

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Cerdas

Sistem Cerdas adalah ilmu pengetahuan yang bergerak dalam pembuatan mesin yang cerdas. Mesin yang cerdas dapat diartikan sebagai mesin yang dapat melakukan hal-hal tertentu yang memerlukan kecerdasan apabila dilakukan oleh manusia . Sistem pakar atau sebutan sistem cerdas dengan menerapkan beberapa metode, maka akan menghasilkan sistem yang dapat memberikan rekomendasi solusi terbaik . Sistem cerdas telah banyak diterapkan diberbagai lembaga maupun institusi, dengan penyajian serta mengembangkan *knowledge-based* dan penerapan rule yang ditawarkan oleh para ahli dapat memberikan solusi dalam pengambilan keputusan dengan cepat.(Akbar & Mukhtar, 2020)

2.2 Metode Pembelajaran

Metode dapat diartikan cara-cara atau langkah-langkah yang digunakan dalam menyampaikan suatu gagasan, pemikiran atau wawasan yang disusun secara sistematis dan terencana serta didasarkan pada teori, konsep, dan prinsip tertentu yang terdapat dalam berbagai disiplin ilmu terkait. Menurut Abdul Majid, metode adalah cara yang digunakan untuk mengimplementasikan rencana yang sudah disusun dalam kegiatan nyata agar tujuan yang telah disusun tercapai secara optimal. Sedangkan pembelajaran adalah suatu proses yang dilakukan individu untuk memperoleh suatu perubahan perilaku yang baru secara keseluruhan, sebagai hasil dari pengalaman individu sendiri dalam interaksi dengan lingkungannya.(Suparyanto dan Rosad (2015, 2020).

2.3 Algoritma Naive Bayes

Naive Bayes merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Algoritma menggunakan teorema *Bayes*

dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas. (Manalu et al., 2017)

Naive Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output. Dengan kata lain, diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas individu. Keuntungan penggunaan *Naive Bayes* adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (Training Data) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. *Naive Bayes* sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan. (Manalu et al., 2017).

Langkah-langkah algoritma *Naive Bayes* menurut Suntoro (2019), sebagai berikut:

1. Siapkan dataset.
2. Hitung probabilitas sebelumnya untuk label kelas yang diberikan
3. Temukan probabilitas Peluang dengan setiap atribut untuk setiap kelas
4. Masukkan nilai ini dalam Formula Bayes dan hitung probabilitas posterior.
5. Lihat kelas mana yang memiliki probabilitas lebih tinggi, mengingat input milik kelas probabilitas lebih tinggi.

Persamaan dari teorema Bayes adalah : (Manalu et al., 2017)

$$P(H|X) = \frac{P(H) \cdot \prod_{i=1}^n P(X_i|H)}{P(X)} \dots\dots\dots \text{persamaan (1)}$$

Di mana :

X : Data dengan class yang belum diketahui

H : Hipotesis data merupakan suatu class spesifik

$P(H|X)$: Probabilitas hipotesis H berdasar kondisi X (posteriori probabilitas)

$P(H)$: Probabilitas hipotesis H (prior probabilitas)

$P(X|H)$: Probabilitas X berdasarkan kondisi pada hipotesis H

$P(X)$: Probabilitas X

Keuntungan *Naive Bayes* adalah bahwa metode ini dapat membutuhkan jumlah data pelatihan (Training Data) yang kecil dengan menentukan estimasi parameter yang diperlukan untuk proses pengklasifikasian. *Naive Bayes* ialah metode

pengklasifikasian probabilistik yang sederhana. Untuk klasifikasi *Bayes* sederhana yaitu dikenal dengan *naïve Bayesian Classifier* yang dapat diasumsikan bahwa efek di suatu nilai atribut di kelas yang diberikan ialah bebas unruk atribut-atribut lain. Kelebihan *Naive Bayes* ialah hanya memerlukan sejumlah kecil data latih agar menentukan parameter mean yang varians dari variabel agar diperlukan untuk klasifikasi. *Naive Bayes* adalah metode supervised document classification agar dapat membutuhkan data training sebelum melakukan proses klasifikasi. (Lestari et al., 2022)

2.4 Aplikasi

Pengertian aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi *user* dengan menggunakan kemampuan komputer. (aF. Listianto, F. Fauzi, R. Irviani, K. Kasmi, 2017)

Sedangkan menurut (Susanti & Haevi, 2018) Aplikasi merupakan suatu perangkat lunak yang mempermudah pemakai dalam menyelesaikan tugasnya sehingga adanya proses input menjadi *output*.

