

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Sistem

Sistem yang berdasar pada prosedur ialah suatu struktur kerja dari beberapa langkah yang saling terhubung, yang bersatu mengerjakan tujuan tertentu. Sedangkan sistem yang berfokus pada prosedur bisa diartikan sebagai setiap kesatuan, baik secara konseptual maupun fisik, terdiri atas elemen - elemen yang saling bergantung (Duha & Juliani, 2020).

Sistem merupakan gabungan banyak elemen, komponen yang saling berhubungan, dengan tujuan memperoleh hasil tertentu yang diinginkan (Maydianto & Ridho, 2021). Sistem memiliki pendekatan yang difokuskan pada rangkaian prosedur kerja yang saling terhubung, berkolaborasi dan terorganisir untuk mencapai sasaran yang diinginkan. Rangkaian prosedur tersebut terdapat panduan dengan langkah-langkah yang berurutan mengenai apa, siapa, kapan dan bagaimana cara kerjanya (Prehanto, 2020:3).

Berdasarkan penjelasan di atas, kesimpulannya sistem merupakan suatu kesatuan yang terdiri atas berbagai komponen yang saling berhubungan serta memerlukan satu dengan lainnya guna memperoleh tujuan tertentu. Adapula sistem sebagai jaringan prosedur kerja yang

terhubung dan tersusun secara teratur, yang mencakup langkah-langkah mengenai apa yang dikerjakan.

2. *Machine Learning*

Machine Learning (ML) adalah suatu disiplin yang berkaitan dengan perancangan serta analisis algoritma sehingga nantinya komputer dapat belajar. Ketika algoritma generik ini mendapatkan suatu set data, maka ia akan menciptakan aturan, model atau inferensi berdasarkan data - data tersebut (Id, 2021:6).

Machine Learning merupakan cabang ilmu yang mengeksplorasi bagaimana algoritma komputer dapat mengetahui pola-pola yang ada pada data, bertujuan untuk merubah bermacam-macam jenis data untuk dijadikan tindakan nyata dengan meminimalisir keterlibatan manusia. *Machine Learning* dapat menciptakan mesin yang belajar dari data yang disediakan, yang kemudian dapat menarik keputusan secara mandiri (Kurniawan, 2022:5).

Machine learning merupakan aspek dari kecerdasan buatan yang memerlukan data-data yang akurat dalam proses belajarnya. *Machine learning* mampu menghasilkan keputusan yang tepat, cepat, dan dapat menawarkan solusi untuk berbagai permasalahan. *Machine learning* memiliki kemampuan untuk belajar secara mandiri dan mengambil keputusan tanpa perlu diprogram berkali - kali oleh manusia, ini terjadi karena pengalaman berbagai data yang dimilikinya (Maulani et al., 2025:1).

Berdasarkan penjelasan di atas dapat disimpulkan *Machine Learning* ialah elemen kecerdasan buatan yang memfokuskan diri pada algoritma komputer untuk belajar dari data, mengidentifikasi pola, dan secara otomatis menciptakan model atau aturan. Dengan menggunakan data yang ada, *machine learning* memungkinkan sistem untuk melakukan keputusan atau prediksi tanpa banyak campur tangan manusia, sehingga dapat menyediakan solusi yang cepat dan akurat untuk berbagai masalah.

3. *Website*

Web yang sering disebut situs *website* adalah suatu jaringan informasi universal yang menghubungkan sejumlah besar komputer melalui protokol internet. Situs *website* menjadi elemen fundamental dari internet karena memungkinkan pemakai mengakses, menemukan, dan menyebarluaskan bermacam-macam jenis konten digital, seperti tulisan, visual, video, serta data lain, menggunakan peramban *website* (Mukhlis et al., 2023:2).

Situs *website* berisi sekumpulan laman yang menyimpan beragam informasi seperti tulisan, visual, video, dan lainnya, dapat diakses di manapun, kapanpun oleh siapa saja dengan dukungan teknologi internet (Rahmi et al., 2023). Selain itu, data dapat disimpan secara terpusat, sehingga jauh lebih mudah digunakan untuk berbagai tujuan. Seringkali, aplikasi *website* merupakan bentuk interaksi utama bagi pengguna untuk mengontrol data mereka dan aliran data melalui sistem untuk mendapatkan hasil yang diinginkan (Balsam & Mishra, 2025).

Berdasarkan penjelasan di atas, kesimpulannya *website* adalah kumpulan laman yang mengandung beragam jenis informasi seperti tulisan, visual, video, dan data lainnya, diakses melalui internet menggunakan peramban. *Website* memungkinkan pengguna untuk mendapatkan, mengatur, membagikan informasi dengan mudah dari manapun dan kapan saja, serta mendukung penyimpanan data yang terpusat untuk mempermudah pengelolaan dan interaksi antara pengguna dengan sistem.

4. Sistem Kredit Semester (SKS)

SKS atau Sistem Kredit Semester memberikan peluang bagi siswa untuk memilih mata pelajaran dan mengatur waktu studi, sehingga mereka dapat menentukan dan merencanakan cara belajar mereka sendiri. Pelaksanaan SKS memberikan kesempatan kepada siswa untuk belajar, meraih level kemampuan tertinggi menurut bakat, minat, dan ritme belajar mereka (Zainuri, 2022).

Pengaturan beban studi bagi siswa diatur dalam Permendikbud No. 81 A Tahun 2013, yang menjelaskan bahwa kurikulum di setiap satuan pendidikan disusun dalam bentuk paket serta sistem SKS. SKS diimplementasikan dengan cara mengolah pembelajaran secara beragam dengan penyediaan entitas pembelajaran lengkap untuk tiap mata pelajaran yang diikuti oleh siswa (Dewi et al., 2023).

Program SKS jelas berbeda dari sistem pendidikan konvensional (paket). Dalam sistem konvensional, semua siswa mendapatkan porsi yang sama dalam menjalani pendidikan sesuai dengan yang telah ditentukan oleh

sekolah. Sementara itu, SKS memberi kesempatan kepada siswa untuk memilih pelajaran sesuai dengan kemampuan mereka dalam satu semester (Jaenuri & Mulyani, 2024).

