

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Penyakit Paru – Paru

Paru-paru adalah salah satu organ vital manusia yang memiliki peran penting dalam sistem pernapasan. Penyakit yang menyerang organ ini tergolong berbahaya karena dapat menular dengan cepat. Kurangnya perhatian masyarakat terhadap kondisi ini menyebabkan angka kematian akibat penyakit paru-paru cukup tinggi. Beberapa jenis penyakit yang sering terjadi meliputi pneumonia, legionnaires, efusi pleura, tuberkulosis (TB), pneumotoraks, asma, penyakit paru obstruktif kronis, bronkitis kronis, emfisema, silikosis, dan asbestosis.

Kesadaran masyarakat terhadap bahaya penyakit paru-paru serta gejalanya masih rendah. Jika dibiarkan tanpa penanganan yang tepat, kondisi ini dapat berujung pada kematian. Jumlah penderita penyakit paru-paru di Indonesia masih tinggi. Berdasarkan data Globocan atau International Agency for Research on Cancer tahun 2012, tercatat 25.322 kasus kanker paru-paru pada pria dan 9.374 kasus pada wanita di Indonesia. (Wenda et al., 2023).

2.1.2 Penyebab penyakit paru – paru

Penyakit paru-paru umumnya muncul akibat paparan udara tercemar yang mengandung debu, asap, virus, atau bakteri, sehingga memicu infeksi pada saluran pernapasan. Bayi hingga orang dewasa berisiko mengalami penyakit ini, yang proses

penyembuhannya tidaklah mudah. Penting untuk memahami jenis penyakit paru-paru berdasarkan jumlah kasus di Indonesia serta gejala yang dialami pasien. Salah satu penyakit paru-paru yang berbahaya adalah PPOK. Berdasarkan laporan World Health Organization yang diterbitkan WHO pada 2008, PPOK diperkirakan menjadi penyebab kematian ketiga pada 2020, sementara tingkat kecacatannya diproyeksikan naik dari peringkat ke-9 ke peringkat ke-5 pada tahun yang sama (Metode & Bayes, 2022).

2.1.3 Metode Klasifikasi

Klasifikasi dalam machine learning bertujuan memprediksi kategori atau label data baru dengan mengidentifikasi pola dari data latihan. Proses ini melibatkan penggunaan teknik atau algoritma tertentu untuk menentukan kategori data berdasarkan pola yang telah dipelajari sebelumnya. Tujuan utamanya adalah mengelompokkan data ke dalam kelas yang sesuai berdasarkan fitur atau atribut yang tersedia (Farid, 2021).

Metode Klasifikasi terdapat beberapa algoritma diantaranya *Decision Tree*, *Naive Bayes Classifier*, *K-Nearest Neighbors* (KNN), *Support Vector Machines* (SVM) dan *Random Forest*.

2.1.3.1 Naïve bayes

Naive Bayes merupakan metode klasifikasi yang memiliki probabilitas tinggi dalam menghasilkan perhitungan sistem yang optimal. Teknik ini mengandalkan konsep probabilitas serta statistik yang diperkenalkan oleh ilmuwan Inggris, Thomas Bayes. Prinsip utamanya adalah memprediksi peluang masa depan berdasarkan

pengalaman sebelumnya, yang dikenal sebagai Teorema Bayes. Karakteristik utama dari Naïve Bayes Classifier terletak pada asumsi yang sangat kuat (naïf) mengenai independensi setiap kondisi atau kejadian (Wahyu, 2023) .

Keunggulan metode ini terletak pada kemampuannya dalam mengestimasi parameter meskipun jumlah data pelatihan terbatas. Setiap variabel dianggap saling bebas, sehingga proses klasifikasi hanya memerlukan varian satu variabel dalam satu kelas tanpa harus mempertimbangkan keseluruhan matriks kovarians.