2.5 Klasifikasi Data

Klasifikasi data adalah proses mengasosiasikan karakteristik metadata ke setiap aset di kawasan digital, yang mengidentifikasi jenis data yang terkait dengan aset tersebut. Klasifikasi data sendiri adalah memiliki berbagai macam jenis yang dibedakan menjadi 5 jenis, mulai dari klasifikasi data berdasarkan sumber data, cara memperolehnya, waktu pengumpulan, jenis (jenis data primer dan sekunder misalnya), dan sifat data (Kristanto, 2018).

2.6 Website

Website atau disingkat *web*, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio dan animasi lainnya yang disediakan melalui jalur koneksi internet. (Christian et al., 2018)

Website merupakan kumpulan dari halaman-halaman situs yang terangkum dalam sebuah domain atau subdomain pada world wide web (www). Biasanya sebuah website terdiri dari dokumen yang ditulis dalam format HTML (Hyper Text Markup Language). Halaman web dapat memanggil elemen-elemen lain berupa teks, gambar, video, dan sebagainya. Dengan adanya website informasi dari sebuah organisasi atau kelompok dapat dibaca dan dilihat oleh user lain yang ingin mengaksenya. Sehingga tanpa menghubungi pihak terkait secara langsung mereka dapat mengetahui informasi yang ingin didapatkan.(Veri et al., 2021).

2.7 Unified Modeling Language (UML)

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML adalah salah satu standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.(Hendrastuty et al., 2021). Perancangan Sistem Terdapat beberapa bagian dalam perancangan UML (*Unified Modelling Language*) yaitu sebagai berikut:

2.7.1 Use Case Diagram

Use Case diagram merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya. Selanjutnya, *Use Case* tidak hanya sangat penting pada tahap analisis, tetapi juga sangat penting untuk perancangan, untuk mencari kelas-kelas yang terlibat dalam aplikasi, serta untuk melakukan pengujian (Arfida & Wibowo, 2018) *Use case diagram* memiliki beberapa simbol sebagai berikut:

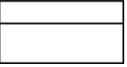
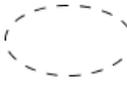
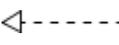
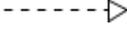
Tabel 2. 1 Simbol *Usecase Diagram*

Simbol	Keterangan
Use Case 	Menggambarkan dari urutan urutan aksi yang akan ditampilkan sistem dan menghasilkan suatu hasil bagi suatu aktor
Actor 	Mendeskripsikan peran yang pengguna mainkan saat berinteraksi dengan <i>use case</i>
Asosiasi 	Mendeskripsikan relasi antara <i>use case</i> dengan aktor dalam proses berbasis komputer
Include 	Suatu <i>use case</i> memungkinkan untuk menggunakan fungsional yang disediakan oleh <i>use case</i> lainnya
Extend 	Memungkinkan hubungan <i>use case</i> berpotensi untuk memperluas fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.

2.7.2 *Class Diagram*

Class diagram merupakan spesifikasi yang menghasilkan objek ketika dipakai, dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan sistem (properti/atribut), dan layanan yang menyediakan status operasional (metode/fungsi). Class diagram menggambarkan struktur dan deskripsi class, *package*, dan hubungan di antara mereka, seperti *containment*, pewarisan, asosiasi, dan lain lain. Diagram Kelas (Class Diagram) juga merupakan hubungan antar kelas dan penjelasan detail tiaptiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. (Irawan & Simargolang, 2018)

Tabel 2. 2 *Class Diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Generalization</i>	Relasi dimana objek anak(<i>descendent</i>) Berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>NaryAssociation</i>	Untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Menggambarkan dari urutan urutan aksi yang akan sistem tampilkan dan menghasilkan suatu hasil yang terukur bagi suatu actor
	<i>Realization</i>	Suatu objek yang benar benar melakukan operasi
	<i>Dependency</i>	Relasi perubahan terjadi dalam suatu elemen mandiri (<i>independent</i>) dan mempegaruhi elemen yang bergantung pada elemen yang tidak mandiri
	<i>Association</i>	Hubungan antara objek satu dengan objek lainnya

2.7.3 Activiy Diagram

Activity diagram mendeskripsikan macam macam alur aktivitas dalam sistem yang sedang dirancang, bagaimana alur berawal masing-masing, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat mendreskripsikan proses paralel yang mungkin terjadi pada beberapa eksekusi perintah. *Activity* Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. (Irawan & Simargolang, 2018)

