

## PENERAPAN *DECISION TREE* DENGAN PENYEIMBANGAN DATA *IMBALANCE* MENGGUNAKAN *UPSAMPLING* DALAM PREDIKSI PENYAKIT *LIVER*

Agung Fazriansyah<sup>1\*</sup>; Yuris Alkhalifi<sup>2</sup>; Ainun Zumarniansyah<sup>3</sup>

Sistem Informasi<sup>1,3</sup>, Informatika<sup>2</sup>

Universitas Bina Sarana Informatika, Jakarta, Indonesia<sup>1,2,3</sup>

[www.bsi.ac.id](http://www.bsi.ac.id)<sup>1,2,3</sup>

[agung.fzr@bsi.ac.id](mailto:agung.fzr@bsi.ac.id)<sup>1</sup>, [yuris.yak@bsi.ac.id](mailto:yuris.yak@bsi.ac.id)<sup>2</sup>, [ainun.azm@bsi.ac.id](mailto:ainun.azm@bsi.ac.id)<sup>3</sup>

(\*) Corresponding Author



Ciptaan disebarluaskan di bawah Lisensi Creative Commons Atribusi-NonKomersial 4.0 Internasional.

**Abstract**— *Acute liver disease has a significant impact on liver function and is often only detected at an advanced stage due to the lack of patient awareness for early examination. One of the challenges in treating liver disease is the delay in diagnosis, where many patients do not notice the early symptoms until their condition has worsened. Therefore, a predictive system is needed that can identify liver disease patients early on, allowing for regular check-ups and timely treatment. In this study, a classification model was developed using a machine learning approach, specifically the Decision Tree algorithm, by balancing the data in the minority class through upsampling. The research results show that this model is capable of predicting liver disease status with an accuracy rate of 89.22%, a recall of 88.45%, a precision of 83.21%, and an f1-score of 85.78%. In addition, the ROC-AUC value of 0.89 is categorized as a good classification. This model achieved a higher accuracy score than other studies with similar datasets. This system is expected to help improve early detection and expedite the treatment of liver disease patients.*

**Keywords:** *classification, decision tree, imbalance data, liver disease.*

**Abstrak**— Penyakit *liver* (hati) akut memiliki dampak signifikan terhadap fungsi hati dan sering kali baru terdeteksi pada tahap lanjut karena kurangnya kesadaran pasien untuk melakukan pemeriksaan dini. Salah satu tantangan dalam pengobatan penyakit hati adalah keterlambatan diagnosis, di mana banyak pasien tidak menyadari gejala awal hingga kondisi mereka sudah memburuk. Untuk itu, diperlukan sistem prediksi yang mampu mengidentifikasi penderita penyakit hati sejak dini, sehingga memungkinkan dilakukan pemeriksaan rutin dan pengobatan tepat waktu. Dalam penelitian ini, dikembangkan model klasifikasi dengan pendekatan *machine learning* yakni menggunakan algoritma *Decision Tree* dengan menyeimbangkan data pada kelas minoritas dengan menggunakan *upsampling*. Hasil penelitian menunjukkan bahwa model ini mampu memprediksi status penyakit hati dengan tingkat akurasi mencapai 89.22%, *recall* sebesar 88.45%, *precision* sebesar 83.21% dan *f1-score* sebesar 85.78%. Selain itu, nilai ROC-AUC sebesar 0.89 yang dikategorikan sebagai *good classification*. Model ini mendapatkan nilai akurasi yang lebih tinggi dari penelitian lainnya dengan *dataset* yang serupa. Sistem ini diharapkan dapat membantu meningkatkan deteksi dini dan mempercepat penanganan pasien penyakit hati.

**Kata kunci:** *klasifikasi, decision tree, data imbalance, penyakit hati.*

### PENDAHULUAN

*Liver* (hati) adalah organ manusia yang perannya sangat penting karena organ ini berfungsi untuk metabolisme serta produksi protein penting dalam tubuh, namun jumlah orang yang menderita penyakit *liver* (hati) meningkat setiap tahun, gaya hidup yang tidak sehat seperti pola makan dan

faktor risiko lainnya menjadi penyebab utama (Aldana & Wibowo, 2024). dengan meningkatnya penyakit hati oleh karena itu pentingnya memahami faktor risiko dan pengobatan guna untuk mencegah kerusakan hati secara efisien. Kerusakan pada hati dapat menyebabkan gagal hati, kanker hati dan penurunan fungsi hati atau kerusakan jaringan hati lebih lanjut (Desiani, 2022).

