**TESIS**

# KLASIFIKASI PENYAKIT JANTUNG MENGGUNAKAN METODE SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE DAN RANDOM FOREST CLASIFIER


Disusun oleh:

**Nama** : **Hidayat**
**NIM** : **20.51.1392**
**Konsentrasi** : **Informatics Technopreneurship**



**PROGRAM STUDI S2 TEKNIK INFORMATIKA**
**PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA**
**YOGYAKARTA**
**2024**

**TESIS**

# KLASIFIKASI PENYAKIT JANTUNG MENGGUNAKAN METODE SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE DAN RANDOM FOREST CLASIFIER

# HEART DISEASE CLASSIFICATION USING SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE AND RANDOM FOREST CLASSIFIER

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister


Disusun oleh:

**Nama** : **Hidayat**
**NIM** : **20.51.1392**
**Konsentrasi** : **Informatics Technopreneurship**


**PROGRAM STUDI S2 TEKNIK INFORMATIKA**
**PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA**
**YOGYAKARTA**
**2024**

## HALAMAN PENGESAHAN

# KLASIFIKASI PENYAKIT JANTUNG MENGGUNAKAN METODE SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE DAN RANDOM FOREST CLASIFIER

# HEART DISEASE CLASSIFICATION USING SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE AND RANDOM FOREST CLASSIFIER

Dipersiapkan dan Disusun oleh

**Hidayat**

**20.51.1392**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 3 April 2024

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 April 2024
**Rektor**

**Prof. Dr. M. Suyanto, M.M.**
**NIK. 190302001**

## HALAMAN PERSETUJUAN

## KLASIFIKASI PENYAKIT JANTUNG MENGGUNAKAN METODE SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE DAN RANDOM FOREST CLASIFIER

## HEART DISEASE CLASSIFICATION USING SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE AND RANDOM FOREST CLASSIFIER

Dipersiapkan dan Disusun oleh

**Hidayat**

**20.51.1392**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 3 April 2024

| **Pembimbing Utama** | **Anggota Tim Penguji** |
|---|---|
| **Dr. Andi Sunyoto, M.Kom** | **Dhani Ariatmanto, S.Kom., M.Kom., Ph.D.** |
| **NIK. 190302052** | **NIK. 190302197** |
| | **Alva Hendi Muhammad, S.T., M.Eng., Ph.D.** |
| **Pembimbing Pendamping** | **NIK. 190302493** |
| **Hanif Al Fatta, M.Kom., Ph.D** | **Dr. Andi Sunyoto, M.Kom** |
| **NIK. 190302096** | **NIK. 190302052** |

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 April 2024
**Direktur Program Pascasarjana**

**Prof. Dr. Kusrini, M.Kom.**
**NIK. 190302106**

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

**Nama mahasiswa : Hidayat**
**NIM : 20.51.1392**
**Konsentrasi : Informatics Technopreneurship**

Menyatakan bahwa Tesis dengan judul berikut:
**KLASIFIKASI PENYAKIT JANTUNG MENGGUNAKAN METODE SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE DAN RANDOM FOREST CLASIFIER**

Dosen Pembimbing Utama : Dr. Andi Sunyoto, M.Kom
Dosen Pembimbing Pendamping : Hanif Al Fatta, M.Kom., Ph.D

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 3 April 2024
Yang Menyatakan,


Hidayat

## HALAMAN PERSEMBAHAN

Segala puji bagi Allah, Tuhan semesta alam, *Alhamdu lillahi Robbil 'Alamin* Tesis ini selesai dipersembahkan dengan penuh rasa hormat dan terima kasih kepada:

- Allah SWT, atas segala rahmat dan karunia-Nya yang tiada terkira dan takjub, sehingga penulis dapat menyelesaikan tesis ini dengan lancar.
- Untuk Abak dan Amak, Kedua Orang Tua tercinta, yang selalu memberikan doa, motivasi, dan dukungan tiada henti dalam setiap langkah penulis.
- Untuk Keluarga Uda andi, ayuk tri, Da ersa, teti, Kak adi, ni yanti, Sidi dayat, tiara, anak keponakan terimakasih telah mensupport hingga bisa berada pada titik proses ini dan anandaku Aflah Al-Fatih Hidayat.
- Teman seperjuangan diperantauan, kehidupan cita-cita, cinta & terimakasih atas proses ini, semangat, dan do'anya, sukses dan berkah untuk kita semuanya.

Penulis menyadari bahwa karya ilmiah ini masih jauh dari kesempurnaan. Oleh karena itu, penulis menerima dengan terbuka segala kritik dan saran yang membangun demi perbaikan di masa depan.

## HALAMAN MOTTO

**PERBAIKI SHALATMU,**

**MAKA ALLAH AKAN PERBAIKI KUALITAS HIDUPMU**

"Jadikanlah sabar dan shalat sebagai penolongmu. Dan sesungguhnya yang demikian itu sungguh berat, kecuali bagi orang-orang yang khusyu'"

(QS. Al-Baqarah: 45).

## KATA PENGANTAR

Puji syukur kehadirat Allah SWT, atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan karya ilmiah ini dengan lancar. Sholawat serta salam senantiasa tercurah kepada junjungan Nabi Muhammad SAW, beserta keluarga, sahabat, dan pengikutnya hingga akhir zaman.

Penulis menyadari bahwa dalam penyusunan tesis ini masih terdapat banyak kekurangan dan keterbatasan. Oleh karena itu, penulis dengan terbuka menerima kritik dan saran yang membangun demi perbaikan di masa depan.

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

- Allah SWT, atas segala rahmat dan karunia-Nya yang tiada terkira.
- Kedua Orang Tua tercinta, yang selalu memberikan doa, motivasi, dan dukungan tiada henti dalam setiap langkah penulis.
- Dr. Andi Sunyoto, M.Kom, Dr. Hanif Al Fatta, Ph.D atas bimbingan, arahan, dan masukan selama proses penyusunan tesis ini. Seluruh Dosen dan admisi MTI Amikom, atas ilmu pengetahuan dan dedikasinya yang telah penulis terima selama masa perkuliahan.

Penulis berharap karya ilmiah ini dapat bermanfaat bagi pembaca dan memberikan sumbangsih bagi kemajuan ilmu pengetahuan.

Yogyakarta, 16 Mei 2024

Penulis

**DAFTAR ISI**

## DAFTAR TABEL

## DAFTAR GAMBAR

# INTISARI

Penelitian ini bertujuan untuk meningkatkan akurasi dalam klasifikasi penyakit jantung dengan mengintegrasikan Synthetic Minority Over-sampling Technique (SMOTE) dan algoritma Random Forest Classifier. Data yang digunakan berasal dari pasien yang telah didiagnosis dengan penyakit jantung atau tidak. Tahap awal penelitian fokus pada penanggulangan masalah ketidakseimbangan kelas dengan mengaplikasikan SMOTE, menciptakan sampel sintetis dari kelas minoritas. Proses berlanjut dengan normalisasi data menggunakan metode min-max normalisasi, yang diikuti oleh proses klasifikasi menggunakan Random Forest Classifier untuk melatih model dalam melakukan klasifikasi.

Hasil penelitian menunjukkan bahwa pendekatan ini efektif meningkatkan kemampuan model dalam mengidentifikasi kasus penyakit jantung. Evaluasi model menghasilkan akurasi yang lebih baik, mencapai 99%, yang menandai peningkatan 1% dari akurasi penelitian sebelumnya yang mencapai 98%. Dengan demikian, integrasi SMOTE dan Random Forest Classifier membawa kontribusi positif dalam meningkatkan ketepatan diagnosis penyakit jantung pada dataset pasien yang dianalisis. Hasil terbaik ini mencerminkan keberhasilan metode tersebut dalam mengatasi tantangan klasifikasi pada data kesehatan, memperkuat potensi aplikasinya dalam praktik medis untuk meningkatkan deteksi dini penyakit jantung.

Kata kunci: Klasifikasi, Penyakit Jantung, Synthetic Minority Over-sampling Technique (SMOTE), Random Forest Classifier.

## *ABSTRACT*

*This research aims to increase accuracy in heart disease classification by integrating the Synthetic Minority Over-sampling Technique (SMOTE) and the Random Forest Classifier algorithm. The data used comes from patients who have been diagnosed with heart disease or not. The initial stage of the research focused on overcoming the problem of class imbalance by applying SMOTE, creating synthetic samples from minority classes. The process continues with data normalization using the min-max normalization method, followed by a classification process using the Random Forest Classifier to train the model to carry out classification.*

*The research results show that this approach is effective in increasing the model's ability to identify cases of heart disease. The model evaluation resulted in better accuracy, reaching 92%, which marked a 2% increase from the accuracy of previous research which reached 90%. Thus, the integration of SMOTE and Random Forest Classifier brings a positive contribution to increasing the accuracy of heart disease diagnosis in the analyzed patient dataset. These excellent results reflect the method's success in overcoming classification challenges in health data, strengthening its potential application in medical practice to improve early detection of heart disease.*



*Keyword: Classification, Heart Disease, Synthetic Minority Over-sampling Technique (SMOTE), Random Forest Classifier.*

# BAB I

# PENDAHULUAN

## 1.1. Latar Belakang Masalah.

Penyakit kardiovaskular atau penyakit yang disebabkan adanya gangguan pada jantung dan pembuluh darah, penyakit ini masih menjadi ancaman dunia (global threat) dan merupakan penyakit yang berperan utama sebagai penyebab kematian nomor satu di seluruh dunia(RI, 2019). Data Organisasi Kesehatan Dunia (WHO) menyebutkan, lebih dari 17 juta orang di dunia meninggal akibat penyakit jantung dan pembuluh darah(KEMKES, 2017).

Penyakit jantung merupakan penyakit yang memiliki kondisi ketika bagian jantung yang meliputi pembuluh darah jantung, selaput jantung, katup jantung, dan otot jantung mengalami gangguan, penyakit ini disebabkan oleh berbagai hal, seperti sumbatan pada pembuluh darah jantung, peradangan, infeksi, atau kelainan bawaan(alodokter, 2023). Rata-rata biaya pengobatan ketika seseorang terindikasi penyakit jantung ketika melakukan pengecekan jantung memakan biaya 3-4 juta, operasi by pass jantung 150-300 jt, ring jantung 50 hingga 100 jt (Alvin, 2019).

Terdapat beberapa riset terkait penyakit jantung ini dimana riset tersebut mengungkapkan bahwa penyakit jantung ialah salah satu penyakit yang perlu mendapatkan perhatian serius karena Serangan jantung yang parah atau terlambat ditangani bisa menyebabkan beberapa komplikasi berbahaya. Komplikasi tersebut antara lain gangguan irama jantung atau aritmia, gagal jantung, syok kardiogenik, dan henti jantung(Pittara, 2022).

Beberapa cara yang dapat dilakukan untuk dapat membantu para petugas medis dalam menemukan apakah seseorang terindikasi penyakit jantung atau tidak agar ketika pasien yang mengalami penyakit tersebut ini dapat ketahui dengan cepat, salah satunya ialah dengan penggunaan Machine learning, dengan penggunaan ini terbukti mampu untuk dapat menyelesaikan topik klasifikasi, dan optimasi dalam pembuatan sebuah system penyedia layanan kesehatan(Widiastiwi & Ernawati, 2021). Sebagai contoh menangani pasien yang terinfeksi penyakit jantung untuk dapat memprediksi dari data yang dihasilkan oleh industry kesehatan sehingga dapat membantu dan menyelematkan nyawa seseorang dalam jangka panjang, dan paling tidak dapat mempersingkat waktu untuk dapat mengetahui pasien terindikasi penyakit karena terbantukan dengan metode machine learning yang digunakan (Bianto et al., 2020).

Ada beberapa penelitian sebelumnya dengan study kasus yang sama yaitu klasifikasi penyakit jantung, dari penelitian yang ada menggunakan beberapa metode machine learning untuk dapat memprediksi seseorang terindikasi penyakit jantung yaitu random forest classifier, ann, svm, naïve bayes, support vector machine, decision tree dll sebagainya, hasil penelitian sebelumnya memiliki hasil akurasi yang paling baik adalah 98% dengan menggunakan teknik seperti pre-prosessing data, penentuan hyperparameter, kombinasi metode balancing data dll.

Berdasarkan hasil yang didapatkan pada proses indentifikasi masalah, maka topik yang diangkat dalam penelitian ini adalah penyakit jantung, data yang dipakai dalam pengolahan penelitian ini adalah heart disease dataset yang diambil dari kaggel, dataset tersebut mengalami imbalance class dimana jumlah kategori

terindikasi penyakit (1) sebanyak 526 dan tidak terindikasi penyakit (0) sebanyak 499, imbalance kelas dapat mempengaruhi model saat klasifikasi, model hanya dapat menentukan kelas mayoritas dan kemungkinan besar kelas minoritas yang diprediksi, akan diprediksi sebagai kelas mayoritas(Reinert Yosua Rumagit, 2019). Dengan terjadinya imbalance pada dataset, maka diterapkan metode Synthetic Minority Over-Sampling Technique untuk dapat mengatasi dataset yang mempunyai masalah imbalance class, dimana kelas target memiliki jumlah yang lebih kecil dibandingkan dengan kelas target lain(Jason Brownlee, 2020).

Penelitian ini, saat melakukan klasifikasi penyakit jantung metode yang digunakan yaitu algoritma random forest. algoritma tersebut dapat mengurangi masalah overfitting yang sering terjadi pada model yang kompleks, ini dilakukan dengan menggabungkan prediksi dari banyak pohon keputusan yang terkondisi secara independen, yang mencegah model terlalu "menghafal" data pelatihan (Liaw & Wiener, 2002). Random Forest dapat memberikan informasi tentang pentingnya setiap variabel dalam prediksi, dengan menggunakan ukuran penting ini, kita dapat mengidentifikasi variabel yang paling berpengaruh dalam prediksi dan mengambil tindakan yang sesuai(Cutler et al., 2012). Random Forest relatif mudah digunakan dan memiliki parameter yang intuitif. Selain itu, algoritma ini memiliki kemampuan bawaan untuk menangani masalah seperti pengaturan parameter dan penyesuaian model(Prasad et al., 2006).

Penelitian-penelitian sebelumnya dengan study kasus yang sama memiliki hasil akurasi yang paling baik yaitu 98%. Rata-rata, penelitian sebelumnya tidak melakukan langkah-langkah penting dalam memproses data mereka. Misalnya,

tidak melakukan penanganan nilai kosong dalam atribut dataset, yang dapat mengakibatkan bias dalam hasil klasifikasi(Maula et al., 2022). Selain itu, juga tidak menerapkan normalisasi data, yang diperlukan untuk menghindari perbedaan skala yang mungkin ada dalam dataset. Jika perbedaan skala ini tidak diatasi, model machine learning cenderung menghasilkan hasil yang tidak optimal(Irawan & Wahono, 2015). Terakhir, seringkali penelitian sebelumnya tidak mengatasi masalah ketidakseimbangan data dalam dataset mereka. Ini dapat menyebabkan model cenderung hanya memprediksi kelas mayoritas dan mengurangi akurasi klasifikasi secara keseluruhan(Prasetio & Pratiwi, 2015).

Tujuan dari penelitian ini ialah meningkatkatkan hasil akurasi pada klasifikasi penyakit jantung menggunakan metode SMOTE dan machine learning dengan algoritma random forest untuk mendapatkan hasil akurasi yang lebih baik dari akurasi yang dihasilkan sebelumnya. Di dalam penelitian ini diusulkannya penggunaan Teknik preprosessing data seperti replace missing value dan normalisasi data, penggunaan algoritma random forest, metode SMOTE, dan Confusion matrix untuk dapat menilai performa dari model yang di pakai pada proses klasifikasi penyakit jantung ini.

**1.2. Rumusan Masalah.**

Berdasarkan uraian latar belakang diatas, maka dirumuskan suatu masalah yang akan dipecahkan/diselesaikan pada penelitian ini. Rumusan masalah yang diangkat sebagai berikut :

a. Apakah dengan penggunaan metode SMOTE dapat meningkatkan akurasi klasifikasi yang dilakukan oleh metode Random Forest?
b. Apakah penerapan machine learning dengan metode random forest untuk proses klasifikasi dapat meningkatkan akurasi?

**1.3. Batasan Masalah.**

Batasan masalah dari penelitian ini sebagai berikut :

a. Data yang di gunakan dalam penelitian ini adalah data public yang terdapat pada situs kaggle. Data tersebut merupakan data penyakit jantung yang dimiliki oleh David Lapp.
b. Metode yang di gunakan pada proses klasifikasi ini adalah random forest.
c. Metode yang digunakan untuk menangani imbalance data adalah metode Synthetic Minority Over-Sampling Technique (SMOTE).
d. Evaluasi metode yang digunakan Confusion Matrix.
e. Software yang digunakan dalam membantu proses penelitian ini adalah Google Colabolatory dan Microsoft Excel.

**1.4. Tujuan Penelitian.**

Berikut ini merupakan tujuan dari penelitian yang dilakukan :

a. Penerapan metode machine learning dengan algoritma Random Forest pada proses klasifikasi untuk dapat meningkatkan akurasi.

b. Penerapan metode keseimbangan dataset Synthetic Minority Over-Sampling Technique (SMOTE) untuk mengatasi ketidakseimbangan class pada dataset dan mampu meningkatkan akurasi metode.

**1.5. Manfaat Penelitian.**

Bagian ini memuat penjelasan tentang:

a. Dapat mengetahui hasil klasifikasi kemungkinan terindikasi penyakit jantung dari total pasien yang ada, dengan menggunakan metode machine learning yang dipakai.
b. Mengetahui penggunaan metode yang dipakai, untuk dapat menghasilkan akurasi yang lebih baik.
c. Dapat menjadikan bahan pertimbangan atau pemilihan metode yang tepat, saat pembuatan system pelayanan kesehatan, khususnya prediksi penyakit jantung.
d. Dapat mengembangkan technology dengan metode machine learning lainnya untuk mengoptimalkan prediksi dalam bidang kesehatan, tepatnya prediksi penyakit jantung.
e. Dapat di jadikan referensi pada penelitian selanjutnya terkait dengan prediksi penyakit jantung.

