BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Tabel 2.1 Tabel Penelitian Terdahulu

Nama Jurnal	Judul Artikel	Penulis	Metode	Hasil
Applied Sciences (Switzerland), Vol. 10, No. 13, 2020 [11]	*An Early Warning System to Detect At-Risk Students in Online Higher Education*	David Bañeres, M. Elena Rodríguez, Ana Elena Guerrero-Roldán, dan Abdulkadir Karadeniz	Algoritma *Supervised Learning: Prediction* (*Naïve Bayes*, *Decision Tree*, *K-Nearest Neighbors*, dan *Support Vector Machine*)	Hasil dari penelitian ini menyimpulkan bahwa sistem EWS mampu memprediksi kemungkinan kegagalan siswa dalam suatu *course* dengan tepat, serta algoritma klasifikasi risiko yang akan diterapkan sebagai model *Gradual-At-Risk* (GAR) berhasil mengklasifikasikan tingkat risiko secara tepat dalam dua contoh kasus pengujian.
International Journal of Educational Technology in Higher Education, Vol. 16, No. 1, 2019 [14]	*Using Learning Analytics to Develop Early-Warning System for At-Risk Students*	Gökhan Akçapınar, Arif Altun, dan Petek Aşkar	Algoritma *Supervised Learning: Classification* (*Classification Tree*, *CN2 Rules*, *Naïve Bayes*, *Neural Network*, kNN, *Random Forest*, SVM)	Penelitian ini menunjukkan potensi sistem peringatan dini dalam pembelajaran *online* untuk membantu siswa berisiko gagal dalam mencapai kesuksesan. Model klasifikasi terbaik adalah model dengan algoritma kNN yang menunjukkan akurasi tinggi, yaitu 89%, dalam mengidentifikasi siswa berisiko gagal.
Applied Sciences, Vol. 12, No. 19, 2022 [17]	*Clustering Analysis for Classifying Student Academic Performance in Higher Education*	Ahmad Fikri Mohamed Nafuri, Nor Samsiah Sani, Nur Fatin Aqilah Zainudin, Abdul Hadi Abd Rahman, dan Mohd Aliff	Algoritma *Unsupervised Learning: Clustering* (K-Means, BIRCH, dan DBSCAN) Metrik evaluasi *Silhouette Coefficient*, *Davies-Bouldin*	Pengoptimalan algoritma K-Means pada Model B (KMoB) memiliki kinerja terbaik di antara semua model dalam membentuk 5 kelompok mahasiswa B40 berdasarkan indeks *Silhouette*,

Nama Jurnal	Judul Artikel	Penulis	Metode	Hasil
			Index, dan *Calinski-Harabasz Index*	Davies-Bouldin, dan Calinski-Harabasz.
Applied Sciences, Vol. 12, No. 1, 2022 [18]	*Automatic Clustering of Students by Level of Situational Interest Based on Their EEG Features*	Ernee Sazlinayati Othman, Ibrahima Faye, dan Aarij Mahmood Hussaan	Algoritma *Unsupervised Learning: Clustering* (K-Means dan DBSCAN)	Model dengan algoritma K-Means memiliki performa yang lebih baik dalam membentuk 3 kelompok mahasiswa dan berhasil mencapai parameter kinerja maksimal sebesar 100%.
Data, Vol. 7, No. 11, 2022 [19]	*Density-Based Unsupervised Learning Algorithm to Categorize College Students into Dropout Risk Levels*	Miguel Angel Valles-Coral, Luis Salazar-Ramírez, Richard Injante, Edwin Augusto Hernandez-Torres, Juan Juárez-Díaz, Jorge Raul Navarro-Cabrera, Lloy Pinedo, dan Pierre Vidaurre-Rojas	Algoritma *Unsupervised Learning: Clustering* (K-Means, DBSCAN, dan HDBSCAN) Metrik evaluasi *Silhouette Index*, *Davies-Bouldin Index*, dan *Calinski-Harabasz Index*	Algoritma HDBSCAN menghasilkan model dengan performa terbaik yakni hasil validitas yang lebih tinggi (koefisien *Silhouette* dan indeks Davies–Bouldin) dari total tiga metrik evaluasi.
Lontar Komputer: Jurnal Ilmiah Teknologi Informasi, Vol. 11, No. 1, 2020 [20]	*Customer Segmentation Based on RFM Model Using K-Means, K-Medoids, and DBSCAN Methods*	Rahma Wati Br Sembiring Berahman, Fahd Agodzo Mohammed, dan Kankamol Chairuang	Algoritma *Unsupervised Learning: Clustering* (K-Means, K-Medoids, dan DBSCAN) Metrik evaluasi *Silhouette Coefficient* dan *Davies-Bouldin Index*	Hasil validitas model menunjukkan bahwa model K-Means dengan 2 *cluster* memiliki performa terbaik jika dibandingkan dengan algoritma lainnya, berdasarkan metrik *Silhouette* dan DBI.
JISKA (Jurnal Informatika Sunan Kalijaga), Vol. 7, No. 2, 2022 [21]	*Comparative Study of K-Means Clustering Algorithm and K-Medoids Clustering in Student Data Clustering*	Qomariyah dan Maria Ulfah Siregar	Algoritma *Unsupervised Learning: Clustering* (K-Means dan K-Medoids) Metrik evaluasi *Davies-Bouldin Index*	Berdasarkan nilai DBI, model *clustering* yang memiliki performa terbaik adalah model K-Means dengan 5 *cluster*.