1. Cara Kerja Naïve Bayes : Cara kerja dari metode Naïve Bayes Classifier menggunakan perhitungan probabilitas.

2. Kelebihan Naïve bayes :

1. Cepat dan sederhana.
2. Efisien untuk data dengan jumlah fitur yang besar.
3. Bekerja baik pada data kategorikal dan data teks.

3. Kekurangan Naïve bayes :

1. Asumsi independensi antar fitur jarang terjadi di dunia nyata.
2. Kurang akurat pada data dengan fitur yang sangat bergantung satu sama lain.

2.1.3.2 Support Vector Machines (SVM)

Algoritma *Support Vector Machine*(SVM) berperan dalam mencari hyperplane optimal, yakni fungsi yang mampu memisahkan dua kelas secara efektif. Prosesnya melibatkan upaya untuk memaksimalkan margin atau jarak antara pola pelatihan serta batas keputusan.

SVM tidak hanya digunakan untuk klasifikasi dua kelas, tetapi juga dapat menangani kasus dengan jumlah kelas yang lebih banyak. Pendekatan One-vs-All melatih model SVM untuk setiap kelas dengan menganggapnya sebagai kelas positif, sementara kelas lainnya dianggap negatif. Sebaliknya, pendekatan One-vs-One membangun model SVM untuk setiap kombinasi pasangan kelas. Hasil akhir prediksi diperoleh dari keputusan yang dihasilkan oleh model-model tersebut.

(Sephya., 2023).

1. Cara Kerja SVM : Membuat model yang memisahkan kelas dengan hyperplane dalam ruang fitur, menggunakan data pelatihan untuk menentukan hyperplane terbaik.

2. Kelebihan SVM :

1. Efektif dalam ruang fitur yang memiliki dimensi tinggi.
2. Tidak sensitif terhadap dimensi fitur yang tinggi (*curse of dimensionality*).
3. Dapat menangani dataset dengan jumlah fitur yang lebih besar daripada jumlah sampel.
4. Cocok untuk dataset yang memiliki pola yang kompleks dan non-linear dengan menggunakan kernel.

3. Kekurangan SVM :

1. Kompleksitas komputasi yang tinggi terutama pada dataset besar.
2. Memerlukan penyesuaian parameter yang cermat, seperti parameter C dan pilihan kernel, untuk mendapatkan hasil yang optimal.

3. Tidak intuitif untuk diinterpretasikan secara langsung seperti *KNN*.

2.1.3.3 Matriks Confusion

Matriks Confusion adalah alat evaluasi yang berperan dalam analisis klasifikasi di machine learning. Alat ini menampilkan kinerja model melalui perbandingan antara hasil prediksi dan kelas asli dari data yang diuji. Kategori dalam matriks ini terdiri atas *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) Type I Error
	0 (Negative)	FN (False Negatif) Type II Error	TN (True Negative)

Gambar 2.1 Matriks Confusion

Dr. Adi Wibowo dari Universitas Indonesia menekankan pentingnya matriks confusion dengan menyatakan bahwa alat ini memberikan gambaran jelas mengenai kemampuan model klasifikasi dalam mengenali kelas-kelas dalam dataset. Memahami hasil prediksi model melalui matriks confusion memungkinkan evaluasi akurasi serta kinerja keseluruhan model. Pernyataan tersebut mengilustrasikan peran

matriks confusion sebagai instrumen utama dalam menilai performa model klasifikasi serta signifikansi interpretasi hasil prediksi.

2.1.4 CRISP - DM

CRISP-DM adalah singkatan dari Cross Industry Standard Process for Data Mining. Pakar di Indonesia telah menyampaikan pandangan mereka mengenai metodologi ini. Contohnya, Profesor Bambang Riyanto dari Universitas Indonesia menyatakan bahwa CRISP-DM menawarkan pendekatan yang sistematis serta terstruktur dalam mengelola proyek data mining, memungkinkan organisasi memahami dan memanfaatkan data dengan lebih efektif. Pernyataan tersebut menegaskan signifikansi CRISP-DM dalam pengembangan proyek data mining di Indonesia serta menunjukkan manfaatnya dalam mendukung pengambilan keputusan berbasis data.