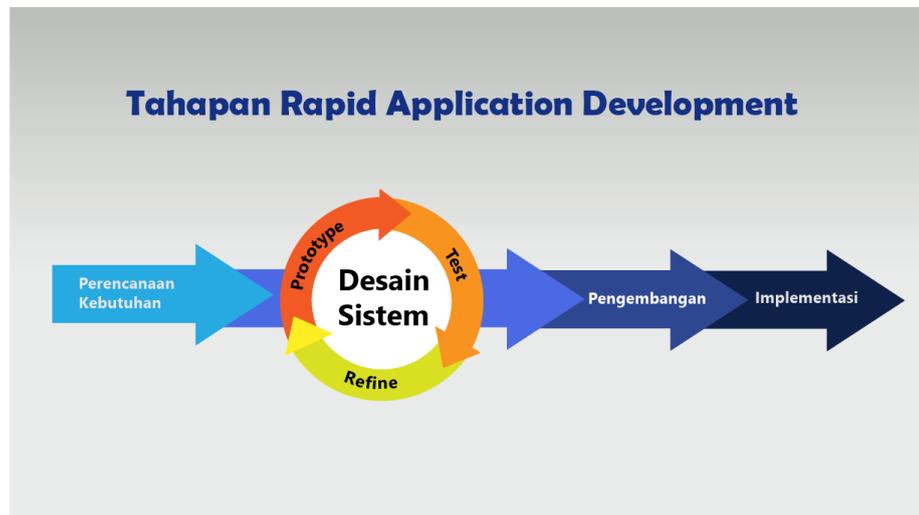
Tabel 2. 3 *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Activity</i>	Mendeskripsikan bagaimana masing-masing kelas antarmuka saling berinteraksi satusamalah.
	<i>InitialNode</i>	Bagaimana objek dibentuk atau diawali.
	<i>ActivityFinal Node</i>	Bagaimana objek dibentuk dan diakhiri.
	<i>ForkNode</i>	Satu aliran yang padatahap tertentu berubah menjadi beberapa aliran.

2.8 Metode Pengembangan Sistem

RAD (*Rapid Application Development*), yang merupakan metode Prototyping yang melibatkan pengguna dalam pengembangan aplikasi secara terus-menerus, hingga sistem baru dipresentasikan. Kelebihan utama dari metode RAD adalah mempermudah proses integrasi dan waktu pengembangan aplikasi bisa lebih cepat dan efektif. Langkah langkah metode *RAD* secara sebagai

berikut:



Gambar 2. 1 Tahapan Metode *Rapid Application Development*

Tahapan proses proses dari metode RAD sebagai berikut:

1. Perencanaan Kebutuhan.

Fase ini merupakan fase awal pengembangan sistem dimana dilakukan identifikasi masalah dan pengumpulan data dari pengguna dengan tujuan untuk menentukan maksud atau tujuan akhir dari sistem dan kebutuhan informasi yang diperlukan. Pada tahap ini, partisipasi kedua belah pihak sangat penting untuk menentukan kebutuhan pengembangan sistem.

2. Desain Sistem.

Pada tahap perancangan sistem, motivasi pengguna yang terlibat sangat penting untuk mencapai tujuan, karena pada tahap ini dilakukan proses perancangan dan proses perbaikan desain jika masih terdapat proses perancangan yang tidak sesuai dengan kebutuhan pengguna pada tahap sebelumnya. Keluaran dari fase ini adalah spesifikasi perangkat lunak, yang mencakup organisasi dalam sistem secara umum, struktur data, dan lain-lain.

3. Proses pengembangan dan pengumpulan feedback.

Selama fase ini, desain sistem yang dikembangkan dan disetujui diubah menjadi versi beta aplikasi, yang mengarah ke rilis final. Selama fase ini, programmer harus melanjutkan aktivitas pengembangan dan berintegrasi

dengan bagian lain, sambil terus mempertimbangkan umpan balik dari pengguna atau pelanggan. Jika prosesnya berjalan dengan baik, maka dapat melanjutkan ke tahap berikutnya, sedangkan jika aplikasi yang dikembangkan tidak memenuhi persyaratan, programmer kembali ke tahap desain sistem.

4. Implementasi atau penyelesaian produk.

Fase ini merupakan fase dimana programmer mengimplementasikan rancangan sistem yang telah disepakati pada fase sebelumnya. Sebelum mengimplementasikan sistem, program terlebih dahulu melalui proses pengujian, dimana ditemukan kesalahan dalam pengembangan sistem. Pada tahap ini biasanya dilakukan pemberian umpan balik terhadap sistem yang dibuat dan mendapatkan sistem yang disetujui.

2.9 Pengujian *Black Box*

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program (Herdiansah et al., 2020).

