BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang merujuk pada [6] oleh Imaniar Ikko Mulya Rizky, Suhendro Yusuf Irianto, dan Sriyanto pada tahun 2023 dengan judul "Perbandingan Kinerja Algoritma *Naive Bayes*, *Support Vector Machine* dan *Random forest* untuk Prediksi Penyakit Ginjal Kronis" Hasil pengujian menunjukkan bahwa pada split data (data training 0.7% dan data testing 0.3%), pemrosesan klasifikasi menggunakan algoritma *Naive Bayes* sebesar 97.14%, sedangkan pemrosesan klasifikasi menggunakan algoritma *Support Vector Machine* sebesar 92.50% dan untuk pemrosesan klasifikasi menggunakan algoritma *Random Forest* sebesar 99.64%. Penelitian ini menunjukkan bahwa algoritma Random Forest memiliki kinerja terbaik dalam memprediksidalam Penyakit Ginjal Kronis.

Penelitian dengan judul "Perbandingan Algoritme C5.0 dan Random Forest menggunakan Data Bank" oleh Rizka Aulia dan Muhamad Fadli. Hasil penelitian perbandingan algoritme C5.0 dan *random forest* untuk data latih dan data uji menggunakan *K–fold cross validation*, menghasilkan tingkat akurasi diatas 90%. Algoritme C5.0 lebih unggul untuk tingkat akurasi data latih sebesar 90.4 % , nilai akurasi terbaik untuk algoritme C5.0 terjadi pada fold ke 10 . Sedangkan *random forest* lebih unggul untuk data uji sebesar 97%, nilai ini konvergen dari 1- fold [7].

Penelitian yang merujuk pada [8] oleh Yunada Wiratama, RZ Abdul Aziz pada tahun 2024 dengan judul "Perbandingan Prediksi Penyakit Stunting Balita Menggunakan Algoritma Support Vektor Machine dan Random Forest"

Dalam penelitian ini, kami menerapkan dua metode machine learning untuk di bandingkan mana yang lebih baik klasifikasi dari dua metode ini, yaitu Random Forest dan Support Vector Machine (SVM), untuk melakukan klasifikasi penyakit stunting pada balita. Data yang digunakan merupakan data publik berjumlah 97.873 data. Setelah melalui tahap preprocessing, seperti pembersihan data, normalisasi, dan pembagian data, data dibagi menjadi set pelatihan dan pengujian. Model Random Forest dan SVM kemudian dilatih dengan menggunakan set pelatihan dan dievaluasi menggunakan metrik seperti akurasi, precision, dan recall. Hasil analisis menunjukkan bahwa kedua metode memiliki kinerja yang baik dalam mengklasifikasikan stunting pada balita, dengan hasil Random Forest mencapai akurasi 0,9997 dan SVM mencapai akurasi 0,9951. Temuan

Penelitian yang dilakukan oleh Mario Utomo, Rastri Prathivi pada tahun 2024 dengan judul "Perbandingan Algoritma Support Vector Machine dan Decision Tree untuk Klasifikasi Performa Perusahaan Mario Berdasarkan Kualitas Tidur" Penelitian ini membandingkan Support Vector Machine dan Decision Tree dengan pendekatan One Against All dalam mengklasifikasikan performa perusahaan. Feature yang digunakan untuk klasifikasi performa perusahaan ini terdiri dari 3 rasio keuangan yaitu rasio profitabilitas (ROA), likuiditas (CR), dan leverage (DER). Labelling atau target dalam klasifikasi dibagi menjadi 3 kategori yaitu normal, baik, dan kurang baik. Pada penelitian ini akan mempertimbangkan evaluasi seperti accuracy, cross validation, dan confusion matrix. Hasil kinerja algoritma Support Vector Machine menghasilkan akurasi sebesar 86,67%, sedangkan pada algoritma Decision Tree menghasilkan akurasi sebesar 93,33%.[9]