2.1.4.1 Business Understanding

1. Tujuan Bisnis : Meningkatkan efisiensi dan akurasi proses rujukan rontgen bagi pasien di Puskesmas Cimuning untuk deteksi dini penyakit paru-paru.
2. Objektif Data Mining : Dokter dapat menggunakan gejala pasien untuk memprediksi risiko penyakit paru-paru dengan sistem pendukung keputusan berbasis data mining.
3. Kebutuhan Pemangku Kepentingan : Sumber daya puskesmas dan rumah sakit dapat dioptimalkan dengan sistem yang akurat yang membantu tenaga medis menentukan pasien yang benar-benar memerlukan pemeriksaan rontgen.

2.1.4.2 Data Understanding

Pengumpulan data terkait, termasuk Batuk Berdahak, Frekuensi Sesak, Nyeri Dada, Riwayat Merokok, serta Pola Napas. Melakukan analisis eksploratif guna mengidentifikasi pola, distribusi, serta hubungan antar fitur dalam data. Mengecek kualitas data melalui pemeriksaan missing values, outliers, dan inkonsistensi.

2.1.4.3 Data Preparation

Normalisasi data, penanganan outlier, serta identifikasi nilai yang hilang perlu dilakukan apabila diperlukan. Karakteristik paling relevan dalam memprediksi penyakit paru-paru dini harus ditentukan. Pembagian data ke dalam subset pengujian dan pelatihan perlu mempertimbangkan proporsi yang sesuai.

2.1.4.4 Modeling

- a. **Naive Bayes** untuk model probabilistik sederhana yang mampu memproses data diskrit dengan cepat.
- b. **Support Vector Machine(SVM)** untuk memetakan pasien dengan pola gejala yang kompleks ke dalam dua kelas (terindikasi atau tidak).

2.1.4.5 Evaluation

Evaluasi model *KNN* dan *SVM* menggunakan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* pada subset pengujian. Bandingkan kinerja kedua model dan identifikasi model yang memberikan hasil terbaik untuk prediksi dini penyakit paru - paru.

2.1.4.6 Deployment

Naive Bayes dan Support Vector Machine digunakan untuk mengoptimalkan sistem pendukung keputusan dalam membantu karyawan Puskesmas Cimuning menentukan rujukan rontgen bagi pasien. Pelatihan diberikan kepada tenaga medis agar mampu memanfaatkan hasil prediksi sistem dalam proses pengambilan keputusan.

Pengecekan rutin dilakukan guna memastikan performa sistem tetap sesuai dengan kondisi aktual. Model dapat mengalami pembaruan apabila data terbaru mengindikasikan perubahan pola gejala atau karakteristik pasien.

2.1.5 Waterfall

Dr. Winston W. Royce memperkenalkan Metode Waterfall pada tahun 1970 sebagai pendekatan sistematis dan berurutan dalam pengembangan perangkat lunak. Setiap tahap harus diselesaikan secara menyeluruh sebelum melanjutkan ke tahap berikutnya, menyerupai aliran air terjun dari satu tingkat ke tingkat lain. Metode Waterfall terdiri atas beberapa tahapan, antara lain:

2.1.5.1 Requirement Analysis (Analisis Kebutuhan)

Tujuan dari tahap ini adalah untuk menemukan dan mencatat kebutuhan sistem dan pengguna. Proses ini membantu memahami keinginan pengguna dan menetapkan cara sistem bekerja. Analisis dilakukan melalui wawancara, survei, dan peninjauan dokumen yang relevan.

2.1.5.2 *System Design* (Desain Sistem)

Setelah menganalisis serta memahami kebutuhan, tahap selanjutnya ialah merancang sistem yang akan digunakan. Tahapan ini melibatkan pemilihan teknologi, perancangan database, antarmuka pengguna, serta komponen sistem lainnya. Optimalisasi perancangan sistem berperan krusial dalam keberhasilan implementasi.