Nama Jurnal	Judul Artikel	Penulis	Metode	Hasil
IAENG International Journal of Computer Science, Vol. 50, No. 3, 2023 [22]	*Revisiting Fuel Subsidies in Indonesia using K-Means, PAM, and CLARA*	Fajar Agung Prasetyo, Rezzy Eko Caraka, Yunho Kim, Noor Ell Goldameir, Sulistyowati, Avia Enggar Tyasti, Prana Ugiana Gio, Faisal Anggoro, Muthia Ramadhani, dan Bens Pardamean	Algoritma *Unsupervised Learning: Clustering* (K-Means, PAM, dan CLARA) Metrik evaluasi *Silhouette Coefficient, Dunn Index*, dan *Connectivity Coefficient*	Hasil penelitian menunjukkan bahwa model *clustering* K-Means dengan 2 klaster (k=2) memiliki performa terbaik dengan nilai *Silhouette Coefficient* dan *Dunn Index* tertinggi dibandingkan algoritma lainnya.
ICIC Express Letters, Part B: Applications, Vol. 13, No. 3, 2022 [23]	*Covid-19 Clustering by Province: A Case Study of Covid-19 Cases in Indonesia*	Ferry Vincenttius Ferdinand, Johan Sebastian, Christopher Nata, Friska Natalia, dan Stevanus Adiwena	Algoritma *Unsupervised Learning: Clustering* (K-Means, K-Medoids, dan *Gaussian Mixture Model*) Metrik evaluasi *Calinski-Harabasz* untuk K-Means, *Gap-Statistics* untuk *K-Medoids, dan Bayesian Information Criterion* (BIC) untuk GMM.	Kesimpulan dari studi menunjukkan bahwa model dengan performa terbaik dalam pengelompokan data berdasarkan parameter kasus dan korban jiwa pada 8 periode waktu dapat ditemukan pada algoritma *Gaussian Mixture Model* (GMM) berdasarkan rasio Wvar/Bvar. Di sisi lain, untuk parameter pemulihan, algoritma pembentuk model *clustering* terbaik adalah K-Medoids.
Journal of Computational Science, Vol. 51, 2021 [26]	*Clustering of Graphs Using Pseudo-Guided Random Walk*	Zahid Halim, Hussain Mahmood Sargana, Aadam, Uzma, dan Muhammad Waqas	Metrik evaluasi *Silhouette Coefficient* (SC), *Davies-Bouldin Index* (DBI), *Calinski-Harabasz Index* (CHI), *Dunn Index* (DI), *Modularity Index* dan *Normalized Cut*	Metrik evaluasi yang digunakan untuk menilai performa 8 model *clustering* dengan pendekatan *random-walk* menunjukkan bahwa model *Pseudo-Guided Random Walk* (PGRW) dalam penelitian ini unggul dengan nilai DBI, DI, CHI, dan *modularity index* yang lebih baik.

Berdasarkan tabel 2.1 di atas, penelitian ini akan mengembangkan sistem intervensi yang dapat memberikan peringatan dini sekaligus menyediakan fungsi

pemantauan karena telah terbukti dapat mendeteksi potensi dan menurunkan tingkat kegagalan siswa dalam menjalankan studi mereka, sebagaimana yang dijelaskan sebagai hasil pada penelitian terdahulu [11] dan [14]. Namun, perbedaan utama penelitian ini dibandingkan dengan penggunaan algoritma klasifikasi atau prediksi dalam membangun model untuk diterapkan pada sistem, seperti dalam dua penelitian tersebut, adalah penggunaan teknik pengelompokan *(clustering)* untuk memberikan label dan mengelompokkan data ke dalam klaster. Penelitian ini akan menggunakan algoritma *clustering* yang juga digunakan pada penelitian [17]–[23] untuk mengembangkan suatu *Early Intervention Warning and Monitoring System* (EIWMS). Secara spesifik, tiga jenis algoritma *clustering* yang digunakan sebagai algoritma utama untuk membagi kelompok mahasiswa berdasarkan perkembangan studi mereka adalah K-Means, K-Medoids, dan DBSCAN.

Poin perbedaan antara penelitian ini dengan penelitian [17]–[19] dan [21] adalah objektif penelitian dan jenis data mahasiswa yang digunakan. Penelitian ini berfokus pada pengembangan sistem EIWMS dari hasil identifikasi kelompok siswa berdasarkan perkembangan studi menggunakan data satuan kredit semester yang diemban selama studi mahasiswa dan nilai capaian akademik mereka (IPK atau IPS). Di sisi lain, penelitian [17] bertujuan untuk pengelompokkan siswa berdasarkan data mengenai latar belakang dan capaian akademik siswa untuk mengurangi tingkat *drop-out* universitas. Fokus pada penelitian [18] adalah mengelompokkan siswa berdasarkan tingkat minat situasional mereka selama pembelajaran berdasarkan hasil deteksi psikologis dari data sinyal *electroencephalography* (EEG). Penelitian [19] berfokus pada pengelompokan tingkat risiko *drop-out* siswa berdasarkan data perilaku maupun gangguan psikologis yang dialami. Penelitian [21] bertujuan untuk memetakan persebaran mahasiswa berdasarkan data asal daerah, asal sekolah, semester, dan IPK. Sementara itu, perbedaan antara penelitian ini dengan penelitian [20], [22], dan [23] terletak pada objek penelitiannya, yaitu data mahasiswa. Pembeda penelitian ini dengan tujuh penelitian terdahulu dengan topik *clustering* tersebut juga terletak pada pemilihan algoritma yang digunakan.