Tabel 2.1 Penelitian Terkait

No.	Judul Penelitian	Metode	Akurasi	Jumlah Dataset
1.	Perbandingan Kinerja Algoritma *Naive Bayes*, *Support Vector Machine* dan *Random forest* untuk Prediksi Penyakit Ginjal Kronis	*Naïve Bayes, SVM,* dan *Random Forest*	*Naïve Bayes* (97.14%), *SVM* (92.50%), *Random Forest* (99.64%).	400 Data
2.	Perbandingan Algoritme C5.0 dan *Random Forest* menggunakan Data Bank	C5.0 dan *Random Forest*	C5.0 (85.5%), *Random Forest* (97%)	45211 Data
3.	Perbandingan Prediksi Penyakit Stunting Balita Menggunakan Algoritma *Support Vektor Machine* dan *Random Forest*	*Support Vektor Machine* dan *Random Forest*	*Support Vektor Machine (*99,97%) dan *Random Forest (*99,51%*)*	22855 Data
4.	Perbandingan Algoritma *Support Vector Machine* dan *Decision Tree* untuk Klasifikasi Performa Perusahaan	*Support Vector Machine* dan *Decision Tree*	*Support Vector Machine* (86,67%) dan *Decision Tree* (93,33%)	150 Data

2.2 Landasan Teori

a. Penyakit *Liver*

Penyakit liver, atau yang sering disebut penyakit hati, merupakan gangguan kesehatan serius yang memengaruhi fungsi hati sebagai organ vital dalam tubuh manusia. Berdasarkan data dari Organisasi Kesehatan Dunia (WHO), sekitar 1,5 juta kematian setiap tahun disebabkan oleh penyakit liver, dengan hepatitis dan sirosis sebagai penyebab utama. Di Indonesia, prevalensi penyakit hepatitis B dan C mencapai 7,1% dari populasi, menjadikan penyakit liver sebagai salah satu masalah kesehatan utama.

Hati memiliki peran penting dalam berbagai proses metabolisme, termasuk memproduksi protein plasma, menyimpan energi dalam bentuk glikogen, mendetoksifikasi zat-zat berbahaya seperti alkohol dan obat-obatan, serta memproduksi empedu yang membantu pencernaan lemak. Gangguan pada fungsi hati dapat menyebabkan komplikasi serius, seperti gagal hati, hipertensi portal, dan bahkan kanker hati.

Penyakit liver dapat disebabkan oleh berbagai faktor, seperti:

- Infeksi virus: Virus hepatitis (Hepatitis A, B, C, D, dan E) menjadi penyebab utama peradangan hati.
- Konsumsi alkohol berlebihan: Alkohol dapat merusak sel-sel hati, menyebabkan peradangan kronis dan sirosis.
- Obesitas dan sindrom metabolik: Penumpukan lemak di hati (non-alcoholic fatty liver disease) dapat berkembang menjadi fibrosis dan sirosis.

- Gangguan genetik: Kondisi seperti hemochromatosis dan penyakit Wilson memengaruhi metabolisme hati.

Deteksi dini terhadap risiko penyakit liver menjadi sangat penting untuk mencegah kerusakan lebih lanjut. Langkah-langkah seperti pemeriksaan laboratorium (ALT, AST, bilirubin), pencitraan (USG, elastografi), dan evaluasi gaya hidup dapat membantu mengidentifikasi individu dengan risiko tinggi. Pencegahan dan intervensi dini terbukti dapat mengurangi angka morbiditas dan mortalitas akibat penyakit liver.

b. Machine Learning

Machine learning (ML) adalah cabang kecerdasan buatan yang memungkinkan sistem untuk belajar dan meningkatkan kinerja secara otomatis melalui data tanpa pemrograman eksplisit. Teknologi ini memanfaatkan algoritma untuk mengidentifikasi pola dan membuat prediksi atau keputusan. Tiga pendekatan utama ML adalah supervised learning, yang menggunakan data berlabel; unsupervised learning, yang mengungkap pola tersembunyi dalam data tanpa label; dan reinforcement learning, yang melatih agen melalui umpan balik berupa reward atau punishment.

Komponen utama dalam ML mencakup dataset, fitur, model, dan metrik evaluasi. Dataset menyediakan data untuk pelatihan dan pengujian, sementara fitur adalah atribut yang digunakan untuk membangun model. Model merupakan representasi matematis yang dirancang untuk membuat prediksi, dengan kinerja dievaluasi menggunakan metrik seperti akurasi, *precision,* dan *recall* .