Berdasarkan penjelasan di atas, kesimpulannya Sistem Kredit Semester (SKS) ialah suatu metode pembelajaran yang memberikan siswa kebebasan untuk memilih mata pelajaran dan mengatur jumlah serta waktu penyelesaian studi sesuai kemampuan, potensi, minat, dan ritme belajar masing-masing. SKS memungkinkan proses belajar yang lebih beragam dan terencana, sehingga siswa dapat mengembangkan potensi mereka dan mencapai tingkat belajar yang optimal.

5. *Random Forest*

Untuk menghasilkan prediksi menggunakan *Random Forest*, sebuah data akan melewati semua pohon keputusan. Prediksi akhir dari *Random Forest* untuk data dilakukan melalui pemungutan suara mayoritas atau rata-rata, berdasarkan *output* dari keseluruhan pohon keputusan. Karena algoritma *Random Forest* memanfaatkan sampel *bootstrap* untuk mengembangkan setiap pohon keputusan, beberapa data tidak termasuk dalam pembentukan pohon tertentu. Dengan memperlakukan sampel sampel *out-of-bag* (OOB) ini sebagai data yang perlu diprediksi, maka hal itu dapat memberikan estimasi tentang tingkat kesalahan prediksi dari hutan yang telah dibuat (Hu & Szymczak, 2023).

Perbedaannya antara *Random Forest* dan metode ensemble pohon lainnya seperti *bagging* atau *boosting* adalah bahwa pohon-pohon dalam

Random Forest bersifat independen. Setiap pohon dipilih menggunakan sampel *bootstrap* dan di setiap node, sekumpulan prediktor acak dipertimbangkan untuk menentukan pembagian yang terbaik. Ini mengurangi korelasi antar pohon (Barreñada et al., 2024).

Salah satu kelebihan dari algoritma *random forest* adalah fleksibilitasnya. Metode ini dapat digunakan untuk kedua jenis masalah, yaitu regresi dan klasifikasi. Selain itu, algoritma ini memiliki tingkat keterbacaan yang baik, sehingga tidak sulit untuk dipahami. *Overfitting* adalah isu penting dalam *machine learning*. *Random forest* mengatasi masalah ini dengan mengkombinasikan beberapa pohon keputusan, sehingga menghasilkan pengurangan bias dan volatilitas (Salman et al., 2024).

Berdasarkan penjelasan di atas, kesimpulannya *Random Forest* merupakan algoritma *machine learning* yang menghasilkan prediksi dengan mengkombinasikan hasil dari beberapa pohon keputusan melalui mekanisme voting atau rata-rata. Setiap pohon dibangun secara independen menggunakan sampel *bootstrap* dan pemilihan fitur secara acak, sehingga mampu mengurangi korelasi antar pohon dan meminimalkan risiko *overfitting*. *Random Forest* dapat digunakan untuk klasifikasi maupun regresi, serta mampu menghasilkan model yang akurat dan stabil.

6. *Cross Validation*

Cross validation sering digunakan untuk evaluasi model *machine learning* yang tangguh. Data pelatihan dibagi menjadi beberapa bagian

sesuai dengan metode *Cross validation* yang dipilih. Model dilatih menggunakan semua bagian data kecuali satu bagian (fold pelatihan) dan diuji pada satu bagian yang tersisa (fold uji). Proses ini diulang hingga setiap bagian data digunakan sebagai fold uji sekali. Hasil evaluasi performa model diperoleh dari rata-rata skor yang dihasilkan, diberi bobot berdasarkan jumlah titik data di masing-masing fold uji (Sweet et al., 2023).

Cross validation digunakan untuk membantu memperoleh keseimbangan antara bias dan *variance* pada model. Keseimbangan tersebut dicapai dengan meminimalkan kedua komponen sehingga model mampu menghasilkan prediksi lebih baik pada data baru. Biasanya, peningkatan nilai bias biasanya diikuti oleh penurunan *variance*, dan sebaliknya (Allgaier & Pryss, 2024).

Pada metode *K-Fold Cross Validation*, data terbagi dalam K bagian yang memiliki ukuran relatif serupa. Tiap-tiap bagian bergantian digunakan untuk data pengujian, dan untuk K-1 bagian lainnya dipakai untuk data pelatihan. Tahap ini diulang sampai seluruh satuan pernah digunakan sebagai data pengujian. Nilai kesalahan prediksi kemudian diperoleh dari rata-rata hasil pengujian pada seluruh *fold* (Burzykowski et al., 2023).

Berdasarkan penjelasan di atas dapat disimpulkan *Cross Validation* merupakan metode untuk mengevaluasi performa model secara lebih akurat dengan memanfaatkan beberapa pembagian data latih dan uji. Metode ini membantu menghasilkan model yang lebih stabil, menekan risiko

overfitting, dan menaikkan kemampuan model dalam melakukan prediksi pada data yang belum pernah digunakan.

7. *Confusion Matrix*

Confusion Matrix adalah sebuah tabel persegi dengan ukuran $N \times N$, N merepresentasikan jumlah kategori *output*. Baris di tiap matriks menunjukkan jumlah *instance* dari kelas yang diprediksi, sementara tiap kolom menunjukkan jumlah *instance* dalam kelas yang sebenarnya. Tabel ini memberikan informasi rincian kelas demi kelas mengenai jumlah prediksi akurat dan tidak akurat yang dihasilkan oleh pengklasifikasi dalam tugas klasifikasi. Kategorisasi dapat berupa biner atau *multiclass* (Sathyanarayanan & Tantri, 2024).

Confusion Matrix (CM) diartikan sebagai sebuah matriks yang terdiri dari bilangan bulat positif sebagai elemennya, yang menggambarkan jumlah pasangan kombinasi label kelas yang diprediksi dan aktual. CM dapat diperoleh dengan menghitung berdasarkan pasangan dari himpunan kelas yang diprediksi dan yang aktual (Markoulidakis & Markoulidakis, 2024:3).

Confusion matrix merupakan alat yang biasa dipakai pada bidang data mining atau Sistem Pendukung Keputusan untuk menilai akurasi. Saat mengevaluasi kinerja dengan *confusion matrix*, hasil klasifikasi digolongkan ke empat kategori, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) (Wulandari & Sifaunajah, 2019:5).

