



BAB 2 LANDASAN TEORI

2.1 Penelitian Terdahulu

Beberapa penelitian sebelumnya yang membahas tentang sistem pakar dengan menggunakan metode *decision tree*. Penelitian yang pertama dilakukan oleh Maryam & Ariono (2022), yang berjudul “Sistem Pakar Pengklasifikasi Stadium Kanker Serviks Berbasis Mobile Menggunakan Metode *Decision Tree*”. Proses klasifikasi pada penelitian tersebut menggunakan data rekam medis kanker serviks sejumlah 200 *record*. Metode *decision tree* yang digunakan menunjukkan hasil nilai *accuracy* 85.50%, *recall* 85.40%, dan *precision* 86.74%. Bahasa pemrograman pada aplikasi ini yaitu Flutter. Hasil dari penelitian tersebut adalah pengembangan aplikasi *mobile* yang membantu masyarakat mendiagnosis tingkat stadium kanker serviks dengan mudah dan memberikan saran tindakan berdasarkan stadium yang dialami.

Penelitian kedua dilakukan oleh Santosa dkk. (2018) dengan judul “Implementasi Algoritma Decision Tree C.45 Untuk Diagnosa Penyakit *Tuberculosis* (TB)”. Penelitian ini menggunakan 100 data pasien dengan pembagian 50:50 untuk data *training* dan data *testing*. Hasil akurasi yang diperoleh sebesar 90% menggunakan metode *Confusion Matrix*. Penelitian ini menghasilkan sebuah sistem yang diharapkan dapat membantu masyarakat dalam memperoleh informasi yang cepat dan tepat untuk mendiagnosis penyakit Tuberkulosis.

Penelitian ketiga dilakukan oleh Wiyanto dkk. (2021), yang berjudul “Penerapan Sistem Pakar Berbasis Android Dengan Metode *Decision Tree* Untuk Memprediksi *Postpartum Haemorrhage* Pada Wanita Hamil”. Pada penelitian tersebut hasil klasifikasi dengan menggunakan algoritma *decision tree* berhasil mengelola *dataset* dan menghasilkan tiga



klasifikasi, yaitu PPH (*Postpartum Hemorrhage*), NO PPH, dan *Potential* PPH, yang sebelumnya hanya memiliki dua klasifikasi. Hasil pengujian menggunakan RapidMiner menunjukkan akurasi sebesar 93,91% pada data *training* dan 94,86% pada data *testing*. Penelitian ini menghasilkan sebuah sistem pakar berbasis android dengan menggunakan algoritma *decision tree*.

Penelitian keempat dilakukan oleh Hartanto (2023) dengan judul “Penerapan Algoritma *Decision Tree* Untuk Seleksi Penerima Beasiswa (Studi Kasus: SMPN 1 Soreang)”. Penelitian ini menggunakan data yang berisi nilai siswa sebanyak 237 orang. Dari proses *modelling* didapatkan nilai akurasi sebesar 92%. Hasil dari perhitungan menggunakan algoritma *decision tree* ini dibuat untuk memprediksi akurasi untuk keputusan seleksi penerimaan beasiswa di SMPN 1 Soreang.

Penelitian selanjutnya dilakukan oleh Akbar dkk. (2022), yang berjudul “Implementasi Algoritma *Decision Tree* C4.5 dan *Support Vector Regression* untuk Prediksi Penyakit Stroke”. Penelitian tersebut menggunakan 5110 *dataset* pasien penderita stroke yang diperoleh dari *kaggle open datasets*. Hasil penelitian ini mengidentifikasi pengidap penyakit stroke pada variabel yang didapati dan menganalisisnya memakai algoritma data mining *Decision Tree* C4.5 menghasilkan nilai *root mean square error* terhadap rasio 70 : 30 bernilai 0.235, sedangkan untuk algoritma *Support Vector Regression* terhadap rasio 70 : 30 bernilai 0.399.