Penelitian ini juga akan memanfaatkan beberapa metrik validasi *clustering* internal untuk membandingkan performa model dan memilih model dengan algoritma *clustering* terbaik dan paling cocok untuk mengelompokkan data mahasiswa UMN. Terdapat 4 (empat) metrik evaluasi model yang akan digunakan pada penelitian ini, sebagaimana yang juga digunakan pada penelitian [17], [19]–[22], dan [26], yakni *Silhouette Score* (SH), *Davies-Bouldin Index* (DBI), *Calinski-Harabasz Index* (CHI), dan *Dunn Index* (DI). Namun, keenam penelitian terdahulu tersebut tidak menggunakan keempat metrik evaluasi secara bersamaan. Oleh karena itu, perbedaan dan kebaruan penelitian ini dibandingkan dengan sepuluh penelitian terdahulu yang tercantum dalam Tabel 2.1 terletak pada komposisi jenis dan pemilihan algoritma, metode evaluasi, serta objek penelitian. Dengan demikian, penelitian ini memberikan kontribusi baru dalam bidang *clustering* data mahasiswa, terutama dalam konteks identifikasi pola berdasarkan perkembangan studi mereka.

2.2. Tinjauan Teori

2.2.1. Kelulusan Mahasiswa

Kelulusan adalah langkah akhir bagi setiap mahasiswa untuk menyelesaikan pendidikannya di perguruan tinggi dan terjun ke dunia kerja. Kelulusan mahasiswa juga termasuk sebagai salah satu besaran penilaian dalam Standar Penjaminan Mutu Internal (SPMI) yang dapat menentukan akreditasi instansi perguruan tinggi [27]. Dilansir dari Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT), angka kelulusan mahasiswa tepat waktu telah menjadi tolak ukur utama dalam menentukan kualitas dan efisiensi kegiatan pembelajaran yang dilakukan oleh perguruan tinggi [28]. Mahasiswa dapat dinyatakan lulus apabila telah mengambil dan menyelesaikan satuan kredit atau SKS sejumlah tetapan program studi, menjalankan kegiatan praktik magang ataupun penelitian, menyelesaikan tugas akhir atau skripsi, serta persyaratan lainnya yang ditentukan oleh perguruan tinggi naungan mereka.

2.2.2. *Learning Analytics*

Learning Analytics merupakan area penelitian interdisipliner baru yang menyelidiki perolehan, pengukuran, analisis, dan pelaporan data mengenai performa siswa dan kondisi lingkungan pembelajarannya [29]. Tujuan dari tipe analitik ini adalah untuk mengidentifikasi fakta terkait perilaku, aktivitas, atau lingkungan siswa yang kemudian akan digunakan dalam mengoptimalkan proses pembelajaran atau menjawab permasalahan akademis. *Learning analytics* mengimplementasikan pendekatan berbasis data yang dikembangkan dari hasil evaluasi dan penilaian kemajuan, motivasi, sikap, serta kepuasan peserta didik. Penerapan analitik pembelajaran dapat memberikan manfaat seperti: 1) menafsirkan perilaku belajar yang tidak biasa; 2) menemukan pola pembelajaran yang efektif; 3) mengungkap kesalahan dan upaya pembelajaran tidak efektif; 4) menerapkan intervensi yang bermanfaat dan relevan; serta 5) meningkatkan pemahaman peserta didik tentang kemajuan dan perkembangan akademik mereka [30].

2.2.3. Sistem Intervensi Peringatan Dini dan Pemantauan

Sistem Intervensi Peringatan Dini dan Pemantauan, atau *Early Intervention Warning and Monitoring System* (EIWMS), merupakan turunan dari konsep sistem *Early Warning System* (EWS). EWS mengacu pada sistem memberikan peringatan kepada penggunanya mengenai potensi ancaman yang bertujuan untuk menghindari permasalahan sekecil apapun sebelum menjadi ancaman nyata. Dalam bidang pendidikan, EWS merupakan sistem yang mengimplementasikan metode dan alat untuk mendeteksi risiko putus sekolah dan menerapkan intervensi yang tepat untuk meningkatkan tingkat retensi dan kesuksesan siswa [11].

EIWMS merupakan sistem dengan strategi sistematis komprehensif yang memanfaatkan data historis dan kontekstual siswa, serta teknologi analitik pembelajaran *(learning analytics)*, untuk mengidentifikasi siswa yang berisiko mengalami kegagalan akademik, memberikan intervensi