**1.6. Hipotesis.**

Penelitian ini menerapkan dua metode yaitu metode *SMOTE* dan metode Random Forest. Penggunaan metode *SMOTE* diterapkan untuk dapat mengatasi ketidakseimbangan kelas pada dataset penelitian ini. Dataset penelitian ini memiliki dua class dengan jumlah classnya tidak seimbang sehingga metode SMOTE akan menambahkan kelas minoritas agar sama dengan kelas mayoritas dengan cara menciptakan data buatan, data buatan atau sintesis tersebut di buat berdasarkan *k-tetangga* terdekat. Penerapan algoritma *Random Forest* untuk melakukan klasifikasi penyakit jantung. Algoritma ini akan membangun beberapa decision tree dan menggabungkannya demi mendapatkan prediksi yang lebih stabil dan akurat. 'Hutan' yang dibangun oleh Random Forest adalah kumpulan decision tree di mana biasanya dilatih dengan metode bagging. Ide umum dari metode bagging adalah kombinasi model pembelajaran untuk meningkatkan hasil keseluruhan.

Penelitian ini, penulis akan mengklasifikasikan penyakit penyakit jantung dengan menggunakan metode *SMOTE* dan machine learning dengan algoritma *random forest*, diharapkan bahwa penggunaan metode balancing dataset dan metode klasifikasi yang digunakan yaitu *Random forest* mampu meningkatkan akurasi pada proses klasifikasi ini, selain itu diharapkan agar mendapatkan akurasi yang lebih baik dari pada penelitian yang dilakukan sebelumya oleh Ramalingam.,Dkk (2018) yaitu diatas 98%.

# BAB II

# TINJAUAN PUSTAKA

## 2.1. Tinjauan Pustaka.

Beberapa penelitian terbaru telah menguji efektivitas metode machine learning dalam klasifikasi penyakit jantung. Penelitian-penelitian tersebut mencakup penggunaan berbagai algoritma machine learning untuk memprediksi risiko dan jenis penyakit jantung pada pasien. Hasil-hasil studi ini menunjukkan potensi yang besar dalam meningkatkan diagnosis dan pengelolaan penyakit jantung melalui pendekatan teknologi berbasis AI. Diharapkan bahwa implementasi lebih lanjut dari metode ini dapat memberikan kontribusi signifikan dalam upaya pencegahan dan pengobatan penyakit jantung. Beberapa penelitian mengenai klasifikasi penyakit jantung dengan menggunakan metode machine learning, sebagai berikut :

Prediksi penyakit jantung menggunakan metode machine learning dilakukan oleh Mohan,S,Dkk.(Mohan et al., 2019). Hasil dari penelitian ini adalah metode hybrid random forest with a linear model (HRFLM) terbukti menjadi cukup akurat dalam prediksi penyakit jantung, dengan ramalan akurasi 88,4%. kekurangan penelitian ini yaitu menghapus data yang hilang pada fitur tertentu sehingga sebagian class dataset hilang hal ini dapat membuat model hanya mengetahui data yang mempunyai class dengan jumlah lebih banyak, lalu tidak memperhatikan data yang diolah apakah terjadi imbalance atau tidak. berikutnya tidak melihat skala

data, jika terdapat selisih yang begitu besar maka model tidak optimal dalam melakukan clasification.

Penelitian berikutnya mengenai Prediksi Penyakit Jantung Menggunakan machine learning yang dilakukan oleh Bhowmick, Dkk.(Bhowmick et al., 2022). Hasil penelitian adalah akurasi KNN sangat efisien dibandingkan dengan algoritma lainnya saat memprediksi penyakit jantung, knn mempunyai akurasi 87%. kekurangan dari penelitian ini yaitu hampir sama dengan yang dilakukan oleh penelitian sebelumnya menghapus data yang hilang pada fitur tertentu sehingga sebagian class dataset hilang, tidak memperhatikan data yang diolah apakah terjadi imbalance atau tidak, dan tidak melihat skala data, jika terdapat selisih yang begitu besar.

Hal yang sama dilakukan oleh penelitian sebelumnya yaitu penelitian mengenai Perancangan Sistem Klasifikasi Penyakit Jantung Mengunakan Naïve Bayes oleh Ari, B, M.,Dkk. (Bianto et al., 2020). hasil dari penelitian ini adalah rata-rata akurasi pengujian senilai 90,61%, rata-rata hasil nilai presisi senilai 87,44% dan rata-rata nilai recall senilai 87,95%. penelitian ini kekurangannya sama dengan penelitian sebelumnya ditambah dengan pengujian hanya menggunakan satu metode machine learning saja sehingga tidak mengetahui metode manakah yang tepat untuk melakukan klasifikasi penyakit jantung.

Selanjutnya penelitian mengenai Klasifikasi Penyakit Jantung Pada Dataset Imbalanced Class menggunakan Algoritme Stacking yang dilakukan oleh Nurmasani,A,Dkk.(Nurmasani & Pristyanto, 2021). hasil dari penelitian ini yaitu algoritma stacking mampu menghasilkan kinerja dari sisi akurasi TPR, TNR,

GMean dan AUC yang lebih baik dibandingkan single classifier lainnya. pengujian ini mendapatkan akurasi tertinggi sebesar 90%. Penelitian ini tidak memperhatikan nilai fitur dataset yang memiliki skala yang besar sehingga membutuh normalisasi data.

Berikutnya penelitian yang dilakukan oleh Ramalingam, Dkk. (Ramalingam et al., 2018), penelitian ini melakukan prediksi penyakit jantung menggunakan pembelajaran mesin teknik. metode yang digunakan pada penelitian ini yaitu naïve bayes, Knn, Decicion Tree, Svm, dan Random Forest. yang dihasilkan oleh penelitian ini menunjukan algoritma svm lebih baik dalam melakukan klasifikasi di banding algoritma lainnya. akurasi yang dimiliki oleh metode terbaik adalah 98%. kekurangan penelitian ini sama dengan penelitian pertama yang dilakukan sebelumnya.

Terakhir penelitian yang hampir sama caranya dengan penelitian ke-satu dan ke-empat sebelumnya, penelitian ini dilakukan oleh Shanta, M. S, Dkk. (Shiva Shanta Mani & Manikandan, 2020). Penelitian ini melakukan beberapa pengujian klasifikasi menggunakan beberapa metode, maka metode yang paling baik dalam melakukan clasification yaitu Support Vector Machine (SVM) dengan akurasi 72,6%.

Terdapat banyak metode yang diuji ketika melakukan klasifikasi penyakit jantun. Metode yang dapat diterapkan untuk proses klasifikasi ini adalah Support Vector Machine, Decision Tree, Naïve Bayes, dll. Selain itu rata-rata yang tidak dilakukan oleh penelitian sebelumnya yaitu tidak menerapkan metode untuk mengatasi class tidak seimbang, lalu tidak menerapkan teknik normalisasi data, dan

teknik pre-prosessing untuk mengatasi data kosong pada fitur dataset. hal ini dapat membuat proses klasifikasi menjadi tidak efektif. Pada penelitian kali ini penulis menggunakan metode Synthetic Minority Over-Sampling Technique (SMOTE) untuk mengatasi ketidakseimbangan data, dan menggunakan algoritma random forest untuk proses klasifikasi pada dataset penyakit jantung. Penelitian ini bertujuan meningkatkan peforma akurasi model dari model yang digunakan pada klasifikasi kemungkinan penyakit jantung.

**2.2. Keaslian Penelitian.**

Klasifikasi Penyakit Jantung Menggunakan Metode Synthetic Minority Over-Sampling Technique dan Random Forest Clasifier

Tabel 2. 1 Matriks Literatur Review Dan Posisi Penelitian.

| No | Judul | Peneliti, Media Publikasi, dan Tahun | Tujuan Penelitian | Kesimpulan | Saran atau Kelemahan | Perbandingan |
|---|---|---|---|---|---|---|
| 1 | Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques | Mohan,S, Dkk.,(2019) IEEE Access | Prediksi Penyakit Jantung Menggunakan Teknik Hybrid Machine Learning | HRFLM terbukti cukup akurat dalam memprediksi penyakit jantung, dengan akurasi prediksi sebesar 88,4%. | • Ganti nilai yang hilang.<br>• Implementasi metode untuk mengatasi data yang tidak seimbang.<br>• Implementasi normalisasi data. | Penelitian yang diajukan menggunakan metode random forest untuk melakukan proses klasifikasi, metode SMOTE untuk proses balancing class dataset, dan menerapkan teknik pre-prosessing data seperti :<br>• Normalisasi data menggunakan min-max normalization<br>• Replace missing value.<br>Dimana penelitian sebelumnya tidak melakukan teknik yang sama seperti dengan teknik yang ada pada penelitian yang diajukan. |

Tabel 2.1 (lanjutan) Matriks Literatur Review Dan Posisi Penelitian.

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | Heart Disease Prediction Using Machine Learning Algorithms | Singh, A & Kumar, R..(2020) International Conference on Electrical and Electronics Engineering (ICE3-2020) | Menguji beberapa metode machine learning pada proses prediksi penyakit jantung. | Akurasi KNN sangat efisien dibandingkan denga n algoritma lain saat memprediksi penyakit jantung. Knn mendapatkan akurasi sebesar 87%. | • Melakukan normalisasi data karena data yang di proses memiliki value dengan skala berbeda pada fitur dataset.<br>• Mengatasi balancing data<br>• dan cleaning dataset yang memiliki missing value. | Penelitian yang diajukan mengatasi fitur dataset yang memiliki skala yang berbeda sehingga perlu di normalisasi. maka dengan itu penelitian yang diajukan menggunakan teknik normalisasi min-max normalisasi, lalu peneltian yang diajukan juga mengatasi data imbalance dan missing value. |

Tabel 2.1 (lanjutan) Matriks Literatur Review Dan Posisi Penelitian.

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | Perancangan Sistem Klasifikasi Penyakit Jantung Mengunakan Naïve Bayes | Ari, B, M.,Dkk (2019) Creative Information Technology Journal | Penelitian ini bertujuan untuk pembuatan sistem klasifikasi penyakit jantung menggunakan Naïve Bayes Classifier. | Pembuatan sistem ini menyimpulkan nilai hasil akurasi dengan rata-rata akurasi senilai 90,61%, rata-rata hasil nilai presisi senilai 87,44% dan rata-rata nilai recall senilai 87,95% dengan konfigurasi data yang terdapat pada UCI Machine Learning yang berisi 2 kelas klasifikasi dan 15 atribut. | • Jangan menghapus data yang memiliki missing value pada dataset, digantikan dengan mengubah data tersebut dengan mean atau yang lainnya.<br>• implementasi metode untuk mengatasi data imbalance.<br>• mengatasi skala data yang terjadi pada dataset | Penelitian yang diajukan mengimplementasi teknik seperti :<br>• Replace Missing value<br>• mengatsai data dengan class yang tidak balance<br>• mengatasi nilai skala data yang jauh berbeda dalam fitur dataset. |

Tabel 2.1 (lanjutan) Matriks Literatur Review Dan Posisi Penelitian.

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | Algoritme Stacking Untuk Klasifikasi Penyakit Jantung Pada Dataset Imbalanced Class | Nurmasani,A.,Dkk (2021) Jurnal Pseudocode. | Meningkatkan kinerja algoritma stacking untuk klasifikasi penyakit jantung. | Algoritme stacking mampu menghasilkan kinerja dari sisi akurasi TPR, TNR, G-Mean dan AUC yang lebih baik dibandingkan single classifier lainnya. pengujian ini mendapatkan akurasi tertinggi sebesar 90%. | • memperhatikan skala data, jika terdapat selisih yang begitu besar maka model tidak optimal dalam melakukan classification. | Penelitian yang diajukan mengimplementasi teknik seperti :<br>• Replace Missing value<br>• mengatasi nilai skala data yang jauh berbeda dalam fitur dataset.<br>Dibandingkan dengan penelitian ini tidak adanya teknik tersebut. |
| 5 | Heart disease prediction using machine learning techniques : a survey | V,R,V.,Dkk (2018) International Journal of Engineering & Technology | Menguji beberapa metode machine learning untuk dapat melakukan klasifikasi penyakit jantung | Hasil dari penelitian ini yaitu algoritma svm lebih baik dalam melakukan klasifikasi dibanding algoritma lainnya. akurasi yang dimiliki oleh metode terbaik adalah 98%. | • Memperhatikan data yang diolah apakah terjadi imbalance atau tidak. | Penelitian yang diajukan mengatasi data yang tidak seimbang sedangkan penelitian ini tidak sehingga dapat membuat proses klasifikasi tidak optimal. |

Tabel 2.1 (lanjutan) Matriks Literatur Review Dan Posisi Penelitian.

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | Heart Disease Prediction Using Machine Lerning Anupama | Yadav,A.,Dkk (2021) International Research Journal of Engineering and Technology (IRJET) | Menguji beberapa metode machine learning dengan data seimbang. | Hasil dari penelitian ini dengan pengujian terhadap beberapa algoritma maka metode yang paling baik dalam melakuan clasification yaitu Support Vector Machine (SVM) dengan akurasi 72,6%. | • memperhatikan skala data, jika terdapat selisih yang begitu besar maka model tidak optimal dalam melakukan classification. | Penelitian yang diajukan mengatasi fitur dataset yang memiliki skala yang berbeda sehingga perlu di normalisasi. maka dengan itu penelitian yang diajukan menggunakan teknik normalisasi min-max normalisasi, lalu peneltian yang diajukan juga mengatasi data imbalance dan missing value menggunakan metode SMOTE. |

Pada Table 2.1 dipaparkan matriks literatur review dan posisi penelitian. Dengan topik penelitian yang dilakukan oleh beberapa peneliti sebelumnya yaitu memprediksi penyakit jantung dengan penggunaan beberapa metode machine learning yang berbeda-beda (Svm,Decicion three,naïve bayes,dll), begitu juga dengan penanganan imbalance data dengan penerapan beberapa metode untuk dapat mengatasinya seperti Random oversampling, dan Random undersampling, hingga tahap preprosessing data yang dilakukan memiliki perbedaan antara peneliti sebelumnya. Beberapa Penelitian yang dilakukan untuk klasifikasi penyakit jantung masih ada yang belum menerapkan metode untuk penanganan ketidakseimbangan data pada proses klasifikasi, adapun juga beberapa penelitian sudah

menerapkan metode Random oversampling, dan Random undersampling untuk balancing dataset. Di sisi lain pada tahap preprosessing data pada beberapa penelitian yang masih mengatasi data kosong dengan cara menghapus data tersebut pada dataset untuk pengolahan klasifikasi sehingga masih belum menerapakan pengubahan dengan rata-rata ataupun dengan cara lainnya pada data yang bernilai kosong dalam sebuah atribut pada dataset yang dimiliki, sehingga nantinya output yang dihasilkan tidak bias.

Pada penelitian ini, penulis akan menyempurnakan penelitian yang sedang dilakukan, berdasarkan kekurangan-kekurangan dari penelitian sebelumnya,dan melakukan pengujian dengan mengatur metode yang digunakan berupa penyetelan parameter, dll hingga mendapatkan peforma akurasi model yang digunakan pada proses klasifikasi penyakit jantung ini menjadi lebih baik yaitu penelitian mengenai Prediksi Jantung Menggunakan Metode Synthetic Minority Over-Sampling Technique Dan Random Forest.

## 2.3. Landasan Teori.

Klasifikasi penyakit jantung menggunakan metode machine learning, teknik Synthetic Minority Over-Sampling Technique (SMOTE), dan evaluasi menggunakan confusion matrix. Pertama, klasifikasi penyakit jantung dengan machine learning melibatkan penerapan algoritma machine learning untuk memprediksi keberadaan atau jenis penyakit jantung berdasarkan data klinis pasien. Metode ini memungkinkan identifikasi pola kompleks dalam data yang mungkin sulit dikenali oleh manusia. Kedua, SMOTE adalah teknik oversampling yang digunakan untuk menangani ketidakseimbangan kelas dalam dataset, khususnya dalam kasus di mana kelas minoritas (misalnya, pasien dengan penyakit jantung) memiliki representasi yang kurang. SMOTE menciptakan sampel sintetis dari kelas minoritas untuk meningkatkan keseimbangan dataset. Ketiga, evaluasi menggunakan confusion matrix adalah metode yang umum digunakan untuk mengukur kinerja model klasifikasi. Confusion matrix memberikan gambaran tentang seberapa baik model dapat memprediksi kelas-kelas yang benar dan salah dari dataset. Dengan memadukan ketiga komponen ini, penelitian ini bertujuan untuk mengembangkan model klasifikasi yang akurat dalam memprediksi penyakit jantung dan mengevaluasi kinerjanya menggunakan metrik yang sesuai seperti akurasi, presisi, recall, dan F1-score.

### 2.3.1. Penyakit Jantung.

Penyakit jantung merupakan kondisi ketika jantung manusia mengalami gangguan. Terdapat beberapa jenis penyakit jantung, antara lain(Fadli, 2022):

1. Penyakit jantung koroner, merupakan suatu penyakit jantung yang terjadi akibat penyempitan pembuluh darah di jantung.
2. Penyakit jantung bawaan, merupakan suatu masalah jantung yang ditemukan sejak bayi, yang paling umum ditemukan adalah kebocoran katup jantung.
3. Infeksi jantung (endokarditis), merupakan suatu infeksi pada lapisan dalam jantung.
4. Gagal jantung, merupakan suatu kegagalan otot jantung untuk memompakan darah secara memadai ke seluruh tubuh.
5. Aritmia, merupakan suatu gangguan irama jantung yang menyebabkan denyut jantung tidak normal.