Tabel 2.1 Penelitian Terdahulu

No	Peneliti, Tahun & Judul	Perbedaan	Persamaan	Hasil
1	Maryani dan Huan Wen Ariono, (2022), "Sistem Pakar Pengklasifikasi Stadium Kanker Serviks Berbasis Mobile Menggunakan Metode <i>Decision Tree</i> "	1. Menggunakan 200 rekam medik. 2. Platform <i>mobile</i> .	1. Metode <i>decision tree</i> .	Aplikasi <i>mobile</i> diagnosis tingkat stadium kanker serviks dengan nilai akurasi algoritma <i>decision tree</i> 85.50%, <i>recall</i> 85.40%, dan <i>precision</i> 86.74%.
2	Iwan Santosa, Hamidatur Rosiyah, dan Maza Rahmanita, (2019) "Implementasi Algoritma <i>Decision Tree</i> C.45 Untuk Diagnosis Penyakit <i>Tuberculosis</i> (TB)"	1. Menggunakan 100 data pasien.	1. Platform web. 2. Metode <i>decision tree</i> .	Sistem diagnosis penyakit <i>Tuberculosis</i> dengan akurasi sebesar 90% menggunakan metode <i>Confusion Matrix</i> .



Lanjutan Tabel 2.1 Penelitian Terdahulu

No	Penelitian, Tahun & Judul	Perbedaan	Persamaan	Hasil
3	Wiyanto, Mutiara Ihdin, Maulida, dan Sifa Fauziah, (2021), "Penerapan Sistem Pakar Berbasis Android Dengan Metode <i>Decision Tree</i> Untuk Memprediksi <i>Postpartum Haemorrhage</i> Pada Wanita Hamil"	1. Terdapat tiga klasifikasi. 2. Platform mobile.	1. Metode <i>decision tree</i> .	Sistem pakar berbasis Android prediksi <i>Postpartum Haemorrhage</i> dengan akurasi algoritma <i>decision tree</i> sebesar 93,91% pada data <i>training</i> dan 94,86% pada data <i>testing</i> .
4	Putri Adriani Hartanto, (2021) "Penerapan Algoritma <i>Decision Tree</i> Untuk Seleksi Penerimaan Beasiswa (Studi Kasus: SMPN 1 Soreang)"	1. Menggunakan data nilai siswa sebanyak 237 orang. 2. Terbatas hanya pada perhitungan.	1. Metode <i>decision tree</i> .	Prediksi akurasi untuk keputusan seleksi penerimaan beasiswa di SMPN 1 Soreang, dengan nilai akurasi sebesar 92%.



Lanjutan Tabel 2.1 Penelitian Terdahulu

No	Peneliti, Tahun & Judul	Perbedaan	Persamaan	Hasil
5	Firmansyah Akbar, Hanif Wira Saputra, Adhitya Karek Maulaya, Muhammad Fikri Hidayat, Rahmadden, (2022) "Implementasi Algoritma <i>Decision Tree</i> C4.5 dan <i>Support Vector Regression</i> untuk Prediksi Penyakit Stroke"	1. Menggunakan 5110 <i>open dataset</i> . 2. Terbatas pada perhitungan.	1. Metode <i>decision tree</i> .	Identifikasi pengidap stroke menggunakan algoritma <i>decision tree</i> C4.5 menghasilkan <i>root mean square error</i> sebesar 0.235 pada rasio data 70:30, sementara algoritma <i>Support Vector Regression</i> menghasilkan nilai 0.399 pada rasio yang sama.



2.2 Kajian Pustaka

2.2.1 Sistem Pakar

Menurut Hita (2020), sistem pakar (*expert system*) adalah program komputer yang dirancang untuk meniru kemampuan seorang ahli dalam menyelesaikan masalah tertentu. Ahli atau pakar adalah individu dengan pemahaman mendalam tentang suatu bidang, seperti dokter, penasihat keuangan, teknisi mesin mobil, atau spesialis komputer. Definisi ini sejalan dengan pandangan Koli dkk. (2022), yang menyatakan bahwa sistem pakar mereplikasi proses pemecahan masalah yang biasanya dilakukan oleh seorang ahli dan mengaplikasikannya pada komputer. Sistem ini dirancang untuk bidang pengetahuan tertentu, yang fokus pada kemampuan khusus yang menyerupai keahlian manusia dalam suatu aspek.

