

**OPTIMASI MODEL *RANDOM FOREST* UNTUK DETEKSI PENYAKIT
GAGAL JANTUNG DENGAN *GRIDSEARCHCV***

SKRIPSI

Oleh :
MUHAMMAD AZRA DWI RIZKY
NIM. 210605110097



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**OPTIMASI MODEL *RANDOM FOREST* UNTUK DETEKSI PENYAKIT
GAGAL JANTUNG DENGAN *GRIDSEARCHCV***

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
MUHAMMAD AZRA DWI RIZKY
NIM 210605110097

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

HALAMAN PERSETUJUAN

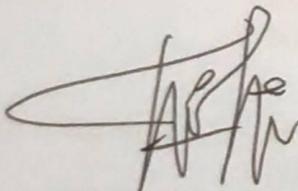
OPTIMASI MODEL *RANDOM FOREST* UNTUK DETEKSI PENYAKIT
GAGAL JANTUNG DENGAN *GRIDSEARCHCV*

SKRIPSI

Oleh :
MUHAMMAD AZRA DWI RIZKY
NIM 210605110097

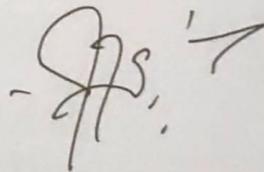
Telah Diperiksa dan Disetujui untuk
Diuji: Tanggal: 5 Juni 2025

Pembimbing I,



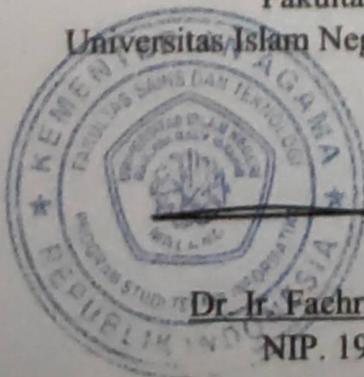
Tri Mukti Lestari, M.Kom
NIP. 19911108 202012 2 005

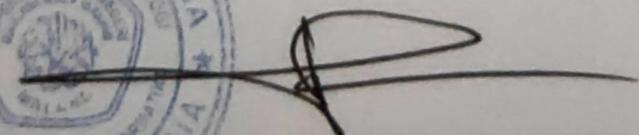
Pembimbing II,



Khadijah Fahmi Hayati Holle, M.Kom
NIP. 19900626 202203 2 002

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Ir. Fachrul Kurniawan, M.MT., IPU
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

**OPTIMASI MODEL *RANDOM FOREST* UNTUK DETEKSI PENYAKIT
GAGAL JANTUNG DENGAN *GRIDSEARCHCV***

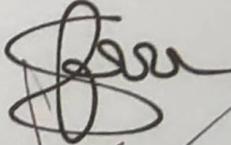
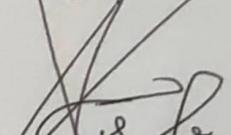
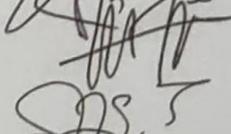
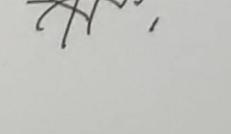
SKRIPSI

Oleh:

MUHAMMAD AZRA DWI RIZKY
NIM. 210605110097

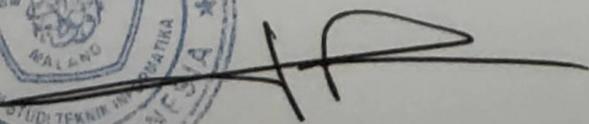
Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 23 Juni 2025

Susunan Dewan Penguji

Ketua Penguji	: <u>Dr. M. Amin Hariyadi, M.T</u> NIP. 19670018 200501 1 001	()
Anggota Penguji I	: <u>Dr. Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004	()
Anggota Penguji II	: <u>Tri Mukti Lestari, M.Kom</u> NIP. 19911108 202012 2 005	()
Anggota Penguji III	: <u>Khadijah Fahmi Hayati Holle, M.Kom</u> NIP. 19900626 202203 2 002	()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Ir. Fachrul Kurniawan, M.MT., IPU
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Azra Dwi Rizky
NIM : 210605110097
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Optimasi Model *Random Forest* Untuk Deteksi Penyakit Gagal Jantung Dengan *GridsearchCV*.

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 15 Juni 2025
Yang membuat pernyataan,



Muhammad Azra Dwi Rizky
NIM. 210605110097

MOTTO

“Just Get Good”

HALAMAN PERSEMBAHAN

Puji Syukur atas kehadiran Allah Subhanahu wa ta'ala, karena berkat rahmat dan petunjuk-Nya penulis dapat menyelesaikan skripsi ini. Shalawat serta salam kepada Rasulullah Shallallahu 'alaihi wasallam, yang telah membawa kita dari zaman jahiliyah menuju addinul Islam.

Penulis mempersembahkan karya ini kepada orang tua, keluarga, dan teman-teman yang telah membantu, mendukung dan menemani penulis selama berkuliah.

KATA PENGANTAR

Bismillahirrahmaanirrahiim, Assalamu'alaikum wr. wb.