		True Values	
		True	False
Prediction	True	TP Correct result	FP Unexpected result
	False	FN Missing result	TN Correct absence of result

Gambar 2. 1 *Confusion Matrix* Klasifikasi
Sumber : Wulandari & Sifaunajah (2019:5)

Berdasarkan penjelasan di atas, kesimpulannya *Confusion Matrix* merupakan sarana untuk menilai kinerja dan efektivitas model klasifikasi dengan cara mengomparasikan hasil prediksi dengan data yang sesungguhnya. Evaluasi dalam *confusion matrix* memanfaatkan empat elemen utama untuk menentukan seberapa tinggi akurasi dan kinerja model klasifikasi, baik dalam situasi biner maupun *multiclass*.

8. ROC-AUC (*Receiver Operating Characteristic – Area Under Curve*)

ROC merupakan metode yang umum digunakan untuk mengevaluasi dan membandingkan performa model prediksi melalui perhitungan AUC (*Area Under Curve*). Selain dihitung nilai AUC-nya, ROC juga disajikan dalam bentuk kurva untuk memudahkan interpretasi kemampuan model dalam membedakan kelas yang berbeda. Evaluasi ROC dapat dilakukan dengan menganalisis nilai AUC, sensitivitas rata-rata (*average sensitivity*), dan spesifisitas rata-rata (*average specificity*) yang digunakan untuk mengukur performa model klasifikasi (Carrington et al., 2023).

Nilai ROC-AUC diperoleh dari luas area di bawah kurva ROC yang dibentuk oleh hubungan antara sensitivitas dan spesifisitas. Perhitungan ini mempertimbangkan area yang menunjukkan tingkat sensitivitas dan spesifisitas yang memadai maupun area yang menunjukkan performa klasifikasi yang kurang memadai, sehingga dapat memberikan gambaran menyeluruh mengenai kemampuan model klasifikasi (Chicco & Jurman, 2023).

Selain kurva ROC, *Area Under the Curve* (AUC) juga merupakan metrik yang sering digunakan untuk mengevaluasi performa model *machine learning*. Nilai AUC memberikan satu nilai tunggal yang dapat digunakan untuk membandingkan kinerja beberapa model klasifikasi. Pada dataset yang memiliki ketidakseimbangan kelas yang tinggi (*imbalanced dataset*), metrik *Precision-Recall AUC* (PR-AUC) dapat digunakan karena lebih berfokus pada kemampuan model dalam mengklasifikasikan kelas minoritas (Imani & Arabnia, 2023).

Berdasarkan penjelasan di atas dapat disimpulkan ROC-AUC adalah metode evaluasi untuk mengukur kemampuan pada model klasifikasi dalam mengidentifikasi antar kelas. Evaluasi dilakukan melalui kurva ROC yang menggambarkan hubungan antara sensitivitas dan spesifisitas, sedangkan nilai AUC digunakan sebagai ukuran performa model secara keseluruhan. Semakin tinggi nilai AUC dan semakin mengarah ke angka 1, maka semakin efektif kemampuan model untuk klasifikasi.

9. *Database*

Database merupakan sekumpulan catatan atau informasi yang terorganisir dalam sistem komputer dan disusun dengan cara tertentu sehingga data tersebut dapat ditemukan dan diakses dengan mudah (Silalahi, 2022:2). Basis data juga diartikan sebagai sekumpulan data yang tersusun, disimpan, dan diatur dengan cara yang sistematis agar dapat diakses, diperbaharui, dan dikelola dengan mudah (Usman et al., 2025:1).

Basis data relasional merumuskan model data yang mendasari banyak aplikasi yang berbasis data, dan merupakan sistem pertama yang menawarkan pendekatan deklaratif tingkat tinggi untuk menentukan jenis data yang harus disimpan dan tindakan yang harus diambil. Secara internal terdiri dari implementasi yang kompleks dengan algoritma yang rumit dan berbagai teknik, basis data relasional menawarkan antarmuka yang sederhana, intuitif, dan berbasis tabel, yang umumnya dikenal oleh sebagian besar pengguna akhir melalui aplikasi *spreadsheet* (Panda, 2025:6).

Berdasarkan penjelasan di atas dapat disimpulkan *database* ialah himpunan informasi yang disusun dengan terorganisir di dalam sistem komputer, memungkinkan penyimpanan, pengelolaan, pembaruan, dan akses yang mudah. Salah satu tipe dari basis data ini adalah basis data relasional, yang menyusun informasi dalam format tabel dan memberikan pendekatan yang teratur serta mudah dipahami untuk mengelola data.

10. *Extreme Programming (XP)*

Extreme Programming bertujuan mengembangkan perangkat lunak dengan cara yang cepat dan efektif, prinsip-prinsipnya seperti pengujian berulang, integrasi yang sering serta respon cepat terhadap perubahan kebutuhan (Widarti et al., 2023:79). XP memberikan kesan sebagai sekumpulan ide yang klasik dan mudah, serta tidak memberikan dampak apapun terhadap proses pengembangan aplikasi (Suryantara, 2017:24).

Tiga prinsip penting yang diterapkan dalam XP untuk menghasilkan sistem yang berhasil adalah pengujian yang dilakukan secara rutin, pengkodean yang sederhana yang dilakukan oleh para pengembang, dan kolaborasi yang erat dengan pengguna untuk mempercepat pembangunan sistem. Pengujian serta praktik pengkodean yang efektif menjadi pusat dari XP. Kode diuji setiap hari dan diletakkan dalam lingkungan pengujian yang terintegrasi (Santoso & Migunani, 2021:15).

Berdasarkan penjelasan di atas dapat disimpulkan *Extreme Programming* merupakan metode pengembangan perangkat lunak yang mengutamakan kecepatan, efisiensi, dan kemampuan untuk menyesuaikan diri dengan perubahan kebutuhan. Prinsip utama XP mencakup pengujian yang dilakukan secara terus-menerus, integrasi sistem yang dilakukan secara berkala, penulisan kode yang sederhana oleh para pengembang, serta kerja sama yang erat dengan pengguna.