Penelitian ini membahas topik penyakit jantung yang lebih mengarah ke jenis penyakit jantung coroner. Penyakit jantung coroner adalah penyakit yang disebabkan oleh penyumbatan pembuluh darah utama yang mengalirkan pasokan oksigen, darah, dan nutrisi untuk jantung(Makarim, 2023). Umumnya, kondisi ini merupakan dampak dari adanya plak kolesterol dan peradangan pada pembuluh darah arteri di jantung. Terdapat beberapa penyebab terjadinya penyakit jantung koroner di antaranya sebagai berikut (Hospitals, 2023):

1. Hipertensi (tekanan darah tinggi)
2. Diabetes
3. Berat badan berlebih
4. Peradangan pada pembuluh darah
5. Kebiasaan merokok
6. Kadar kolesterol dan trigliserida tinggi

### 2.3.2. Klasifikasi.

Klasifikasi adalah suatu proses pengelompokan data dimana data yang digunakan tersebut sebelumnya sudah mempunyai kelas label atau target. Biasanya algoritma-algoritma untuk menyelesaikan masalah klasifikasi dikategorisasikan ke dalam supervised learning atau pembelajaran yang diawasi(Adiezwara Ramadhan Frananda, 2021). Proses klasfikasi akan melakukan proses terhadap sebuah data dan menentukan bahwa data tersebut akan masuk pada kategori kelas yang mana(Ervina, 2019). Adapun langkah-langkah dalam proses klasifikasi yaitu (Algonz D.B. Raharja, 2022):

1. Membangun model dari data training pada dataset pengolahan, lalu data tersebut sudah mempunyai label yang sudah diketahui sebelumnya. algoritma klasifikasi diterapkan untuk membuat model berdasarkan data training.
2. Melakukan evaluasi terhadap model yang di hasilkan, untuk mengetahui seberapa baik kinerja dari metode yang dipakai pada model tersebut.

### 2.3.3. Machine Learning.

Machine learning (ML) adalah mesin yang dikembangkan agar bisa belajar dengan sendirinya tanpa arahan dari penggunanya, pembelajaran mesin dikembangkan berdasarkan disiplin ilmu lainnya seperti statistika, matematika dan data mining sehingga mesin dapat belajar dengan menganalisa data tanpa perlu di program ulang atau diperintah(Dicoding Intern, 2020). Machine learning merupakan bagian dari kecerdasan buatan atau artificial intelligence (AI) yang mampu mempelajari data dengan sendirinya dengan algoritma yang terus berkembang sehingga tidak perlu diprogram ulang secara berkala. Semakin banyak

data yang digunakan untuk melatih machine learning, maka semakin baik machine learning tersebut(Dqlab, 2022).

Terdapat tiga cabang utama dalam machine learning diantaranya(Royal Society of Great Britain, 2017):

1. Suvervised machine learning.

   Suatu system yang dilatih sesuai dengan data yang telah diberi label dari awal. Kemudian dari label tersebut membagikan setiap titik data dalam satu atau beberapa kelompok. Dari data yang ada,lalu system mempelajari bagaimana data tersebut dipelajari atau dikenal sebagai data training terstruktur, dan dari data training itulah system memprediksi atau mengklasifikasikan data test atau data uji.

2. Unsupervised learning.

   Kebalikan dari pembelajaran sebelumnya adalah Suvervised machine learning, dimana Unsupervised learning ini adalah pembelajaran tanpa pengawasan dengan artinya analisis yang dilakukan pada data yang diangkat dimana data tersebut tanpa label, ini bertujuan untuk mendeteksi karakteristik dari suatu titik data yang kurang lebih serupa antara satu sama lain, misalnya meng-cluster data dan menetapkan data dari cluster tersebut.

3. Reinforcement learning.

   Dimana pembelajaran yang memperkuat belajar berdasarkan dengan pengalaman, yang berada dalam pembelajaran supervised dan unsupervised.

**2.3.4. Random Forest Clasifier.**

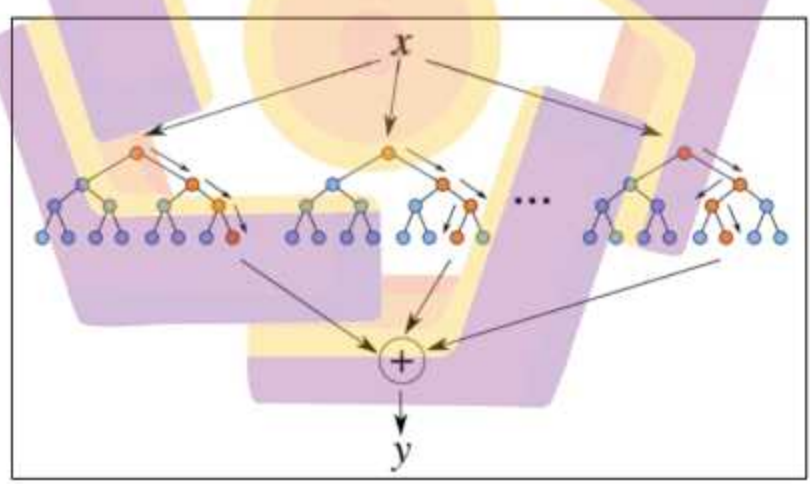
Algoritma Random Forest pertama kali diperkenalkan oleh Breiman (2001). Random Forest memiliki dua fungsi pemecahan masalah, yaitu klasifikasi dan regresi. Hutan Acak dapat digunakan untuk banyak tipe data, mis. Diskrit, kontinyu, kombinasi multivariat dan data kelangsungan hidup. Random Forest dapat mendeteksi interaksi antara variabel dependen dan independen serta mengeksplorasi data dengan fleksibilitasnya. Jika analisis dilakukan dengan menggunakan struktur acak, tidak ada asumsi tertentu yang harus dipenuhi (Wulansari, 2018).

Random Forest adalah metode yang terdiri dari serangkaian pohon terstruktur, masing-masing memberikan unit suara kelas, dan hasilnya didasarkan pada sebagian besar keputusan. Teknik dasar yang digunakan Random Forest adalah pohon keputusan. Dengan kata lain, hutan acak adalah kumpulan pohon keputusan yang digunakan untuk mengklasifikasikan dan memprediksi data dengan memasukkan input ke akar di atas dan kemudian menghitung ke daun di bawah (Haristu, 2019). Pohon yang digunakan dalam metode ini dapat tumbuh menjadi tanaman dengan menanam setiap pohon dengan cara yang sama. Random Forest menggunakan strategi ensambel bagging yang dapat mengatasi masalah overfitting yang terjadi saat data train kecil (Samudra, 2019). Hasil random forest analysis untuk klasifikasi berupa bentuk tiap pohon pada hutan binaan, sedangkan hasil prediksi diperoleh dari rata-rata tiap pohon (Lingga P, 2017).

Random Forest merupakan hasil pengembangan dari metode Classification and Regression Tree (CART), yang menggunakan metode bag or bootstrap

aggregation dan random feature selection. Bagging merupakan salah satu teknik yang dapat digunakan untuk memperbaiki hasil dari suatu algoritma klasifikasi. Metode bagging ini didasarkan pada metode ensemble (Oceano, 2019). Algoritma metode hutan acak dibagi menjadi dua bagian. Yang pertama adalah menghasilkan pohon "k" untuk membuat hutan acak. Yang kedua adalah membuat prediksi menggunakan hutan yang dihasilkan secara acak. Langkah-langkah penerapan metode Random Forest adalah:

1. Membuat data sampel dengan cara pengambilan acak dengan pengembalian dari dataset.
2. Gunakan sampel data untuk membangun pohon ke i (i=1, 2, 3, ..., k)
3. Ulangi langkah 1 dan 2 sebanyak k kali




Gambar 2. 1 Random Forest.

Sumber: Morioh.com

Komputasi yang digunakan dalam membangun pohon keputusan menggunakan metode CART terdiri dari verifikasi informasi yang menggambarkan

ukuran atribut yang digunakan untuk mengklasifikasikan setiap simpul pohon. Misalkan N adalah node yang memisahkan setiap kelas data yang dilambangkan dengan D berdasarkan atributnya. Split node dijalankan berdasarkan atribut dengan informasi validasi tertinggi. Rumus untuk menerima informasi konfirmasi adalah sebagai berikut:

$$Gain\ (A) = Info\ (D) - In(D) \quad (2.1)$$

Nilai info (D) dapat diperoleh menggunakan rumus 2 dan 3 untuk mendapatkan nilai info $_A$(D) :

$$Inf0(\mathrm{D}) = \sum_{i=0}^{n} pilog2(pi) \quad (2.2)$$

Keterangan :

n = jumlah kelas target

pi = proporsi kelas i terhadap partisi D

$$Inf0_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} x\ Info(D_j) \quad (2.3)$$

Keterangan :

v = jumlah partisi

$D_j$ = total partisi ke j

D = total baris pada semua partisi

Nilai validasi data untuk atribut yang berisi nilai kontinu atau numerik harus menentukan titik pisah terbaik untuk mengelompokkan nilai. Resolusi terbaik ditemukan dengan menyortir data terlebih dahulu. Kemudian median atau rata-rata dari setiap pasangan nilai yang berdekatan dianggap sebagai titik pembagian yang memungkinkan untuk digunakan. Jika atribut A adalah atribut bernilai kontinu,

semua nilai A diurutkan kemudian dirata-ratakan, sehingga kemungkinan jumlah partisi pada Persamaan 3.3 adalah dua atau. v = 2 (j = 1 dan 2) (Haristu, 2019).

Terdapat parameter yang bisa diatur dan diimplementasikan saat proses klasifikasi dilakukan menggunakan random forest. Nilai parameter diatur guna mendapatkan model yang optimal, pengaturan dari parameter ini terdapat pada beberapa metode machine learning biasanya disebut *hyperparameter* (Putatunda & Rama, 2018). *Hyperparameter* dipakai untuk mengatur macam-macam aspek dalam meachine learning yang dapat mempengaruhi performa dan model yang dihasilkan. Percarian *hyperparameter* dilakukan secara manual atau dengan menguji kumpulan *hyperparameter* (Claesen & De Moor, 2015).

Tabel 2. 2 Parameter pada Metode Random forest.
Sumber: (https://towardsdatascience.com)

| Parameter | Keterangan |
|---|---|
| n_estimators | jumlah pohon di forest |
| max_features | jumlah maksimum fitur yang dipertimbangkan untuk memisahkan node. |
| max_depth | jumlah maksimal level di setiap pohon keputusan |
| min_samples_split | jumlah min titik data yang ditempatkan di node sebelum node dipisah. |
| min_samples_leaf | jumlah min titik data yang diperbolehkan dalam simpul daun |
| Bootstrap | metode pengambilan sampel titik data (dengan atau tanpa penggantian) |

### 2.3.5. Synthetic Minority Over-Sampling Technique (SMOTE).

Synthetic Minority Over-Sampling Technique (SMOTE) merupakan salah satu turunan dari oversampling, metode SMOTE ini pertama kali diperkenalkan oleh Nithes V. Chawla untuk dapat mengatasi ketidakseimbangan kelas dari suatu data (Kovács et al., 2020). Metode SMOTE menambah kelas minoritas agar sama dengan kelas mayoritas dengan cara menambahkan data buatan, data buatan atau sintesis tersebut di buat berdasarkan *k-tetangga* terdekat (Sabiq Sofyan, 2013).

Cara kerja dari SMOTE ini ialah pertama mengambil selisih antara vector dari fitur kelas minoritas dan nilai dari Nearest Neighbor dari kelas minoritas lalu kalikan nilai tersebut dengan angka acak antara 0 sampai 1, Berikutnya hasil dari penjumlahan tersebut di tambahkan dengan vector lainnya sehingga mendapatkan hasil baru dari vector, sebagaimana didefinisikan kedalam persamaan berikut (Kasanah et al., 2019):

$$X_{syn} = X_i + (X_{knn} - X_i) \times \delta \quad (2.4)$$

Keterangan :

$X_{syn}$ = data synthesis yang akan diciptakan

$X_i$ = data yang akan di replikasi

$^{x}knn$ = data yang memiliki jarak dari data $X_i$

$\delta$ = angka acak antara 0 sampai 1

### 2.3.6. Evaluasi Metode.

Setelah proses analisis dilakukan maka didapatkann-nya hasil klasifikasi. Berikutnya melakukan penilaian atau evaluasi nilai klasifikasi yang paling baik. Secara umum pengukuran kinerja metode klasifikasi dilakukan yaitu dengan

membandingkan antara prediksi yang dihasilkan dengan variable data *testing* sebagai data sebenarnya.

**2.3.6.1. Confusion Matrix.**

Confusion matrix memberikan perincian terkait kesalahan pada hasil klasifikasi dengan metode yang di gunakan. Confusion matrix adalah tabel yang berisikan perhitungan yang didasari pada evaluasi model klasifikasi berdasarkan jumlah study kasus yang di klasikasikan yang diprediksi benar dan salah (Kuncahyo Setyo Nugroho, 2019).

Tabel 2. 3 Confusion Matrix.
Sumber: (Doreswamy & Hemanth, 2011)

| *Classification* | *Predicted Positive* | *Predicted Negative* |
|---|---|---|
| *Actual Positive* | *True Positive (TP)* | *False Negative (FN)* |
| *Actual Negative* | *False Positive (FP)* | *True Negative (TN)* |

Dalam pengukuruan kinerja menggunakan *confusion matrix* terdapat empat bagian untuk mengidentifikasi suatu prediksi, berikut diantaranya (Doreswamy & Hemanth, 2011) :

1. TP (True Positive) adalah jumlah data dengan nilai aktual positif dan nilai prediksi positif.
2. TN (True Negative) adalah jumlah data dengan nilai actual positif dan nilai prediksi negative.
3. FP (False Positive) adalah jumlah data dengan nilai actual negatif dan nilai prediksi positif.
4. FN (False Negative) adalah jumlah data dengan nilai actual negatif dan nilai prediksi negatif.

Terdapat beberapa nilai evaluasi yang sering di pakai pada klasifikasi biner. Dapat dilihat berdasarkan nilai *confusion matrix* (Sokolova & Lapalme, 2009):

1. *Accuracy* (ACC) adalah efektivitas dari hasil yang didapatkan dalam proses klasifikasi

$$Accuracy\ (\%) = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (2.5)$$

2. *Precision* (PREC) adalah persentase dari label data dengan label positif yang dihasil dari proses klasifikasi.

$$Precision\ (\%) = \frac{(TP)}{(TP+FP)} \quad (2.6)$$

3. *Recall* (REC) atau *sensitivity* adalah efektivitas dari pengklasifikasi dalam mengidentifikasi label positif

$$Recall\ (\%) = \frac{(TP)}{(TP+FN)} \quad (2.7)$$

4. *F1-Score* adalah perbandingan rata-rata presisi dan recall yang dibobotkan

$$F1\text{-}Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (2.8)$$

# BAB III

# METODE PENELITIAN

## 3.1. Jenis, Sifat, dan Pendekatan Penelitian.

Jenis penelitian ini dapat dikategorikan sebagai penelitian eksperimental, yang bertujuan untuk menguji efektivitas penggunaan metode Synthetic Minority Over-Sampling Technique (SMOTE) dan Random Forest Classifier saat melakukan klasifikasi penyakit jantung.

Sifat penelitian ini memiliki sifat kuantitatif karena melibatkan pengolahan dan analisis data numerik untuk mengklasifikasikan jenis penyakit jantung. Pendekatan penelitian ini berfokus pada pengembangan dan evaluasi model klasifikasi menggunakan metode SMOTE dan Random Forest Classifier.

Penelitian ini menggunakan data publik yang tersedia mengenai pasien dengan riwayat penyakit jantung dari kaggle. Kaggle adalah lingkungan dan komunitas machine learning online. Ini memiliki kumpulan data standar yang coba dimodelkan oleh ratusan atau ribuan individu atau tim. Data penelitian tersebut harus mencakup fitur-fitur yang relevan seperti usia, jenis kelamin, tekanan darah, kadar kolesterol, riwayat merokok, dan lain-lain, lalu masuk pada proses preprocessing langkah ini melakukan preprocessing terhadap data termasuk pembersihan data, dan pengubahan format data. Pengolahan data dengan SMOTE setelah preprocessing, akan dilakukan pengolahan data menggunakan metode SMOTE untuk menyeimbangkan dataset yang tidak seimbang. metode SMOTE akan menghasilkan sampel sintetis dari kelas minoritas untuk meningkatkan

representasi kelas tersebut dalam dataset. Setelah itu dilakukan proses normalisasi data menggunakan metode min-max normalization proses ini dilakukan untuk normalisasi data yang terdapat skala value pada fitur dataset. berikutnya masuk pada Pembentukan Model dengan Random Forest Classifier Setelah dataset seimbang dan normal, akan dilakukan pembentukan model klasifikasi menggunakan metode Random Forest Classifier. Pohon keputusan akan dibentuk dan diintegrasikan menjadi ensemble untuk menghasilkan prediksi yang akurat. dan yang terakhir yaitu proses evaluasi model proses ini akan dilakukan evaluasi performa model menggunakan metrik-metrik seperti akurasi, presisi, recall, f1-score dengan confusion matrix. evaluasi ini akan memberikan informasi tentang kemampuan model dalam mengklasifikasikan jenis penyakit jantung. Hasil dari evaluasi model akan dianalisis dan diinterpretasikan untuk menentukan sejauh mana metode SMOTE dan Random Forest Classifier efektif dalam klasifikasi penyakit jantung. Hasil ini dapat memberikan wawasan penting dalam diagnosis dan perawatan penyakit jantung.