2.1.5.3 *Implementation* (Implementasi)

Pada tahap ini, desain yang telah dibuat diterjemahkan ke dalam kode program. Pengembang menulis kode dan membuat modul-modul sistem sesuai dengan spesifikasi yang telah dirancang. Implementasi harus dilakukan dengan hati-hati untuk menghindari kesalahan dan memastikan bahwa semua kebutuhan telah terpenuhi.

2.1.5.4 *Testing* (Pengujian)

Usai pengembangan sistem, tahap berikutnya ialah pengujian guna memastikan sistem berfungsi sebagaimana yang diharapkan. Proses ini mencakup pengujian unit, integrasi, sistem, serta penerimaan. Tujuan utama pengujian adalah mengidentifikasi serta memperbaiki bug, sekaligus memastikan sistem sesuai dengan kebutuhan pengguna.

2.1.5.5 *Deployment* (Penerapan)

Setelah sistem lulus pengujian, tahap berikutnya adalah penerapan sistem ke lingkungan produksi. Ini melibatkan instalasi perangkat lunak, migrasi data, dan

pelatihan pengguna. Penerapan harus dilakukan dengan hati-hati untuk meminimalkan gangguan terhadap operasi bisnis.

2.1.5.6 *Maintenance* (Pemeliharaan)

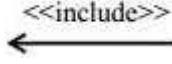
Tahap akhir mencakup pemeliharaan sistem. Setelah sistem diterapkan, pemantauan serta perawatan diperlukan agar tetap berfungsi optimal serta menyesuaikan dengan kebutuhan pengguna yang dapat berubah seiring waktu. Pemeliharaan meliputi perbaikan bug, peningkatan fitur, serta penyesuaian terhadap perubahan lingkungan.

2.1.6 Pemodelan Sistem UML

2.1.6.1 Use Case Diagram

Menurut Martin Fowler (2003) mengatakan “Use Case Diagram adalah teknik merekam persyaratan fungsional sebuah sistem”. Use case mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri. Komponen - komponen pembentuk diagram *use case* dijelaskan pada tabel 2.1

Tabel 2.1 Use Case Diagram

Komponen UseCase	Keterangan
 Aktor	Komponen yang merepresentasikan individu atau objek, seperti perangkat atau sistem lain, yang berinteraksi dengan sistem.
 <i>Use Case</i>	Gabaran fungsional suatu sistem untuk membantu pengguna memahami dan memahami fungsinya
 <i>Association</i>	Abstraksi hubungan antara aktor dan <i>Use Case</i> .
 <i>Generalisasi</i>	Menandakan keahlian aktor dalam berperan pada <i>Use Case</i> .
 <i><<include>></i>	Menunjukan bahwa suatu <i>Use Case</i> seluruhnya merupakan fungsionalitas dari usecase lainnya.

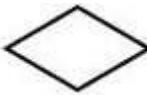
	Menunjukkan bahwa suatu usecase merupakan tambahan fungsional dari <i>Use Case</i> lainnya jika suatu kondisi sudah terpenuhi.
--	--

2.1.6.2 Activity Diagram

Activity diagram, dalam bahasa Indonesia diagram aktivitas, merupakan diagram yang dapat mensimulasikan aktivitas sistem. Alur proses sistem digambarkan dalam susunan vertikal. Diagram ini menjelaskan logika prosedural, proses bisnis, dan alur kerja setiap proses sistem; analisis masih menggunakan Support Vector Machine dan Naive Bayes (Setyawan et al., 2021). Komponen-komponen pembentuk diagram *activity* dijelaskan pada tabel 2.2

Tabel 2. 2 Simbol-simbol Diagram Activity

Activity Diagram	Keterangan
 Status Awal	Start State, sebagai tanda awal proses dari <i>activity</i> diagram
 <i>Activity</i>	Menampung event atau aktivitas pada proses sistem.