Segala puji dan syukur yang tak terhingga penulis panjatkan kepada Allah SWT atas rahmat dan petunjuk-Nya. Berkat karunia-Nya, penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Optimasi Model *Random Forest* Untuk Deteksi Penyakit Gagal Jantung Dengan *Gridsearchcv*”. Oleh karena itu, dengan penuh rendah hati, penulis mengucapkan terima kasih kepada:

1. Prof. Dr. H. M. Zainuddin, M. A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan, M.MT, IPU., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, atas motivasi dan arahan yang diberikan selama proses perkuliahan.
4. Tri Mukti Lestari, M.Kom, selaku dosen Pembimbing 1 yang telah membimbing dan mengajarkan saya banyak pelajaran berharga hingga selesai.
5. Khadijah Fahmi Hayati Holle, M.Kom, selaku dosen pembimbing II, yang dengan penuh kesabaran membimbing penulis hingga selesai
6. Dr. M. Amin Hariyadi, M.T dan Dr. Irwan Budi Santoso, M.Kom, selaku dosen penguji saya, yang telah memberikan kritik dan saran yang berharga selama proses ujian skripsi.
7. Keluarga penulis, terutama ibu dan ayah telah memberikan dorongan motivasi luar biasa sehingga penulis mampu sampai pada titik ini.

8. Seluruh teman-teman Angkatan 2021 Teknik Informatika, terutama Sulthon dan anak-anak Pesma yang telah banyak membantu, mendukung, dan memotivasi dalam menyelesaikan penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan, sehingga masukan, kritik, dan saran yang membangun dari pembaca sangat diharapkan untuk perbaikan di masa mendatang. Semoga karya ini dapat bermanfaat bagi para pembaca sekalian serta memberikan kontribusi positif dalam pengembangan ilmu pengetahuan.

Malang, 15 Juni 2025

Penulis

DAFTAR ISI

HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
ABSTRAK	xiii
ABSTRACT	xiv
المخلص	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Pernyataan Masalah	5
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
BAB II STUDI PUSTAKA	6
2.1 Penelitian Terkait.....	6
2.2 Penyakit Gagal Jantung	11
2.3 <i>Random Forest</i>	12
2.4 <i>GridSearchCV</i>	14
BAB III METODE PENELITIAN	16
3.1 Data Eksperimen yang Digunakan.....	17
3.2 <i>Exploratory Data Analysis (EDA)</i>	18
3.3 <i>Preprocessing</i>	28
3.3.1 Penanganan <i>Outlier</i>	28
3.3.2 <i>One-Hot Encoding</i>	29
3.3.3 <i>Label Encoding</i>	30
3.4 <i>Split Data</i>	31
3.5 Desain Sistem	31
3.6 Implementasi <i>GridsearchCV</i>	32
3.7 Implementasi Algoritma <i>Random forest</i>	34
3.8 Skenario Uji Coba.....	38
3.9 Evaluasi.....	40
BAB IV HASIL DAN PEMBAHASAN	43
4.1 Hasil Pengujian.....	43
4.1.1 Pengujian tanpa <i>GridsearchCV</i>	43
4.1.2 Pengujian dengan <i>GridsearchCV</i>	52
4.1.3 Pembahasan	61
BAB V KESIMPULAN	75
5.1 Kesimpulan.....	75
5.2 Saran	76
DAFTAR PUSTAKA	77
LAMPIRAN	80

DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi cara kerja <i>Random Forest</i>	13
Gambar 3. 1 prosedur penelitian	17
Gambar 3. 2 Pengecekan <i>Missing Value</i>	18
Gambar 3. 3 Distribusi Target	19
Gambar 3. 4 Distribusi Data <i>Categorical</i>	20
Gambar 3. 5 distribusi Pasien berdasarkan <i>Sex</i>	21
Gambar 3. 6 distribusi Pasien berdasarkan <i>ChestPainType</i>	22
Gambar 3. 7 distribusi Pasien berdasarkan <i>RestingECG</i>	23
Gambar 3. 8 distribusi Pasien berdasarkan <i>ST_Slope</i>	24
Gambar 3. 9 distribusi Pasien berdasarkan <i>ExerciseAngina</i>	25
Gambar 3. 10 distribusi Pasien berdasarkan <i>FastingBS</i>	26
Gambar 3. 11 Distribusi Fitur <i>Numerical</i>	27
Gambar 3. 12 Penanganan <i>Outlier</i> pada <i>Cholesterol</i>	29
Gambar 3. 13 Fitur yang telah dirubah dengan <i>Label Encoding</i>	30
Gambar 3. 14 Desain Sistem.....	31
Gambar 3. 15 Flowchart Algoritma <i>GridsearchCV</i>	32
Gambar 3. 16 Flowchart <i>Random Forest</i>	38
Gambar 4. 1 <i>confusion matrix</i> model A1 - NE tanpa <i>GridsearchCV</i>	44
Gambar 4. 2 <i>confusion matrix</i> model A1 - E tanpa <i>GridsearchCV</i>	46
Gambar 4. 3 <i>confusion matrix</i> model B1 - NE tanpa <i>GridsearchCV</i>	47
Gambar 4. 4 <i>Confusion matrix</i> model B1 - E tanpa <i>GridsearchCV</i>	48
Gambar 4. 5 <i>confusion matrix</i> model C1 - NE tanpa <i>GridsearchCV</i>	50
Gambar 4. 6 <i>confusion matrix</i> model C1 – E tanpa <i>GridsearchCV</i>	51
Gambar 4. 7 <i>confusion matrix</i> model A2 - NE dengan <i>GridsearchCV</i>	53
Gambar 4. 8 <i>confusion matrix</i> model A2 – E dengan <i>GridsearchCV</i>	54
Gambar 4. 9 <i>confusion matrix</i> model B2 - NE dengan <i>GridsearchCV</i>	56
Gambar 4. 10 <i>confusion matrix</i> model B2 – E dengan <i>GridsearchCV</i>	57
Gambar 4. 11 <i>confusion matrix</i> model C2 - NE dengan <i>GridsearchCV</i>	59
Gambar 4. 12 <i>confusion matrix</i> model C2 dengan <i>GridsearchCV</i>	60
Gambar 4. 13 Perbandingan Akurasi Tanpa dan Dengan <i>GridsearchCV</i>	61
Gambar 4. 14 Perbandingan Presisi Tanpa dan Dengan <i>GridsearchCV</i>	62
Gambar 4. 15 Perbandingan <i>Recall</i> Tanpa dan Dengan <i>GridsearchCV</i>	64
Gambar 4. 16 Perbandingan <i>F1-score</i> Tanpa dan Dengan <i>GridsearchCV</i>	65