### 3.2. Metode Pengumpulan Data.

Dataset yang digunakan pada penelitian ini diambil dari kaggle yaitu Heart Disease Dataset. Dataset ini merupakan dataset publik digunakan untuk memprediksi kemungkinan pasien terkena penyakit jantung atau tidak berdasarkan parameter input seperti usia, seks, jenis nyeri dada, tekanan darah, kolesterol, gula darah, hasil elektrokardiografi, detak jantung maksimum, latihan diinduksi angina, oldpeak, oldpeak = st depresi yang diinduksi oleh latihan relatif terhadap istirahat,

kemiringan segmen st latihan puncak, dan jumlah kapal utama (0-3) diwarnai oleh flourosopy. Setiap baris dalam data memberikan informasi yang relevan tentang pasien.

Dataset penyakit jantung ini mempunyai 13 atribut. Dengan variabel independen adalah age, sex, chest pain type (4 values), resting blood pressure, serum cholestoral in mg/dl, fasting blood sugar > 120 mg/dl, resting electrocardiographic results (values 0,1,2), maximum heart rate achieved, exercise induced angina, oldpeak = ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, number of major vessels (0-3) colored by flourosopy. dan untuk variabel dependent ialah target. kategori pasien pada variabel dependent berjumlah 2 kategori yaitu pasien terindikasi penyakit jantung dan tidak terindikasi penyakit. Tabel 3.1 Dataset menggambarkan informasi fitur dan class pada dataset yang digunakan.

Tabel 3. 1 Fitur Dataset.

| No | Fitur | Keterangan |
|---|---|---|
| 1 | Age | Menyimpan usia pasien |
| 2 | Sex | (1 = laki-laki; 0 = perempuan) |
| 3 | Chest Pain Type | jenis nyeri dada |
| 4 | Resting Blood Pressure | tekanan darah istirahat (dalam mm Hg saat masuk rumah sakit) |
| 5 | serum cholestoral in mg/dl | Kolesterol serum dalam mg/dl |
| 7 | fasting blood sugar | (gula darah puasa > 120 mg/dl) (1 = benar; 0 = salah) |
| 8 | resting electrocardiographic results | hasil elektrokardiografi istirahat |
| 9 | maximum heart rate achieved | detak jantung maksimum tercapai |
| 10 | exercise induced angina | latihan diinduksi angina (1 = ya; 0 = tidak ada) |
| 11 | oldpeak = ST depression induced by exercise relative to rest | Depresi ST yang diinduksi oleh olahraga relatif terhadap istirahat |
| 12 | the slope of the peak exercise ST segment | kemiringan segmen ST latihan puncak |
| 13 | number of major vessels (0-3) colored by flourosopy | jumlah kapal utama (0-3) diwarnai oleh flourosopy |
| Class | Target | bilangan bulat bernilai 0 = tidak ada penyakit dan 1 = penyakit |

### 3.3. Metode Analisis Data.

Metode analisis data yaitu Synthetic Minority Over-Sampling Technique (SMOTE) dan Random Forest Classifier, adalah teknik yang digunakan dalam klasifikasi penyakit jantung. Berikut adalah langkah-langkah yang dilakukan dalam penelitian ini :

1. Pendahuluan

Pada bagian ini, menjelaskan tentang latar belakang penelitian, tujuan penelitian, serta gambaran umum tentang metode yang akan digunakan, yaitu

Synthetic Minority Over-Sampling Technique (SMOTE) dan Random Forest Classifier.

2. Pengumpulan Data

   Penjelasan tentang sumber data yang digunakan dalam penelitian ini, termasuk karakteristik data, variabel yang diamati, dan cara pengumpulan data.

3. Preprocessing Data

   Langkah-langkah preprocessing data yang dilakukan, termasuk penanganan nilai yang hilang, normalisasi data, dan pengkodean variabel kategorikal.

4. Synthetic Minority Over-Sampling Technique (SMOTE)

   Penjelasan tentang konsep dasar dari SMOTE dan bagaimana teknik ini digunakan untuk menangani ketidakseimbangan kelas dalam dataset. Langkah-langkah implementasi SMOTE dalam penelitian ini juga akan dijelaskan.

5. Random Forest Classifier

   Penjelasan tentang konsep dasar dari Random Forest Classifier dan mengapa algoritma ini dipilih untuk klasifikasi penyakit jantung. Langkah-langkah implementasi Random Forest Classifier, termasuk pengaturan hyperparameter, juga akan diuraikan.

6. Pembagian Data

   Penjelasan tentang bagaimana dataset dibagi menjadi set pelatihan dan set pengujian, serta proporsi pembagian yang digunakan.

7. Pelatihan Model

   Proses pelatihan model menggunakan algoritma Random Forest Classifier pada dataset yang telah diproses dan dilakukan oversampling menggunakan

SMOTE. Langkah-langkah pelatihan, termasuk teknik validasi yang digunakan, akan dijelaskan.

8. Evaluasi Model

   Metrik evaluasi yang digunakan untuk mengevaluasi kinerja model, seperti akurasi, presisi, recall, dan f1-score akan dijelaskan di sini. Hasil evaluasi dari model yang telah dilatih juga akan disajikan.

9. Analisis Hasil

   Interpretasi hasil dari model klasifikasi yang telah dilatih, termasuk fitur-fitur yang paling berpengaruh dalam klasifikasi penyakit jantung.

10. Kesimpulan

    Ringkasan dari temuan penelitian, implikasi praktis, serta saran untuk penelitian selanjutnya.

Dalam menunjang penelitian ini, penelitian menggunakan beberapa alat bantu berupa seperangkat komputer dan software sebagai berikut :

a. Processor intel core i5
b. Ram 4 GB
c. Harddisk 1 TB
d. SSD 240 GB
e. Microsoft Excel
f. Google Colaboratory

**3.4. Alur Penelitian.**

Alur penelitian dapat dilihat pada gambar 3.1. terdapat tahapan yang dilakukan dalam penelitian ini diantaranya :

a. Identifikasi masalah

Proses ini merupakan tahap dimana penulis mencari permasalahan yang ada dengan mencari sumber informasi permasalahan berupa artikel terkait dan jurnal peneltian terkait dengan sumber yang terpercaya.

b. Penyusunan proposal thesis

Proses ini merupakan tahap dimana penulis mencari permasalahan yang ada dengan mencari sumber informasi permasalahan berupa artikel terkait dan jurnal peneltian terkait dengan sumber yang terpercaya.

c. Studi pustaka

Proses ini, penulis mencari tahu informasi dengan membaca jurnal penelitian dan buku yang diaggap relevan dengan permasalahan yang akan diangkat.

d. Pengambilan dataset

Setelah tahap identifikasi masalah dan studi literatur, tahap berikutnya yang akan dilakukan adalah mengambil dataset di kaggle sesuai dengan topik yang dipilih sebelumnya. File yang digunakan pada penelitian ini adalah file yang berekstensi .csv. Kemudian pada tahap studi literatur, dan experiment terhadap beberapa metode untuk menyelasaikan topik yang diangkat, maka penulis telah menentukan algoritma yang digunakan untuk mencari algoritma mana yang

paling efisien terhadap klasifikasi kemungkinan penyakit jantung. Penulis menggunakan algoritma Random Forest dan SMOTE pada penelitian ini.

e. Pre-Prosessing data

Sebelum masuk pada pengolahan data maka perlu mekukan preprocessing data untuk memeriksa dan memperbaiki kesalahan yang ditemukan pada data yang di gunakan dalam penelitian ini.

f. Balancing data

Data yang di gunakan dalam pengolahan penelitian ini mengalami ketidakseimbangan kelas. Pada tahap ini menerapkan metode SMOTE untuk mengatasi ketidakseimbangan kelas.

g. Normalisasi data

Proses mengubah atribut numerik dalam dataset sehingga memiliki skala yang seragam atau sebanding. proses ini menggunakan metode min-max normalisasi

h. Split data

Pada tahap ini membagi dataset menjadi dua bagian yakni bagian yang digunakan untuk training data, dan untuk testing data, pembagian yang dilakukan yaitu berjumlah 80/20, yang dimana 80% untuk data training dan 20% untuk data testing

i. Random forest klasifikasi

Tahap ini adalah melakukan klasifikasi terhadap data yang sudah melewati tahapan sebelumnya yaitu Pre-processing data, Balancing data, dan normalisasi data

j. Tunning Parameter

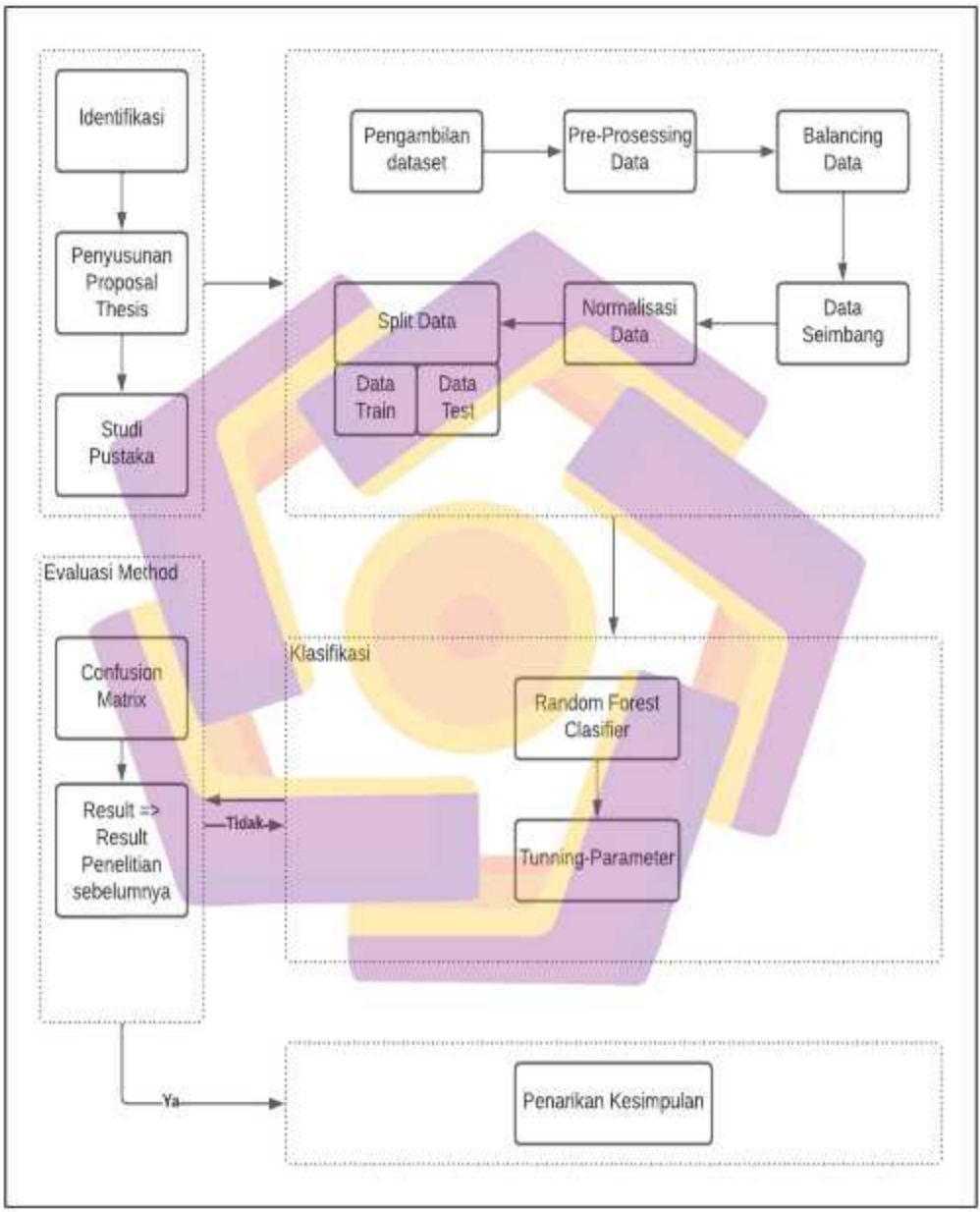
Tahap ini ialah mengatur berbagai macam aspek dalam machine learning yang sangat berpengaruh pada performa dan model yang dihasilkan yaitu mengatur *hyperparameter*, Pencarian *hyperparameter* dilakukan secara manual, dengan menguji kumpulan *hyperparameter* pada parameter yang ditentukan sebelumnya.

k. Evaluasi Method

Menilai kinerja model pada proses klasifikasi, tahapan ini menggunakan confusion matrix untuk mengukur kinerja model yang di gunakan dalam proses klasifikasi, jika hasil akurasi dari klasifikasi melebihi hasil akurasi yang dihasilkan pada penelitian sebelumnya maka dilanjutkan pada proses berikutnya, jika tidak maka kembali pada proses sebelumnya yaitu proses klasifikasi dan mengatur parameter hingga mendapatkan hasil yang optimal.

l. Penarikan Kesimpulan

Tahap ini merupakan tahap pemberian kesimpulan berdasarkan hasil dari pengujian yang telah dilakukan. Hasil penelitian berupa fakta yang diperoleh metode yang di terapkan untuk pediksi penyakit jantung. Hasil pengujian dan evaluasi dijadikan kesimpulan akhir mengenai metode balancing data, algoritma Random forest untuk proses klasifikasi, dan parameter yang diujikan.




Gambar 3. 1 Alur Penelitian.

# BAB IV

# HASIL PENELITIAN DAN PEMBAHASAN

Berdasarkan penelitian dan tinjauan literatur sebelumnya, bagian ini menyajikan penjelasan mengenai hasil-hasil yang diperoleh dari proses komputasi. Diskusi ini mencakup langkah-langkah utama dalam penelitian ini, mulai dari pengumpulan dataset, pra-pemrosesan data, penanganan ketidakseimbangan kelas pada dataset menggunakan metode SMOTE, normalisasi data, pembagian dataset, penerapan klasifikasi dan pengaturan parameter dari metode random forest, serta evaluasi hasil menggunakan confusion matrix guna mengukur kinerja model yang dihasilkan selama tahap klasifikasi.

## 4.1. Dataset.

Penelitian ini menggunakan dataset penyakit jantung yang berasal dari Kaggle https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset?resource=download. Dataset ini berasal dari database Cleveland Amerika Serikat, database ini satu-satunya database yang digunakan oleh peneliti Machine Learning hingga saat ini, data di dalam database ini berasal dari hasil Analisa medis yang di lakukan sebelumnya(Janosi, 1988). Data ini dipakai untuk melatih dan menguji model saat melakukan klasifikasi khususnya pada klasfikasi penyakit jantung. berikut ini merupakan tampilan dataset yang dapat dilihat pada tabel 4.1.

Tabel 4. 1 Dataset

| age | Sex | cp | trestbps | chol | Fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 52 | 1 | 0 | 125 | 212 | 0 | 1 | 168 | 0 | 1 | 2 | 2 | 3 | 0 |
| 53 | 1 | 0 | 140 | 203 | 1 | 0 | 155 | 1 | 3.1 | 0 | 0 | 3 | 0 |
| 70 | 1 | 0 | 145 | 174 | 0 | 1 | 125 | 1 | 2.6 | 0 | 0 | 3 | 0 |
| 61 | 1 | 0 | 148 | 203 | 0 | 1 | 161 | 0 | 0 | 2 | 1 | 3 | 0 |
| 62 | 0 | 0 | 138 | 294 | 1 | 1 | 106 | 0 | 1.9 | 1 | 3 | 2 | 0 |
| 58 | 0 | 0 | 100 | 248 | 0 | 0 | 122 | 0 | 1 | 1 | 0 | 2 | 1 |
| 58 | 1 | 0 | 114 | 318 | 0 | 2 | 140 | 0 | 4.4 | 0 | 3 | 1 | 0 |
| 55 | 1 | 0 | 160 | 289 | 0 | 0 | 145 | 1 | 0.8 | 1 | 1 | 3 | 0 |

Pada tabel 4.1 dataset ini mempunyai 2 class yaitu 0 (tidak ada penyakit) sebanyak 499 data dan 1 (penyakit) sebanyak 526 data. Jumlah class pada dataset ini tentunya mengalami ketidakseimbangan kelas karena terdapat sebuah kelas pada kelas target yang memiliki jumlah lebih besar (kelas mayoritas) dibandingkan dengan kelas lain pada kelas target yang memiliki jumlah lebih kecil (kelas minoritas). Maka dengan itu diterapkan-nya metode Synthetic Minority Over-Sampling Technique (SMOTE) untuk dapat mengatasi dataset mempunyai masalah imbalance kelas.

**4.2. Preprosessing Data.**

Sebelum masuk pada proses berikutnya perlu dilakukan teknik pre-prosessing data, agar dari proses ini menghasilkan data yang menjadi syarat sebagai bahan pengolahan proses klasifikasi yang efektif. Dengan preprosessing data ini dilakukan agar metode yang digunakan mendapatkan hasil yang baik dalam proses klasifikasi, terdapat sebuah teknik dalam melakukan preprosessing ini yang sesuai dengan pola dataset yang terjadi pada penelitian yaitu melakukan pengubahan nilai kosong yang terjadi pada dataset. Dataset penelitian ini terdapat nilai kosong pada

fitur dataset yaitu pada atribut thalach sebanyak 36 data yang kosong, bisa dilihat pada gambar 4.1 berikut ini.

```
1  data.isna().sum()

age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach    36
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
```

Gambar 4. 1 Missing Value

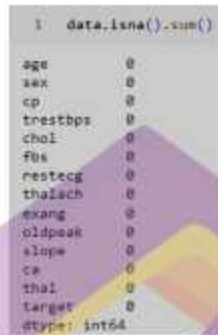
Atribut missing value tersebut diubah dengan rata-rata atribut thalach mengalami penyakit dan thalach tidak mengalami penyakit menjadi pilihan, dibandingkan dengan menghapus baris data yang terdapat nilai kosong, karena tidak ada data yang terbuang. hasil dari proses ini terdapat pada gambar 4.2 berikut.

```
1  mean_thalach_has_jantung = data[data['target']==1]['thalach'].mean()
2  mean_thalach_has_jantung

158.61023622047244

1  mean_thalach_has_No_jantung = data[data['target']==0]['thalach'].mean()
2  mean_thalach_has_No_jantung

138.985446985447

data.loc[data['target']==1,'thalach'] =
data.loc[data['target']==1, 'thalach'].fillna(mean_thalach_has_jantung)
data.loc[data['target']==0,'thalach'] =
data.loc[data['target']==0, 'thalach'].fillna(mean_thalach_has_No_jantung)
```

Gambar 4. 2 Hasil Replace Missing Value.