11. *Black Box Testing*

Black box testing merupakan metode yang memungkinkan seorang *software engineer* dan penguji untuk memperoleh berbagai kondisi masukan yang sepenuhnya berdasarkan semua spesifikasi fungsional dari suatu aplikasi (Pratama et al., 2023). Walaupun pengujian ini dapat menemukan kelemahan dalam sistem, mungkin pengujian ini tidak bisa mendeteksi semua kesalahan, terutama yang berkaitan dengan struktur kode yang ada di dalam (Maspupah, 2024).

Pengujian *black-box* berhubungan dengan evaluasi yang dilakukan pada antarmuka aplikasi perangkat lunak. Metode ini digunakan untuk menunjukkan bahwa fitur-fitur perangkat lunak dapat berjalan dengan baik, *input* diterima dengan benar dan menghasilkan *output* yang akurat, serta integritas informasi *eksternal* (Sari, 2021:188).

Berdasarkan definisi di atas, kesimpulannya *black box testing* ialah metode pengujian perangkat lunak yang menekankan pada fungsi dan spesifikasi aplikasi, tanpa mempertimbangkan struktur atau kode yang mendasarinya. Metode ini dilakukan dengan memberikan berbagai jenis input untuk memvalidasi sistem mampu menerima masukan dengan tepat serta menghasilkan keluaran yang sesuai.

12. *System Usability Scale (SUS)*

Usability testing membantu mengidentifikasi permasalahan dan kekurangan yang terdapat pada suatu produk. Pengujian ini memberikan informasi mengenai pengalaman pengguna saat berinteraksi dengan sistem,

serta membantu memahami kebutuhan dan harapan pengguna terhadap pengembangan sistem di masa mendatang (Mulia et al., 2023)

SUS menggunakan kuesioner yang mengukur tiga aspek penting dalam penggunaan teknologi. Aspek pertama adalah efektivitas, yang mengukur kemampuan pengguna dalam mencapai tujuan yang diinginkan saat menggunakan sistem. Aspek kedua adalah efisiensi, yaitu mengukur seberapa besar usaha dan sumber daya yang dikeluarkan pengguna untuk mencapai tujuan tersebut. Aspek ketiga adalah kepuasan, yang menilai tingkat kepuasan pengguna terhadap pengalaman mereka selama menggunakan sistem (Cheah et al., 2022).

System Usability Scale (SUS) dapat digunakan sebagai metode yang cepat untuk mengukur persepsi pengguna terhadap tingkat kegunaan suatu sistem komputer. Metode ini telah terbukti sebagai alat evaluasi usability yang andal dan efektif dalam menilai serta membandingkan tingkat kegunaan antar sistem (Dyayu et al., 2023).

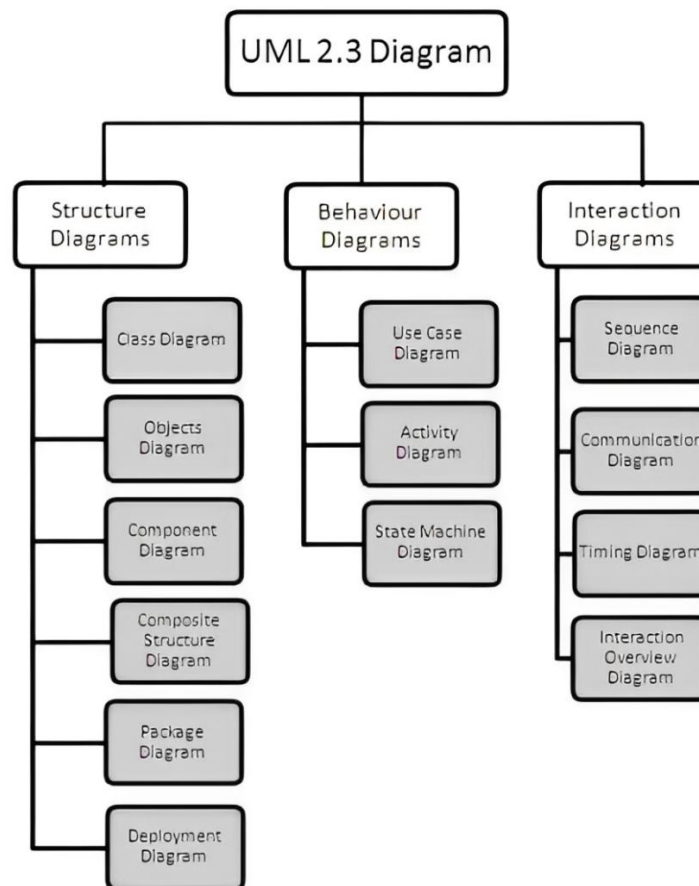
Berdasarkan definisi di atas, *System Usability Scale* (SUS) merupakan metode pengujian *usability* yang digunakan untuk mengukur tingkat kegunaan suatu sistem berdasarkan persepsi pengguna secara cepat dan andal. Melalui kuesioner SUS, dapat dievaluasi tiga aspek penting, yaitu efektivitas, efisiensi, dan kepuasan pengguna dalam menggunakan sistem serta kebutuhan dan harapan mereka terhadap pengembangan sistem di masa mendatang.

13. *Unified Modeling Language* (UML)

Unified Modeling Language adalah bahasa visual gambar/grafik untuk menggambarkan, mendefinisikan, mengembangkan, dan mendokumentasikan dari sebuah sistem dalam *software development* yang berorientasi OO (*Object-Oriented*) tentang struktur dan perilaku sistem (Sumirat, 2023:73).

UML merupakan bahasa standar yang umum dipakai dalam ranah industri untuk menjelaskan kebutuhan, melakukan analisis dan perancangan, menggambarkan struktur pemrograman berorientasi objek. UML lahir karena kebutuhan akan pemodelan visual yang membantu menetapkan, memaparkan, mengembangkan, dan mendokumentasikan sistem perangkat lunak (Hasanah & Untari, 2020:64).

