43].

$$Presisi = \frac{TP}{(TP + FP)} \quad (2.7)$$

Rumus 2.6 Rumus Metrik Presisi [43]

Berikut penjelasan dari Rumus 2.6:

1) TP *(True Positive)*: Hasil klasifikasi model dengan benar memprediksi kelas positif

2) FP (*False Positive*): Hasil klasifikasi model salah memprediksi kelas negatif sebagai positif

3) Spesifisitas

Spesifisitas merupakan suatu metrik atau ukuran yang dipakai untuk melihat dan mendapatkan perbandingan jumlah total nilai yang diprediksi negatif dibandingkan jumlah nilai yang seharusnya negatif secara keseluruhan [43].

$$Spesifisitas = \frac{(TN)}{(TN + FP)} \quad (2.8)$$

Rumus 2.7 Rumus Metrik Spesifisitas [43]

Berikut penjelasan dari Rumus 2.7:

1) TN (*True Negative*): Hasil klasifikasi model dengan benar memprediksi kelas negatif
2) FP (*False Positive*): Hasil klasifikasi model salah memprediksi kelas negatif sebagai positif

4) Sensitivitas

Sensitivitas atau *recall* merupakan suatu metrik atau ukuran yang dipakai untuk melihat dan mendapatkan perbandingan jumlah total nilai yang diprediksi positif dibandingkan jumlah nilai yang seharusnya positif secara keseluruhan [43].

$$Sensitivitas = \frac{(TP)}{(TP + FN)} \quad (2.9)$$

Rumus 2.8 Rumus Metrik Sensitivitas [43]

Berikut penjelasan dari Rumus 2.8:

1) TP *(True Positive)*: Hasil klasifikasi model dengan benar memprediksi kelas positif
2) FN (*False Negative*): Hasil klasifikasi model salah memprediksi kelas positif sebagai negatif

5) *F1-Score*

F1-Score adalah *harmonic mean* dari metrik presisi dan *recall*, memberikan gambaran keseluruhan kinerja model. *F1-Score* dinyatakan dalam bentuk skor tunggal yang berkisar antara 0 hingga 1 dimana nilai *F1-Score* yang cenderung tinggi menggambarkan bahwa model memiliki keseimbangan yang baik antara presisi dan *recall* [44].

$$F1 - Score = 2\ x\ \frac{(Precision\ x\ Recall)}{(Precision + Recall)} \quad (2.10)$$

Rumus 2.9 Rumus Metrik *f1-score* [44]

6) *Confusion Matrix*

Confusion matrix telah digunakan dan hadir dalam penggunaannya sebagai metode evaluasi model ilmiah dan aplikasi teknik yang umum untuk digunakan di berbagai bidang [45]. *Confusion matrix* merupakan matriks 2 dimensi, baris-baris menunjukkan label yang sebenarnya dan kolom-kolom menunjukkan label yang diprediksi. *Confusion matrix* digunakan untuk melihat distribusi dari kelas-kelas yang diprediksi dari satu tampilan yang diringkas [46].

7) *User Acceptance Testing* (UAT)

Setelah tahap impelementasi atau *deployment*, tahap evaluasi tidak berakhir pada hanya evaluasi model. Tahap UAT atau *User Acceptance Testing* digunakan untuk memverifikasi oleh *user* terkait *acceptability* dari sistem [47]. Terdapat 3 hal yang di uji pada UAT yaitu mengukur penyesuaian sistem dengan kebutuhan pengguna, membatasi bagaimana sistem yang telah diselesaikan, dan mendapatkan fungsi atau logika bisnis yang belum ditemukan sebelumnya [48]. Pada penelitian ini akan menggunkana skala Likert sebagai acuan pembuatan pertanyaan UAT. Skala Likert adalah metode ukuran yang didapatkan oleh kuesioner terkait

pendapat seseorang terhadap suatu objek atau pertanyaan, dimana jawaban yang diperoleh merupakan skor diantara 1 hingga 5 yang merepresentasikan sikap setuju atau tidak setuju [49].