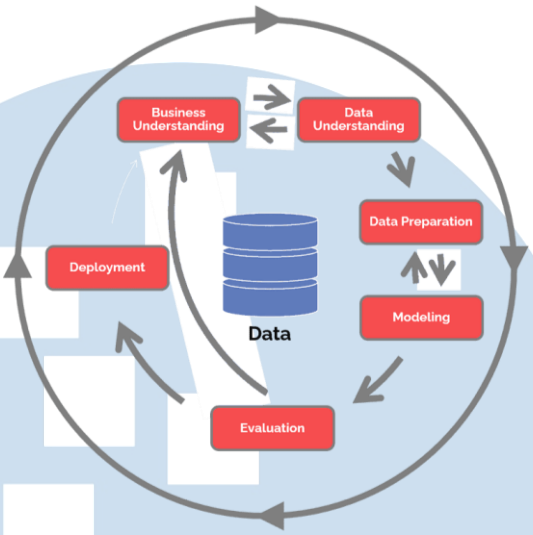
kepada siswa yang berisiko, serta memantau tanggapan siswa yang berisiko terhadap intervensi [12]. Sistem ini dirancang untuk memantau dan membantu menghubungkan siswa dengan bantuan dan sumber daya yang relevan secara tepat waktu untuk memitigasi kegagalan. EIWMS dapat meningkatkan efektivitas dan efisiensi upaya para penasihat akademik dalam menawarkan bantuan bagi siswa dengan risiko kegagalan berdasarkan kecenderungan perilaku akademik siswa [14]. Penerapan dini dari sistem intervensi peringatan dini dan pemantauan studi pada institusional telah terbukti efektif dari segi biaya [31].

2.3. Kerangka Kerja, Algoritma, dan Metode Evaluasi

2.3.1. Kerangka Kerja

2.3.1.1. CRISP-DM

CRISP-DM *(Cross-Industry Standard Process for Data Mining)* merupakan suatu model berisi alur proses yang ditetapkan sebagai standar pelaksanaan dan pengembangan analisis dalam *data mining* maupun pembentukan *data-driven models* (DDM). Penerapan kerangka kerja CRISP-DM dalam pengembangan proyek *data mining* maupun DDM membantu menghindari kesalahan umum dalam pemodelan data, yakni masalah generalisasi terhadap input data baru dan *overfitting* pada model [32]. Kerangka kerja ini terdiri dari enam fase sebagai pedoman dalam merencanakan, mengelola, dan mengimplementasikan metode *machine learning* ke dalam suatu data seperti yang terlihat pada Gambar 2.1 berikut.



Gambar 2.1 Tahapan Kerangka Kerja CRISP-DM
Sumber: *Data Science Process Alliance* (2023) [33]

Berikut ini merupakan penjelasan dari keenam fase yang terdapat dalam CRISP-DM [32], [34], [35].

a. *Business Understanding* (Pemahaman Bisnis)

Fase ini berfokus pada pemahaman tujuan dan persyaratan dari suatu objek bisnis. Fase ini meliputi tiga aktivitas kunci yakni: 1) menentukan tujuan, permasalahan, dan kebutuhan dari objek bisnis secara detail dan menyeluruh; 2) menerjemahkan tujuan bisnis menjadi parameter batasan dan kesuksesan proyek; 3) merumuskan suatu strategi untuk mencapai tujuan atau menjawab permasalahan bisnis tersebut.

b. *Data Understanding* (Pemahaman Data)

Fase ini berfokus pada identifikasi, pengumpulan, dan analisis sekumpulan data yang dibutuhkan untuk mencapai tujuan proyek. Fase ini meliputi empat aktivitas kunci yakni: 1) mengumpulkan dan menyatukan data yang berasal dari berbagai sumber; 2) memeriksa format dan kelengkapan data untuk mengidentifikasi masalah pada data; 3) menganalisis dan menjelajahi data lebih lanjut untuk menemukan hal menarik atau informasi tersembunyi dalam data; 4) menilai dan mengevaluasi kualitas data.

c. *Data Preparation* (Persiapan Data)

Fase ini berfokus pada persiapan data mentah untuk membentuk model penyelesaian masalah kasus bisnis. Terdapat tiga kegiatan yang dapat dilakukan pada fase ini yaitu: 1) menentukan variabel, parameter, atau subset data yang akan digunakan dalam penelitian *(select data);* 2) membersihkan data dengan menghapus atau mengisi *missing value* serta menghilangkan data redundan dan *outlier (data cleaning);* 3) mengubah dan mengonversi entri nilai variabel *(transform data).*

d. *Modeling* (Pemodelan)

Fase ini berfokus pada pembangunan dan pembentukan model dari data yang telah disiapkan dan terbagi sebelumnya menjadi set *training* dan *testing (splitting data).* Terdapat tiga kegiatan yang dapat dilakukan pada fase ini yaitu: 1) memilih teknik dan algoritma pemodelan yang akan digunakan; 2) membentuk model dengan algoritma pilihan; 3) melakukan penyesuaian atau kalibrasi aturan model untuk memperoleh hasil yang optimal. Fase ini dapat diulang untuk menghasilkan model terbaik dari sejumlah model.

e. *Evaluation* (Evaluasi)

Fase ini berfokus pada evaluasi hasil pemodelan data. Adapun, empat aktivitas yang dapat dilakukan pada fase ini yakni: 1) menilai kualitas dan efektivitas model; 2) mengevaluasi apakah model berhasil memenuhi tujuan dan kriteria keberhasilan bisnis; 3) meninjau kembali rangkaian proses yang telah dilakukan dalam menghasilkan model untuk mengidentifikasi kendala; 4) mengambil keputusan selanjutnya terkait penggunaan hasil model.

f. *Deployment* (Penyebaran)

Fase ini berfokus pada pemanfaatan dan pengimplementasian hasil model untuk menyelesaikan permasalahan atau kasus nyata. Fase ini meliputi perencanaan strategi penerapan model dan pelaporan

akhir berupa pembuatan presentasi ataupun laporan. Fase ini juga dapat menghasilkan hasil akhir berupa *software* atau program nyata yang dapat diimplementasikan sebagai solusi dari permasalahan.