DAFTAR TABEL

Tabel 3. 1 Deskripsi Dataset	17
Tabel 3. 2 Sampel Data setelah <i>One-Hot Encoding</i>	30
Tabel 3. 3 <i>Tuning hyperparameter</i>	34
Tabel 3. 4 Split Data	39
Tabel 3. 5 <i>Confusion Matrix</i>	40
Tabel 4. 1 Hasil <i>confusion matrix</i> Model A1 - NE tanpa <i>GridsearchCV</i>	44
Tabel 4. 2 Hasil <i>confusion matrix</i> Model A1 - E tanpa <i>GridsearchCV</i>	46
Tabel 4. 3 Hasil <i>confusion matrix</i> Model B1 - NE tanpa <i>GridsearchCV</i>	47
Tabel 4. 4 Hasil <i>confusion matrix</i> Model B1 - E tanpa <i>GridsearchCV</i>	48
Tabel 4. 5 Hasil <i>confusion matrix</i> Model C1 - NE tanpa <i>GridsearchCV</i>	50
Tabel 4. 6 Hasil <i>confusion matrix</i> Model C1 – E tanpa <i>GridsearchCV</i>	51
Tabel 4. 7 Hasil <i>confusion matrix</i> Model A2 - NE dengan <i>GridsearchCV</i>	53
Tabel 4. 8 Hasil <i>confusion matrix</i> Model A2 – E dengan <i>GridsearchCV</i>	54
Tabel 4. 9 Hasil <i>confusion matrix</i> Model B2 - NE dengan <i>GridsearchCV</i>	56
Tabel 4. 10 Hasil <i>confusion matrix</i> Model B2 – E dengan <i>GridsearchCV</i>	57
Tabel 4. 11 Hasil <i>confusion matrix</i> Model C2 - NE dengan <i>GridsearchCV</i>	59
Tabel 4. 12 Hasil <i>confusion matrix</i> Model C2 dengan <i>GridsearchCV</i>	60
Tabel 4. 13 Proses <i>GridsearchCV</i> pada Model C2	69
Tabel 4. 14 Contoh prediksi dari model C2	70

ABSTRAK

Rizky, Muhammad Azra Dwi. 2025. **Optimasi Model *Random Forest* Untuk Deteksi Penyakit Gagal Jantung Dengan *GridsearchCV***. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Tri Mukti Lestari, M.Kom, (II) Khadijah Fahmi Hayati Holle, M.Kom.

Kata kunci: Deteksi gagal jantung, *Random Forest*, *GridsearchCV*, optimasi *hyperparameter*

Penyakit gagal jantung merupakan salah satu penyebab utama kematian tertinggi di dunia, sehingga deteksi dini menjadi sangat diperlukan. Pada klasifikasi untuk bidang medis, algoritma *Random Forest* banyak digunakan karena kemampuannya dalam menangani data kompleks dan non-linear. Penelitian ini bertujuan untuk mengoptimasi model *Random Forest* dengan menggunakan *GridsearchCV* yang dapat meningkatkan kinerja model dengan mencari kombinasi *hyperparameter* terbaik. Dataset yang digunakan adalah *Heart Failure Prediction Dataset* dari *Kaggle* yang terdiri dari 918 data dan 11 fitur. Evaluasi dilakukan dengan membandingkan performa model sebelum dan sesudah optimasi pada beberapa skenario. Hasil penelitian menunjukkan bahwa optimasi *GridsearchCV* memberikan hasil yang beragam pada tiap skenario uji, dengan model terbaik adalah C2 – NE, yang menghasilkan akurasi 89%, presisi 89.5%, *recall* 89%, dan *F1-score* 89%, serta memberikan peningkatan performa sebesar 2% pada akurasi dan presisi, dan 2.5% pada *recall* dan *F1-score* dibandingkan model tanpa optimasi.

ABSTRACT

Rizky, Muhammad Azra Dwi. 2025. **Optimization of Random Forest Model for Heart Failure Detection Using GridSearchCV**. Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Supervisors: (I) Tri Mukti Lestari, M.Kom, (II) Khadijah Fahmi Hayati Holle, M.Kom

Key words: Heart failure detection, Random Forest, GridSearchCV, hyperparameter optimization.