Di era kemajuan teknologi saat ini, penggunaan canggih dan teknik pemrosesan data modern memungkinkan deteksi dan diagnosa penyakit hati lebih akurat. Deteksi dini gejala penyakit hati sangat penting untuk meningkatkan peluang kelangsungan hidup pasien (Cahyanto et al., 2024; Ghalib & Wasilah, 2023). Oleh karena itu, penelitian ini difokuskan pada identifikasi model yang paling tepat untuk mengklasifikasikan data penyakit hati (Cahyanti et al., 2023). Sehingga deteksi penyakit ini dapat dilakukan lebih awal mengembangkan metode-metode yang efektif dapat memungkinkan para tenaga medis untuk mendeteksi penyakit hati lebih awal dan menerapkan intervensi medis yang lebih cepat dan akurat, yang pada akhirnya meningkatkan kualitas hidup manusia (Prasetyo et al., 2024). Seperti mengetahui seorang pasien memiliki penyakit hati atau tidak (Fadri, 2023). Dalam upaya meningkatkan akurasi prediksi dalam proses deteksi, teknologi pembelajaran machine learning telah mulai diterapkan di bidang medis. Salah satu algoritma yang digunakan dalam analisis data adalah *Decision Tree* (Raharja et al., 2024). Algoritma ini dapat digunakan untuk menganalisa berbagai faktor risiko dan gejala terkait penyakit hati, serta memberikan keputusan yang diagnostik yang akurat. Meskipun demikian dalam penerapannya, algoritma ini perlu dioptimalkan dan disesuaikan dengan karakteristik data yang beragam.

Penelitian (Dritsas & Trigka, 2023) membandingkan berbagai model klasifikasi dan teknik visualisasi yang digunakan untuk memprediksi penyakit hati dengan pemilihan fitur. Penelitian ini menggunakan 14 komparasi algoritma antara lain *Naïve Bayes*, *SVM*, *Logistic Regression*, *ANN*, *J48*, *RT*, *RepTree*, *Random Forest*, *RotF*, *kNN*, *AdaBoostM1*, *Stacking*, *Soft Voting*, dan *Bagging*. Hasil penelitian ini menunjukkan bahwa dari beberapa model algoritma yang diuji, algoritma *Soft Voting* mendapatkan hasil akurasi tertinggi yakni 80,1%, dengan nilai *recall* dan *f1-score* sebesar 80,1%, *precision* sebesar 80,4% dan *AUC* 88,4%. Eksperimen dan analisis komparatif menunjukkan bahwa ini meningkatkan akurasi klasifikasi, mengurangi waktu klasifikasi, dan membantu prediksi penyakit secara lebih efisien.

Penelitian lainnya dilakukan oleh (He, 2023), beberapa model algoritma digunakan dalam penelitian ini antara lain *Logistic Regression* (LR), *Decision Tree* (DT), *Random Forest* (RF), *Gradient Boosted Decision Tree* (GBDT), *XGBoost* (XGB), *LightGBM* (LGBM), and *K-Nearest Neighbors* (KNN) digunakan dalam penelitian ini untuk membangun model prediksi risiko penyakit hati. Akurasi LR, DT, RF, GBDT, XGB, LGBM, dan KNN dapat diperoleh dari eksperimen sebagai 68,67%, 76,51%, 81,93%,

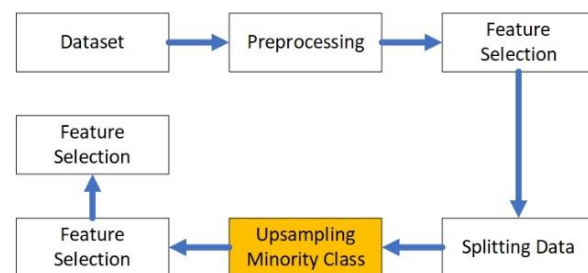
78,31%, 76,51%, 84,13%, dan 75,90%. Dari akurasi tersebut, nilai akurasi tertinggi diperoleh sebesar 84,13% dalam menggunakan algoritma LGBM. Model ini membantu dokter membuat keputusan klinis yang lebih baik dan meningkatkan kelangsungan hidup pasien.

Penelitian lainnya (Jeyalakshmi & Rangaraj, 2021) melakukan penelitian untuk memperkenalkan metode yang memungkinkan prediksi penyakit hati yang akurat dan dapat diandalkan melalui penggunaan metode pembelajaran mendalam yang disesuaikan. Penelitian ini menggunakan *Modified Convolutional Neural Network based Liver Disease Prediction System* (MCNN-LDPS) yang diperkenalkan untuk hasil prediksi penyakit hati yang akurat. Hasil penelitiannya menunjukkan bahwa metode MCNN-LDPS yang diuji menghasilkan akurasi sebesar 86,70%, *f1-score* sebesar 70,02%, *precision* sebesar 84,35% dan *recall* sebesar 59,85%.

Berdasarkan beberapa penelitian sebelumnya tersebut, maka pada penelitian ini melakukan penerapan algoritma dengan menggunakan *Decision Tree* untuk menghasilkan prediksi dari penyakit liver dengan menggunakan data publik khusus untuk mendeteksi penyakit liver atau tidak untuk meningkatkan keakuratan indentifikasi penyakit liver dengan menyeimbangkan data yang *imbalance* (tidakimbang) dengan menggunakan *Upsampling Minority Class*.

## BAHAN DAN METODE

Pada setiap penelitian terdapat metode yang menjelaskan setiap tahapan penelitiannya. Adapun metode pada penelitian ini dapat dilihat pada Gambar 1.