Pada gambar 4.2 merupakan ilustrasi dari proses mengubah nilai missing value pada fitur dataset. Setelah proses replace missing value dilakukan maka

kembali untuk pengecekan nilai missing value pada dataset apakah sudah berhasil diganti atau tidak, yang dapat dilihat pada gambar 4.3.

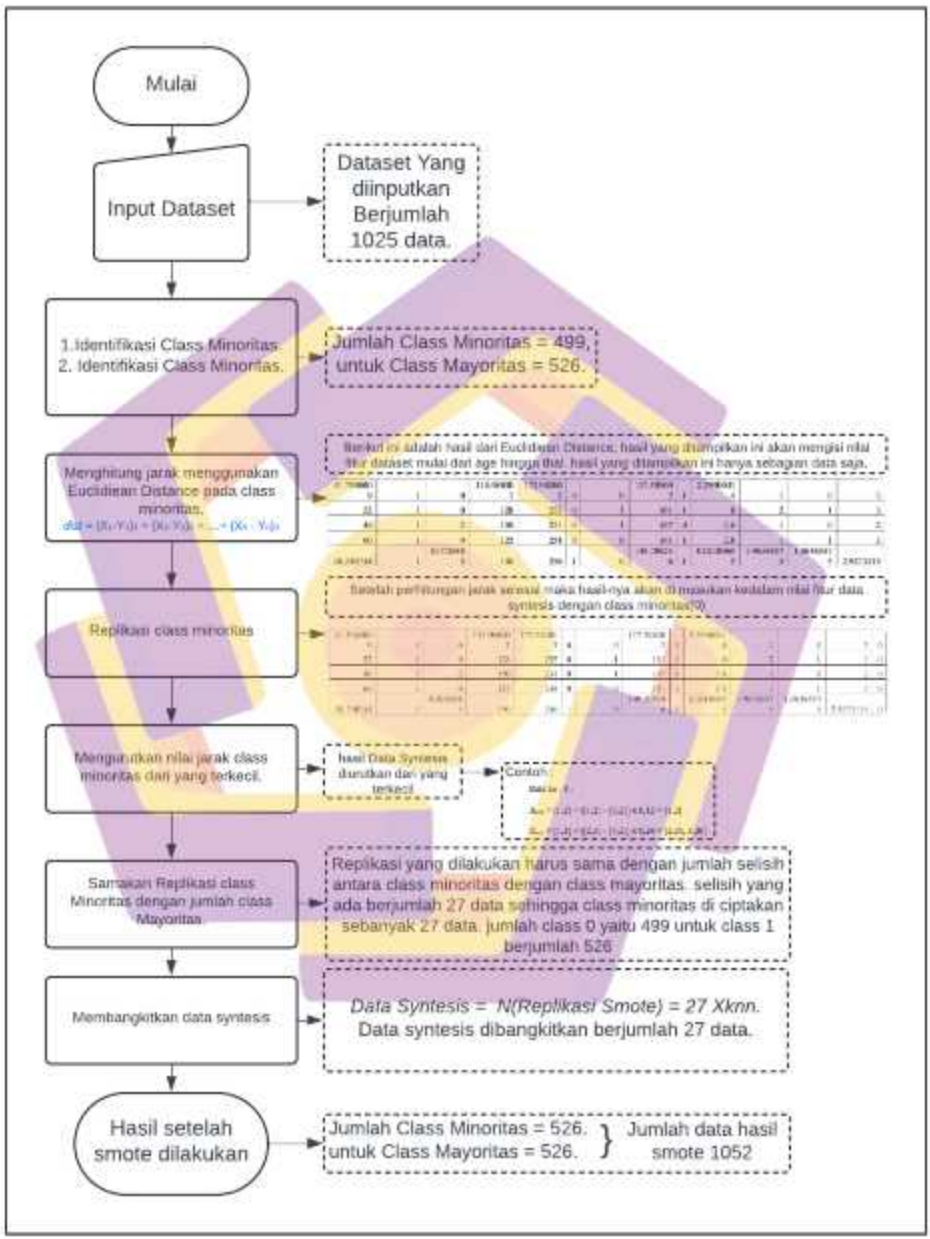


```
1  data.isna().sum()

age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

Gambar 4. 3 Berhasil Mengatasi Missing value.

**4.3. Balancing Data.**

Dataset yang digunakan penelitian ini memiliki Jumlah class yang mengalami ketidakseimbangan kelas karena terdapat sebuah kelas pada kelas target yang memiliki jumlah lebih besar (kelas mayoritas) dibandingkan dengan kelas lain pada kelas target yang memiliki jumlah lebih kecil (kelas minoritas), kelas tersebut yaitu 0 (tidak ada penyakit) sebanyak 499 data dan 1 (penyakit) sebanyak 526 data, dengan adanya ketidakseimbangan class pada dataset ini, maka dengan itu diterapkan-nya metode Synthetic Minority Over-Sampling Technique (SMOTE) untuk mengatasi dataset mempunyai masalah imbalance kelas. Metode SMOTE akan menambah kelas minoritas agar sama dengan kelas mayoritas dengan menambahkan data buatan, data buatan atau sintesis tersebut di buat berdasarkan *k-tetangga* terdekat(Abd Mizwar A. Rahim, 2022).




Gambar 4. 4 Alur Balancing Data.

Pada gambar 4.4 Merupakan alur saat melakukan balancing data dari sebuah dataset yang dinilai sebagai dataset yang mengalami imbalance class , mengatasi imbalance ini dengan penggunaan metode SMOTE, berikut ini penjelasan setiap langkah-langkah melakukan balancing dataset menggunakan metode SMOTE :

- Pertama melakukan import dataset yang dijadikan sebagai bahan proses Analisa klasifikasi.
- Kedua mengidentifikasi apakah dataset tersebut terjadi imbalance data, jika terdapat class mayoritas dan minoritas maka dataset tersebut dinilai sebagai data yang terjadi imbalance class.
- Metode SMOTE menambah kelas minoritas agar sama dengan kelas mayoritas dengan cara menambahkan data buatan, data buatan atau sintesis tersebut di buat berdasarkan *k-tetangga* terdekat. Dengan persamaan

$$X_{syn} = X_i + ( X_{knn} - X_i) \times \delta .$$

Ket :

- Xsyn: Data sintetis yang dihasilkan.
- Xi: Data minoritas yang sedang diproses.
- Xknn: Salah satu dari k-tetangga terdekat dari Xi. K-tetangga terdekat adalah data minoritas lain yang memiliki jarak terdekat dengan Xi dalam ruang fitur.

Sebuah nilai antara 0 dan 1 yang menentukan seberapa jauh jarak antara Xi dan Xknn akan diambil. Semakin besar nilai , semakin besar pula jarak yang diambil, menghasilkan data sintetis yang lebih berbeda dari Xi. Sedangkan semakin

kecil nilai , semakin dekat jarak yang diambil, menghasilkan data sintetis yang lebih mirip dengan Xi.

- Hasil data buatan yang dihasilkan oleh perhitungan jarak akan diurutkan dari terkecil hingga yang terbesar nilainya.
- Yang terakhir membangkitkan data syntesis tersebut pada class minoritas agar sama dengan class mayoritas

Hasil setelah dilakukan SMOTE, maka jadilah dataset yang sudah seimbang classnya. Berikut contoh perhitungan SMOTE pada dataset sebelum dilakukan proses SMOTE.

Tabel 4. 2 Dataset Tidak Seimbang

| No | Atribut 1 | Atribut 2 | Kelas |
|---|---|---|---|
| 1 | 1 | 2 | 1 |
| 2 | 2 | 3 | 1 |
| 3 | 4 | 3 | 1 |
| 4 | 6 | 2 | 2 |
| 5 | 6 | 4 | 2 |
| 6 | 5 | 4 | 2 |
| 7 | 4 | 4 | 2 |
| 8 | 5 | 6 | 2 |
| 9 | 6 | 3 | 2 |
| 10 | 4 | 5 | 2 |
| 11 | 6 | 7 | 2 |
| 12 | 5 | 3 | 2 |

Pada Tabel 4.2 diketahui kelas minoritas atau pada kelas 1 berjumlah 3, dan kelas mayoritas atau pada kelas 2 berjumlah 9, tahapan yang dilakukan untuk memperbanyak kelas minoritas menghitung jarak menggunakan euclidean distance dapat dihitung dengan persamaan (4.1).

$$dist = \sqrt{(X_1-Y_1)^2 + (X_2-Y_2)^2} + \ldots + (X_n - Y_n)^2 \quad (4.1)$$

Ket :

- Dist : Jarak antara dua titik.
- X1, X2, ..., Xn : Koordinat titik pertama dalam ruang n-dimensi.
- Y1, Y2, ..., Yn : Koordinat titik kedua dalam ruang n-dimensi.

Rumus tersebut bekerja dengan cara menghitung kuadrat dari selisih setiap dimensi antara titik pertama (X1, X2, ..., Xn) dan titik kedua (Y1, Y2, ..., Yn), kemudian menjumlahkan semua kuadrat tersebut. Akar kuadrat dari jumlah tersebut kemudian diambil untuk mendapatkan jarak euclidean yang merupakan jarak lurus terpendek antara dua titik dalam ruang n-dimensi.

Agar mendapatkan instance tetangga terdekat Xknn dengan setiap kelas minoritas lainnya, berikut ini replikasi setiap kelas minoritas sebagai berikut :

Data ke-1 dari setiap kelas minoritas :

$$\left(\begin{bmatrix}1 & 1\\ 2 & 2\end{bmatrix}\right) = \sqrt{(1-1)^2 + (2-2)^2} = \sqrt{0}$$

$$\left(\begin{bmatrix}1 & 1\\ 2 & 3\end{bmatrix}\right) = \sqrt{(1-2)^2 + (2-3)^2} = \sqrt{2}$$

$$\left(\begin{bmatrix}1 & 4\\ 2 & 3\end{bmatrix}\right) = \sqrt{(1-4)^2 + (2-3)^2} = \sqrt{10}$$

Jarak dari setiap data ke-1 dengan data minoritas diurutkan dari yang terkecil = 0,2,10. berikut Data ke-2 dari setiap kelas minoritas :

$$d\left(\begin{bmatrix}2 & 1\\ 3 & 2\end{bmatrix}\right) = \sqrt{(2-1)^2 + (3-2)^2} = \sqrt{2}$$

$$d\left(\begin{bmatrix}2 & 2\\ 3 & 3\end{bmatrix}\right) = \sqrt{(2-2)^2 + (3-3)^2} = \sqrt{0}$$

$$d\left(\begin{bmatrix}2 & 4\\ 3 & 3\end{bmatrix}\right) = \sqrt{(2-4)^2 + (3-3)^2} = \sqrt{4}$$

Jarak dari setiap data ke-2 dengan data minoritas diurutkan dari yang terkecil = 0,2,4. berikut Data ke-3 dari setiap kelas minoritas :

$$d\left(\begin{bmatrix}4 & 1\\ 3 & 2\end{bmatrix}\right) = \sqrt{(4-1)^2 + (3-2)^2} = \sqrt{10}$$

$$d\left(\begin{bmatrix}4 & 2\\ 3 & 3\end{bmatrix}\right) = \sqrt{(4-2)^2 + (3-3)^2} = \sqrt{4}$$

$$d\left(\begin{bmatrix}4 & 4\\ 3 & 3\end{bmatrix}\right) = \sqrt{(4-4)^2 + (3-3)^2} = \sqrt{0}$$

Hasil dari proses SMOTE dapat dilihat pada gambar 4.3 dibawah ini. Jarak dari setiap data ke-3 dengan data minoritas diurutkan dari yang terkecil = 0,4,10. Karena kelas mayoritas berjumlah 9 maka replikasi dari setiap data minoritas harus di replikasi sebanyak dua kali N dari SMOTE = 200.

Langkah berikutnya membangkitkan data sintesis, menggunakan persamaan (2.1). berikut ini perhitungan data synthesis pada kelas minoritas, dengan jumlah N(replikasi SMOTE) = 200 Xknn yang digunakan adalah acak data dari banyak N.

Data ke - 1 :

$X_{syn}$ = [1,2] + ([1,2] – [1,2]) x 0,12 = [1,2]

$X_{syn}$ = [1,2] + ([2,3] – [1,2]) x 0,26 = [2,26, 3,26]

Data Ke - 2 :

$X_{syn}$ = [2,3] + ([4,3] – [2,3]) x 0,31 = [2,62, 3]

$X_{syn}$ = [2,3] + ([1,2] – [2,3]) x 0,21 = [1,79 2,79]

Data Ke – 3 :

$X_{syn}$ = [4,3] + ([1,2] – [4,3]) x 0,34 = [2,98, 2,66]

$X_{syn}$ = [4,3] + ([2,3] – [4,3]) x 0,17 = [3,66, 3]

berikut ini hasil setelah dilakukan SMOTE, yang dapat dilihat pada tabel Tabel 4.3.

Tabel 4. 3 Dataset Setelah Dilakukan SMOTE

| No | Atribut 1 | Atribut 2 | Kelas |
|---|---|---|---|
| 1 | 1 | 2 | 1 |
| 2 | 2 | 3 | 1 |
| 3 | 4 | 3 | 1 |
| 4 | 6 | 2 | 2 |
| 5 | 6 | 4 | 2 |
| 6 | 5 | 4 | 2 |
| 7 | 4 | 4 | 2 |
| 8 | 5 | 6 | 2 |
| 9 | 6 | 3 | 2 |
| 10 | 4 | 5 | 2 |
| 11 | 6 | 7 | 2 |
| 12 | 5 | 3 | 2 |
| 13* | 1 | 2 | 1 |
| 14* | 2.26 | 3.26 | 1 |
| 15* | 2.62 | 3 | 1 |
| 16* | 1.79 | 2,79 | 1 |
| 17* | 2.98 | 2.66 | 1 |
| 18* | 3,66 | 3 | 1 |

*) data syntesis baru

Berikut ini hasil balancing class pada dataset penyakit jantung yang dapat dilihat pada tabel 4.4 Hasil Balancing data dengan SMOTE.

Tabel 4. 4 Hasil Balancing Data Dengan SMOTE.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 41.7940009 | 1 | 0 | 115.980003 | 172.980003 | 0 | 0 | 117.588002 | 1 | 2.29900014 | 1 | 0 | 3 | 0 |
| 52 | 1 | 0 | 128 | 255 | 0 | 1 | 161 | 1 | 0 | 2 | 1 | 3 | 0 |
| 46 | 1 | 2 | 150 | 231 | 0 | 1 | 147 | 0 | 3.6 | 1 | 0 | 2 | 0 |
| 60 | 1 | 0 | 125 | 258 | 0 | 0 | 141 | 1 | 2.8 | 1 | 1 | 3 | 0 |
| 48.290754 | 1 | 0.0726885 | 130 | 256 | 1 | 0 | 149.709246 | 1 | 0.0218065 | 1.96365575 | 1.96365575 | 2.9273115 | 0 |
| 46.5065523 | 1 | 1.0131045 | 113.921373 | 245.960687 | 0 | 0.50655225 | 148.052418 | 0 | 0.3947582 | 2 | 0 | 2.49344775 | 0 |
| 57 | 1 | 1 | 154 | 232 | 0 | 0 | 164 | 0 | 0 | 2 | 1 | 2 | 0 |
| 47 | 1 | 2 | 108 | 243 | 0 | 1 | 152 | 0 | 0 | 2 | 0 | 2 | 0 |
| 59 | 1 | 3 | 170 | 288 | 0 | 0 | 159 | 0 | 0.2 | 1 | 0 | 3 | 0 |
| 49.9139191 | 0.18101348 | 0 | 132.172162 | 305.724054 | 0 | 0.81898652 | 142.905067 | 1 | 0.98278383 | 1 | 0.54304043 | 3 | 0 |
| 55.4761052 | 1 | 0 | 110 | 239 | 0 | 0.70477896 | 130.723537 | 1 | 2.32764634 | 1 | 1 | 3 | 0 |
| 57 | 1 | 0 | 130 | 131 | 0 | 1 | 115 | 1 | 1.2 | 1 | 1 | 3 | 0 |
| 50.9207711 | 0.81584577 | 0 | 141.053532 | 200.920771 | 0 | 0.36830846 | 126.736617 | 1 | 1.10256965 | 1 | 0.18415423 | 3 | 0 |
| 69 | 1 | 2 | 140 | 254 | 0 | 0 | 146 | 0 | 2 | 1 | 3 | 3 | 0 |
| 59 | 1 | 3 | 134 | 204 | 0 | 1 | 162 | 0 | 0.8 | 2 | 2 | 2 | 0 |

Pada tabel 4.4 di atas merupakan hasil dari proses balancing kelas pada dataset menggunakan SMOTE, hasil dari SMOTE ini menambahkan hasil pada kelas minoritas pada kelas 0 (tidak ada penyakit) agar seimbang dengan kelas mayoritas yaitu kelas 1 (penyakit). jumlah yang ditambahkan yaitu berjumlah 27 data. Berikut ini adalah code program dari proses metode Smote yang dapat dilihat pada gambar 4.5 dibawah ini.