 <i>Decision</i>	<i>Decision</i> , digunakan ketika terjadi pemilihan 2 kondisi event pada diagram
 <i>Join</i>	Penggabungan dimana lebih dari satu aktivitas lalu digabungkan menjadi satu
 <i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
 <i>Status Akhir</i>	<i>End State</i> , sebagai tanda akhir dari <i>activity diagram</i> .

2.1.7 Pemrograman Sistem

Berikut beberapa pemrograman terkait yang terdapat pada penelitian ini:

2.1.7.1 Python

Naive Bayes dan *Support Vector Machine* tetap digunakan dalam berbagai analisis data berkat fleksibilitasnya. Python terkenal karena filosofi "*batteries included*," yang berarti bahasa ini menyediakan pustaka serta modul standar yang kaya fitur. Hal ini memungkinkan penggunaanya untuk mengembangkan aplikasi tanpa perlu menulis kode dari awal. Kemudahan dalam pembelajaran serta dukungan dari komunitas yang aktif menjadikan Python pilihan utama bagi pemula maupun profesional di dunia pemrograman.

Popularitas Python dalam analisis data didukung oleh kemudahan penggunaannya serta aksesibilitasnya bagi berbagai kalangan. Beragam pustaka yang tersedia memiliki fungsi spesifik yang dapat dimanfaatkan siapa saja pada berbagai sistem operasi sebagai bahasa pemrograman open source, Python memberi kebebasan bagi pengguna untuk mengembangkan dan membagikan kode sesuai kebutuhan (Ua et al., 2023).

2.1.7.2 SQLITE

SQLite dipilih sebagai sistem basis data dalam penelitian ini karena bersifat ringan, portabel, dan tidak memerlukan konfigurasi server. Teknologi ini cocok untuk aplikasi penelitian yang membutuhkan penyimpanan data lokal dengan pengelolaan yang sederhana. Keunggulan self-contained memungkinkan seluruh data tersimpan

dalam satu file database, sehingga proses pengelolaan serta transfer data antar sistem menjadi lebih praktis.

2.1.7.3 Flask

Flask merupakan framework web mikro yang dikembangkan menggunakan bahasa pemrograman Python. Tujuan utama dari framework ini adalah memungkinkan pembuatan aplikasi web secara cepat serta efisien, mengutamakan kesederhanaan dan fleksibilitas. Pengembang dapat memanfaatkan Flask untuk membangun berbagai jenis aplikasi, mulai dari blog sederhana, *API RESTful*, hingga sistem yang lebih kompleks. Salah satu kelebihannya terletak pada kemampuannya untuk berintegrasi secara mudah dengan beragam ekstensi serta library Python lainnya. Hal ini memungkinkan pengembang memiliki kendali penuh terhadap struktur dan fungsionalitas aplikasi web yang dibuat.

2.2 Kajian Penelitian Terdahulu

2.2.1 Paper 1

Pada paper yang berjudul “**Implementasi Algoritma Decision Tree dan Support Vector Machine untuk Klasifikasi Penyakit Kanker Paru**” yang ditulis oleh Dhini Sephya, Kharisma Rahayu, Salsabila Rabbani, Vindi Fitria, Rahmaddeni, Yuda Irawan, Regiolina Hayami.

Penelitian ini bertujuan membandingkan kinerja algoritma Decision Tree serta *Support Vector Machine*(SVM) dalam mengklasifikasikan atau mendeteksi dini

penyakit kanker paru-paru. Fokus utama penelitian adalah menemukan algoritma yang lebih akurat dan efektif dalam memprediksi atau mengklasifikasikan risiko kanker paru-paru agar hasilnya dapat dimanfaatkan sebagai dukungan dalam diagnosis awal di sektor kesehatan.