Sumber: (Hasil Penelitian, 2024)

Gambar 1. Metode Penelitian

### Dataset

Penelitian ini menggunakan data publik dari Kaggle. *Dataset* ini label kelas yang digunakan untuk membagi kelompok menjadi pasien hati (penyakit hati) atau tidak (tidak ada penyakit). Adapun jumlah data pada *dataset* ini berjumlah 583 data

baris dan 11 kolom. Dataset ini terdapat 2 label yang terdiri dari 416 data terdeteksi penyakit hati (*Desease*) dan 167 data yang tidak mempunyai penyakit hati (*No Desease*). Dari total data, terdapat 441 data pasien laki-laki dan 142 data pasien perempuan lalu untuk setiap pasien yang usianya melebihi 89 terdapat sebagai yang berusia 90 (UCI Machine Learning & Crawford, 2018).

**Preprocessing**

Sebelum melakukan tahapan eksperimen, dilakukan *preprocessing* ini adalah untuk memastikan bahwa data memenuhi persyaratan model dan memaksimalkan performa model (Rina et al., 2024). Adapun tahapan *preprocessing* pada penelitian ini antara lain:

1. *Handling Missing Values*

Tahapan ini merupakan proses penting dalam *preprocessing* data yang bertujuan untuk menemukan dan menangani nilai yang hilang dalam kumpulan data. Nilai yang hilang dapat terjadi karena berbagai alasan, seperti kesalahan input data, data yang tidak tersedia, atau kendala teknis selama pengumpulan data (Dhimas Irnawan et al., 2021). Mengabaikan nilai yang hilang dapat menyebabkan bias atau kesalahan dalam model pembelajaran mesin, jadi penting untuk menanganinya dengan benar.

2. *Feature Selection*

Tahapan ini untuk memilih fitur (kolom atau variabel dalam *dataset*) yang paling relevan dan penting untuk digunakan dalam pelatihan model pembelajaran mesin (Armaya, 2024). Dengan memilih hanya fitur yang relevan, kita dapat meningkatkan efisiensi, akurasi, dan interpretabilitas model, sekaligus mengurangi kompleksitas dan risiko *overfitting*.

**Splitting Data**

Saat data sudah bersih dan relevan dengan penelitian, data akan dilakukan pembagian data (*split data*). Untuk pelatihan, validasi, dan pengujian model, proses membagi *dataset* menjadi beberapa subset. Proses ini sangat penting untuk mengevaluasi kinerja model dan memastikan bahwa model dapat digeneralisasi dengan baik ke data baru (Hidayat et al., 2024).

**Upsampling Minority Class**

Metode penyeimbangan data, juga dikenal sebagai *data balancing*, digunakan ketika *dataset* tidak seimbang, di mana satu kelas memiliki jumlah sampel yang lebih sedikit (kelas minoritas), ditambahkan jumlah sampelnya sehingga jumlah sampelnya mendekati atau sama dengan jumlah

sampel kelas mayoritas (Singgalen, 2022). Tujuan utamanya adalah membuat model pembelajaran mesin dapat belajar pola lebih baik dari kelas minoritas daripada hanya bias pada kelas mayoritas.

**Modelling**

Model yang akan digunakan pada penelitian ini adalah model algoritma *Decision Tree C4.5*. *Decision Tree C4.5* merupakan sebuah diagram alir yang menyerupai struktur pohon. Setiap node internal menunjukkan pengujian terhadap suatu atribut, setiap cabang menunjukkan output dari pengujian tersebut, dan node daun menunjukkan distribusi kelas atau kelas-kelas (Qadrini et al., 2021). Algoritma *Decision Tree C4.5* juga menangani kasus di mana sebuah atau lebih atribut tidak memiliki nilai. Namun, atribut kelas tidak boleh kosong, hanya diskret (Nasrullah, 2021).

**Evaluasi Model**

Untuk memastikan bahwa model akurat, dapat diandalkan, dan dapat digeneralisasi dengan baik pada data uji, proses evaluasi model dilakukan untuk menilai kemampuan model untuk memprediksi label pada *dataset*. Metode evaluasi model yang akan digunakan pada penelitian ini adalah tabel *confusion matrix*, nilai akurasi, *precision*, *recall*, *f1-score*, dan kurva ROC-AUC.

1. *Confusion Matrix*

*Confusion Matrix* adalah metode evaluasi dalam perhitungan akurasi *data mining*, pendekatan ini berbentuk tabel untuk mengklasifikasikan jumlah data uji yang benar dan salah. (Normawati & Prayogi, 2021). Metode ini dapat dijadikan acuan perhitungan lainnya, seperti akurasi, *precision*, *recall*, *f1-score* dan lainnya. *Confusion Matrix* dapat dilihat pada Tabel 1.

Tabel 1. *Confusion Matrix*

	Aktual Positif	Aktual Negatif
Prediksi Positif	TP	FN
Prediksi Negatif	FP	TN

Sumber: (Normawati & Prayogi, 2021)

Keterangan:

*True Positive (TP)* : Banyak data nilai sebenarnya adalah positif dengan kelas prediksi sebagai nilai positif.