The role of the gaming industry is immense in today's economy by making important contributions to economic growth. According to Fortune Business Insights In 2023, the value of the global gaming market surpassed USD \$200 billion, showing very strong growth exceeding many traditional entertainment sectors. User reviews are necessary for the continued growth of the gaming industry. However, while reviews provide many insights, their unstructured and subjective nature often poses a challenge for developers. The purpose of this research is to identify the main aspects of the *game* reviews which are divided into 3 main categories namely *gameplay*, *Graphics*, and *Story* from user reviews automatically using *neural networks*. Among the 5 models trained based on the test scenario, Model 4 with a *training* and *testing* data ratio of 60:40, a *learning rate* of 0,01, and an error tolerance of 1e-10 provides the best performance with an *Accuracy* percentage of 90% and an average percentage of *precision* 89%, *recall* 91%, *F1-score* 90%

الملخص

رسالة **GridSearchCV** رضكي، محمد أزرا دوي. ٢٠٢٥. تحسين نموذج غابة القرار لاكتشاف فشل القلب باستخدام
.تخرج. برنامج دراسة هندسة المعلومات، كلية العلوم والتكنولوجيا، الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانغ
المشرفان: (١) تري موكتي ليستاري، ماجستير في الحاسوب، (٢) خديجة فهمي حياتي هول، ماجستير في الحاسوب

تحسين معاملات النموذج، **GridSearchCV**، الكلمات المفتاحية: اكتشاف فشل القلب، غابة القرار

يعد مرض قصور القلب أحد الأسباب الرئيسية للوفاة في العالم، مما يجعل الكشف المبكر عنه أمرًا لا غنى عنه. في التصنيف الطبي، تُستخدم خوارزمية الغابة العشوائية على نطاق واسع نظرًا لقدرتها على التعامل مع البيانات المعقدة وغير الخطية. تهدف هذه التي يمكنه تحسين أداء النموذج من خلال إيجاد أفضل **GridsearchCV** الدراسة إلى تحسين نموذج الغابة العشوائية باستخدام والتي تتكون من **Kaggle** مجموعة من الملعلمات الفائقة. مجموعة البيانات المستخدمة هي مجموعة بيانات التنبؤ بقصور القلب من 918 بيانات و 11 سمة. تم إجراء التقييم من خلال مقارنة أداء النموذج قبل التحسين وبعده في عدة سيناريوهات. أظهرت النتائج 918 والذي، **C2 - NE** أعطى نتائج متباينة في كل سيناريو اختبار، حيث كان أفضل نموذج هو **GridsearchCV** أن تحسين وقدم تحسينًا في الأداء بنسبة **F1** أدى إلى دقة 89% ودقة 89.5% واستدعاء 89% من التذكر و 89% من درجة مقارنة بالنموذج بدون تحسين **F1** في الدقة والدقة و 2.5% في التذكر ودرجة 2%

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penyakit gagal jantung merupakan salah satu penyebab utama kematian global, dengan 17,9 juta kematian setiap tahun atau sekitar 31% dari total kematian global (WHO, 2021). Di Indonesia, gagal jantung termasuk dalam 10 besar penyakit tidak menular, dengan angka prevalensi yang terus meningkat dan usia pasien yang relatif lebih muda dibandingkan negara-negara maju. Tingginya angka komplikasi dan kematian penyakit ini sebagian besar disebabkan oleh diagnosis yang terlambat serta kurangnya kesadaran masyarakat terhadap gejala awal, seperti sesak napas dan kelelahan ekstrem. Gagal jantung didefinisikan sebagai sindrom klinis yang diakibatkan oleh abnormalitas struktur atau fungsi jantung, sering kali disertai dengan gejala kongesti paru atau sistemik. Penyakit ini diperparah oleh faktor risiko perilaku, seperti pola makan yang buruk, kurang aktivitas fisik, konsumsi tembakau, serta faktor lingkungan seperti polusi udara. Faktor-faktor ini dapat memicu peningkatan tekanan darah, kadar glukosa darah, kadar lipid darah, serta obesitas, yang secara signifikan meningkatkan risiko komplikasi serius seperti stroke, serangan jantung, dan gagal jantung kronis.

Dalam Islam, kesehatan merupakan salah satu hal yang paling penting, dimana kita sebagai umat manusia harus selalu menjaga kesehatan. Al-Qur'an sendiri pada Surah Al-Baqarah ayat 57 menjelaskan bahwa:

وَوَلَّلْنَا عَلَيْكُمُ الْغَمَامَ وَأَنزَلْنَا عَلَيْكُمُ الْمَنَّاءَ وَالسَّلْوَىٰ ۗ كُلُوا مِن طَيِّبَاتِ مَا رَزَقْنَاكُمْ ۗ وَمَا ظَلَمُونَا وَلَٰكِن

كَانُوا أَنفُسَهُمْ يَظْلِمُونَ

"Kami menaungi kamu dengan awan dan Kami menurunkan kepadamu manna dan salwa. Makanlah (makanan) yang baik-baik dari rezeki yang telah Kami berikan kepadamu. Mereka tidak menzalimi Kami, tetapi justru merekalah yang menzalimi diri sendiri." (QS. Al-Baqarah: 57).

Ayat ini memberikan pelajaran penting bahwa salah satu cara menjaga tubuh tetap sehat adalah dengan mengonsumsi makanan yang baik, bergizi, dan halal. Islam mengajarkan bahwa tubuh adalah amanah yang harus dijaga, sehingga memilih makanan yang sehat merupakan bentuk rasa syukur kepada Allah atas nikmat rezeki yang diberikan. Selain itu, menjaga kesehatan tidak hanya terkait dengan asupan makanan, tetapi juga dengan menjaga kebersihan, menjalankan pola hidup seimbang, dan menghindari hal-hal yang dapat merusak tubuh dan mendatangkan penyakit seperti mengonsumsi makanan atau minuman yang berbahaya (TafsirWeb.com, n.d.).