Penelitian ini terdiri dari beberapa tahapan, dimulai dengan proses pengumpulan data medis pasien yang memiliki potensi terkena kanker paru-paru. Data yang diperoleh kemudian diolah guna memastikan kualitasnya melalui proses pembersihan, konversi ke dalam format numerik, serta normalisasi. Setelah tahap ini selesai, algoritma *Decision Tree* dan *Support Vector Machine*(SVM) digunakan untuk melakukan prediksi dan klasifikasi tingkat risiko kanker paru-paru. Kinerja dari kedua algoritma ini dinilai berdasarkan metrik akurasi, presisi, recall, serta F1-score, sehingga memungkinkan analisis perbandingan efektivitasnya dalam mendeteksi kanker paru-paru pada tahap awal. Langkah terakhir dalam penelitian ini adalah analisis terhadap hasil evaluasi guna menentukan algoritma yang paling optimal untuk mendukung diagnosis dini di fasilitas kesehatan.

Kajian dalam jurnal ini berfokus pada perbandingan efektivitas antara algoritma *Decision Tree* dan *Support Vector Machine*(SVM) dalam mendeteksi kanker paru-paru. Hasil penelitian mengungkap bahwa salah satu dari kedua algoritma tersebut (baik SVM maupun Decision Tree) menunjukkan keunggulan dalam hal akurasi, presisi, atau recall, menjadikannya lebih efektif dalam memprediksi risiko kanker paru-

paru berdasarkan gejala pasien. Algoritma yang terbukti lebih unggul direkomendasikan untuk diterapkan dalam sistem pendukung keputusan medis, sehingga dapat membantu tenaga kesehatan dalam mengambil keputusan secara lebih cepat dan akurat. Kesimpulan ini memperkuat potensi penerapan teknik data mining dalam meningkatkan deteksi dini kanker paru-paru di sektor kesehatan.

Beberapa keterbatasan dalam penelitian ini mencakup jumlah serta variasi dataset yang mungkin kurang mencerminkan populasi secara luas, sehingga dapat memengaruhi generalisasi model. Selain itu, penelitian ini hanya membandingkan *Decision Tree* dan *SVM* tanpa mempertimbangkan algoritma lain yang mungkin memiliki performa lebih baik. Optimasi parameter model yang belum dilakukan secara mendalam juga dapat berkontribusi terhadap akurasi yang diperoleh. Di dunia nyata, implementasi model ini dapat menghadapi tantangan akibat beragamnya kondisi klinis, serta keterbatasan dalam evaluasi yang hanya berfokus pada akurasi, presisi, dan recall, tanpa mempertimbangkan aspek kesalahan diagnosis yang sangat penting dalam praktik medis.

2.2.2 Paper 2

Paper berjudul “**KLASIFIKASI PENYAKIT PARU-PARU DENGAN MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER**” yang ditulis oleh Muhammad Yusril Haffandi, Elin Haerani, Fadhilah Syafria, dan Lola Oktavia

membahas klasifikasi penyakit paru-paru menggunakan algoritma Naïve Bayes Classifier.

Penelitian ini bertujuan untuk menghasilkan klasifikasi yang akurat guna mendukung diagnosis dini, sehingga tenaga medis dapat mengambil keputusan lebih cepat dan tepat dalam menangani penyakit paru-paru. Algoritma Naïve Bayes Classifier diterapkan untuk mengklasifikasikan data pasien berdasarkan gejala yang dialami. Evaluasi model dilakukan dengan membagi data menjadi 100 data training dan 34 data testing dengan rasio 7:3. Pengujian menggunakan RapidMiner menunjukkan bahwa model mencapai tingkat akurasi 97,06%, yang membuktikan efektivitas Naïve Bayes dalam mengklasifikasikan penyakit paru-paru berdasarkan gejala pasien.

Hasil utama dari penelitian ini menunjukkan bahwa algoritma Naïve Bayes memiliki tingkat akurasi tinggi dalam klasifikasi penyakit paru-paru, dengan pencapaian akurasi sebesar 97,06%. Pengujian dilakukan menggunakan dataset yang terdiri dari 100 data training dan 34 data testing dalam platform RapidMiner. Temuan ini membuktikan bahwa Naïve Bayes merupakan metode yang efektif dan dapat diandalkan dalam mendukung diagnosis dini, serta berpotensi menjadi alat bantu dalam sistem pendukung keputusan medis.