Sistem pakar tidak dimaksudkan untuk menggantikan peran seorang ahli atau pakar dengan komputer. Walaupun sistem ini dirancang dengan cara mengintegrasikan keahlian seorang pakar ke dalam sistem komputer, fungsinya hanya sebagai alat bantu untuk membantu manusia dalam menemukan solusi atas masalah yang dihadapi. Basis pengetahuan dalam sistem pakar diperoleh dari pengalaman seorang ahli atau teori-teori yang relevan pada bidang tertentu. Oleh karena itu, sistem pakar memiliki batasan dalam penerapannya, (Alam & Nurcahyo, 2022).

2.2.2 Stres

Stres adalah kondisi yang dialami seseorang ketika terdapat ketidaksesuaian antara tuntutan yang dihadapi dan kemampuan untuk mengatasinya, (Seto dkk., 2020). Pada mahasiswa, tingkat stres memiliki hubungan yang signifikan dengan motivasi belajar mereka



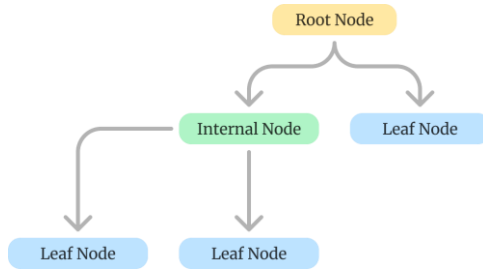
dalam kehidupan kampus. Kusumah dkk. (2022) mendefinisikan stres sebagai gangguan mental yang muncul akibat tekanan. Penelitian tersebut juga menunjukkan bahwa stres dapat berdampak negatif, seperti menurunnya kemampuan akademik, memburuknya kondisi kesehatan, munculnya depresi, serta gangguan pada pola tidur.

2.2.3 Decision Tree

Algoritma *Decision Tree* merupakan metode klasifikasi yang memanfaatkan data untuk menemukan hubungan antara variabel target dan variabel input, (Hartanto, 2023). Pohon keputusan dikenal sebagai metode yang kuat dan populer dalam klasifikasi prediktif atau pengelompokan, (Akbar dkk., 2022). Algoritma ini memiliki dua keunggulan utama: pertama, konsepnya yang jelas dan mudah dipahami, terutama karena pohon keputusan yang dihasilkan dari proses pelatihan dapat menjelaskan cara kerja model klasifikasi data; kedua, algoritma ini mudah diimplementasikan karena menggunakan pendekatan rekursif, (Wiyanto dkk., 2021).

Menurut Sijabat (2015), *decision tree* memiliki 3 jenis *node*, yaitu:

1. *Root Node* adalah *node* yang berada di posisi paling atas. *Node* ini tidak menerima *input*, dan bisa tidak memiliki *output* atau memiliki lebih dari satu *output*.
2. *Internal Node* adalah *node* percabangan yang memiliki satu *input* dan minimal dua *output*.
3. *Leaf Node* adalah *node* akhir yang memiliki satu *input* dan tidak memiliki *output*.



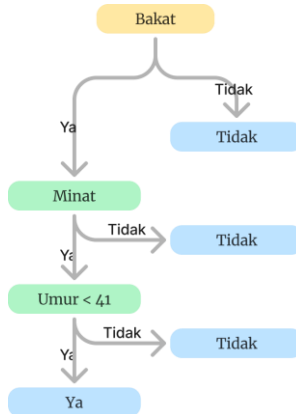
Gambar 2.1 Pohon Keputusan

Berikut contoh kasus untuk memudahkan penjelasan tentang *decision tree*. Pada tabel 2.2, akan berisi prediksi apakah seseorang tersebut akan menjadi seorang penyanyi, berdasarkan umur, bakat, dan minat.

Tabel 2.2 Contoh Kasus Penyanyi

Umur	Bakat	Minat	Penyanyi
20	Ya	Ya	Ya
41	Tidak	Tidak	Tidak
19	Ya	Ya	Ya
45	Ya	Ya	Tidak
22	Ya	Tidak	Tidak
34	Tidak	Ya	Tidak
42	Ya	Tidak	Tidak
27	Tidak	Ya	Tidak
33	Ya	Tidak	Tidak
43	Ya	Ya	Tidak

Berdasarkan tabel 2.2, seseorang tidak dapat menjadi seorang penyanyi, apabila tidak memiliki bakat terlepas meskipun dia memiliki keunggulan lain. Dan seseorang yang memiliki bakat, minat dan berumur kurang dari 41 tahun, bisa menjadi penyanyi. Hal ini dapat digambarkan menjadi pohon keputusan seperti gambar 2.2.