UML (*Unified Modeling Language*) merupakan bahasa yang dapat dimanfaatkan untuk menganalisis dan merancang desain berbasis objek. Pengembangan UML disusun berdasarkan kebutuhan model visual untuk mengatur spesifikasi, menjelaskan, membangun, dan mendokumentasikan sistem perangkat lunak (Andriyanto, 2022:2).



Gambar 2. 2 Bagan UML
Sumber : Hasanah & Untari (2020:66)

Berdasarkan definisi di atas, kesimpulannya UML adalah suatu cara visual untuk memodelkan dalam pembuatan perangkat lunak berbasis objek. UML memiliki tujuan untuk mempresentasikan, menetapkan, merancang, dan mencatat sistem dengan cara yang teratur dan sistemik.

14. Flowchart

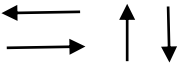
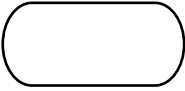
Flowchart merupakan representasi grafis pada tahapan atau urutan tindakan dalam program yang sangat berpengaruh dalam menyelesaikan masalah tertentu yang membutuhkan analisis dan pemahaman yang terus-menerus. Diagram alur dimanfaatkan untuk menunjukkan kegiatan baik

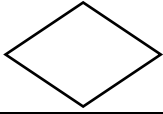
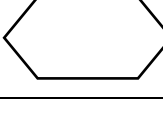
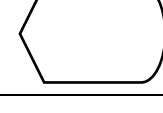
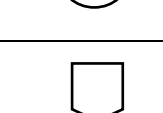
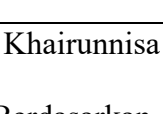
yang dilakukan secara manual maupun dalam proses pengolahan (Elfaladonna & Isa, 2024:15).

Flowchart berfungsi untuk menggambarkan keseluruhan proses dengan cara yang jelas, sehingga lebih mudah menemukan masalah atau kesalahan dalam proses tersebut, serta membantu merancang atau memperbaiki alur kerja. Selain itu, flowchart juga bisa digunakan untuk menjelaskan proses secara langsung dan mudah dimengerti oleh semua individu yang terlibat (Rusli & Sany, 2023:12).

Setiap flowchart terhubung dengan suatu proses atau sistem yang spesifik. Proses ini dimulai dengan memasukkan data atau bahan ke dalam sistem dan mengikuti semua langkah yang diperlukan untuk mengonversi input menjadi output akhir. Simbol-simbol flowchart khusus menggambarkan proses yang berlangsung, tindakan yang diambil di masing-masing tahap, serta hubungan antara berbagai tahap tersebut (Khairunnisa et al., 2023:87).

Tabel 2. 1 Simbol Flowchart

No	Bagan	Nama	Fungsi
1		<i>Flow</i>	Alur dari program
2		<i>Terminator</i>	Memulai / mengakhiri sistem
3		<i>Proses</i>	Menyatakan proses
4		<i>Input / Output</i>	<i>Input / Output</i> data

No	Bagan	Nama	Fungsi
5		<i>Decision</i>	Menyatakan suatu kondisi
6		<i>Storage Data</i>	Menyatakan <i>input</i> dari disk atau disimpan ke disk
7		<i>Preparation</i>	Inisialisasi / pemberian nilai awal
8		<i>Display</i>	Menyatakan alat <i>output</i>
9		<i>Document</i>	<i>Input</i> / <i>Output</i> berbentuk dokumen
10		<i>Manual Operation</i>	Proses manual
11		<i>On Page Connector</i>	Sambungan proses di laman yang sama
12		<i>Off Page Connector</i>	Sambungan proses ke laman berbeda

Sumber : Khairunnisa et al., (2023:90)

Berdasarkan definisi di atas, kesimpulannya *flowchart* adalah diagram untuk menunjukkan aliran atau urutan tahapan dalam sebuah proses atau sistem secara visual. Dengan adanya simbol-simbol tertentu, flowchart membantu untuk menjelaskan keterkaitan antara langkah-langkah dari masukan sampai menghasilkan keluaran.

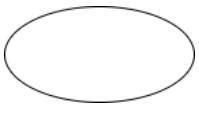
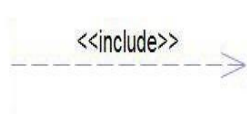
15. Use Case Diagram

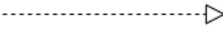
Secara umum, *use case* disusun di awal proses pengembangan perangkat lunak atau sistem. *Use Case* menjelaskan jenis-jenis aktivitas

yang terjadi di sistem dan siapa memiliki akses untuk mengakses fitur-fitur dalam sistem. *Use Case* memberikan ilustrasi tentang hubungan interaksi antara sistem dan pengguna Niqotaini et al., (2023:1).

Use case diagram adalah representasi visual yang menunjukkan posisi pengguna serta cara kerjanya dalam sistem. *Use case* diagram juga mampu memperlihatkan hubungan antara pengguna dan sistem serta menjelaskan rincian kasus penggunaan (Destriana et al., 2022:5). *Use case* diagram memvisualisasikan berbagai fitur yang diharapkan ada dalam sistem. Yang menjadi fokus adalah apa yang dilakukan oleh sistem, bukan bagaimana cara melakukannya. Tiap kasus penggunaan menggambarkan hubungan antara aktor dan sistem (Habibi et al., 2020:50).

Tabel 2. 2 Simbol *Use Case* Diagram

No	Nama	Simbol	Keterangan
1	<i>Actor</i>		Mewakili pengguna dalam berinteraksi dengan sistem.
2	<i>Usecase</i>		Fitur atau tindakan yang ditunjukkan oleh sistem sebagai unit yang berkomunikasi satu sama lain, termasuk aktor. <i>Usecase</i> dinamai atau dijelaskan dengan kata kerja.
3	<i>Include</i>		Menghubungkan <i>usecase</i> tambahan yang menuju <i>usecase</i> yang ditambahkan. <i>Usecase</i> yang ditambahkan akan memerlukan <i>usecase</i> ini agar bisa berfungsi atau menjadi syarat untuk menjalankan fungsinya. Arah pada panah <i>include</i> merujuk pada <i>usecase</i> tambahan.