Keterbatasan penelitian ini terletak pada penggunaan tunggal algoritma Naïve Bayes, sementara metode lain berpotensi memberikan hasil yang berbeda atau lebih optimal dalam klasifikasi penyakit paru-paru. Meskipun tingkat akurasinya tinggi,

tidak adanya perbandingan dengan algoritma lain membuat hasil penelitian ini belum dapat mengidentifikasi metode yang paling sesuai untuk kondisi tertentu.

2.2.3 Paper 3

Paper berjudul “KOMPARASI ALGORITMA DECISION TREE, SVM, NAIVE BAYES DALAM PREDIKSI PENYAKIT LIVER” yang ditulis oleh Diva Nabila Herisnan, Apriliani, Eric Dadynata, Rahmaddeni, dan Lusiana Efrizoni membahas perbandingan tiga algoritma dalam mendiagnosis penyakit liver. Penelitian ini bertujuan untuk menentukan algoritma yang memiliki tingkat akurasi serta keandalan terbaik, sehingga dapat membantu tenaga medis dalam mendeteksi penyakit liver dengan lebih cepat dan tepat. Selain itu, penelitian ini juga memberikan panduan bagi pengguna dalam memilih algoritma klasifikasi yang paling sesuai berdasarkan aspek akurasi dan kemudahan interpretasi model. Metodologi yang digunakan dalam penelitian ini mencakup tahap preprocessing untuk membersihkan serta menyiapkan data sebelum analisis, termasuk penanganan data yang hilang serta penghapusan duplikasi. Seleksi fitur kemudian diterapkan untuk menyaring variabel yang kurang berpengaruh terhadap prediksi, dengan tujuan mengurangi risiko overfitting. Setelah proses seleksi fitur selesai, data dibagi menjadi data latih dan data uji dengan rasio 50:50, 70:30, serta 90:10 guna mengevaluasi performa model dalam berbagai skenario pembagian data. Ketiga algoritma, yaitu *Decision Tree*, *SVM*, serta *Naive Bayes*, diterapkan pada data yang telah diproses.

Evaluasi dilakukan dengan mengukur akurasi, precision, recall, serta F1-score di setiap rasio pembagian data guna menentukan algoritma yang paling optimal dalam memprediksi penyakit liver. Hasil utama penelitian ini menunjukkan bahwa performa setiap algoritma berbeda tergantung pada rasio pembagian data yang digunakan. Berdasarkan pengujian, Decision Tree cenderung memiliki akurasi lebih tinggi dibandingkan SVM serta Naive Bayes. Penelitian ini memiliki keterbatasan karena hanya membandingkan tiga algoritma, sementara metode klasifikasi lainnya berpotensi memberikan hasil yang lebih baik. Selain itu, variasi dataset yang terbatas juga memengaruhi cakupan temuan serta penerapan hasil penelitian dalam kondisi dunia nyata.

2.2.4 Kesimpulan

Paper pertama hingga ketiga membahas perbandingan kinerja algoritma Naive Bayes, SVM, dan Decision Tree dalam klasifikasi penyakit paru-paru serta liver. Hasil analisis menunjukkan bahwa masing-masing algoritma memiliki keunggulan berbeda bergantung pada karakteristik dataset, yang berhubungan dengan deteksi dini penyakit di Puskesmas Cimuning. Paper kedua secara khusus meneliti efektivitas Naïve Bayes dalam mendeteksi penyakit paru-paru dengan tingkat akurasi tinggi, sehingga algoritma ini berpotensi dimanfaatkan untuk diagnosa awal.

Ketiga paper tersebut memberikan dasar bagi penelitian skripsi yang berfokus pada perbandingan kinerja Naive Bayes dan SVM dalam deteksi dini penyakit paru-paru. Bukti empiris yang disajikan dalam penelitian-penelitian tersebut menguatkan

potensi kedua algoritma dalam analisis penyakit sekaligus menyoroti keterbatasan yang harus diperhatikan, seperti variasi dataset serta optimasi parameter algoritma.