Meskipun penyakit jantung merupakan salah satu penyebab utama kematian di Indonesia, kesadaran masyarakat terhadap bahaya penyakit ini masih sangat rendah. Banyak masyarakat yang tidak menyadari risiko yang mereka hadapi dan kurang memperhatikan gejala-gejala awal yang muncul, seperti nyeri dada atau mudah lelah. Hal ini membuat diagnosis penyakit jantung sering kali terlambat dilakukan, sehingga risiko komplikasi serius menjadi lebih tinggi. Oleh karena itu, penerapan teknologi *machine learning* untuk deteksi dini penyakit jantung dapat menjadi solusi efektif dalam meningkatkan kesadaran serta kualitas penanganan

medis. Teknologi ini lebih unggul dibandingkan metode konvensional yang sering memakan waktu, sangat bergantung pada keahlian tenaga medis, dan rentan terhadap kesalahan manusia, yang dapat menurunkan akurasi diagnosis, terutama pada tahap awal gejala yang tidak spesifik (Muhammad et al., 2020)..

Penelitian terkait deteksi penyakit gagal jantung telah dilakukan dengan berbagai pendekatan dan algoritma. Misalnya, (García-Ordás et al., 2023) mengeksplorasi metode *deep learning* dengan *feature augmentation* untuk meningkatkan kinerja model, sementara (Akther et al., 2024) berfokus pada pengembangan algoritma yang konsisten pada berbagai dataset serta integrasinya ke dalam sistem berbasis web. Selain itu, (Hermawan et al., 2024) menggunakan algoritma SVM dan *Logistic Regression* untuk meningkatkan akurasi diagnosis penyakit gagal jantung dan penelitian oleh (Tamba & others, 2022) yang membuat sistem prediksi penyakit gagal jantung dengan menggunakan algoritma *Random Forest*, dan mendapat hasil akurasi model yang memuaskan. Meskipun berbagai penelitian telah dilakukan, masih terdapat ruang untuk pengembangan lebih lanjut, khususnya dalam hal optimasi *hyperparameter* seperti penggunaan *GridsearchCV* dan pemanfaatan fitur-fitur klinis yang relevan untuk meningkatkan akurasi prediksi.

Oleh karena itu, peneliti memilih untuk mengusulkan model pendeteksi penyakit gagal jantung menggunakan algoritma *Random Forest* yang juga dioptimasi dengan *GridsearchCV*. Dalam penelitian ini, algoritma *Random Forest* digunakan karena kemampuannya yang andal dalam menangani dataset dengan banyak fitur dan kompleksitas tinggi. *Random Forest* bekerja dengan membangun

sejumlah *decision tree* pada subset data yang berbeda, kemudian menggabungkan prediksi dari masing-masing *tree* untuk meningkatkan akurasi sekaligus mengurangi risiko *overfitting* (Genuer et al., 2020). Keunggulan lain dari algoritma ini adalah kemampuannya dalam menangani data yang tidak linier serta menghasilkan *feature importance*, yang memberikan wawasan tentang fitur mana yang paling relevan dalam prediksi. Hal ini sangat penting dalam konteks medis untuk memahami variabel-variabel yang memengaruhi kondisi pasien. Untuk meningkatkan kinerja model, penelitian ini juga menerapkan *GridSearchCV*, sebuah teknik optimasi *hyperparameter* yang secara sistematis mencari kombinasi parameter terbaik. Dengan *GridSearchCV*, model dapat dioptimalkan untuk mencapai akurasi yang lebih tinggi, mencegah *overfitting*, dan meningkatkan generalisasi pada data baru.

Melalui penelitian ini, diharapkan hasil klasifikasi yang dihasilkan oleh algoritma *Random Forest* dapat membantu mengidentifikasi individu yang berisiko secara lebih akurat. Dengan demikian, langkah-langkah preventif dan intervensi dini dapat segera diambil oleh tenaga medis. Selain itu, hasil deteksi ini dapat menjadi referensi tambahan untuk masyarakat umum agar lebih peduli terhadap kesehatan jantungnya dan segera melakukan konsultasi medis jika ditemukan gejala-gejala yang mencurigakan. Harapannya, dengan adanya sistem deteksi berbasis *machine learning* ini, masyarakat dapat lebih waspada terhadap penyakit jantung dan berperan aktif dalam menjaga kesehatannya.

1.2 Pernyataan Masalah

- a. Bagaimana performa model *Random Forest* yang dioptimasi dengan *GridsearchCV* dalam melakukan deteksi penyakit gagal jantung?
- b. Seberapa besar peningkatan performa model *Random Forest* sebelum dan sesudah menggunakan *GridsearchCV*?

1.3 Batasan Masalah

Dataset yang digunakan diambil dari data bank *Kaggle* dengan judul “*Heart Failure Prediction Dataset*” (fedesoriano, 2021). Dataset ini dikumpulkan dari beberapa dataset terdahulu yang terdiri dari Cleveland, Hungarian, Switzerland, Long Beach VA, dan Stalog. Dataset berisi 918 record dengan 11 fitur yang merupakan ciri-ciri umum penyakit gagal jantung.

1.4 Tujuan Penelitian

- a. Untuk mengetahui bagaimana performa model *Random Forest* yang dioptimasi dengan *GridsearchCV* dalam melakukan deteksi penyakit gagal jantung.
- b. Untuk mengetahui seberapa besar peningkatan performa model *Random Forest* sebelum dan sesudah menggunakan *GridsearchCV*.