```
[ ]  1  from imblearn.over_sampling import SMOTE
     2  smote = SMOTE()

[ ]  1  XTrainSmote,YTestsmote = smote.fit_resample(x.astype('float'),y)
     2  YTrainSmote,XTestsmote = smote.fit_resample(x,y)

     1  XTrainSmote.to_csv("xvariabel.csv")

[ ]  1  XTestsmote.to_csv("labeldataset.csv")

setelah di download lalu di gabungkan dalam satu file csv yang mempunyai nama file databalace.csv

Normalisasi data menggunakan min-max scaler

[ ]  1  dataBalance = pd.read_csv("datasetBalance.csv")
     2  dataBalance
```

Gambar 4. 5 Code Program Smote.

Pada gambar 4.5 menjelaskan bahwa proses balancing data ini menggunakan pustaka imbalanced-learn untuk menangani ketidakseimbangan kelas dalam dataset dengan menggunakan metode oversampling SMOTE. Pertama, objek SMOTE diciptakan dan digunakan untuk melakukan oversampling pada dataset pelatihan (XTrainSmote) dan labelnya (YTestsmote). Kemudian, oversampling dilakukan lagi pada dataset uji (XTestsmote) dan labelnya (YTrainSmote). Hasilnya disimpan dalam file CSV terpisah, "xvariabel.csv" untuk

dataset pelatihan yang di-oversampling dan "labeldataset.csv" untuk dataset uji yang di-oversampling.

**4.4. Normalisasi Data.**

Hasil balancing pada dataset ini masih mempunyai atribut dengan nilai skala yang berbeda jauh, contohnya nilai pada kolom age dan nilai pada kolom cp yang dapat dilihat pada tabel 4.1 maka perlu dilakukan standarisasi untuk memiliki skala yang sama saat membangun model pembelajaran mesin. Teknik normalisasi data yang digunakan adalah Min Max normalization yang merupakan metode normalisasi dengan melakukan transformasi linier terhadap data asli sehingga menghasilkan keseimbangan nilai perbandingan antar atribut, atribut-atribut ini ketika dikonversi dan menghasilkan perhitungan yang similaritas, maka nantinya dapat berada di rentang 0 hingga 1.

Keseimbangan tersebut adalah nilai skala antar atribut yang terdapat dalam dataset. Metode ini dapat menggunakan rumus sebagai berikut (Wahanani et al., 2020) :

$$N = \frac{MinRange + (X - MinValue)(MaxRange - MinRange)}{MaxValue - MinValue} \quad (4.2)$$

Dimana :

N = Normalisasi Min_Max

MinRange = Nilai Konversi Kecil Yang ditentukan

MaxRange = Nilai Konversi Terbesar yang ditentukan

MaxValue = Nilai Terbesar pada atribut yang dibandingkan

MinValue = Nilai Terkecil pada atribut yang dibandingkan

Berikut ini contoh normalisasi pada data ke 1031 yang dapat dilihat pada gambar 4.6 contoh hasil normalisasi data. contoh perhitungan normalisasi Min-Max pada atribut age dengan atribut cp.

Diketahui :

MinRange = 0

MaxRange = 1

X = 42.290754

MinValue = 0

MaxValue = 9

Normalisasi = 0+(42.290754-0)(1-0)/9-0 = 4,698972666

Berikut ini hasil standarisasi menggunakan teknik normalisasi min_max pada dataset penelitian ini, dapat dilihat pada gambar 4.6

```
array([[0.5       , 0.        , 0.        , ..., 1.        , 0.
        0.66666667],
       [0.35416667, 1.        , 0.        , ..., 1.        , 0.
        1.        ],
       [0.625     , 1.        , 1.        , ..., 0.5       , 0.
        1.        ],
       ...,
       [0.66666667, 0.        , 0.        , ..., 1.        , 0.
        0.66666667],
       [0.3125    , 1.        , 0.33333333, ..., 1.        , 0.
        1.        ],
       [0.39583333, 1.        , 0.33333333, ..., 0.        , 0.
        1.        ]])
```

Gambar 4. 6 Hasil Normalisasi.

Berikut ini adalah code program dari proses normalisasi menggunakan yang dapat dilihat pada gambar 4.7 dibawah ini.

```
from sklearn.preprocessing import MinMaxScaler

Scaler = MinMaxScaler()

from sklearn.model_selection import train_test_split
Xtrain,Xtest,Ytrain,Ytest = train_test_split (X,Y,random_state=20, test_size=0.2)

Scaler.fit_transform(Xtrain,Ytrain)
```

Gambar 4. 7 Code Program Normalisasi dengan Min-Max Scaller.

Pada gambar 4.7 menyampaikan bahwa proses normalisasi menggunakan pustaka scikit-learn untuk melakukan penskalaan fitur menggunakan metode Min-Max Scaling dan membagi dataset menjadi data pelatihan dan data uji menggunakan fungsi train_test_split. Pertama, objek MinMaxScaler dibuat untuk melakukan penskalaan. Selanjutnya, train_test_split digunakan untuk membagi dataset menjadi data pelatihan (Xtrain dan Ytrain) dan data uji (Xtest dan Ytest) dengan ukuran data uji sebesar 20% dari dataset dan menggunakan seed (random_state) 20 untuk memastikan reproduksibilitas.

**4.5. Split Data.**

Proses ini dilakukan untuk membagikan dataset menjadi data training dan data testing, pembagian pada penelitian ini dibagikan data training dan data testing menjadi 80/20 dan 70/30. Dengan jumlah pembagian tersebut mempunyai tujuan melihat model dalam memprediksi ketika mempunyai data test dengan jumlah 211 dan dengan jumlah 316, secara umum model machine learning mendapatkan hasil akurasi yang baik jika memiliki jumlah data testing yang sedikit dan data training yang banyak. Maka dalam penelitian ini meningkatkan data test dan menguji apakah model mendapatkan hasil yang baik atau tidak. Pada Tabel 4.5 menggambarkan pembagian data yang di lakukan.

Tabel 4. 5 Train/Test Split

| Keterangan | Data Training | Data Testing | Total |
|---|---|---|---|
| **Proporsi** | 80% | 20% | 100% |
| | 70% | 30% | |
| **Jumlah** | 841 | 211 | 1052 |
| | 736 | 316 | |

Pada Tabel 4.5 di jelaskan bahwa pembagian data training dan data testing yang dilakukan yaitu membagikan/split data menjadi 80/20 dan 70/30, 80% untuk data training berjumlah 841 data dan 20% untuk data testing berjumlah 211 data, split yang kedua yaitu 70% dengan jumlah data training 736 dan data testing 316, dengan itu jumlah keseluruhan data dari dataset berjumlah 1052 total data.

**4.6. Klasifikasi Random Forest.**

Penggunaan metode dalam penelitian ini untuk melakukan klasifikasi adalah metode Random Forest, berikut ini contoh langkah-langkah penyusunan Algoritma Random Forest saat melakukan klasifikasi dengan sebuah dataset obat dengan dua fitur (misalnya, dosis dan durasi) dan dua kelas (misalnya, efektif dan tidak efektif) menggunakan algoritma Random Forest. Dalam contoh ini, kita akan membuat dua pohon keputusan dalam ensemble Random Forest dengan contoh datasetnya penyakit jantung.

Tabel 4. 6 Dataset Penyakit Jantung

| No | Usia | Kolesterol | Tekanan Darah | Penyakit Jantung |
|---|---|---|---|---|
| 1 | 63 | 145 | 233 | Tidak |
| 2 | 37 | 130 | 250 | Ya |
| 3 | 41 | 120 | 204 | Ya |
| 4 | 55 | 175 | 198 | Tidak |
| 5 | 45 | 130 | 240 | Ya |
| 6 | 65 | 122 | 220 | Tidak |
| 7 | 71 | 190 | 260 | Tidak |
| 8 | 30 | 150 | 185 | Ya |

Tahapan klasifikasi menggunakan Random Forest:

1. Pembuatan Sampel Bootstrap (Bootstrap Sampling):
   a) Mulai dengan membuat beberapa sampel bootstrap dari dataset. Misalnya, 3 sampel bootstrap.
   b) Setiap sampel bootstrap dibuat dengan memilih secara acak sebagian data dari dataset dengan penggantian.
   c) Sampel bootstrap yang mungkin :
      - Sampel 1: (63, 145, 233, Tidak), (41, 120, 204, Ya), (45, 130, 240, Ya), (30, 150, 185, Ya), (63, 145, 233, Tidak).
      - Sampel 2: (37, 130, 250, Ya), (55, 175, 198, Tidak), (71, 190, 260, Tidak), (30, 150, 185, Ya), (65, 122, 220, Tidak).
      - Sampel 3: (37, 130, 250, Ya), (45, 130, 240, Ya), (55, 175, 198, Tidak), (65, 122, 220, Tidak), (71, 190, 260, Tidak).
2. Pembuatan Pohon Keputusan (Decision Tree Building):
   a) Untuk setiap sampel bootstrap, kita bangun pohon keputusan.

b) Proses ini melibatkan pemilihan fitur terbaik untuk setiap split berdasarkan kriteria seperti Information Gain atau Gini Impurity.

c) Misalkan kita bangun 3 pohon keputusan untuk setiap sampel. Contoh kali ini akan mencontohkan satu pohon keputusan untuk setiap sampel.

3. Pemilihan Pohon:

a) Misalkan kita memilih untuk membuat 3 pohon keputusan.

4. Prediksi:

a) Setelah kita memiliki pohon-pohon keputusan, kita lakukan prediksi untuk setiap sampel uji menggunakan setiap pohon.

b) Misalnya, untuk setiap pohon pada setiap sampel, kita dapatkan prediksi sebagai berikut :

- Prediksi untuk Sampel 1: [Tidak, Ya, Ya, Ya, Tidak]
- Prediksi untuk Sampel 2: [Ya, Tidak, Tidak, Ya, Tidak]
- Prediksi untuk Sampel 3: [Tidak, Ya, Tidak, Tidak, Tidak]

5. Aggregasi Prediksi:

a) Setelah mendapatkan prediksi dari setiap pohon, kita ambil mayoritas prediksi untuk setiap sampel sebagai hasil akhir.

b) Misalnya, mayoritas prediksi untuk setiap sampel adalah sebagai berikut :

- Hasil klasifikasi untuk Sampel 1: "Tidak" (3 dari 5 pohon memprediksi "Tidak").
- Hasil klasifikasi untuk Sampel 2: "Ya" (2 dari 5 pohon memprediksi "Ya")

- Hasil klasifikasi untuk Sampel 3: "Tidak" (4 dari 5 pohon memprediksi "Tidak").

Klasifikasi kemungkinan pasien terindikasi penyakit Jantung atau tidak sebagaimana yang dilakukan dalam penelitian ini. Untuk proses klasifikasi langkah utama yang dilakukan ialah tuning parameter. Hasil dari tuning parameter dapat dilihat pada tabel 4.7 dibawah ini.

Tabel 4. 7 Tuning Parameter Metode Random Forest 80/20.

| Pengujian pada split data 80/20 | | | | | |
|---|---|---|---|---|---|
| No | *Max_Dept* | *n_estimators* | *min_ samples_ leaf* | *min_ samples_ split* | Acc |
| 1 | 30 | 250 | 30 | 30 | 97% |
| 2 | 29 | 230 | 28 | 28 | 99% |
| 3 | 28 | 210 | 27 | 27 | 96% |
| 4 | 27 | 200 | 26 | 26 | 94% |
| 5 | 26 | 180 | 24 | 25 | 92% |
| 6 | 25 | 170 | 22 | 22 | 91% |
| 7 | 24 | 160 | 20 | 23 | 88% |
| 8 | 23 | 120 | 18 | 20 | 85% |
| 9 | 21 | 140 | 17 | 18 | 86% |
| 10 | 22 | 110 | 16 | 19 | 82% |