*False Positive (FP)* : Banyak nilai sebenarnya negatif, tetapi kelas prediksi nilainya positif.

*False Negative (FN)* : Banyak data nilai sebenarnya adalah positif, sementara kelas prediksi memiliki nilai negatif.

*True Negative* (TN) : Banyak data nilai sebenarnya adalah negatif diklasifikasikan dengan nilai negatif.

## 2. Accuracy

*Accuracy* atau Akurasi adalah salah satu cara untuk menguji algoritma berdasarkan seberapa dekat nilai prediksi dengan nilai sebenarnya (Karnadi & Handhayani, 2024). Akurasi ditunjukkan pada persamaan 1.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (1)$$

## 3. Precision

*Precision* atau presisi merupakan salah satu cara untuk menguji algoritma adalah dengan membandingkan jumlah data yang benar yang diambil sistem dengan jumlah seluruh data yang benar maupun salah. Metode ini dikenal sebagai pengujian *precision* atau presisi (Karnadi & Handhayani, 2024). Presisi ditunjukkan pada persamaan 2.

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

## 4. Recall

*Recall* adalah salah satu cara untuk menguji algoritma adalah dengan membandingkan jumlah data benar yang diterima sistem dengan jumlah total data benar yang diambil atau tidak diambil system (Karnadi & Handhayani, 2024). *Recall* ditunjukkan pada persamaan 3.

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

## 5. F1-score

*F1-score* adalah metrik evaluasi dalam pembelajaran mesin yang digunakan untuk mengevaluasi keseimbangan antara *precision* dan *recall*. *F1-score* merupakan metrik harmonik dari *precision* dan *recall*, dan memberikan nilai tunggal yang menunjukkan kinerja model pada tugas klasifikasi, terutama dalam kasus di mana distribusi kelas dataset tidak seimbang (Karnadi & Handhayani, 2024). *F1-score* ditunjukkan pada persamaan 4.

$$f1 - score = 2x \frac{precision*recall}{precision+recall} \times 100\% \quad (4)$$

## 6. ROC-AUC

ROC (*Receiver Operating Characteristic*) adalah grafik dua dimensi yang menggunakan garis horizontal *false* positif dan vertikal *true* positif (Firmansyah et al., 2024). Sedangkan AUC (*Area Under Curve*) adalah seberapa baik model dapat membedakan kelas positif dan negatif diukur oleh

area di bawah kurva ROC (AUC). ROC-AUC ini bermanfaat untuk dataset tidak seimbang karena berfokus pada kemampuan model untuk membedakan kelas positif dan negatif terlepas dari distribusi kelas. Terdapat 5 tingkat akurasi pada AUC, yang dapat dilihat pada Tabel 2.

Tabel 2. Tabel Klasifikasi Akurasi AUC

No	Nilai	Keterangan
1	90% - 100%	<i>Excellent Classification</i>
2	80% - 90%	<i>Good Classification</i>
3	70% - 80%	<i>Fair Classification</i>
4	60% - 70%	<i>Poor Classification</i>
5	50% - 60%	<i>Failure Classification</i>

Sumber: (Yunitasari et al., 2021)

## HASIL DAN PEMBAHASAN

### Dataset

Jumlah data pada *dataset* ini berjumlah 583 data baris dan 11 kolom. Sebanyak 416 data terdeteksi penyakit hati '*Deasease*' dan 167 data berlabel '*No Desease*'. Adapun deksripsi dari *dataset* ini dapat dilihat pada Tabel 3.

Tabel 3. Deskripsi Dataset

No	Nama Atribut	Tipe	Deskripsi
1	<i>Age</i>	Numerik	Usia Pasien yang tidak dan terkena penyakit <i>Liver</i>
2	<i>Gender</i>	Objek	Berisi Keterangan Pria dan Wanita
3	<i>Total Bilirubin</i>	Float	Total pigmen berwarna kuning yang terbentuk dari penguaraan sel darah merah
4	<i>Direct_Bilirubin</i>	Float	<i>Bilirubin</i> langsung
5	<i>Alkaline_Phosphatase</i>	Numerik	<i>Alkali Fosfatase</i>
6	<i>Alamine Aminotransferase</i>	Numerik	<i>Alamine Aminotransferase</i>
7	<i>Aspartate_Aminotransferase</i>	Numerik	<i>Aspartate_Aminotransferase</i>
8	<i>Total_Protiens</i>	Float	Total Protein
9	<i>Albumin</i>	Float	<i>Albumin</i>
10	<i>Albumin_and_Globulin_Ratio</i>	Float	Rasio <i>Albumin</i> dan <i>Globulin</i>
11	<i>Dataset</i>	Numerik	<i>Dataset</i> : field yang digunakan untuk membagi data menjadi dua set (pasien dengan penyakit hati, atau tidak ada penyakit)

Sumber: (UCI Machine Learning & Crawford, 2018)

### Preprocessing

Untuk tahapan *processing* pada penelitian ini, akan menggunakan 2 tahapan antara lain *Handling Missing Values* dan *Feature Selection*.