2.4 Alat Penelitian

2.4.1 Python

Python merupakan salah satu bahasa pemrograman tingkat tinggi dimana bahasa ini bersifat umum, interpretatif, dan dinamis. Python memiliki beberapa kelebihan dan beragam fitur yang menjadikannya unggul jika dibandingkan dengan bahasa pemrograman lainnya, seperti [50]:

1) Python merupakan pemrograman dengan penggunaan kode yang lebih sederhana dan sedikit jika dibandingkan pemrograman lain. Python juga memiliki struktur bahasa yang mudah dipelajari [51].
2) Python bahasa pemrograman yang memiliki sifat *open-source* dimana bahasa ini dapat digunakan oleh berbagai *operating system*, seperti Linux, Windows, dan Mac OS [50].
3) Python memiliki kepustakaan atau *library* yang beragam dan besar, serta menyediakan modul untuk berbagai jenis kebutuhan, seperti pengolahan data, *software development*, hingga untuk pelatihan model *machine learning* [51].

2.4.2 HTML

Hyper Text Markup Language atau umumnya disebut HTML merupakan suatu bahasa pemrograman yang dapat digunakan dalam pembentukan sebuah *website* yang isinya dapat berupa susunan teks, gambar, formulir, atau konten multimedia jenis lainnya [52]. HMTL dikelola pemakaiannya oleh *World Wide Web Consortium* atau W3C dimana bahasa pemrograman ini menggunakan *tag-tag* yang disusun untuk menampilkan elemen-elemen tertentu pada sebuah *website* agar sesuai dengan layout yang diinginkan. Pada umumnya, file HTML disimpan dengan ekstensi *.html* [53].

2.4.3 CSS

Cascading Style Sheet atau yang pada umumnya disingkat sebagai CSS merupakan tipe bahasa pemrograman yang dapat mendukung bahasa pemrograman HTML dari segi desain tata letak konten [54]. Umumnya *file* CSS akan terpisah dari *file* HTML untuk mengatur dan mengontrol warna, bentuk, tipe tulisan, dan sebagainya. Kode CSS memiliki 3 bagian umum dalam mendukung gaya pembuatan *website* yaitu *Selector*, *property*, dan *value* [53].

2.4.4 PHP

Bahasa pemrograman PHP atau *Hypertext Preprocessor* adalah bahasa pemrograman yang juga digunakan untuk mendukung pembuatan *website* yang akan digunakan tahap *deployment* pada penelitian ini. PHP merupakan salah satu bahasa dengan sifat *server-side* yang dapat digunakan bersamaan dengan HTML. Fungsi dari penggunaan PHP adalah untuk menjadi penghubung antara *database server* dan suatu situs untuk mendapatkan, mengolah, dan menampilkan data [55].

2.4.5 Visual Studio Code

Dalam pembuatan proyek penelitian ini, *software* Visual Studio Code (VSCode) digunakan untuk mengayomi seluruh bahasa pemrograman yang digunakan. Visual Studio Code merupakan *text editor* bersifat *open-source* yang tersedia untuk Windows, macOS, dan Linux, dimana *tools* ini pertama kali diumumkan pada April 2015 oleh Microsoft pada konferensi Build [54]. VSCode memiliki banyak fitur yang membuatnya menjadi pilihan populer bagi para pengembang, antara lain [51]:

1) VSCode mendukung untuk berbagai bahasa pemrograman, termasuk Python, JavaScript, Java, C++, dan C#.
2) VSCode memiliki fungsi *debugger* bawaan yang memungkinkan pengembang untuk menemukan dan memperbaiki bug dalam kode mereka.
3) VSCode memiliki integrasi Git bawaan yang memungkinkan pengembang untuk mengelola kode mereka dengan mudah.

4) VSCode memiliki banyak ekstensi yang tersedia yang dapat ditambahkan pada *text editor*.