2.3.1.2. *Machine Learning*

Machine Learning (ML) mulai berkembang sebagai subbidang dari konsep *Artificial Intelligence* (AI) pada pertengahan abad kedua puluh, yang melibatkan implementasi algoritma pembelajaran mandiri untuk mengumpulkan informasi dari data [36]. *Machine learning* berfokus dalam menemukan dan mempelajari pola dari data untuk meningkatkan kinerja atau optimisasi proses [37]. Pembelajaran mesin dapat mengidentifikasi dan menganalisis pola pada data seperti korelasi non-linier, interkoneksi, dimensi dasar pada data, atau subkelompok variabel tanpa diprogram secara eksplisit. Pembelajaran mesin menyediakan metode yang lebih efektif untuk mengekstrak wawasan dalam data untuk terus meningkatkan kinerja model dan membuat keputusan berdasarkan data *(data driven)*. Secara singkat, *machine learning* dapat didefinisikan sebagai pendekatan komputasi yang terotomasi dan fleksibel untuk menemukan pola dalam struktur data besar. Terdapat dua jenis metode pembelajaran mesin yang paling sering digunakan yakni *unsupervised learning* dan *supervised learning*. Kehadiran label dalam subset data pelatihan membedakan dua kelompok metode utama ini.

2.3.1.2.1. *Unsupervised Learning*

Unsupervised Learning adalah jenis pembelajaran mesin yang mengidentifikasi struktur dan pola dalam data yang tidak berlabel atau tanpa target fitur [38]. *Unsupervised Learning* sering diimplementasikan sebelum membangun model dengan algoritma *supervised learning* untuk menghasilkan label data. Beberapa pendekatan yang menggunakan jenis algoritma ini antara lain: *clustering,* asosiasi, dan *dimension reduction.*

Clustering digunakan untuk menemukan pengelompokan alami dalam data yang tidak berlabel dan kemudian memberi label pada setiap nilainya. Asosiasi digunakan untuk mengungkap aturan-aturan yang dapat menjelaskan hubungan antar atribut dalam data secara tepat. Terakhir, *dimension reduction* biasa digunakan sebagai teknik *data preprocessing* untuk menghilangkan *noise* yang dapat mengganggu akurasi prediksi beberapa algoritma, serta mengompresi data menjadi ukuran yang lebih kecil dengan tetap mempertahankan informasi penting di dalamnya [39].

2.3.1.2.2. *Supervised Learning*

Supervised Learning adalah jenis pembelajaran mesin yang melibatkan pembelajaran fungsi untuk memetakan karakteristik *input* (pengukuran) x ke variabel *output* y [38]. Algotitma *supervised learning* meliputi klasifikasi dan regresi, yang memerlukan data berlabel untuk membuat prediksi atau mengklasifikasikan objek [37]. Tujuan klasifikasi adalah melatih model untuk memprediksi label kelas variabel respons y sesuai label, yang dapat berupa nilai diskrit atau kategorikal. Sebaliknya, regresi bertujuan untuk memprediksi nilai dari variabel respon y berupa nilai kontinu berdasarkan korelasinya dengan variabel prediktor x.

2.3.2. Algoritma *Clustering*

2.3.2.1. K-Means

K-Means adalah salah satu algoritma *partitioning clustering* yang digunakan untuk menghasilkan klaster atau label dalam data. *Partition-based clustering* merupakan metode pengelompokan data di mana sejumlah objek akan dikumpulkan terlebih dahulu untuk kemudian dipartisi menjadi beberapa grup berisi nilai data yang serupa [40]. K-Means termasuk ke dalam

algoritma *distance-based* yang memanfaatkan metrik Euclidean atau kosinus dalam perhitungan jarak ke titik data lainnya. Algoritma K-Means mengelompokkan n data ke dalam K klaster terpisah melalui iteratif proses yang bersifat konvergen, dan kinerjanya bergantung pada nilai *centroid* k awal [41]. Pemilihan nilai *centroid* atau jumlah klaster yang tidak tepat dapat menyebabkan proses pembagian kelompok data yang tidak stabil dan meningkatkan jumlah iterasi, sehingga kompleksitas waktu dan ruang pun meningkat. Berikut adalah urutan langkah-langkah dalam iterasi algoritma K-Means:

1) Awali dengan menentukan jumlah klaster 'k' dan pusat klaster secara acak.
2) Hitung jarak antara setiap titik data dengan pusat-pusat klaster yang telah ditentukan.
3) Tempatkan setiap titik data ke dalam klaster dengan jarak terkecil ke pusat klaster.
4) Hitung kembali pusat klaster baru berdasarkan rata-rata dari semua titik data yang termasuk dalam klaster tersebut dengan rumus berikut.

$$C_j = \frac{1}{|S_j|} \sum_{i \in S_j} x_i$$

Rumus 2.1 Rumus Pembaruan Pusat Klaster K-Means

Keterangan Rumus 2.1:

C_j : pusat klaster ke-j

S_j : himpunan data yang termasuk dalam klaster ke-j

x_i : titik data ke-i

5) Ulangi langkah 2 hingga 4 sampai tidak ada perubahan lagi dalam penyusunan klaster.

6) Proses iteratif berhenti ketika tidak ada lagi perubahan dalam penempatan titik data ke dalam klaster dan model *clustering* dianggap telah konvergen.