1. Handling Missing Values

Pada tahapan ini terdapat 4 nilai yang kosong yang akan dihapus yakni pada kolom 'Albumin\_and\_Globulin\_Ratio', sehingga tidak ada lagi kolom yang kosong seperti yang terlihat pada Gambar 3.

```
[21]: liver_data.isnull().sum()
Age 0
Gender 0
Total_Bilirubin 0
Direct_Bilirubin 0
Alkaline_Phosphatase 0
Alamine_Aminotransferase 0
Aspartate_Aminotransferase 0
Total_Protiens 0
Albumin 0
Albumin_and_Globulin_Ratio 4
Disease_Status 0
dtype: int64

Albumin_and_Globulin_Ratio has 4 na values

[22]: liver_data.isnull().sum()
Age 0
Gender 0
Total_Bilirubin 0
Direct_Bilirubin 0
Alkaline_Phosphatase 0
Alamine_Aminotransferase 0
Aspartate_Aminotransferase 0
Total_Protiens 0
Albumin 0
Albumin_and_Globulin_Ratio 0
Disease_Status 0
dtype: int64

No missing values
```

Sumber: (Hasil Penelitian, 2024)  
Gambar 3. Check Handle Missing Values

2. Feature Selection

Pemilihan fitur dilakukan agar pemrosesan data dengan model dapat berjalan dengan hasil yang valid dengan menghapus fitur label 'Gender' karena tidak relevan dan label 'Desease Status' yang akan digunakan sebagai label prediksi, sehingga data yang tersisa dapat dilihat pada Gambar 5.

```
#Drop Disease Status, also drop Gender so we can normalize other data
X = liver_data.drop(['Disease Status', 'Gender'], axis = 1)
y = liver_data['Disease Status']

X.head()
```

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphatase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio
0	65	0.7	0.1	127	16	18	6.6	3.3	0.00
1	62	1.0	0.5	400	84	100	7.8	3.2	0.74
2	61	7.1	4.1	400	80	88	7.8	3.3	0.89
3	62	1.0	0.4	102	14	20	6.8	3.4	1.00
4	72	3.9	2.0	100	27	69	7.3	2.4	0.40

Sumber: (Hasil Penelitian, 2024)  
Gambar 5. Seleksi Fitur

Splitting Data

Dataset akan dibagi menjadi 80% data untuk latihan (training data) dan 20% data untuk data uji (test data) yang dapat dilihat pada Gambar 6.