Pengembangan ML melibatkan pemahaman masalah, eksplorasi dan pra-pemrosesan data, pemilihan algoritma, pelatihan model, serta evaluasi dan implementasi. Aplikasi ML mencakup berbagai bidang, seperti prediksi penyakit di bidang kesehatan, deteksi penipuan di sektor keuangan, dan sistem rekomendasi dalam pemasaran. Dengan kemampuannya untuk mengolah data besar secara efisien, ML telah menjadi komponen kunci dalam inovasi teknologi modern.

c. *Random Forest*

Random forest adalah kombinasi dari prediktor tree (pohon). Setiap tree(pohon) bergantung pada nilai vektor acak yang diambil sampelnya secara independen dan dengan distribusi yang sama untuk semua tree (pohon) [10]. Random Forest juga merupakan salah satu metode ensemble yang sangat populer dalam pembelajaran mesin, yang digunakan untuk tugas klasifikasi dan regresi. Metode ini bekerja dengan membangun sejumlah besar pohon keputusan (decision trees) yang bersifat "lemah" (weak learners) secara independen dan kemudian menggabungkan hasil dari pohon-pohon tersebut untuk membuat prediksi yang lebih kuat dan lebih akurat. Dalam proses ini, setiap pohon keputusan dibangun menggunakan teknik bagging (bootstrap aggregating), yang berarti bahwa setiap pohon dibangun dengan mengambil sampel acak dengan penggantian (bootstrapping) dari data pelatihan. Teknik ini mengurangi varians dan meningkatkan akurasi model tanpa meningkatkan bias secara signifikan.

Setiap pohon keputusan dalam Random Forest menggunakan subset acak dari fitur untuk membuat keputusan pemisahan (split) di setiap

node, yang membantu mencegah korelasi antar pohon dan meningkatkan keberagaman dalam hutan. Proses ini menghasilkan berbagai pohon yang beragam, yang secara kolektif bekerja lebih baik dibandingkan dengan model pohon keputusan tunggal. Prediksi akhir diperoleh dengan cara voting (untuk klasifikasi) atau rata-rata (untuk regresi), yang secara signifikan meningkatkan stabilitas dan ketahanan model terhadap overfitting, terutama pada data dengan dimensi yang sangat tinggi dan rumit.

Kelebihan utama dari Random Forest adalah kemampuannya untuk menangani dataset yang besar dan tidak terstruktur, seperti data teks atau gambar. Selain itu, Random Forest juga memiliki kemampuan untuk menangani data yang hilang (missing values) dan mengidentifikasi fitur yang paling penting dalam model. Namun, kelemahan dari algoritma ini adalah interpretabilitas yang rendah karena jumlah pohon yang sangat besar membuat model menjadi kompleks dan sulit untuk dipahami.

d. C.5.0

C5.0 adalah pengembangan dari algoritma C4.5 yang terkenal, yang digunakan untuk membangun pohon keputusan dalam pembelajaran mesin. C5.0 adalah algoritma yang sangat efisien untuk tugas klasifikasi, dan dikenal karena keakuratannya yang tinggi serta kemampuannya dalam menghasilkan model yang lebih ringkas dan lebih cepat dibandingkan dengan algoritma pohon keputusan lainnya. C5.0 menggunakan gain informasi dan rasio gain informasi untuk memilih fitur terbaik dalam memisahkan kelas-kelas data. Proses ini

memungkinkan C5.0 untuk membangun pohon keputusan yang lebih stabil dan lebih akurat.

Salah satu fitur utama dari C5.0 adalah teknik pemangkasan pohon (pruning), yang bertujuan untuk mengurangi kompleksitas model dan menghindari overfitting dengan menghilangkan cabang-cabang pohon yang tidak memberikan kontribusi signifikan terhadap prediksi. C5.0 juga mendukung teknik boosting, yang dapat meningkatkan akurasi model dengan menggabungkan beberapa model lemah menjadi model yang lebih kuat. Keuntungan lainnya adalah C5.0 menghasilkan aturan yang dapat dibaca dan dipahami manusia, sehingga sangat berguna dalam aplikasi yang memerlukan interpretasi model, seperti dalam aplikasi medis dan analisis bisnis.