1.5 Manfaat Penelitian

1. Penelitian ini diharapkan dapat membantu mengklasifikasi penyakit gagal jantung sehingga dapat meningkatkan akurasi sistem deteksi dini, dalam tindakan preventif, diagnosis dan pengobatan yang lebih efektif.
2. Penelitian ini diharapkan bisa menjadi sumber referensi untuk penelitian selanjutnya.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Klasifikasi adalah tindakan mengatur suatu objek dalam kelompok atau kategori berdasarkan prinsip atau kriteria yang telah disetujui. Dalam *machine learning*, klasifikasi digunakan untuk membagi data atau objek ke dalam kelas tertentu berdasarkan kemiripan fitur atau karakteristik yang dimiliki oleh masing-masing data. Proses ini melibatkan analisis terhadap atribut atau variabel yang relevan dari data, kemudian menentukan kelompok yang paling sesuai untuk setiap objek (Schalock et al., 2021).

Tujuan utama dari klasifikasi adalah untuk memprediksi label atau kelas secara akurat dari data baru yang belum diketahui dengan menggunakan model yang dibangun berdasarkan data yang sudah diketahui kelasnya (Religia et al., 2019). Klasifikasi dapat diaplikasikan pada berbagai jenis data, seperti teks, gambar, sinyal, maupun data numerik, dan dapat digunakan untuk menyelesaikan berbagai masalah, yang mana dalam konteks penelitian ini adalah deteksi penyakit gagal jantung.

Penelitian terkait deteksi penyakit gagal jantung telah banyak dilakukan sebelumnya dengan berbagai metode, seperti *preprocessing data*, optimasi *hyperparameter*, dan seleksi fitur, telah diterapkan untuk meningkatkan akurasi model. Penjelasan mengenai penelitian-penelitian yang relevan dan berkontribusi terhadap penelitian ini disajikan dengan detail lebih lengkap pada Tabel 2.1.

Penelitian oleh (García-Ordás et al., 2023) membangun sistem prediksi risiko penyakit jantung dengan melakukan *preprocessing* data yang unik. Fitur usia dihapus dan diganti dengan tiga kategori baru, yaitu *young*, *adult*, dan *elder*. Langkah serupa diterapkan pada fitur tekanan darah (*Resting BP*) dan kolesterol, di mana keduanya dikonversi menjadi kategori risiko (rendah, sedang, tinggi). Model prediksi dikembangkan menggunakan algoritma CNN dengan SAE, menunjukkan pentingnya peran *preprocessing* data dalam meningkatkan kinerja model serta memberikan pendekatan yang inovatif untuk klasifikasi penyakit gagal jantung.

Penelitian oleh (Akther et al., 2024) memanfaatkan empat dataset berbeda dari *Kaggle* untuk melakukan klasifikasi penyakit gagal jantung. Dua teknik seleksi fitur, yaitu PCA dan Lasso, digunakan untuk membandingkan performa model dengan tiga skenario: tanpa seleksi fitur, menggunakan PCA, dan menggunakan Lasso. Hasilnya, didapat algoritma *XGBoost* dengan PCA mendapat hasil tertinggi dibanding dengan skenario lainnya Model terbaik ini diintegrasikan ke dalam aplikasi web untuk mendukung implementasi praktis.

Penelitian oleh (Hermawan et al., 2024) melakukan prediksi gagal jantung menggunakan algoritma *Support Vector Machine (SVM)* dan *Logistic Regression*. Dalam penelitian ini, peneliti juga melakukan perhitungan linearitas dalam dataset yang digunakan. Hasil analisis menunjukkan bahwa data tidak sepenuhnya linear, sehingga pada hasil klasifikasi menunjukkan bahwa algoritma SVM mendapat performa lebih baik dibandingkan dengan *Logistic Regression*.

Penelitian oleh (Sepharni et al., 2022) berfokus pada pengembangan sistem prediksi penyakit gagal jantung menggunakan algoritma C4.5 serta mengevaluasi

keefektifannya. Dalam prosesnya, peneliti menggunakan lima fitur yang telah dipilih melalui tahap *feature selection* dan menerapkan teknik *k-fold cross-validation* untuk pembagian data. Hasil penelitian menunjukkan bahwa algoritma C4.5 mampu memberikan performa yang baik dalam memprediksi penyakit gagal jantung.

Penelitian oleh (Baghdadi et al., 2023) mengusulkan algoritma *Catboost* beserta beberapa algoritma lainnya untuk mengembangkan model prediksi dini penyakit kardiovaskular (CVD). Pada prosesnya, peneliti melakukan pembagian data 70% sebagai training data dan 30% sebagai testing data. Selain itu, peneliti juga melakukan optimasi *hyperparameter* untuk 4 algoritma. Hasilnya, algoritma *Catboost* yang telah di tune bisa mendapatkan hasil lebih baik dibandingkan algoritma lainnya.

Penelitian oleh (Tamba & others, 2022) memanfaatkan algoritma Random Forest yang dioptimalkan menggunakan *GridSearchCV* dan *K-Fold Cross Validation* untuk klasifikasi penyakit gagal jantung. Hasil penelitian menunjukkan bahwa penerapan *GridSearchCV* berhasil meningkatkan akurasi model. Temuan ini membuktikan bahwa optimasi hyperparameter dengan *GridSearchCV* efektif dalam meningkatkan kinerja model machine learning.