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

- a. Fungsi – fungsi yang tidak benar atau hilang,
- b. Kesalahan interface
- c. Kesalahan dalam struktur data atau akses eksternal
- d. Kesalahan kinerja
- e. Inisialisasi dan kesalahan terminasi

Mengemukakan ciri-ciri *black box testing*, diantaranya sebagai berikut:

1. *Black box* testing berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
2. *Black box* testing bukan teknik alternatif daripada *white box testing*. Lebih dari pada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.

3. *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur *internal* dari *sistem* atau komponen yang dites. Juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*.

2.10 Penelitian Terdahulu

Berikut adalah penelitian terdahulu yang menjadi inspirasi dan referensi bagi peneliti dalam penelitian ini.

Tabel 2. 4 penelitian terdahulu

Nama	Judul	Terbit	Metode /Algoritma	Uraian
Teddy Pratama, Yulmaini	Implementasi Algoritma Naïve Bayes Untuk Menentukan Konsentrasi Skripsi dan Rekomendasi Bahasa Pemrograman	2018	Algoritma Naïve Bayes	Algoritma Naive Bayes dapat digunakan untuk menentukan Konsentrasi Skripsi dan Rekomendasi Bahasa Pemrograman sehingga dapat menentukan konsentrasi skripsi dan Bahasa pemrograman yang diambil oleh mahasiswa pada saat melakukan penelitian Skripsi.
Patmi Kasih , Intan Nur Farida	Sistem Bantu Pemilihan Dosen Pembimbing Tugas Akhir Berdasarkan Kategori Pilihan dan Keahlian Dosen menggunakan Naïve Bayes	2017	Naive Bayes	Sistem Bantu Pemilihan Dosen Pembimbing Tugas Akhir Berdasarkan Kategori Pilihan dan Keahlian Dosen Menggunakan Naïve Bayes telah berhasil dibuat dan berjalan dengan baik sesuai dengan hipotesa awal yaitu bisa memberikan rekomendasi dosen pembimbing di Prodi Teknik Informatika UN PGRI Kediri.
Rizaldi Akba Mukhtar	Perancangan E-Tracer Study berbasis Sistem Cerdas	2020	Metode Simple	penelitian ini berhasil merancang e-tracer study yang dibangun dengan framework codeigniter

Nama	Judul	Terbit	Metode /Algoritma	Uraian
			Additive Weighting	dan bahasa pemrograman pendukung terdiri dari HTML, CSS, Jquery, Java Script, JSON, AJAX, Bootstrap sebagai media dalam perancangan antar muka. Sedangkan PHP sebagai server side dan MySQL sebagai database. Pada proses perancangan dilakukan melalui tahapan rumusan masalah, literatur studi, pengumpulan datam analisis kebutuhan, desai system dan pengujian aplikasi. Pada proses pengujian menggunakan metode blackbox dan diukur dengan menggunakan metode usability test pada pengukuran 5 (lima) komponen terdiri dari; learnability, efficiency, memorability, errors, dan satisfaction yang didapatkan melalui penyebaran kuisisioner kepada mahasiswa dan alumni. e-tracer
Indria & Kasih	Perancangan Sistem Bantu Penentuan Metode Pembelajaran Terbaik Menggunakan Algoritma Weighted Product	2023	Weighted Product	Hasil dari penelitian ini adalah sebuah implementasi program rekomendasi metode pembelajaran. Kriteria yang digunakan adalah: Kemampuan Siswa, Motivasi Belajar, Situasi kelas, Fasilitas Belajar, Kemampuan Guru. Sedangkan alternatif hasil adalah: Metode Ceramah, Diskusi, Demonstrasi, Tanya Jawab, Eksperimen,

Nama	Judul	Terbit	Metode /Algoritma	Uraian
				Resitasi, Karyawisata, dan Inquiry. Dari hasil simulasi yang telah dilakukan, diperoleh hasil perbandingan yang pertama adalah alternatif metode ceramah, sehingga metode ini dapat digunakan sebagai rekomendasi bagi guru dalam menentukan metode pembelajaran terbaik yang dapat diterapkan ke siswa.
Rizky Prayudhi et al	Sistem Pendukung Keputusan Rekomendasi Penggunaan Media Pembelajaran Online Menggunakan Metode <i>Analytical Hierarchy Process</i> (Ahp)	2022	<i>Analytical Hierarchy Process</i>	Sistem pendukung keputusan rekomendasi ini, diharapkan bisa membantu dosen dalam menentukan media pembelajaran online yang efektif untuk digunakan dalam proses pembelajaran.