No	Nama	Simbol	Keterangan
4	<i>Extend</i>		Tambahan fitur dari <i>use case</i> utama yang hanya berjalan pada kondisi tertentu.
5	<i>Association</i>		Penghubung <i>actor</i> dengan <i>use case</i> yang dapat diakses atau jalankan.
6	<i>Generalization</i>		Sebagai penghubung <i>actor</i> atau <i>use case</i> yang memiliki kesamaan sifat, sebagai pewarisan atas fungsi/peran yang lebih umum.
7	<i>Sistem</i>		Menentukan paket yang menggambarkan sistem secara terbatas.

Sumber : Niqotaini et al., (2023:1-3)

Berdasarkan definisi di atas, kesimpulannya *use case* diagram adalah jenis diagram yang diterapkan pada fase awal pembangunan sistem untuk menunjukkan berbagai fungsi yang ada dalam sistem, serta interaksi antara pengguna dengan sistem. Diagram ini menitikberatkan pada kemampuan sistem, bukan cara pelaksanaannya, sehingga membantu untuk memahami hubungan antara pengguna dan sistem serta memudahkan untuk mengenali kebutuhan dan fitur yang diinginkan dari sistem.

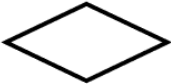
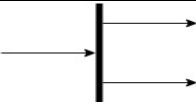
16. Activity Diagram

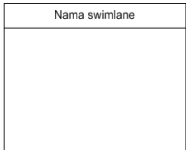
Activity diagram merupakan gambaran yang memperlihatkan aliran kerja dari bermacam kegiatan pengguna atau sistem, individu yang terlibat dalam kegiatan tersebut, serta urutan aliran dari kegiatan-kegiatan ini

(Destriana et al., 2022:5). *Activity* diagram menunjukkan banyak jalur aktifitas dalam sistem yang sedang dibuat, cara setiap jalur dimulai, potensi keputusan yang dapat muncul, serta bagaimana jalur tersebut berakhir. *Activity* diagram menggambarkan tahapan yang berlangsung bersamaan yang mungkin terjadi dalam beberapa pelaksanaan (Hasanah & Untari, 2020:79).

Fungsi dari *activity* diagram adalah untuk menunjukkan urutan kegiatan yang berlangsung dalam sebuah sistem dan memberikan informasi rinci mengenai proses tersebut. Seluruh diagram aktivitas dapat dibangun berdasarkan satu atau beberapa *use case*. Diagram ini memperjelas alur aktivitas dalam diagram *use case* yang telah dibuat sebelumnya (Niqotaini et al., 2023:19).

Tabel 2. 3 Simbol *Activity* Diagram

No	Nama	Simbol	Keterangan
1	Status Awal / <i>Initial state</i>		Titik mula dalam diagram aktivitas.
2	Aktivitas/ <i>Activity</i>		Interaksi antar kelas antarmuka.
3	Percabangan / <i>Decision</i>		Proses untuk memilih pada aktivitas yang lebih dari satu jenis.
4	Penggabungan / <i>Join</i>		Beberapa kegiatan digabungkan jadi satu.
5	Status Akhir / <i>Final State</i>		Titik akhir atau status akhir.

No	Nama	Simbol	Keterangan
6	Swimlane		Pemisahan untuk organisasi yang bertanggung jawab atas aktivitas.

Sumber : Niqotaini et al., (2023:20-21)

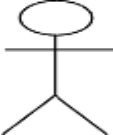
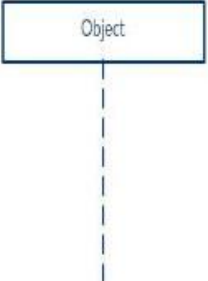
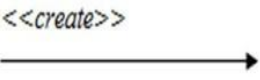
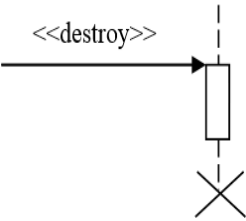
Berdasarkan definisi di atas, kesimpulanya *activity* diagram ialah tipe diagram untuk memperlihatkan serangkaian proses kegiatan yang sedang berlangsung pada sebuah sistem, baik oleh pengguna maupun oleh sistem. Diagram ini menggambarkan alur kegiatan, kemungkinan keputusan yang diambil, serta cara suatu proses dimulai dan diakhiri.

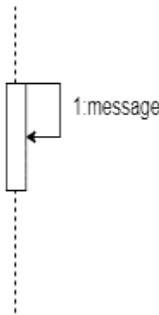
17. *Sequence* Diagram

Sequence diagram menerangkan aktivitas yang dikerjakan oleh objek dalam *use case* diagram dengan menunjukkan tahap hidup objek serta pesan dikirim atau diterima dari objek lainnya. Diagram menggambarkan keterkaitan antar objek-objek dan sistem di sekitarnya, terhitung pengguna serta antarmuka (Niqotaini et al., 2023:28).

Sequence Diagram memvisualisasikan hubungan antara berbagai objek berdasarkan waktu. Tujuannya untuk menampilkan rangkaian pesan yang saling dikirim antar berbagai objek dan interaksi antar objek pada titik tertentu dalam pelaksanaan sistem (Siregar, 2025:121). Fungsi lain dari *Sequence* Diagram adalah untuk menekankan pengenalan metode dalam sistem. *Sequence* Diagram dipakai dalam menjelaskan dan memodelkan *use case* diagram (Sutrisno & Karnadi, 2021).

Tabel 2. 4 Simbol *Sequence Diagram*

No	Nama	Simbol	Keterangan
1	<i>Actor</i>		Mewakili pengguna dalam berinteraksi dengan sistem.
2	Garis Hidup(<i>Lifeline</i>)		Menggambarkan eksistensi dari suatu objek.
3	<i>Object</i>		Menggambarkan objek yang saling bertukar pesan.
4	Pesan tipe <i>create</i>		Menggambarkan objek yang menciptakan sesuatu, dengan panah menunjuk pada objek yang dibuat.
5	Pesan bertipe <i>destroy</i>		Menggambarkan objek yang mengakhiri eksistensi objek lain.