Split the data into train and test data

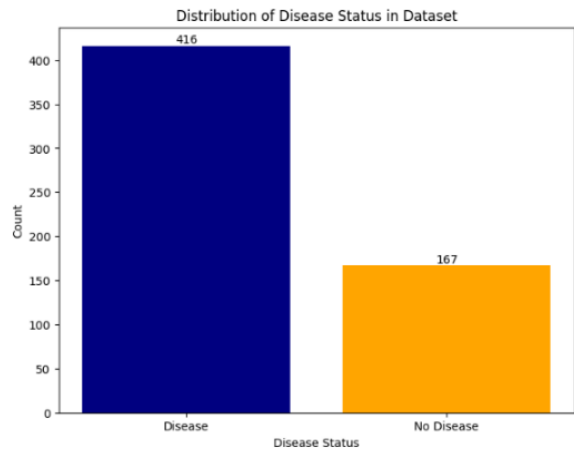
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)
```

Sumber: (Hasil Penelitian, 2024)  
Gambar 6. Splitting Data

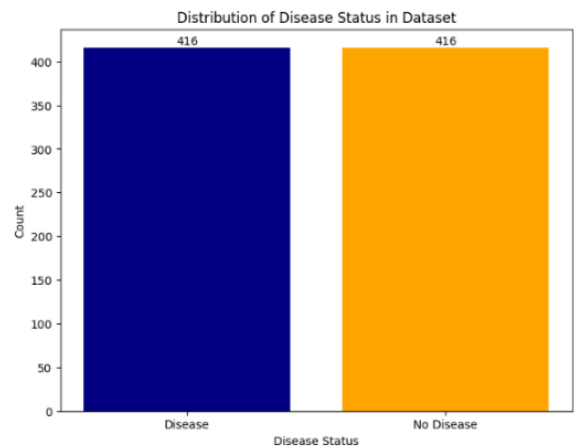
Upsampling Minority Class

Dataset ini mempunyai data yang tidak seimbang (imbalance data) sehingga perlu diseimbangkan terlebih dahulu agar mengatasi Bias terhadap kelas mayoritas dan meningkatkan kemampuan model dalam mendeteksi kelas

minoritas. Label No Disease akan diseimbangkan dengan label Disease. Adapun perbandingan sebelum dan sesudah dilakukan penanganan imbalanced data dapat dilihat pada Gambar 7 & 8.



Sumber: (Hasil Penelitian, 2024)  
Gambar 7. Data Tidak Seimbang (Imbalanced Data)

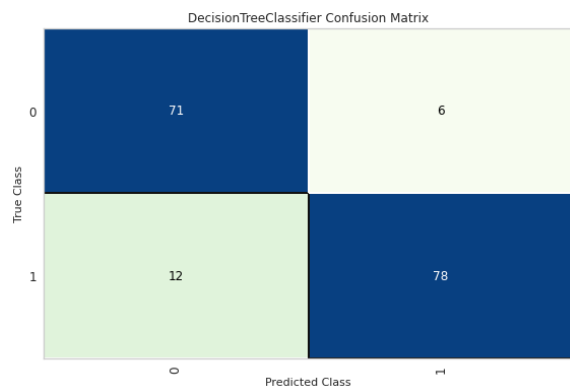


Sumber: (Hasil Penelitian, 2024)  
Gambar 8. Data Seimbang (Balanced Data)

Seperti yang terlihat pada Gambar 7, dataset masih orisinal dan belum dilakukan Upsampling Minority Class, dan setelah dilakukan tahapan Upsampling Minority Class dataset yang memiliki data minor atau lebih sedikit yakni label No Disease lalu disamakan jumlah datanya dengan jumlah yang lebih banyak yakni label Disease yang berjumlah 416. Sehingga data dari kedua label menjadi seimbang dan masing-masing label mempunyai data sejumlah 416 seperti pada Gambar 8.

Evaluate Model

Setelah data dilakukan eksperimen, langkah selanjutnya adalah mengevaluasi model agar dapat melihat hasil yang sudah dilakukan.



Sumber: (Hasil Penelitian, 2024)  
Gambar 9. Confusion Matrix

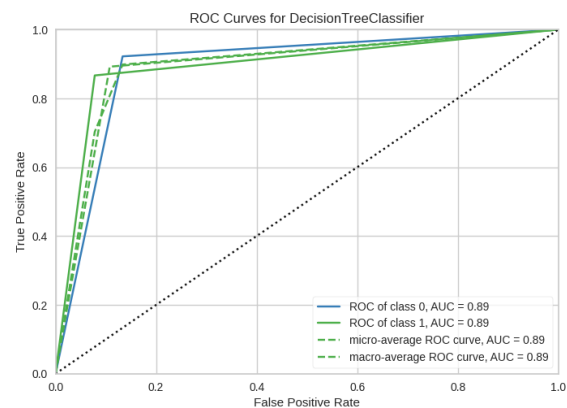
Confusion Matrix yang terlihat pada Gambar 9 menunjukkan bagaimana model *Decision Tree Classifier* bekerja dalam mengkategorikan data ke dalam tugas klasifikasi biner. Model menemukan 71 sampel negatif yang benar (*True Negatives*) dan 78 sampel positif yang benar (*True Positives*). Namun, ada beberapa kesalahan: enam sampel negatif yang salah diprediksi sebagai positif dan dua belas sampel positif yang salah diprediksi sebagai negatif. Meskipun terdapat beberapa kesalahan dalam mengklasifikasikan data, matriks ini menunjukkan bahwa model mendeteksi kedua kelas dengan baik.

Tabel 4. Hasil Pengujian Model Decision Tree

Akurasi	Recall	Precision	f1-score
89,22%	86,67%	92,68%	89,66%

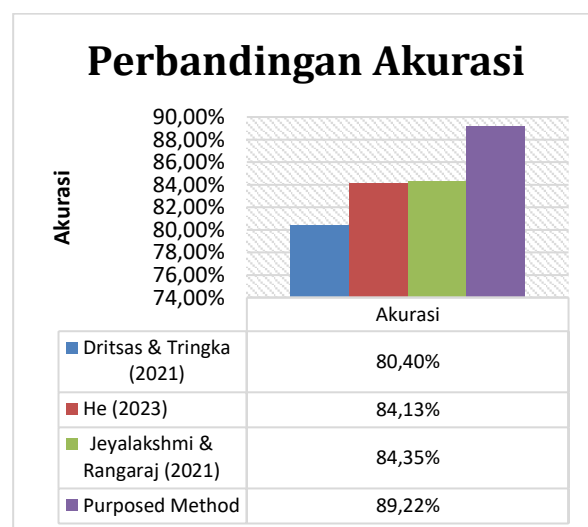
Sumber: (Hasil Penelitian, 2024)

Kinerja model *Decision Tree* untuk menangani ketidakseimbangan kelas pada data yang telah di-*upsampling* ditunjukkan pada Tabel 4. Akurasi model sebesar 89,22%, menunjukkan bahwa model mampu memprediksi sebagian besar sampel dalam dataset dengan benar. *Recall* model sebesar 86,67%, menunjukkan bahwa model mampu mendeteksi kasus positif dengan baik, memastikan bahwa sebagian besar sampel positif ditemukan dengan benar. *Precision* model sebesar 92,86%, menunjukkan bahwa sekitar 83,21% dari semua prediksi positif yang dibuat oleh model adalah benar. Selain itu, nilai *f1-score* sebesar 89,66% menunjukkan keseimbangan yang baik antara *recall* dan *precision*, yang menunjukkan bahwa model memiliki kinerja yang cukup baik untuk menangani ketidakseimbangan data.



Sumber: (Hasil Penelitian, 2024)  
Gambar 11. Grafik ROC-AUC

Grafik ROC-AUC yang terlihat pada Gambar 11 menunjukkan kinerja model *Decision Tree* dalam membedakan kelas positif dan negatif pada tugas klasifikasi biner. Hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) untuk berbagai nilai ambang digambarkan dalam grafik ini. Untuk masing-masing kelas, area di bawah kurva adalah 0,89, yang menunjukkan performa model yang sangat baik dan dikategorikan sebagai *good classification*. Kurva ROC berada di sudut kiri atas grafik, menunjukkan bahwa model memiliki tingkat Tanggapan Positif Asli yang tinggi dengan tingkat Tanggapan Negatif Positif yang rendah, yang merupakan tingkat yang ideal untuk kebanyakan kasus klasifikasi. Pada grafik, garis diagonal menunjukkan baseline untuk tebakan acak (AUC = 0,5), dan model ini secara signifikan melampaui baseline, menunjukkan kemampuan yang jauh lebih besar daripada tebakan acak.