| 11 | 18 | 100 | 12 | 17 | 82% |
| 12 | 16 | 80 | 9 | 15 | 83% |

Tabel 4.7 dijelaskan bahwa parameter-parameter yang diuji pada proses klasifikasi meliputi *max-depth* parameter ini menentukan kedalaman maksimum dari setiap pohon keputusan dalam ensemble. Semakin dalam pohon, semakin

kompleks modelnya, yang dapat menyebabkan overfitting jika tidak diatur dengan benar. Pada saat yang sama, jika terlalu dangkal, model mungkin gagal memahami pola yang rumit dalam data(Akbar & Sanjaya, 2023). *n-estimator* mengontrol jumlah pohon keputusan dalam ensemble. Semakin banyak pohon, semakin baik kemungkinan model untuk melakukan generalisasi dengan baik, karena akan menangkap lebih banyak variasi dalam data. Namun, terlalu banyak pohon bisa memperlambat pelatihan dan meningkatkan penggunaan memori(Zega, 2014). *random-state* parameter ini menentukan biji (seed) untuk inisialisasi pembangunan model. Mengatur nilai ini memastikan reproduktibilitas hasil yang sama setiap kali Anda menjalankan model. Ini penting terutama jika Anda ingin membandingkan kinerja model dengan parameter yang berbeda. *min-samples-leaf* Ini menentukan jumlah sampel minimum yang diperlukan di setiap leaf node dari pohon keputusan. Memperbesar nilai ini dapat mencegah pohon-pohon yang terlalu dalam dan mengurangi kecenderungan overfitting(Setio et al., 2020). *dan min-samples-split* parameter ini menentukan jumlah sampel minimum yang diperlukan untuk membagi sebuah node internal. Ini juga dapat membantu mencegah overfitting dengan memastikan bahwa pembagian yang terjadi hanya ketika ada cukup banyak data yang tersedia(Dimas Ariyoga, 2022).

Pada percobaan yang dilakukan, parameter-parameter tersebut diatur dengan nilai yang tinggi dalam iterasi awal, dan pada iterasi berikutnya, parameter tersebut diatur dengan nilai yang rendah. Percobaan yang dilakukan berjumlah dua belas percobaan dengan menggunakan data seimbang menghasilkan akurasi tertinggi sebesar 99% yaitu pada percobaan ke 2. hasil terbaik pada dua belas

percobaan tersebut ketika memiliki nilai parameter seperti n_estimator yang tidak kurang dari 230 dan tidak lebih besar dari 230, lalu max_dept tidak lebih dari 29 dan tidak kurang dari 29, berikutnya pada parameter min_samples_leaf dan min_samples_split yaitu tidak kurang dan tidak lebih dari 28 maka akurasi yang dihasilkan akan lebih baik jika dibandingkan dengan meningkatkan-nya kedua parameter n_estimator, max_dept ataupun merendahkan dari nilai tersebut begitu juga pada parameter min_samples_leaf dan min_samples_split maka dapat meningkatkan akurasi sebanyak 1% yaitu 99% dari hasil penelitan sebelumnya yang hanya menggunakan algoritma Svm saja tanpa menggunakan metode imbalance yaitu metode SMOTE mendapatkan akurasi 98%, dengan hasil 99% ini merupakan hasil dari percobaan ke-2.

Tabel 4. 8 Tuning Parameter Metode Random Forest 70/30.

| Pengujian pada split data 70/30 | | | | | |
|---|---|---|---|---|---|
| No | *Max_Dept* | *n_estimators* | *min_ samples_ leaf* | *min_ samples_ split* | Acc |
| 1 | 30 | 250 | 30 | 30 | 96% |
| 2 | 29 | 230 | 28 | 28 | 98% |
| 3 | 28 | 210 | 27 | 27 | 94% |
| 4 | 27 | 200 | 26 | 26 | 91% |
| 5 | 26 | 180 | 24 | 25 | 89% |
| 6 | 25 | 170 | 22 | 22 | 87% |
| 7 | 24 | 160 | 20 | 23 | 86% |
| 8 | 23 | 120 | 18 | 20 | 86% |
| 9 | 21 | 140 | 17 | 18 | 87% |
| 10 | 22 | 110 | 16 | 19 | 81% |
| 11 | 18 | 100 | 12 | 17 | 80% |
| 12 | 16 | 80 | 9 | 15 | 79% |

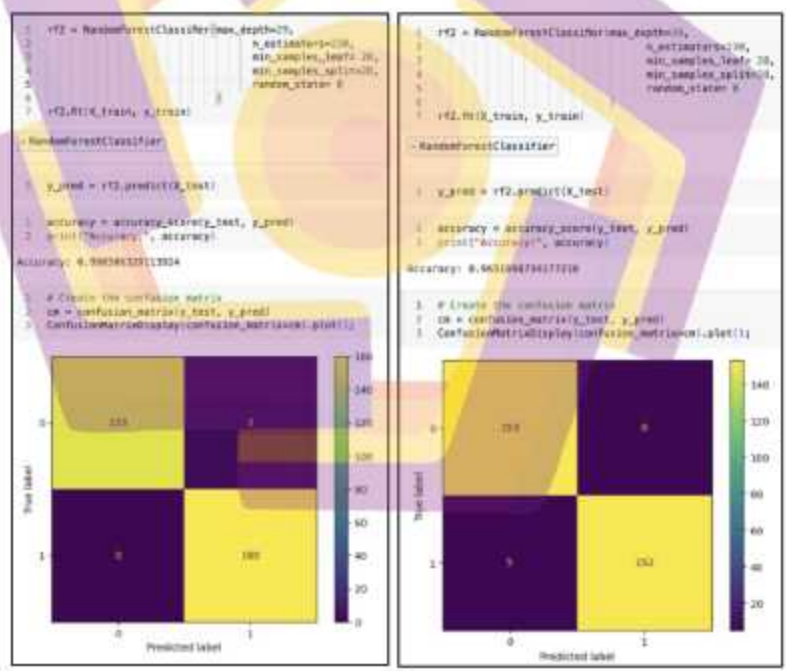
Tabel 4.8 dijelaskan bahwa parameter -parameter yang diuji mempunyai nilai yang sama dengan pengujian menggunakan split data 80/20. Percobaan yang dilakukan juga berjumlah dua belas percobaan dengan menggunakan data seimbang menghasilkan akurasi tertinggi sebesar 97% yaitu pada percobaan ke 2. hasil terbaik pada dua belas percobaan tersebut ketika memiliki nilai parameter seperti n_estimator yang tidak kurang dari 230 dan tidak lebih besar dari 230, lalu max_dept tidak lebih dari 29 dan tidak kurang dari 29, berikutnya pada parameter min_samples_leaf dan min_samples_split yaitu tidak kurang dan tidak lebih dari 28 maka akurasi yang dihasilkan akan lebih baik jika dibandingkan dengan

meningkatkan-nya kedua parameter n_estimator, max_dept ataupun merendahkan dari nilai tersebut begitu juga pada parameter min_samples_leaf dan min_samples_split maka mendapatkan akurasi 98%, akurasi ini masih tetap sama dengan hasil penelitan sebelumnya yang hanya menggunakan algoritma Svm saja tanpa menggunakan metode imbalance yaitu metode SMOTE mendapatkan akurasi 98%, dengan hasil 98% ini merupakan hasil dari percobaan ke-2 menggunakan split 70/30. Hasil dari tuning parameter dapat dilihat pada tabel 4.9 dibawah ini, ini merupakan pengujian ke 3 pada dataset yang belum dilakukan balancing dengan metode smote.

Tabel 4. 9 Tuning Parameter Metode Random Forest 80/20 Tanpa Smote.

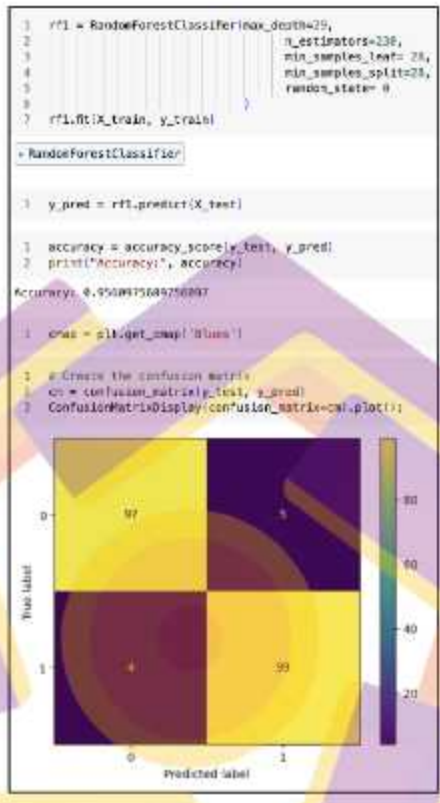
| Pengujian pada split data 80/20 tanpa Smote. | | | | | |
|---|---|---|---|---|---|
| No | *Max_Dept* | *n_estimators* | *min_ samples_ leaf* | *min_ samples_ split* | Acc |
| 1 | 30 | 250 | 30 | 30 | 95% |
| 2 | 29 | 230 | 28 | 28 | 96% |
| 3 | 28 | 210 | 27 | 27 | 89% |
| 4 | 27 | 200 | 26 | 26 | 88% |
| 5 | 26 | 180 | 24 | 25 | 87% |
| 6 | 25 | 170 | 22 | 22 | 85% |
| 7 | 24 | 160 | 20 | 23 | 79% |
| 8 | 23 | 120 | 18 | 20 | 80% |
| 9 | 21 | 140 | 17 | 18 | 79% |
| 10 | 22 | 110 | 16 | 19 | 74% |
| 11 | 18 | 100 | 12 | 17 | 73% |
| 12 | 16 | 80 | 9 | 15 | 73% |

Pengujian ini memiliki kondisi yang sama dengan pengujian yang dilakukan sebelumnya dimulai dari nilai parameter yang diujikan dan juga nilai splitting data yaitu 80:20. Hasil terbaik pada pengujian tanpa smote ini yaitu pada pengujian ke 2 dengan hasil akurasi yang dihasilkan adalah 96%. Hasil ini tidak sebaik hasil pengujian dengan menggunakan data yang sudah diseimbangkan yaitu 99% akurasinya. Berikut ini code program dari pengujian menggunakan metode random forest, code program yang ditampilkan ini merupakan hasil code program terbaik dari pengujian yang berbeda yaitu pengujian dengan split 70/30 menggunakan smote, 80/20 menggunakan smote, dan 80/20 tanpa smote.



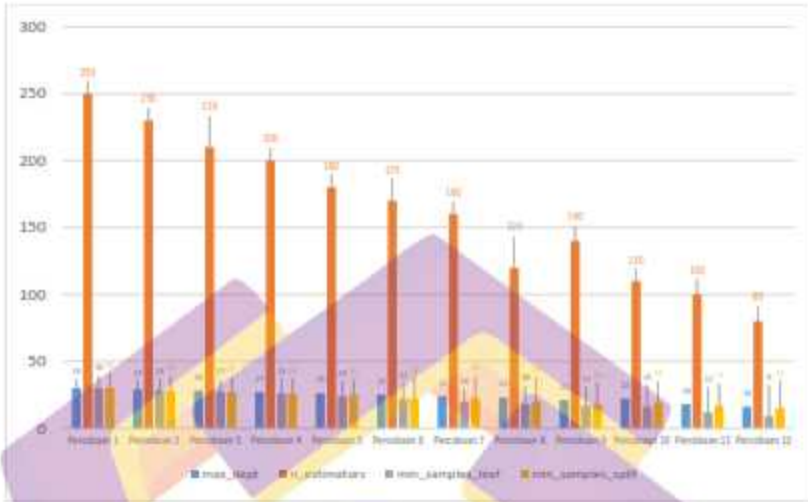

80/20 dengan smote. 70/30 dengan Smote.

Gambar 4. 8 Code Program Klasifikasi Dengan Random Forest.




80/20 tanpa smote

Gambar 4.8 (lanjutan) Code Program Klasifikasi dengan random forest.

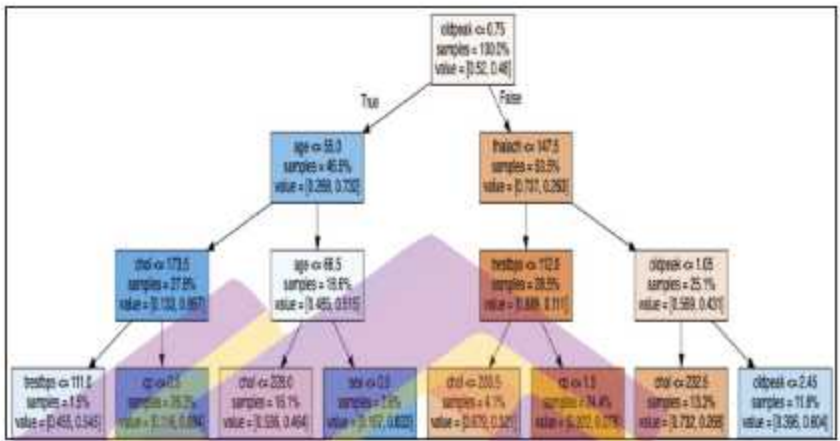



Gambar 4. 9 Nilai Tunning Parameter.

Gambar 4.9 Menunjukan bahwa nilai parameter yang terjadi perendahan nilai adalah parameter max-depth, n-estimator, min-samples-leaf, dan min-samples-split. Percobaan yang dilakukan oleh penelitian ini berjumlah 12 percobaan, dan hasil terbaik-nya terdapat pada percobaan ke-2 dengan hasil akurasi 99%, yang mempunyai kedalaman maksimum pohon dibatasi hingga 29 level, ada 230 pohon ensemble, 28 sampel dalam setiap daun, dan 28 sampel yang memenuhi syarat untuk pemisahan simpul. hasil akurasi dari keseluruhan percobaan menggunakan split data 80/20. Dapat dilihat pada gambar 4.6.


Gambar 4. 10 Akurasi.

Berikut ini pohon yang terbentuk dari metode Random forest pada percobaan yang memiliki hasil terbaik dari penelitian ini yaitu percobaan ke-2 pada pengujian split data 80/20 yang dapat dilihat pada gambar 4.11.




Gambar 4. 11 Pohon Random Forest.

Pada gambar 4.11, merupakan contoh satu pohon yang membatasi kedalaman pohon dalam visualisasi hingga 3 tingkatan dari random forest. Maksud dari visualisasi pohon ini sebagai berikut :

- Jika nilai "oldpeak" kurang dari atau sama dengan 0.75, maka semua sampel yang memenuhi kondisi ini mencakup 100% dari total sampel dalam pohon ini. Nilai "value" [0.52, 0.48] mengindikasikan distribusi kelas. Dalam konteks ini, [0.52, 0.48] mengacu pada dua kelas atau kategori, 52% sampel termasuk dalam kelas 0 dan 48% dalam kelas 1. Lalu dilanjut ke fitur berikutnya yaitu age karena age mendapatkan yang nilai kurang dari 0.75
- Jika "age" kurang dari atau sama dengan 55.0, maka sebagian dari sampel (46.5%) termasuk dalam kelas 1 (nilai [0.268, 0.732]), sedangkan sisanya termasuk dalam kelas 0. Jika "age" lebih dari 55.0, maka sebagian besar sampel (100%) termasuk dalam kelas 0 (nilai [0.52,0.48]). Dll sebagainya jika nilai sebuah fitur kurang dari 0,75.

- Berikutnya jika sebuah fitur memiliki nilai yang lebih dari atau sama dengan 0,75 maka diartikan sebagai false atau membuat cabang baru pada pohon random forest. Contoh-nya pada fitur thalach.
- Ketika nilai "thalach" kurang dari atau sama dengan 147.5, sebagian dari sampel (53.5%) termasuk dalam kelas 0 sedangkan sisanya termasuk dalam kelas 1. Nilai nya adalah [0.737, 0.263] mengindikasikan distribusi kelas di dalam cabang ini. Nilai tersebut merujuk kepada dua kelas atau kategori yang mengindikasikan bahwa sekitar 73.7% dari sampel termasuk dalam kelas 0 dan sekitar 26.3% dalam kelas 1. Begitu juga pada fitur yang lain jika nilainya lebih dari 0,75. Maka fitur tersebut masuk pada cabang dari pohon random forest yaitu cabang yang memiliki nilai lebih dari 0.75 atau diartikan false dari pohon ini.

Akurasi dari dua belas percobaan yang dilakukan menunjukan bahwa Percobaan ke-2 menggunakan split data 80/20 dengan data seimbang mendapatkan nilai akurasi terbaik yaitu 99%, berikut 10 hasil klasifikasi dari percobaan ke- 2 yang dapat dilihat pada gambar 4.12.

Sepuluh Klasifikasi Pada Percobaan ke-2