No	Nama	Simbol	Keterangan
6	Pesan bertipe <i>return</i>		Menggambarkan bahwa suatu objek yang telah melaksanakan metode atau operasi tertentu mengembalikan sesuatu kepada objek yang spesifik, dengan arah panah menunjukkan objek yang menerima kembalian tersebut.
7	Pesan bertipe <i>self</i>		Komponen panah garis putus-putus untuk menjelaskan hubungan antar objek-objek itu sendiri.

Sumber : Niqotaini et al., (2023:30-31)

Berdasarkan penjelasan di atas, kesimpulannya *sequence* diagram adalah tipe diagram yang memvisualisasikan bagaimana objek berinteraksi di dalam suatu sistem sesuai dengan urutan waktu. Diagram ini menampilkan serangkaian pesan yang dikirimkan dan diterima antara objek selama proses berlangsungnya, serta menunjukkan keterkaitan antar objek, pengguna, dan antarmuka sistem.

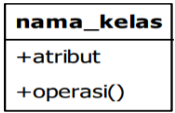
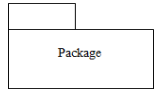
18. Class Diagram

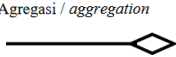
Kelas dan paket yang terdapat dalam sebuah perangkat lunak digambarkan melalui diagram yang disebut sebagai *class* diagram. Representasi secara keseluruhan dari sistem, termasuk kelas dan relasinya, ditampilkan dalam sebuah class diagram. Karakteristik diagram kelas ini bersifat tetap dan menunjukkan hubungan yang terjadi (Niqotaini et al., 2023:38).

Class diagram menggambarkan konfigurasi sistem terkait dengan kelas, atribut, dan batasan yang diterapkan pada kelas (operasi) serta hubungan di antara kelas - kelas tersebut. Diagram kelas dimanfaatkan pada tahap analisis untuk memberikan gambaran tentang entitas statis dalam area permasalahan, dan dalam tahap desain untuk menyajikan gambaran tentang entitas statis (pengklasifikasi) dalam area solusi (Shah & Grant, 2022).

Serangkaian pedoman untuk kelas dan relasi ditetapkan untuk menangani berbagai jenis kalimat demi mengekstrak dengan tepat berbagai komponen *class* diagram UML dari teks yang di tulis dalam bahasa alami. Dengan menganalisis struktur kalimat, pedoman-pedoman ini dapat mengenali kelas, atribut, dan relasinya, lalu mengaitkan elemen-elemen ini ke dalam model UML untuk menciptakan fragmen UML (Meng & Ban, 2024).

Tabel 2. 5 Simbol Class Diagram

No	Nama	Simbol	Keterangan
1	<i>Association</i>		Relasi satu objek dengan objek lain.
2	<i>Public Association</i>		Asosiasi yang mengaitkan lebih dari dua objek dapat dihindari.
3	<i>Class</i>		Kumpulan objek yang memiliki atribut dan operasi yang sama.
4	<i>Package</i>		Kemasan yang terdiri atas satu atau lebih kelas.

5	<i>Direct Association</i>		Hubungan antara kelas-kelas dimana satu kelas memakai kelas lainnya, biasanya juga disertai dengan <i>multiplicity</i>
6	Antar muka / <i>Interface</i>		<i>Interface</i> di pemrograman yang berorientasi objek
7	Generalisasi		Relasi antar kelas dengan konsep dari umum ke khusus
8	<i>Dependency /</i> Kebergantungan		Hubungan ketergantungan antara kelas-kelas.
9	<i>Aggregation /</i> Agrepgasi	 Agregasi / aggregation	Hubungan antar kelas yang menggambarkan makna seluruh bagian.

Sumber : Niqotaini et al., (2023:39-41)

Berdasarkan penjelasan di atas, kesimpulannya *class* diagram ialah diagram yang dimanfaatkan dalam menggambarkan struktur tetap dalam sistem yang mencakup kategori, variabel, fungsi, serta interaksi antara kelas. Diagram ini menyajikan pengertian yang lebih jelas tentang bagaimana cara komponen dalam sistem saling terhubung dan beroperasi.

B. Kajian Empiris

Penelitian ini disusun dengan menggunakan beberapa referensi jurnal yang relevan dengan topik yang diangkat, yaitu Sistem Prediksi Kelayakan Siswa Mengikuti Program SKS 2 Tahun Berbasis *Website* Menggunakan Algoritma *Random Forest*. Referensi ini dijadikan sebagai landasan teoritis dan acuan dalam perancangan sistem agar proses penelitian tersusun secara sistematis dan terarah. Studi literatur dari penelitian terdahulu digunakan untuk

memperkuat dan mematangkan rancangan sistem melalui proses perbandingan, analisis dan evaluasi dari sistem yang telah dikembangkan sebelumnya.

Penelitian sebelumnya yang dilakukan oleh Wakhid et al., (2025) dengan judul “Penerapan Metode SAW pada Sistem Seleksi Siswa Program Akselerasi MAN 2 Kediri” menerapkan metode *Simple Addictive Weighting* (SAW) untuk mengevaluasi dan memilih siswa berdasarkan nilai IQ, nilai semester, nilai ujian serta kepribadian siswa. Tahapan seleksi mencakup normalisasi data dan penentuan bobot untuk setiap kriteria, sehingga dapat menghasilkan peringkat siswa secara objektif dan transparan. Hasil dari penelitian ini menunjukkan bahwa sistem ini mempermudah proses pengambilan keputusan dalam menyeleksi siswa yang layak mengikuti kelas akselerasi dengan efisiensi dengan keakuratan yang lebih baik. Penggunaan metode SAW ini mampu memperbaiki kualitas seleksi melalui pendekatan berbasis data yang sistematis.

Pada penelitian yang dilakukan oleh Lenti et al., (2025) berjudul “*Simple Addictive Weighting* (SAW) Untuk Menentukan Penempatan Kelas Dalam Program Akselerasi Santri” menerapkan metode *Simple Additive Weighting* (SAW) untuk menentukan penempatan kelas bagi santri dalam program akselerasi di Pondok Pesantren. Metode ini memungkinkan penilaian yang lebih tepat berdasar kriteria yang jelas, seperti penguasaan ilmu nahwu, sharaf, kemahiran dalam membaca kitab kuning, menerjemahkan kitab kuning dalam bahasa Arab ke bahasa Indonesia. Hasil penelitian menunjukkan bahwa

penerapan metode SAW meningkatkan objektivitas dalam penempatan kelas serta memberikan transparansi dalam proses pengambilan keputusan.