Sumber: (Hasil Penelitian, 2024)  
Gambar 12. Grafik Perbandingan Akurasi



Dari Gambar 12, dapat dilihat bahwa jika dibandingkan dengan penelitian terdahulu pada (Jeyalakshmi & Rangaraj, 2021) mendapatkan hasil tertinggi sebesar 80,40%, lalu pada penelitian (He, 2023) mendapatkan hasil akurasi sebesar 84,13%, kemudian pada penelitian (Jeyalakshmi & Rangaraj, 2021) mendapatkan hasil akurasi sebesar 87%. Adapun dari pengujian yang sudah dilakukan pada penelitian ini, model *Decision Tree* yang sudah dilakukan penyeimbangan data menggunakan *upsampling* pada kedua kelas mendapatkan nilai akurasi sebesar 89,22%, dengan kata lain yang melebihi model pada penelitian ini mendapatkan hasil kebaruan pada tingkat akurasinya dari penelitian terdahulu.

### KESIMPULAN

Hasil penelitian ini menunjukkan bahwa model *Decision Tree Classifier* dengan menggunakan teknik *upsampling* berhasil dalam memprediksi penyakit liver pada dataset pasien liver India. Berdasarkan hasil eksperimen, model berhasil mencapai akurasi yang tinggi yakni sebesar 89.22%, *recall* sebesar 88.45%, *precision* sebesar 83.21% dan *f1-score* sebesar 85.78%. Selain itu, nilai ROC-AUC sebesar 0.89 dikategorikan sebagai *good classification* dan mengindikasikan bahwa model memiliki kemampuan yang andal dalam membedakan pasien dengan dan tanpa penyakit liver. Hasil ini sangat membantu dalam pengembangan sistem berbasis kecerdasan buatan untuk aplikasi medis, khususnya dalam memprediksi penyakit liver. Sistem ini dapat membantu tenaga medis membuat keputusan yang lebih baik dengan lebih cepat dan akurat. Namun, saran untuk penelitian selanjutnya langkah seperti *pruning*, optimisasi *hyperparameter*, atau penggunaan algoritma ensemble seperti *Random Forest* dapat dilakukan untuk meningkatkan kinerja model pada penelitian berikutnya.

### REFERENSI

- Aldana, S., & Wibowo, J. S. (2024). Penerapan Data Mining Terhadap Klasifikasi Pasien Penderita Penyakit Liver Menggunakan Metode K-Nearest Neighbor. *Progresif: Jurnal Ilmiah Komputer*, 20(1), 124–132. <https://doi.org/10.35889/progresif.v20i1.1376>
- Armaya, A. M. R. (2024). Pengaruh Feature Selection Dan Feature Extraction Dalam Peningkatan Akurasi Klasifikasi Kebakaran Hutan. *JuTI "Jurnal Teknologi Informasi"*, 3(1), 13. <https://doi.org/10.26798/juti.v3i1.1039>
- Cahyanti, F. L. D., Sarasati, F., Astuti, W., & Firasari, E. (2023). Klasifikasi Data Mining Dengan Algoritma Machine Learning Untuk Prediksi Penyakit Liver. *Technologia*, 14(2), 134. <https://doi.org/10.31602/tji.v14i2.10093>
- Cahyanto, H. N., Zulkarnain, O., & Rahagia, R. (2024). Pengembangan Deteksi Dini Dan Asuhan Keperawatan Pada Kanker Menggunakan Artificial Intelligence (AI) Berbasis Web. *Prepotif: Jurnal Kesehatan Masyarakat*, 8. <https://doi.org/10.31004/prepotif.v8i3.34374>
- Desiani, A. (2022). Perbandingan Implementasi Algoritma Naïve Bayes dan K-Nearest Neighbor Pada Klasifikasi Penyakit Hati. *SIMKOM*, 7(2), 104–110. <https://doi.org/10.51717/simkom.v7i2.96>
- Dhimas Irnawan, F., Hidayah, I., & Nugroho, L. E. (2021). Metode Imputasi pada Data Debit Daerah Aliran Sungai Opak, Provinsi DI Yogyakarta. In *Jurnal Nasional Teknik Elektro dan Teknologi Informasi* | (Vol. 10, Issue 4). <https://doi.org/10.22146/jnteti.v10i4.2430>