2.3.2.2. K-Medoids

K-Medoids, atau yang juga dikenal sebagai *Partitioning Around Medoids* (PAM), merupakan jenis algoritma *partitioning clustering* yang melibatkan pengelompokan objek data ke dalam sejumlah klaster yang mencerminkan kesamaan nilai data [42]. Dalam K-Medoids, representasi dari setiap klaster disebut *medoid*, yang merupakan objek data aktual di dalam klaster yang memiliki jarak rata-rata minimum terhadap semua objek lainnya. Algoritma ini berfokus pada pemilihan medoid sebagai pusat klaster untuk meminimalkan total jarak antara medoid dan objek lain di dalam klaster, sehingga membuatnya lebih stabil atau resisten dalam menangani data yang mungkin mengandung variasi nilai ekstrem *(outliers)* [43]. Proses iteratif K-Medoids dilakukan untuk mengoptimalkan lokasi *medoid* dan pembentukan klaster, yang membuatnya lebih tangguh terhadap ketidakpastian dalam pemilihan nilai *centroid* awal. Keseluruhan proses ini memastikan bahwa setiap klaster dikarakterisasi oleh representatif unik dari objek data sehingga meminimalkan dampak *outlier* pada hasil clustering secara signifikan. Berikut adalah tahapan iterasi dalam pembentukan model *clustering* dengan algoritma K-Medoids:

1) Mulailah dengan menentukan jumlah klaster 'k' dan pilih *medoids* awal secara acak.
2) Hitung jarak antara setiap titik data dengan *medoids* yang telah ditentukan.
3) Tetapkan setiap titik data ke dalam klaster dengan jarak terkecil ke *medoids*.

4) Pilih *medoids* baru untuk setiap klaster berdasarkan total jarak minimum ke semua titik data lain dalam klaster menggunakan rumus berikut.

$$D(x_i, x_j) = \sum_{k=1}^{n} |x_{ik} - x_{jk}|$$

Rumus 2.2 Rumus Perhitungan Jarak K-Medoids

Keterangan Rumus 2.2:

$D(x_i, x_j)$: jarak antara dua titik data x_i dan x_j

x_{ik} dan x_{jk} : koordinat atribut ke-k dari setiap titik

5) Ulangi langkah 2 hingga 4 sampai tidak ada perubahan lagi dalam penyusunan klaster.
6) Iterasi dihentikan saat tidak ada lagi perubahan dalam pengelompokan titik data ke dalam klaster.

2.3.2.3. DBSCAN

Density Based Spatial Clustering of Applications with Noise (DBSCAN) merupakan salah satu algoritma *density-based clustering* yang paling umum digunakan. *Density-based clustering* membentuk kelompok data yang secara akurat mencerminkan kepadatan pada suatu set data [44]. DBSCAN menggunakan konsep ε-*neighborhood* untuk memperkirakan kepadatan di sekitar titik data. Dalam DBSCAN, jarak kerapatan atau kedekatan suatu objek harus cukup tinggi untuk menjadi bagian dari sebuah klaster. Batas *cluster* diwakili oleh titik-titik di sekitar inti *cluster*, sedangkan sisanya adalah *noise*. Dapat disimpulkan bahwa algoritma DBSCAN membutuhkan dua parameter agar dapat berjalan, yaitu ε sebagai titik awal klaster dan MinPts sebagai jumlah data minimum yang dibutuhkan untuk membentuk suatu zona padat atau *cluster* [45]. Berikut adalah langkah pembentukan klaster berdasarkan algoritma *clustering* DBSCAN:

1) Langkah awal adalah menentukan parameter jarak batas (epsilon) dan jumlah minimum titik (MinPts).
2) Identifikasi titik inti *(core points)* dengan memeriksa jumlah tetangga dalam radius epsilon.
3) Tentukan klaster dengan menghubungkan titik inti yang saling berdekatan dan tetangganya yang memenuhi syarat.
4) Tambahkan titik-titik yang terhubung ke klaster yang sama dan periksa apakah mereka juga merupakan titik inti.
5) Tandai titik-titik yang tidak termasuk ke klaster mana pun sebagai *noise*.
6) Ulangi langkah-langkah di atas untuk semua titik data hingga seluruh data selesai diproses.

2.3.3. Algoritma Optimasi

2.3.3.1. *Principal Component Analysis* (PCA)

Principal Component Analysis (PCA) merupakan metode analisis statistik yang digunakan untuk mereduksi dimensi data yang kompleks sekaligus mempertahankan informasi penting [46]. PCA terutama berguna dalam skenario penelitian yang melibatkan data dalam jumlah besar, memiliki banyak variabel, dan saling berkorelasi. Tujuan dari penerapan PCA adalah mengidentifikasi fitur yang lebih sedikit namun tetap mewakili data dalam ruang dimensi yang lebih rendah, dengan kerugian informasi minimal. Dalam konteks *clustering*, PCA dapat diimplementasikan pada tahap pra-pemrosesan data maupun visualisasi hasil klaster. PCA membantu mengatasi masalah *over-fitting* dengan mereduksi jumlah fitur (variabel) dalam data. Dengan mengurangi dimensi menggunakan PCA, korelasi antar-variabel dapat dihilangkan, *noise* dapat dikurangi, dan variabilitas yang signifikan dapat dipertahankan [47]. Dalam visualisasi, PCA memungkinkan pemahaman struktur data yang

lebih baik dan interpretasi yang lebih jelas. Hal ini disebabkan karena data yang telah direduksi dimensinya dapat divisualisasikan dalam ruang yang lebih sederhana, seperti ruang dua atau tiga dimensi, memungkinkan pengamatan pola klaster secara lebih efektif. PCA juga memfasilitasi identifikasi pola atau struktur data yang tidak terlihat dalam dimensi aslinya. Berikut adalah tahap penerapan dan rumus dari algoritma PCA:

1) Hitung *mean* (rata-rata) dari setiap fitur (kolom) dalam *dataset*.
2) Normalisasi data dengan mengurangkan *mean* dari setiap fitur.
3) Hitung matriks kovarian antara fitur-fitur dalam *dataset*.
4) Hitung nilai eigen dan vektor eigen dari matriks kovarian.
5) Pilih n vektor eigen dengan nilai eigen tertinggi sebagai komponen utama (n merupakan jumlah dimensi yang diinginkan untuk *dataset* yang telah direduksi).
6) Proyeksikan data ke ruang vektor *eigen* yang dipilih.
7) Data yang diproyeksikan adalah representasi dari *dataset* yang telah direduksi.

$$Y = X \cdot W$$

Rumus 2.3 Rumus Umum PCA

Keterangan Rumus 2.3:

Y : matriks data yang diproyeksikan ke ruang vektor eigen (komponen utama)

X : matriks data asli

W : matriks vektor eigen yang dipilih

2.3.4. Metode Evaluasi

2.3.4.1. *Silhouette Score*

Silhouette Score merupakan salah satu metrik validitas hasil pengelompokan data (*clustering*) yang mengukur seberapa dekat setiap titik data dengan titik data lain dalam suatu kelompok dan seberapa baik pemisahan kelompok satu sama lain [48]. Perhitungan *silhouette score* melibatkan rata-rata jarak dalam suatu kelompok (a) dan rata-rata jarak ke kelompok tetangga terdekat (b) untuk setiap titik data. Skor ini memiliki rentang nilai dari -1 hingga +1. Nilai positif menunjukkan bahwa sampel tersebut jauh dari kelompok tetangga dan berada di kelompok yang benar. Nilai 0 menunjukkan bahwa sampel berada tepat atau sangat dekat dengan batas keputusan antara dua kelompok tetangga, sementara nilai negatif menunjukkan bahwa sampel-sampel tersebut mungkin telah salah diklasifikasikan ke kelompok yang salah. Hasil *cluster* dikatakan semakin baik jika skornya mendekati positif yang mengindikasikan adanya pembagian yang baik antar *cluster* [49]. Berikut adalah rumus perhitungan metrik evaluasi *Silhouette Score.*

$$S = \frac{b - a}{\max(a, b)}$$

Rumus 2.4 Rumus Metrik Evaluasi *Silhouette Score*

Keterangan Rumus 2.4:

a : rata-rata jarak antara titik data dengan titik-titik lain dalam klaster yang sama

b : rata-rata jarak terdekat dari titik data ke klaster lainnya yang berbeda

2.3.4.2. *Davies-Bouldin Index*

Davies-Bouldin Index adalah metrik validasi yang mengevaluasi model *clustering* berdasarkan kesamaan rata-rata

antara suatu kelompok dengan kelompok lain yang paling mirip dengannya [25]. Indeks ini membandingkan jarak dalam kelompok terhadap jarak antar kelompok dengan menjumlahkan jarak rata-rata antara dua kelompok, dibagi oleh jarak antara pusat-pusat kelompok tersebut, untuk mendapatkan nilai maksimum. Skor DBI yang lebih rendah menunjukkan pembagian kelompok yang terdefinisi dengan baik, yakni jarak yang lebih kecil dalam satu kelompok (karakteristik setiap anggota sama) dan jarak yang lebih besar antara kelompok (karakteristik setiap kelompok sudah jelas berbeda) [50]. Berikut adalah rumus perhitungan metrik evaluasi *Davies-Bouldin Index.*

$$DB = \frac{1}{k}\sum_{i=1}^{k} max_{j\neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$

Rumus 2.5 Rumus Metrik Evaluasi *Davies-Bouldin Index*

Keterangan Rumus 2.5:

k : jumlah klaster

σ_i : dispersi dalam klaster ke-i

$d(c_i, c_j)$: jarak antara pusat klaster ke-i (c_i) dan klaster ke-j (c_j)

2.3.4.3. *Calinski-Harabasz Index*

Calinski-Harabasz Index atau *Variance Ratio Criterion* merupakan metrik rasio varians untuk mengevaluasi hasil *clustering* berdasarkan tingkat dispersi atau penyebaran dalam kelompok dan antar kelompok [25]. Nilai kovarian antar kelompok yang semakin besar mengindikasikan semakin tinggi tingkat penyebaran antar kelompok. Sebaliknya, nilai kovarian dalam kelompok yang semakin kecil mengindikasikan semakin erat hubungan atau penyebaran anggota dalam kelompok. Semakin tinggi indeks CH, yang mencerminkan rasio antara kovarian antar dan dalam kelompok, menandakan hasil

clustering yang lebih baik [51]. Berikut adalah rumus perhitungan metrik evaluasi *Calinski-Harabasz Index.*

$$CH = \frac{Tr(B_k)}{Tr(W_k)} \times \frac{(N-k)}{(k-1)}$$

Rumus 2.6 Rumus Metrik Evaluasi *Calinski-Harabasz Index*

Keterangan Rumus 2.6:

$Tr(B_k)$: *trace* dari matriks dispersi antar-klaster

$Tr(W_k)$: *trace* dari matriks dispersi dalam-klaster

N : jumlah titik data

k : jumlah klaster

2.3.4.4. *Dunn Index*

Dunn Index adalah metrik validasi internal yang mengukur tingkat pemisahan dan kerapatan dari hasil pengelompokan algoritma *clustering* [52]. Perhitungan yang digunakan adalah rasio pembagian antara jarak minimum antar kelompok (pemisahan) terhadap jarak maksimum dalam kelompok (kerapatan). Nilai DI yang tinggi mengindikasikan kualitas pengelompokan yang semakin baik dimana terdapat variasi yang kecil antar anggotanya, pemisahan yang baik, serta rata-rata jarak antar kelompok yang lebih besar dibanding varians dalam kelompok. Berikut adalah rumus perhitungan metrik evaluasi *Dunn Index.*

$$D = min_{1 \le i \le k} \left(min_{j \ne i} \left(\frac{d(c_i, c_j)}{max_{l=1}^{k} diam(C_l)} \right) \right)$$

Rumus 2.7 Rumus Metrik Evaluasi *Dunn Index*

Keterangan Rumus 2.7:

$d(c_i, c_j)$: jarak antara pusat klaster ke-i (c_i) dan klaster ke-j (c_j)

$diam(C_l)$: diameter klaster ke-l (C_l)

2.4. Alat Penelitian

2.4.1. Python

Python adalah bahasa pemrograman umum tingkat tinggi yang sering digunakan dalam beberapa tahun terakhir. Python terkenal karena mudah dipelajari, namun tetap dapat memecahkan permasalahan kompleks menggunakan konsep *machine learning* ataupun *deep learning.* Sintaks dalam bahasa pemrograman Python mengutamakan aspek keterbacaan *(readability)* dan kesederhanaan *(simplicity)* kode dalam konsep desainnya. Oleh karena itu, penguna dapat mengembangkan program dengan jumlah kode yang lebih sedikit daripada bahasa pemrograman lainnya. Salah satu karakteristik Python adalah penyediaan dukungan terhadap berbagai konsep pemrograman termasuk prosedural, imperatif, dan fungsional. Python memiliki pustaka standar yang cukup besar dan komprehensif, mendukung sistem bertipe dinamis, dan dapat mengelola memori secara otomatis [53].

2.4.2. Laravel

Laravel merupakan salah satu pengembangan *framework* PHP berdasarkan model desain Model, View, dan Controller (MVC). MVC adalah metode yang teruji efektif dalam mengembangkan aplikasi yang terstruktur dan modular sehingga dapat mengurangi kompleksitas desain arsitektur serta meningkatkan produktivitas karena memberikan fleksibilitas dalam penggunaan ulang kode [54]. Berkat model ini, Laravel memiliki dua karakteristik utama dalam penggunaannya yakni sederhana dan fleksibel. Dalam penggunaannya, Laravel dilengkapi dengan utilitas baris perintah bernama "Artisan," yang berguna untuk proses pengemasan dan instalasi bundel *(package)* [55]. Penggunaan Laravel dapat meningkatkan efisiensi dan kemudahan pengembangan *website* lewat penyediaan *frameworks* dasar, API, *libraries*, *plugin*, serta *extension* yang lengkap. Faktor lain yang berkontribusi terhadap kepopuleran penggunaan Laravel yakni fiturnya yang intuitif, bersifat *open source*, mendukung integrasi SQL maupun layanan basis data lainnya, serta skalabilitas penerapannya yang baik.

2.4.3. Visual Studio Code

Visual Studio Code atau yang biasa disebut sebagai VS Code adalah suatu perangkat lunak *(software)* rancangan Microsoft yang digunakan sebagai alat untuk memfasilitasi aktivitas menulis, memodifikasi, dan mengorganisir kode sumber suatu program komputer *(source code editor)*. Sejak dirilis pada tahun 2015, VS Code menawarkan beragam fitur yang dapat memudahkan pengembangan perangkat lunak seperti penyorotan sintaksis, IntelliSense untuk penyelesaian kode, dukungan *debugging*, integrasi layanan kontrol versi (Git), dan pasar ekstensi yang luas [56]. Visual Studio Code populer digunakan karena ukurannya yang ringan, menyediakan antarmuka *(interface)* yang ramah pengguna, bersifat fleksibel dan *open source*, serta memiliki kompabilitas lintas platform yakni Windows, macOS, dan Linux. VS Code mendukung penggunaan berbagai bahasa pemrograman, mulai dari bahasa populer seperti HTML, CSS, PHP, JavaScript, dan Python hingga bahasa yang sedang berkembang seperti TypeScript dan Rust [57]. Pengembang dapat dengan mudah mengakses dukungan bahasa yang kaya dalam VS Code melalui penggunaan ekstensi, memungkinkan mereka untuk berpindah dengan cepat antar proyek dengan kebutuhan bahasa yang berbeda. Hal ini meningkatkan daya tarik dan utilitas yang luas di kalangan komunitas pengembangan, baik *website*, aplikasi berbasis *cloud*, maupun proyek perangkat lunak lainnya, dengan menyajikan alat yang efisien untuk pengalaman pengkodean yang produktif.