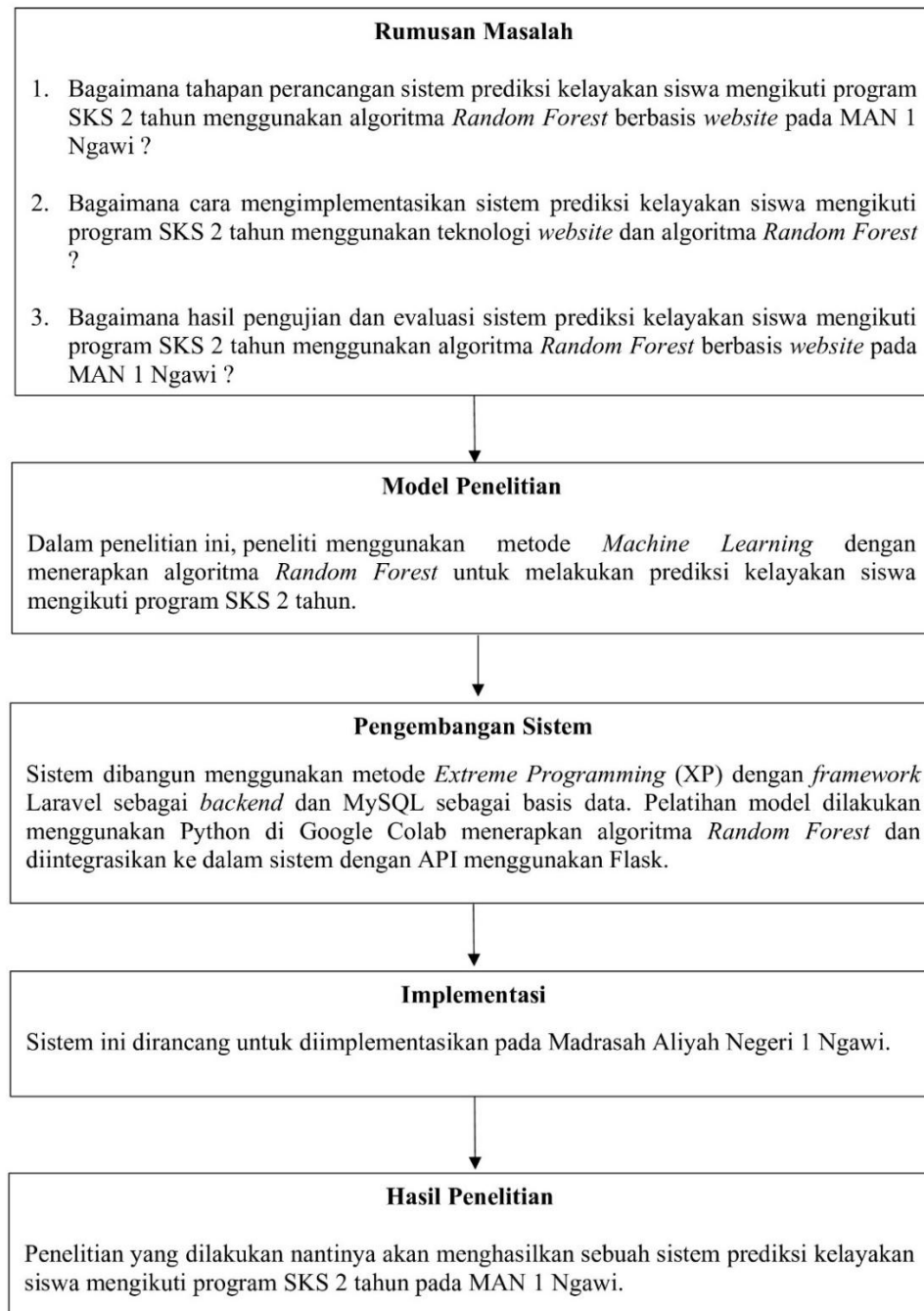
Penelitian oleh Amrullah & Subandi, (2024) dengan judul “Sistem Prediksi Prestasi Akademik Siswa Guna Mendukung Keputusan Pemberian Beasiswa Menggunakan Algoritma C4.5 di SMK Bina Informatika Bintaro” menggunakan algoritma C4.5 untuk membangun model pohon keputusan yang memiliki kemampuan untuk melakukan prediksi pada prestasi akademik siswa berdasarkan data historis. Hasil dari penelitian ini menunjukkan bahwa sistem prediksi prestasi akademik memberikan hasil nilai yang akurat dan penghitungan menunjukkan bahwa sistem ini memiliki kemampuan untuk memberikan prediksi yang cukup akurat serta dapat diandalkan.

Dalam penelitian lain oleh (Rosyad & Pramono, 2025) dengan judul “Rancang Bangun Sistem Pendukung Keputusan Berbasis *Website* Untuk Menentukan Siswa Berprestasi Menggunakan Metode *Simple Addictive Weighting*” menggunakan metode *Simple Additive Weighting* (SAW) untuk mengolah data yang melalui tahapan normalisasi matriks keputusan dan perhitungan nilai preferensi berdasarkan bobot kriteria. Hasil dari penelitian ini menunjukkan bahwa sistem dapat melakukan analisis performa akademik siswa secara terstruktur dan memberikan solusi otomatis yang objektif untuk membantu pihak sekolah menentukan siswa berprestasi berdasarkan data yang telah diinputkan.

Dari beberapa sumber literatur diatas dapat disimpulkan bahwa memilih metode yang sesuai sangat diperlukan dalam proses pembangunan

sistem. Pada proses pembangunan ini, peneliti menerapkan metode *Extreme Programming* dan algoritma *Random Forest* yang dinilai cocok untuk proses seleksi kelayakan siswa dengan batasan-batasan yang ada.

C. Kerangka Berpikir



Gambar 2. 3 Kerangka berpikir