- Dritsas, E., & Trigka, M. (2023). Supervised Machine Learning Models for Liver Disease Risk Prediction. *Computers*, 12(1), 19. <https://doi.org/10.3390/computers12010019>
- Fadri, W. (2023). Jurnal Informasi dan Teknologi Klasifikasi Penyakit Hati dengan Menggunakan Metode Naive Bayes. *Jurnal Informasi Dan Teknologi*, 5(1), 32–37. <https://doi.org/10.37034/jidt.v5i1.230>
- Firmansyah, Y., Kurniawan, R., & Wijaya, Y. A. (2024). Analisis Data Sentimen Pemain Game Role-Playing Game (RPG) Honkai Star Rail dengan Algoritma Naive Bayes. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 6(1). <https://doi.org/10.36499/jinrpl.v6i1.10243>
- Ghalib, F., & Wasilah, W. (2023). Deteksi Dini Kanker Payudara Menggunakan Algoritma K-Nearest Neighbour (KNN) dan Decision Tree C-45. *TEKNIKA*, 17(2), 1–5. <https://doi.org/10.5281/zenodo.8412264>
- He, H. (2023). Research and Application of Different Machine Learning Algorithms in ILPD Risk Prediction Model. *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*, 1330–1334. <https://doi.org/10.1109/ICETCI57876.2023.10176951>
- Hidayat, R., Haris, M., & Simbolon, Z. K. (2024). Implementasi Metode Support Vector Machine (SVM) Pada Klasifikasi Drop Out (DO) Mahasiswa.

- Jeyalakshmi, K., & Rangaraj, R. (2021). Accurate Liver Disease Prediction System using Convolutional Neural Network. *Indian Journal of Science and Technology*, 14(17), 1406–1421. <https://doi.org/10.17485/IJST/v14i17.451>
- Karnadi, B., & Handhayani, T. (2024). Klasifikasi Jenis Buah dengan Menggunakan Metode MobileNetv2 dan Inceptionv3. *Jurnal Eksplorasi Informatika*, 14(1), 35–42. <https://doi.org/10.30864/eksplorasi.v14i1.1067>
- Nasrullah, A. H. (2021). Implementasi Algoritma Decision Tree Untuk Klasifikasi Produk Laris. *Jurnal Ilmiah Ilmu Komputer*, 7(2). <https://doi.org/10.35329/jiik.v7i2.203>
- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. In *Jurnal Sains Komputer & Informatika (J-SAKTI)* (Vol. 5, Issue 2).
- Prasetyo, M. A., Zyen, A. K., & Kusumodestoni, R. H. (2024). Optimasi Algoritma Naive Bayes Berbasis Kernel Untuk Klasifikasi Penyakit Hati. *Jurnal Informatika Teknologi Dan Sains*, 6(3). <https://doi.org/10.51401/jinteks.v6i3.4783>
- Qadrini, L., Seppewali, A., & Aina, A. (2021). Decision Tree Dan Adaboost Pada Klasifikasi Penerima Program Bantuan Sosial. *Jurnal Inovasi Pendidikan*, 7(2), 1959–1966. <https://doi.org/10.47492/jip.v2i7.1046>
- Raharja, A. R., Pramudianto, A., & Muchsam, Y. (2024). Penerapan Algoritma Decision Tree dalam Klasifikasi Data “Framingham” Untuk Menunjukkan Risiko Seseorang Terkena Penyakit Jantung dalam 10 Tahun Mendatang. *Technologia Journal*, 1. <https://doi.org/10.62872/cwgp962>
- Rina, R., Hasan, P. M., Ayu, N., & Adi Saputra, R. (2024). Klasifikasi Keringanan Ukt Mahasiswa Uho Menggunakan K-Nearest Neighbor (KNN). *Jurnal Mahasiswa Teknik Informatika*, 8(6). <https://doi.org/10.36040/jati.v8i6.11757>
- Singgalen, Y. A. (2022). Analisis Sentimen Wisatawan Melalui Data Ulasan Candi Borobudur di Tripadvisor Menggunakan Algoritma Naïve Bayes Classifier. *Building of Informatics, Technology and Science (BITS)*, 4(3). <https://doi.org/10.47065/bits.v4i3.2486>
- UCI Machine Learning, & Crawford, C. (2018). *Indian Liver Patient Records*. Kaggle. <https://www.kaggle.com/datasets/uciml/indian-liver-patient-records>
- Yunitasari, Hopipah, S. H., & Mayasari, R. (2021). Optimasi Backward Elimination untuk Klasifikasi Kepuasan Pelanggan Menggunakan Algoritme k-Nearest Neighbor (k-NN) dan. *Technomedia Journal (TMJ)*, 6(1), 99–110. <https://doi.org/10.33050/tmj.v6i1>