```
hasil_prediksi = pd.DataFrame(y_pred)
print("Hasil Klasifikasi Random Forest : \n - 1 = pasien penyakit jantung \n - 0 = pasien tidak penyakit jantung")
```

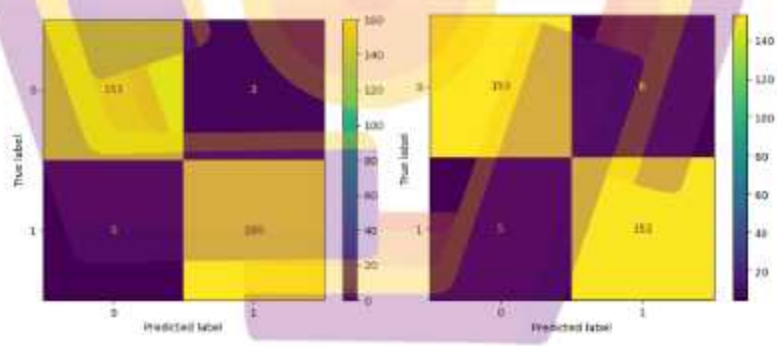
```
Hasil Klasifikasi Random Forest :
 - 1 = pasien penyakit jantung
 - 0 = pasien tidak penyakit jantung    0
0  1
1  0
2  1
3  0
4  1
5  0
6  0
7  1
8  0
9  1
```

Gambar 4. 12 Sepuluh Hasil Klasifikasi Random Forest.

Pada gambar 4.12 ini merupakan hasil klasifikasi yang ditampilkan yang berjumlah 10 data dari hasil test, yang pertama model memprediksi sebagai kelas 1 atau kelas pasien terindikasi penyakit jantung, dan pada data test yang kedua model memprediksi kelas 0 atau kelas pasien yang tidak mengalami penyakit jantung, hasil klasifikasi data test berikutnya dapat dilihat pada gambar 4.12 diatas.

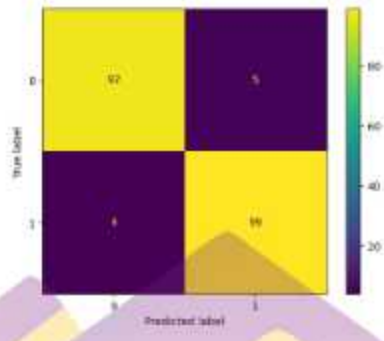
### 4.7. Evaluasi Metode Confusion Matrix.

Percobaan yang dilakukan berjumlah 12 kali, dengan split data 80/20 dengan smote, 70/30 dengan smote dan 80/20 tanpa smote. Berikut ini hasil confusion matrix terbaik dari keseluruhan percobaan yang dilakukan. Dapat dilihat pada Gambar 4.13.




Percobaan ke-2 80/20 dengan smote. Percobaan ke-2 70/30 dengan smote.

Gambar 4. 13 Hasil Confusion Matriks.




Percobaan ke-2 80/20 tanpa smote.

Gambar 4.13 (Lanjutan) Hasil Confusion Matriks.

Gambar 4.13 menunjukan hasil evaluasi terbaik percobaan yang dilakukan menggunakan confusion matrix. Nilai evaluasi terbaik terdapat pada percobaan pada split data 80/20 yang ke-2 , hasil tersebut menyampaikan bahwa sebanyak 3 data diprediksi oleh model yang seharusnya class 0 (tidak terindikasi penyakit) tetapi di prediksi sebagai class 1 (terindikasi penyakit). Eksperimen ke dua mendapatkan hasil terbaik dengan nilai akurasi sebesar 99% yang dapat dilihat pada Gambar 4.6. Akurasi merupakan rasio prediksi yang tepat dalam mengidentifikasi seseorang terindikasi penyakit dan tidak terindikasi penyakit secara keseluruhan dalam dataset. Jumlah data yang diprediksi salah pada pengujian ke dua (pengujian terbaik) adalah 3 data, dan total data yang benar di klasifikasikan adalah 313 data. Terjadi penurunan akurasi mulai dari 1% hingga 2% saat nilai parameter max-depth, n-estimator, min-samples-leaf, dan min-samples-split di kurangi, terjadi juga peningkatan akurasi 2% saat keempat nilai parameter yaitu max-depth, n-estimator, min-samples-leaf, dan min-samples-split ditingkatkan. Keempat nilai parameter tersebut jika melebihi value yang diatur pada pengujian ke-2 maka akurasi terjadi

penurunan sebanyak 2% yaitu menjadi 97% akurasi. Hasil confusion matriks pada percobaan ke-2 yaitu berupa True Positive (TP) sebanyak 153, untuk True Negatif (TN) adalah 160, untuk False Positif (FP) adalah 3, dan untuk False Negative (FN) adalah 0. Hasil dari pengujian dengan data tanpa smote hasilnya tidak sebaik penggunaan data yang sudah seimbang, hasilnya adalah 96% akurasinya, terjadi penurunan 3% akurasi.

Confusion matrix memiliki nilai seperti Nilai True Positive (TP), True Negatif (TN), False Positif (FP), False Negative (FN) maksud antara nilai-nilai hasil confusion matrix tersebut ialah sebagai berikut :

a. True Positive (TP) dimana pasien yang diprediksi penyakit jantung, memang benar secara actual bahwa pasien tersebut terindikasi penyakit jantung.
b. True Negatif (TN) dimana pasien yang di prediksi tidak terindikasi penyakit jantung, dan memang benar bahwa pasien tersebut tidak terindikasi penyakit jantung.
c. False Positif (FP) dimana pasien secara actual nya tidak terindikasi penyakit jantung, tetapi diprediksi terindikasi penyakit jantung.

False Negative (FN) dimana pasien tersebut terindikasi penyakit jantung, tetapi diprediksi tidak penyakit jantung.

**4.8. Clasification Report.**

Hasil pengujian sebelumnya, dievaluasi menggunakan confusion matrix untuk menilai kinerja dari model yang dibangun menggunakan algoritma random forest pada data yang sudah seimbang. terlihat hasil terbaik dari percobaan yang dilakukan dengan beberapa parameter yang di ujikan maka percobaan ke-2 dengan

split data 80/20 memiliki hasil akurasi terbaik, nilai akurasinya adalah 99%. Hasil classification report dari pengujian terbaik dapat diilihat pada gambar 4.14.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.98 | 0.99 | 156 |
| 1 | 0.98 | 1.00 | 0.99 | 160 |
| accuracy | | | 0.99 | 316 |
| macro avg | 0.99 | 0.99 | 0.99 | 316 |
| weighted avg | 0.99 | 0.99 | 0.99 | 316 |

Gambar 4. 14 Hasil Classification Report Percobaan Ke-2.

Dari gambar 4.14 merupakan Hasil Classification Report pada percobaan ke-2 menggunakan data yang seimbang kelasnya. Hasil dari percobaan ke-2 diatas memiliki nilai accuracy 99% yang dimana dengan accuracy ini merupakan rasio pasien yang benar diprediksi penyakit jantung dan Tidak penyakit jantung dari kesuluruhan data yang ada pada dataset, lalu recall mengetahui bahwa Pasien yang diprediksi penyakit jantung dibandingkan dengan keseluruhan Pasien yang sebenarnya penyakit jantung, dengan nilai 100%, Presisi menunjukan rasio Pasien yang benar penyakit jantung yang ada pada dataset dari keseluruhan Pasien yang diprediksi penyakit jantung yaitu 98%, perbandingan rata-rata presisi dan recall yang dibobotkan (F1 Score) yang bernilai 99%.

Hasil accuracy, presisi, recal, dan f1-score. dapat dihitung dengan menggunakan hasil dari confusion matrix yaitu nilai True Positive (TP), True Negatif (TN), False Positif (FP), False Negative (TN). Hasil akurasi percobaan ke-2 ini dapat diketahui dengan menggunakan rumus perhitungan sebagai berikut :

$$Accuracy\ (\%) = \frac{(TP+TN)}{(TP+TN+FP+FN)}\ x\ 100 = \frac{(153+160)}{(153+160+3+0)} = 99\%.$$

Selain nilai akurasi juga dapat mengukur ukuran lainnya seperti Presisi, Recal, dan F1-Score, dengan menggunakan rumus perhitungan sebagai berikut:

$$Recall\ (\%) = \frac{(TP)}{(TP+FP)}\ x\ 100 = \frac{(153)}{(153+0)} = 100\%.$$

$$Precision\ (\%) = \frac{(TP)}{(TP+FN)}\ x\ 100 = \frac{(153)}{(153+3)}\ x\ 100 = 98\%.$$

$$F1\text{-}Score\ (\%) = 2\ x\ \frac{(Precision\ x\ Recall)}{(Precision\ +\ Recall)} = 99\%.$$

Berikut ini code program dari nilai evaluasi Akurasi, Recall, Precision, dan F1-Score pada percobaan terbaik yaitu percobaan ke-2 dengan menggunakan split data 80/20 dengan data yang sudah seimbangkan. Dapat dilihat pada gambar 4.15.


Gambar 4. 15 Code program hasil evaluasi Akurasi, Recall, Precision, dan F1-Score.

Terdapat beberapa teknik pada penelitian ini yang diimplementasikan kedalam experiment, sehingga terjadi peningkatan akurasi yang lebih baik dari akurasi yang dihasilkan sebelumnya, teknik tersebut adalah Teknik preprosessing data yaitu menggantikan nilai kosong yang terdapat pada dataset dan normalisasi data, lalu penerapan metode klasifikasi yang digunakan yaitu metode random forest

dan mengatur parameter yang sesuai dengan pola yang terjadi dalam dataset agar menghasilkan model yang optimal, dan metode untuk dapat mengatasi imbalance kelas yaitu diterapkannya metode *SMOTE*.

Salah satu teknik preprosessing data yang digunakan penelitian ini menggantikan nilai kosong pada atribut dalam dataset, dengan ini maka tidak lagi terdapat missing value dalam variabel dataset, jika terdapat missing value dalam dataset maka model klasifikasi akan menghasilkan output yang bias, tentunya dengan hal ini akan mengurangi hasil akurasi model dalam mengklasifikasikan penyakit jantung ini. adapun teknik lain untuk mengatasi missing value yaitu menghapuskan missing value pada atribut dataset, dengan teknik ini maka banyak data dalam dataset akan terbuang, dibandingkan dengan replace missing value tidak menghapus/membuang banyak data dalam dataset. Berikutnya Teknik pre-prosessing yang diterapkan yaitu normalisasi data. Normalisasi data dilakukan untuk membantu menghindari bias yang dapat muncul ketika variabel-variabel dalam dataset memiliki skala yang berbeda, memperlancar perhitungan algoritma, memungkinkan model untuk konvergen lebih cepat selama pelatihan, meningkatkan akurasi prediksi dengan algoritma yang bergantung pada jarak atau bobot, mencegah overfitting dengan menjaga seimbangnya fitur-fitur, dan bahkan dapat meningkatkan interpretabilitas model.

Metode *SMOTE* digunakan untuk mengatasi imbalance kelas pada dataset, jika model melakukan klasifikasi pada data dengan jumlah kelas yang tidak seimbang maka model hanya dapat mengklasifikasi dengan benar pada kelas mayoritas dan ketika model mempredikasi kelas minoritas maka akan diprediksi

sebagai kelas mayoritas, terdapat beberapa metode selain metode *SMOTE* yang dapat mengatasi imbalance kelas yaitu under-sampling teknik dan over sampling teknik. under sampling teknik menghapus kelas mayoritas secara acak agar menyamakan kelas minoritas sedangkan over sampling teknik menduplikat kelas minoritas secara random agar menyamakan kelas mayoritas. dampak yang terjadi ketika digunakan kedua metode tersebut adalah kehilangan banyak data, dan terjadinya overfitting dalam melakukan klasifikasi.

Tabel 4. 10 Perbandingan Penelitian.

| Author | Metode | Hasil Akurasi |
|---|---|---|
| Mohan,S, Dkk.,(2019) | HRFLM | 88% |
| Singh, A & Kumar, R.,(2020) | KNN | 87% |
| Ari, B, M.,Dkk (2019) | Naïve Bayes Classifier | 83% |
| Liu T, Dkk | DNN Berbasis Autohpo | 87%. |
| Nurmasani,A.,Dkk (2021) | Algoritme stacking | 90% |
| V,R,V.,Dkk (2018) | Naïve Bayes, K – Nearest Neighbour, Decision Tree, Support Vector Machine, Random Forest | 98% |
| Yadav,A.,Dkk (2021) | Svm | 73% |
| Hidayat, Dkk (2023) | Random Forest | 94% |
| **Penelitian Yang diajukan** | **Random Forest & SMOTE** | **99%** |

Pada tabel 4.10 merupakan tabel Perbandingan penelitian sebelumnya dengan penelitian ini. dataset yang diolah dalam penelitian sebelumnya dengan penelitian ini sama yaitu Heart Disease Dataset, terdapat perbedaan antara penelitian sebelumnya dengan penelitian yang dilakukan yaitu penggunaan metode klasifikasi. Metode klasifikasi yang digunakan antara lain yaitu Decision Tree, Svm, Random forest, Naïve Bayes Classifier dll, yang dapat dilihat pada tabel 4.10. dari perbandingan tersebut ketika menggunakan metode random forest dalam melakukan klasifikasi memiliki hasil akurasi yang lebih baik dibandingkan dengan hasil akurasi yang dihasilkan oleh penelitian sebelumnya. hasil dari klasifikasi menggunakan metode Random Forest memiliki hasil akurasi yang lebih baik dari hasil akurasi penelitian sebelumnya yaitu sebesar 99%.

# BAB V

# PENUTUP

## 5.1. Kesimpulan.

Berdasarkan hasil yang didapatkan pada proses analisa yang dilakukan maka dapat ditarik kesimpulan sebagai berikut :

1. Penelitian ini ketika menggunakan metode untuk mengatasi ketidakseimbangan kelas yaitu metode SMOTE, dan untuk proses klasifikasi menggunakan metode Random Forest mampu menaikan akurasi 1% dari penelitian sebelumnya yang menggunakan metode Svm, hasil akurasi yang dihasilkan pada penelitian ini adalah 99%.
2. Penerapan teknik pre-prosessing data seperti normalisasi data dengan min max, dan mengatasi missing value, maka setiap hasil percobaan yang dilakukan terjadi pengurangan hasil eror yaitu fp dan fn. Dengan hasil ini mendapatkan peningkatan hasil evaluasi akurasi sebesar 1%.

## 5.2. Saran.

Penelitian ini melakukan pemilihan hyperparameter masih secara manual pada metode yang digunakan untuk proses klasifikasi, yang dimana akan terdapat banyak uji coba yang dilakukan agar mendapatkan parameter yang optimal untuk proses klasifikasi. Dengan ini saran yang diberikan adalah penerapan Grid Search untuk menemukan parameter terbaik dalam model klasifikasi ini.

## DAFTAR PUSTAKA


Sabiq Sofyan, A. P. (2013). Penerapan Synthetic Minority Oversampling Technique (SMOTE) Terhadap Data Tidak Seimbang Pada Pembuatan Model Komposisi Jamu. Xplore: Journal of Statistics, 1(1), 868–877. Https://doi.org/10.29244/xplore.v1i1.12424

Kasanah, A. N, Dkk. (2019). Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 3(2), 196–201. Https://doi.org/10.29207/resti.v3i2.945

Doreswamy, & Hemanth, K. S. (2011). Performance Evaluation of Predictive Engineering Materials Data Sets. Artificial Intelligent Systems Ans Machine Learning, 3(3), 1–8.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing and Management, 45(4), 427–437. Https://doi.org/10.1016/j.ipm.2009.03.002

Abd Mizwar A. Rahim. (2022). *Prediksi Stroke Menggunakan Metode Synthetic Minority Over-Sampling Technique Dan Xtreme Gradient Boosting*.

Adiezwara Ramadhan Frananda, M. (2021). *Portofolio New Clasification Machine Learning*. Https://bisa.ai/portofolio/detail/MTI3Nw#:~:text=Klasifikasi dalam machine learning adalah,learning atau pembelajaran yang diawasi.

Akbar, H., & Sanjaya, W. K. (2023). Kajian Performa Metode Class Weight Random Forest pada Klasifikasi Imbalance Data Kelas Curah Hujan. *Jurnal Sains, Nalar, Dan Aplikasi Teknologi Informasi*, *3*(1). Https://doi.org/10.20885/snati.v3i1.30

Algonz D.B. Raharja. (2022). *Machine Learning: Pengertian, Cara Kerja, dan 3 Metodenya*. Https://www.ekrut.com/media/apa-itu-machine-learning

Alodokter. (2023). *Pengertian Penyakit Jantung*. Https://www.alodokter.com/penyakit-jantung

Alvin. (2019). *Indonesia, Rincian Biaya Perawatan 4 Penyakit Kritis Penyebab Kematian Tertinggi di*. Https://ilovelife.co.id/blog/rincian-biaya-perawatan-4-penyakit-kritis/

Bhowmick, A., Mahato, K. D., Azad, C., & Kumar, U. (2022). Heart Disease Prediction Using Different Machine Learning Algorithms. *Proceedings - 2022 IEEE World Conference on Applied Intelligence and Computing, AIC 2022*, 60–65. Https://doi.org/10.1109/AIC55036.2022.9848885

Bianto, M. A., Kusrini, K., & Sudarmawan, S. (2020). Perancangan Sistem Klasifikasi Penyakit Jantung Mengunakan Naïve Bayes. *Creative Information Technology Journal*, *6*(1), 75. Https://doi.org/10.24076/citec.2019v6i1.231

Cutler, D. R., Edwards, T. C., Beard, K. H., Cutler, A., Kyle, T., Gibson, J., Lawler, J. J., Beard, H., & Hess, T. (2012). Random Forests for

Classification in Ecology Published by : Ecological Society of America content in a trusted digital archive . We use information technology and tools to increase productivity and facilitate new forms of scholarship . For more informatio. *Ecology*, *88*(11), 2783–2792.

Dicoding Intern. (2020). *Apa itu Machine Learning? Beserta Pengertian dan Cara Kerjanya*. Https://www.dicoding.com/blog/machine-learning-adalah/

Dimas Ariyoga. (2022). *Perbandingan metode seleksi fitur filter, wrapper, dan embedded pada klasifikasi data nirs mangga menggunakan random forest dan support vector machine (svm)*.

Doreswamy, & Hemanth, K. S. (2011). Performance Evaluation of Predictive Engineering Materials Data Sets. *Artificial Intelligent Systems Ans Machine Learning*, *3*(3), 1–8.

Dqlab. (2022). *Apa Itu Machine Learning? Simak Pengertian Hingga Contoh*. Https://dqlab.id/apa-itu-machine-learning-simak-pengertian-hingga-contoh

Ervina. (2019). *Klasifikasi Data: Pengertian, Jenis, Hingga Metodenya*. Https://www.talenta.co/blog/klasifikasi-data-2/

Fadli, R. (2022). *Penyakit Jantung*. Https://www.halodoc.com/kesehatan/penyakit-jantung

Hospitals, M. S. (2023). *Kenali Faktor Risiko Penyakit Jantung Koroner Sejak Dini*. Https://www.siloamhospitals.com/informasi-siloam/artikel/faktor-risiko-penyakit-jantung-koroner

Irawan, E., & Wahono, R. S. (2015). Penggunaan Random Under Sampling untuk Penanganan Ketidakseimbangan Kelas pada Prediksi Cacat Software Berbasis Neural Network. *Journal of Software Engineering*, *1*(2), 92–100.

Janosi, A. S. W. P. M. Dan D. Robert. (1988). *Penyakit jantung*. . Repositori Pembelajaran Mesin UCI.

Jason Brownlee. (2020). *SMOTE for Imbalanced Classification with Python*. Https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Kasanah, A. N., Muladi, M., & Pujianto, U. (2019). Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *3*(2), 196–201. Https://doi.org/10.29207/resti.v3i2.945

Kemkes. (2017). *Penyakit Jantung Penyebab Kematian Tertinggi, Kemenkes Ingatkan CERDIK*. Https://www.kemkes.go.id/article/view/17073100005/penyakit-jantung-penyebab-kematian-tertinggi-kemenkes-ingatkan-cerdik-.html

Kovács, B., Tinya, F., Németh, C., & Ódor, P. (2020). Unfolding the effects of different forestry treatments on microclimate in oak forests: results of a 4-yr experiment. *Ecological Applications*, *30*(2), 321–357. Https://doi.org/10.1002/eap.2043

Kuncahyo Setyo Nugroho. (2019). *Confusion Matrix untuk Evaluasi Model pada Supervised Learning*. Https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f

Liaw, A., & Wiener, M. (2002). Classification and Regression by randomforest. *R News*, *2*(3), 18–22.

Makarim, F. R. (2023). *Penyakit Jantung Koroner*. Https://www.halodoc.com/kesehatan/penyakit-jantung-koroner

Maula, R. A., Gunawan, A. I., Bayu Dewantara, B. S., Al Rasyid, M. U. H., Setiawardhana, S., Saputra, F. A., & Ispianto, J. (2022). Handling Missing Value dengan Pendekatan Regresi pada Dataset Akuakultur Berukuran Kecil. *Jurnal Rekayasa Elektrika*, *18*(3), 175–184. Https://doi.org/10.17529/jre.v18i3.25903

Mohan, S., Thirumalai, C., & Srivastava, G. (2019). Effective heart disease prediction using hybrid machine learning techniques. *IEEE Access*, *7*, 81542–81554. Https://doi.org/10.1109/ACCESS.2019.2923707

Nurmasani, A., & Pristyanto, Y. (2021). Algoritme Stacking Untuk Klasifikasi Penyakit Jantung Pada Dataset Imbalanced Class. *Pseudocode*, *8*(1), 21–26. Https://doi.org/10.33369/pseudocode.8.1.21-26

Pittara. (2022). *BAHAYA SERANGAN JANTUNG*. Https://puskesmas.kuburayakab.go.id/sungai-durian/read/173/bahaya-serangan-jantung#:~:text=Serangan jantung yang parah atau,syok kardiogenik%2C dan henti jantung.

Prasad, A. M., Iverson, L. R., & Liaw, A. (2006). Newer classification and regression tree techniques: Bagging and random forests for ecological prediction. *Ecosystems*, *9*(2), 181–199. Https://doi.org/10.1007/s10021-005-0054-1

Prasetio, R. T., & Pratiwi, D. (2015). *PENERAPAN TEKNIK BAGGING PADA ALGORITMA KLASIFIKASI UNTUK MENGATASI KETIDAKSEIMBANGAN KELAS DATASET MEDIS*. *Vol. II* (Issue 2).

Ramalingam, Dandapath, A., & Karthik Raja, M. (2018). Heart disease prediction using machine learning techniques: A survey. *International Journal of Engineering and Technology(UAE)*, *7*(2.8 Special Issue 8), 684–687. Https://doi.org/10.14419/ijet.v7i2.8.10557

Reinert Yosua Rumagit. (2019). *Imbalanced Dataset*. Https://socs.binus.ac.id/2019/12/26/imbalanced-dataset/

RI, P. K. (2019). *Hari Jantung Sedunia (World Heart Day): Your Heart is Our Heart Too*. Https://p2ptm.kemkes.go.id/kegiatan-p2ptm/pusat-/hari-jantung-sedunia-world-heart-day-your-heart-is-our-heart-too

Royal Society of Great Britain. (2017). *Machine learning : the power and promise of computers that learn by example*.

Sabiq Sofyan, A. P. (2013). Penerapan Synthetic Minority Oversampling Technique (Smote) Terhadap Data Tidak Seimbang Pada Pembuatan Model Komposisi Jamu. *Xplore: Journal of Statistics*, *1*(1), 868–877. Https://doi.org/10.29244/xplore.v1i1.12424

Setio, P. B. N., Saputro, D. R. S., & Winarno, B. (2020). *Klasifikasi dengan Pohon Keputusan Berbasis Algoritme C4.5*. *3*, 64–71. Https://journal.unnes.ac.id/sju/index.php/prisma/

Shiva Shanta Mani, B., & Manikandan, V. M. (2020). Heart disease prediction using machine learning. *Handbook of Research on Disease Prediction Through Data Analytics and Machine Learning*, 373–381. Https://doi.org/10.4018/978-1-7998-2742-9.ch018

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. Https://doi.org/10.1016/j.ipm.2009.03.002

Widiastiwi, Y., & Ernawati, I. (2021). Klasifikasi Penyakit Batu Ginjal Menggunakan Algoritma Decision Tree C4 . 5 Dengan Membandingkan Hasil Uji Akurasi. *Jurnal IKRA-ITH INFORMATIKA*, *5*(2), 128.

Zega, S. A. (2014). Penggunaan Pohon Keputusan untuk Klasifikasi Tingkat Kualitas Mahasiswa Berdasarkan Jalur Masuk Kuliah. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) Yogyakarta*.

Haristu, R. A. (2019). Penerapan Metode Random Forest Untuk Prediksi Win Ratio Pemain Player Unknown Battleground. Yogyakarta: Skripsi.

Lingga P, R. D. (2017). Deteksi Gempa Berdasarkan Data Twitter Menggunakan Decision Tree, Random Forest, dan SVM. Jurnal Teknik ITS, 160.

Claesen, M., & De Moor, B. (2015). *Hyperparameter Search in Machine Learning*. 10–14. Http://arxiv.org/abs/1502.02127.