Kelemahan dari C5.0 terletak pada ketergantungannya pada kualitas data pelatihan. Seperti algoritma pohon keputusan lainnya, C5.0 dapat menghasilkan model yang tidak optimal jika data pelatihan memiliki noise yang tinggi atau data yang hilang *(missing values)*. Selain itu, meskipun C5.0 memiliki kemampuan untuk menghindari overfitting, ia masih dapat menghasilkan pohon yang kompleks pada dataset yang sangat besar.

e. *Support Vector Machine*

Support Vector Machine (SVM) adalah algoritma yang memiliki kesederhanaan dan fleksibilitasnya yang relatif untuk mengatasi berbagai masalah klasifikasi, SVM secara khusus memberikan kinerja prediksi yang seimbang, bahkan dalam studi di mana ukuran sampel mungkin terbatas. SVM bekerja dengan mencari *hyperplane* terbaik yang memisahkan dua kelas dalam ruang fitur. Tujuannya adalah untuk

menemukan hyperplane yang memaksimalkan margin, yaitu jarak antara titik data terdekat dari kedua kelas yang berbeda, yang disebut sebagai support vectors. *Hyperplane* ini berfungsi untuk memisahkan kelas-kelas data, sehingga klasifikasi dapat dilakukan dengan baik.[11]

Salah satu kekuatan utama dari SVM adalah kesederhanaan dan fleksibilitasnya yang relatif untuk mengatasi berbagai masalah klasifikasi, SVM secara khusus memberikan kinerja prediksi yang seimbang, bahkan dalam studi di mana ukuran sampel mungkin terbatas. Kernel yang umum digunakan termasuk linear, polynomial, dan *radial basis function* (RBF). Dengan teknik ini, SVM dapat menyelesaikan masalah klasifikasi yang kompleks, seperti deteksi wajah, pengenalan karakter, dan prediksi penyakit.[11]

Namun, meskipun SVM sangat efektif untuk tugas klasifikasi, algoritma ini memiliki beberapa keterbatasan. Salah satunya adalah kebutuhan untuk memilih kernel yang tepat dan parameter yang sesuai, yang dapat mempengaruhi kinerja model secara signifikan. Selain itu, SVM membutuhkan komputasi yang cukup besar, terutama ketika bekerja dengan dataset yang sangat besar, karena SVM memerlukan operasi matriks yang mahal dalam proses pelatihannya.

f. Klasifikasi

Klasifikasi adalah Teknik penambangan data prediktif yang menggunakan hasil yang diketahui dari kumpulan data yang berbeda untuk membuat prediksi tentang data nilai [12]. Klasifikasi termasuk kedalam supervised learning karena dalam proses klasifikasi terdapat proses pembelajaran dengan data lampau. Dalam klasifikasi, model

pelatihan yang dibangun berdasarkan data yang sudah diberi label digunakan untuk memetakan data input ke dalam kelas tertentu. Secara umum, klasifikasi dapat dilakukan dengan menggunakan berbagai algoritma, termasuk pohon keputusan, algoritma k-NN (k-Nearest Neighbor), SVM, dan Random Forest.[13]

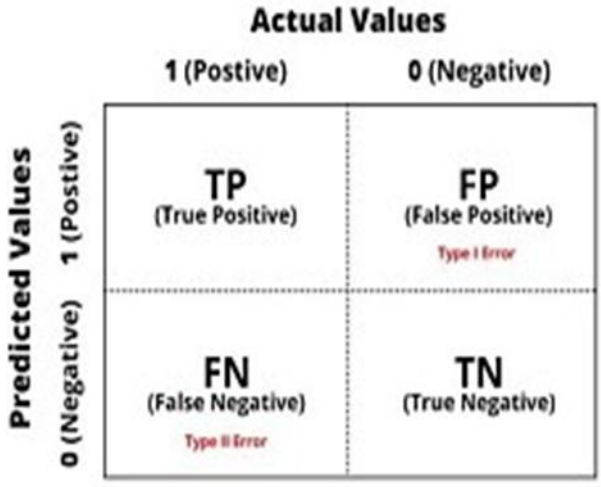
Proses klasifikasi terdiri dari dua tahap utama: pelatihan (training) dan pengujian (testing). Pada tahap pelatihan, model dibangun dengan menggunakan data pelatihan yang telah diberi label. Setelah model dilate h, pada tahap pengujian, model diuji dengan data yang belum pernah dilihat sebelumnya untuk mengevaluasi kinerjanya. Evaluasi kinerja model klasifikasi biasanya dilakukan dengan menggunakan metrik seperti akurasi, precision, recall, dan F1-score, yang masing-masing memberikan gambaran yang berbeda tentang kemampuan model dalam mengklasifikasikan data dengan benar.