Penelitian oleh (Darmawan & Dianta, 2023) mengimplementasikan penggunaan algoritma SVM dengan optimasi *GridsearchCV* untuk memprediksi penyakit serangan jantung. Pada penelitian ini, data dibagi dengan 211 sebagai data training, 45 sebagai data uji dan 46 sebagai data validasi. Pengujian dilakukan 2 kali, yaitu sebelum dan sesudah optimasi dengan *GridsearchCV*. Hasilnya adalah

GridsearchCV berhasil menaikkan hasil akurasi dari prediksi penyakit serangan jantung.

Penelitian oleh (Ahmad et al., 2022) mengeksplorasi pengaruh *GridSearchCV* dalam mengoptimasi performa model machine learning. Beberapa algoritma, seperti LR, KNN, SVM, dan GBC, diuji dengan dan tanpa *GridSearchCV*. Hasil menunjukkan bahwa *GridSearchCV* secara signifikan meningkatkan akurasi model, terutama pada algoritma *XGBoost*. Penelitian ini menggarisbawahi pentingnya *tuning hyperparameter* dalam meningkatkan kinerja model.

Penelitian oleh (Dai et al., 2022) menggunakan algoritma *machine learning* untuk memprediksi risiko kematian di rumah sakit pada pasien *sarcoidosis* yang dirawat karena gagal jantung. Dalam penelitian ini, data dari *National Inpatient Sample* digunakan untuk melatih dan menguji model dengan *split data 70:30*. Analisis fitur mengidentifikasi aritmia dan gangguan cairan-elektrolit sebagai prediktor utama risiko kematian. Penelitian ini menyoroti potensi algoritma *machine learning*, khususnya *Random Forest*, dalam memprediksi kematian di rumah sakit pada pasien gagal jantung dengan *sarcoidosis*.

Tabel 2. 1 Penelitian Terkait

Peneliti	Judul	Objek	Metode	Hasil	Perbedaan Dengan Penelitian Ini
(García-Ordás et al., 2023)	<i>Heart Disease Risk Prediction Using Deep Learning Techniques With Feature Augmentation</i>	Prediksi Resiko Penyakit Jantung	<i>Decision Tree, Random Forest, K-Nearest Neighbors (kNN), AdaBoost, XGBoost, Gaussian Naive Bayes (GNB), Multilayer Perceptron (MLP), Convolutional</i>	CNN dengan SAE mendapatkan hasil akurasi mencapai 90.088%, meningkat 4.4% dibandingkan metode klasik.	Peneliti menggunakan <i>Random Forest</i> dengan <i>GridsearchCV</i> untuk mengoptimalkan hasil model klasifikasi

			<i>Neural Networks (CNN), Multitask Neural Network</i>		
(Akther et al., 2024)	<i>Multi-Faceted Approach to Cardiovascular Risk Assessment by Utilizing Predictive Machine Learning and Clinical Data in a Unified Web Platform</i>	Klasifikasi Penyakit Gagal Jantung, dengan 4 dataset berbeda yang diambil dari <i>Kaggle</i>	<i>Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Naive Bayes, KNN, Gradient Boosting, Extreme Gradient Boosting, MLP, ANN, DNN</i>	<i>Xgboost</i> dengan PCA mendapat akurasi dan <i>F1-score</i> sebesar 99%.	Peneliti menggunakan <i>Random Forest</i> dan <i>GridsearchCV</i> dalam penelitian
(Hermawan et al., 2024)	Prediksi Gagal Jantung Berbasis <i>Machine learning</i> Menggunakan <i>Support Vector Machine</i> dan Regresi Logistik	Prediksi Gagal Jantung	<i>Support Vector Machine</i> dan <i>Logistic Regression</i>	SVM mendapat hasil akurasi lebih tinggi sebesar 85% dibanding LR dengan akurasi 82%.	Peneliti menggunakan algoritma <i>Random Forest</i>
(Sepharni et al., 2022, p. 5)	Klasifikasi Penyakit Jantung Dengan Menggunakan Algoritma C4.5	Klasifikasi Penyakit Gagal Jantung	C4,5	Algoritma C4,5 memberikan hasil akurasi 0,7592, presisi mendapatkan skor rata-rata 0,7933, dan <i>recall</i> mendapatkan nilai rata-rata 0,7657.	Peneliti menggunakan algoritma <i>Random Forest</i>
(Baghdadi et al., 2023)	<i>Advanced machine learning techniques for cardiovascular disease early detection and diagnosis</i>	Prediksi Awal dan Deteksi penyakit <i>Cardiovascular</i>	<i>XGBoost, AdaBoost, LinearDiscriminant, LightGBM, GradientBoosting, Catboost, ExtraTree, KNeighbors, SVM, LogisticRegression, RandomForest, Catboost_tuned</i>	Dari semua algoritma yang digunakan, <i>Catboost_tuned</i> memberikan hasil terbaik dengan akurasi 90.94%	Peneliti hanya menggunakan algoritma <i>Random Forest</i>
(Tamba & others, 2022)	Prediksi Penyakit Gagal Jantung Dengan Menggunakan <i>Random Forest</i>	Prediksi Penyakit Gagal Jantung	<i>Random Forest</i>	<i>GridsearchCv</i> dan <i>K-fold</i> sukses menaikkan hasil akurasi dari 82,6087% menjadi	Peneliti menggunakan 6 skenario dalam pengujian model