Gambar 2.2 Pohon Keputusan Penyanyi

2.2.4 Tes DASS (*Depression Anxiety Stress Scale*)

DASS adalah kepanjangan dari *Depression Anxiety Stress Scale*. DASS merupakan alat ukur berupa kuesioner yang telah menjadi instrumen baku yang dikembangkan oleh Lovibond and Lovibond (1995) dan diusulkan oleh Australian Psychological Society., (Ulfah, 2019). Instrumen ini dimodifikasi dan disesuaikan agar dapat mencerminkan kondisi subjek penelitian secara akurat. Kuesioner DASS terdiri dari 42 pernyataan yang berhubungan dengan tingkat stres, kecemasan, dan depresi individu, (Rais, 2023).

Pengujian yang dilakukan Widyana & Sumiharso (2020), menunjukkan koefisien reliabilitas sebesar 0,954 untuk kategori Depresi, 0,903 untuk kategori Kecemasan, dan 0,917 untuk kategori Stres. Ini menandakan bahwa skala-skala tersebut saling konsisten dan memiliki reliabilitas yang tinggi. Hasil tersebut juga menunjukkan bahwa item-item pada skala ini sesuai dengan versi aslinya. Dengan demikian, penelitian ini membuktikan bahwa versi Indonesia dari DASS-42 memiliki tingkat validitas yang tinggi. Berdasarkan hasil pemeriksaan validitas dan reliabilitas dari *Depression Anxiety Stress*



Scale – Indonesian Form (DAS-IF), menunjukkan bahwa skala ini valid, siap digunakan, dan aman untuk menentukan tingkat depresi, kecemasan, dan stres

Pengujian validitas menggunakan rumus *Pearson Product Moment* menunjukkan kuesioner valid jika $R \text{ hitung} > R \text{ tabel}$ (0,444) dengan signifikansi $> \alpha$ 0,05. Uji validitas pada 20 responden membuktikan kuesioner valid, (Novia, 2018). Reliabilitas diuji dengan *Cronbach's Alpha*, di mana nilai $< 0,6$ dianggap rendah, 0,7 cukup, dan $> 0,8$ baik. Hasil uji menunjukkan reliabilitas tinggi dengan $\alpha = 0,9483$ \ $\alpha = 0,9483$, (Novia, 2018).

2.2.5 Website

Website adalah sekumpulan halaman *web* yang bertujuan untuk menampilkan beragam informasi dalam format teks, gambar, dan audio. Halaman-halaman tersebut terkait dan terbentuk dalam satu domain yang terhubung satu sama lain secara terstruktur, (Riswanto dkk., 2023).

2.2.6 MySQL

MySQL (*My Structure Query Language*) adalah salah satu perangkat lunak sistem manajemen basis data (DBMS) yang menggunakan SQL, di antara berbagai DBMS lainnya seperti Oracle, MS SQL, dan PostgreSQL. MySQL merupakan DBMS bersifat *multithread* dan *multi-user*, serta berbasis *open source* atau gratis dengan lisensi GNU General Public License (GPL), (Windiauliazain, 2023).

2.2.7 RAD

Menurut Kiswanto (2022), metode *Rapid Application Development* (RAD) adalah pendekatan



pengembangan sistem yang menekankan pada siklus perkembangan yang cepat, biasanya memakan waktu 1-3 bulan untuk menyelesaikan sistem. Pada tahap awal, sistem dirancang untuk menetapkan kebutuhan (*requirement*) dan kemudian fokus pada identifikasi kebutuhan aplikasi untuk pengguna. RAD sangat bergantung pada *prototyping* dan partisipasi aktif pengguna. Proses RAD memungkinkan pengguna untuk mengevaluasi model kerja sejak dini, menilai apakah model tersebut telah memenuhi kebutuhan, serta memberikan masukan untuk perubahan yang diperlukan. Berdasarkan masukan pengguna, prototipe diubah, dan proses iterasi terus berlanjut hingga seluruh sistem selesai dan pengguna merasa puas, (Hita, 2020).