Klasifikasi digunakan dalam berbagai bidang, termasuk dalam pengenalan pola (pattern recognition), diagnosis medis, dan analisis sentimen, serta dalam aplikasi lain yang memerlukan keputusan berdasarkan kategori, seperti klasifikasi email spam atau non-spam.

g. *Metrix Evaluation*

Untuk menilai kinerja algoritma yang digunakan dalam penelitian ini, yaitu *Random Forest*, C5.0, dan *Support Vector Machine* (SVM), evaluasi model dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* adalah metode analisis yang menyajikan informasi tentang prediksi model terhadap data uji dan kondisi sebenarnya.

Matriks ini sangat relevan dalam konteks penelitian prediksi risiko penyakit *liver,* yang melibatkan data kategorikal (misalnya, berisiko atau tidak berisiko).



Keterangan :

1) TP (True Positives) : Jumlah kasus yang benar-benar positif dan diprediksi sebagai positif oleh model.

2) FP (False Positives) : Jumlah kasus yang sebenarnya negatif tetapi diprediksi sebagai positif oleh model.

3) FN (False Negatives): Jumlah kasus yang sebenarnya positif tetapi diprediksi sebagai negatif oleh model.

4) TN (True Negatives) : Jumlah kasus yang benar-benar negatif dan diprediksi sebagai negatif oleh model.

Berdasarkan Confusion Matrix, Berikut Rumus untuk menghitung matrik evaluasi klasifikasi:

Pada Tabel 1 TP adalah True Positive, TN adalah True Negative, FP adalah *False Positive* dan FN adalah *False Negative*. Pada penelitian ini performa klasifikasi yang akan dihitung adalah *accuracy*, *precision,* dan *recall*. rumus perhitungan *accuracy*, *precision,* dan *recall.*

a) Accuracy merupakan perhitungan untuk mengukur persentase prediksi yang benar dari keseluruhan data.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

b) Precision merupakan perhitungan untuk mengukur akurasi dan prediksi positif.

$$\textbf{Precision} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FP}}$$

c) Recall adalah mengukur sensitivitas dari model, yaitu seberapa baik model dalam mendeteksi kelas positif.

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}}$$

d) F1-Score adalah perhitungan rata-rata harmonis dari precision dan recall, yang berguna jika terdapat ketidakseimbangan kelas.

$$\textbf{F1 Score} = \textbf{2} \times \frac{\textbf{Precision} \times \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}}$$

h. *Rapid Miner*

Rapidminer merupakan salah satu *software* yang sering digunakan untuk melakukan pengolahan data mining. *RapidMiner* dirancang untuk memungkinkan pengguna melakukan pemodelan data dan analisis tanpa memerlukan banyak keterampilan pemrograman. Dengan menggunakan antarmuka visual yang intuitif, pengguna dapat

mengimpor data, melakukan preprocessing, membangun model pembelajaran mesin, dan mengevaluasi hasilnya dengan mudah.[14]

RapidMiner mendukung berbagai teknik pembelajaran mesin, mulai dari algoritma klasifikasi seperti *Random Forest*, C5.0, dan SVM, hingga teknik clustering dan analisis regresi. Platform ini juga menyediakan alat untuk validasi model, pemilihan fitur, dan visualisasi data. Salah satu keuntungan utama dari RapidMiner adalah kemampuannya untuk menangani *workflow* yang kompleks, memungkinkan pengguna untuk mengotomatisasi dan mengoptimalkan seluruh proses analisis data. Dengan modul-modul yang dapat diperluas, RapidMiner juga dapat digunakan untuk aplikasi spesifik di berbagai bidang, seperti prediksi keuangan, analisis pasar, dan pemrosesan bahasa alami.

Namun, meskipun RapidMiner menawarkan antarmuka grafis yang memudahkan penggunaan, bagi pengguna yang lebih berpengalaman atau yang bekerja dengan dataset yang sangat besar, mungkin lebih memilih menggunakan alat berbasis kode seperti Python atau R, yang menawarkan lebih banyak fleksibilitas dan control.