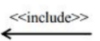
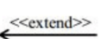
2.2.8 UML

Menurut Hita (2020), pemodelan adalah proses merancang perangkat lunak sebelum tahap pengkodean dimulai. Model dalam perangkat lunak dapat dianalogikan seperti pembuatan cetak biru dalam pembangunan gedung. Unified Modeling Language (UML) adalah suatu "bahasa" yang telah diakui sebagai standar industri untuk menentukan, memvisualisasikan, merancang, dan mendokumentasikan artefak dari sistem perangkat lunak, serta untuk memodelkan bisnis dan sistem non-perangkat lunak lainnya, (Kiswanto, 2022).

2.2.8.1. Use case Diagram

Menurut Sudibyo dkk. (2023), *use case diagram* merupakan abstraksi dari interaksi antara sistem dan aktor. *Use case* berfungsi untuk menggambarkan interaksi tipikal antara admin dan pengguna dalam sebuah sistem, serta cara sistem tersebut digunakan, yang digambarkan melalui diagram. Berikut simbol-simbol yang digunakan dalam *use case diagram*:

Tabel 2.3 Komponen Use case Diagram

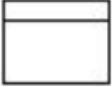
Simbol	Keterangan
	Aktor: Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	Use case: Abstraksi dan interaksi antara sistem dan aktor
	Association: Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	Generalisasi: Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2.2.8.2. Activity Diagram

Activity diagram digunakan untuk menggambarkan serangkaian tindakan dan transisi dasar yang dipicu oleh penyelesaian tindakan dari sumber tertentu. Diagram ini mirip dengan *flowchart*, yang menggambarkan proses yang terjadi antara aktor dan sistem, Sudibyo dkk. (2023). Simbol-simbol yang digunakan dalam *activity* diagram adalah sebagai berikut:



Tabel 2.4 Komponen Activity Diagram

Simbol	Keterangan
	Status awal: Menunjukkan status awal dari diagram aktivitas
	Status akhir: Menunjukkan status akhir dari diagram aktivitas
	Activity: Berisi aktivitas yang dilakukan sistem, <i>activity</i> biasanya diawali dengan kata kerja
	Decision: Percabangan dimana ada pilihan aktivitas yang lebih dari satu
	Swimlane: Pemisah organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
	Line connector: Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

2.2.8.3. Class Diagram

Class Diagram atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class Diagram* juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek, (Purwasih, 2023). Berikut simbol dari komponen pada *class diagram*:



Tabel 2.5 Komponen Class Diagram

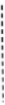
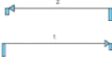
Simbol	Keterangan
	Class: Himpunan dari objek-objek yang berbagi atribut dan operasi yang sama
	Association: Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Directed association: Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain
	Generalization: Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
	Dependency: Relasi antar kelas dengan makna kebergantungan antar kelas
	Aggregation: Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

2.2.8.4. Sequence Diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan suatu sistem yang sesuai dengan *use case* diagram, Sudibyo dkk. (2023). Simbol-simbol yang digunakan pada *sequence* diagram adalah sebagai berikut:



Tabel 2.6 Komponen Sequence Diagram

Simbol	Keterangan
	Actor: Menggambarkan seorang pengguna (<i>user</i>) yang berada di luar sistem dan sedang berinteraksi dengan sistem
	Activation box: merepresentasikan waktu yang dibutuhkan suatu objek untuk menyelesaikan tugasnya
	Lifeline: Menggambarkan aktivitas dari objek
	Objek: Mendokumentasikan perilaku sebuah objek pada sebuah sistem
	Messages: Menggambarkan komunikasi antar objek
	Self Message: Menunjukkan kegiatan memuat proses informasi pada aktivitas sendiri.

2.2.9 Testing

Pengujian perangkat lunak merupakan bagian dari *Software Development Life Cycle* (SDLC) yang digunakan untuk mengidentifikasi kesalahan-kesalahan baik secara fungsional maupun non-fungsional, (Praniffa dkk., 2023). Pengujian perangkat lunak terbagi menjadi dua, yaitu *black box* dan *white box*. *Black box testing* atau dapat disebut juga *Behavioral Testing* adalah pengujian





yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik, (Nurfauziah & Jamaliyah, 2022).