				85,058%.	
(Darmawan & Dianta, 2023)	Implementasi optimasi <i>hyperparameter GridSearchCV</i> pada sistem prediksi serangan jantung menggunakan SVM	Prediksi Penyakit Gagal Jantung	<i>Support Vector Machine (SVM)</i>	<i>GridsearchCV</i> berhasil meningkatkan hasil akurasi klasifikasi dari 83% menjadi 86%	Peneliti menggunakan algoritma <i>Random Forest</i> dan data yang digunakan berbeda
(Ahmad et al., 2022)	<i>Efficient Medical Diagnosis of Human Heart Diseases Using Machine Learning Techniques With and Without GridSearchCV</i>	Klasifikasi Penyakit Gagal Jantung	<i>logistic regression (LR), K-nearest neighbours (K-NN), support vector machine (SVM), and Gradient Boosting Classifier (GBC) with and without Grid-SearchCV</i>	Algoritma <i>XGBoosting</i> dengan <i>GridsearchCv</i> mendapatkan akurasi tertinggi sebesar 99,03% dibandingkan dengan <i>XGBoostin</i> tanpa <i>GridsearchCV</i> yang mendapatkan akurasi 98,05%	Peneliti menggunakan algoritma <i>Random Forest</i> dengan optimasi <i>GridsearchCV</i>
(Dai et al., 2022)	<i>Machine learning predicting mortality in sarcoidosis patients admitted for acute heart failure</i>	Prediksi Tingkat Kematian Pasien Sarcoidosis Yang Dirawat Karena Gagal Jantung Akut	<i>Logistic Regression, Random Forest, XGBoosting</i>	Dari ketiga algoritma yang digunakan, <i>Random Forest</i> dengan <i>GridsearchCV</i> berhasil mendapatkan hasil terbaik dengan sensitivity 60%, specificity 66,4% dan skor AUC sebesar 0,71	Fokus peneliti lebih kepada penyakit gagal jantung

2.2 Penyakit Gagal Jantung

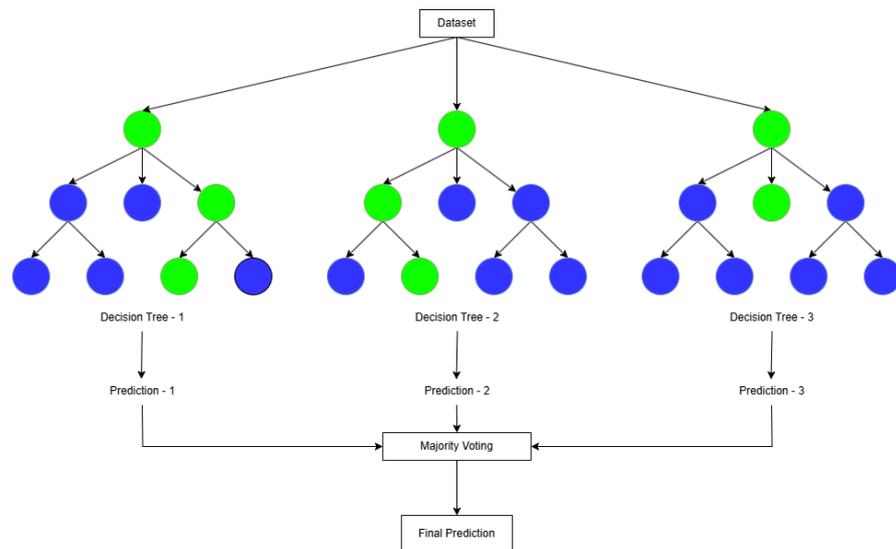
Gagal jantung adalah sindrom klinis yang ditandai oleh munculnya gejala dan tanda fisik akibat ketidakmampuan jantung untuk memenuhi kebutuhan suplai

darah ke seluruh tubuh, yang disebabkan oleh kelainan struktural atau fungsional pada jantung. Kondisi ini dapat mengakibatkan terganggunya fungsi beberapa organ tubuh, atau bahkan menyebabkan fungsi organ berhenti sepenuhnya. Penyakit gagal jantung seringkali merupakan akibat dari berbagai gangguan kardiovaskular, seperti hipertensi, penyakit arteri koroner, atau serangan jantung.

Penyakit ini menjadi salah satu masalah kesehatan yang progresif dengan angka mortalitas dan morbiditas yang tinggi, baik di negara maju maupun berkembang, termasuk Indonesia. Menurut Organisasi Kesehatan Dunia (WHO), penyakit *kardiovaskular*, termasuk gagal jantung, bertanggung jawab atas 17,9 juta kematian setiap tahun, yang mencakup 31% dari total kematian global (WHO, 2021). Di Indonesia, pasien gagal jantung memiliki usia relatif lebih muda dibandingkan dengan pasien di negara-negara Eropa atau Amerika, dengan gejala klinis yang lebih berat. Prevalensi gagal jantung terus meningkat seiring dengan bertambahnya jumlah pasien dengan kerusakan jantung akut yang berkembang menjadi kondisi kronis.

2.3 Random Forest

Random Forest merupakan salah satu algoritma *machine learning* yang populer digunakan. *Random Forest* adalah metode berbasis *ensemble learning* yang meningkatkan hasil akurasi dengan cara menggabungkan banyak *decision tree* yang dilatih secara independen menggunakan teknik *bagging* (Speiser et al., 2019). Algoritma ini pertama kali diperkenalkan oleh *Leo Breiman* pada tahun 2001 sebagai pengembangan dari *decision tree* yang mengatasi kelemahan seperti *overfitting* dan sensitivitas terhadap data *outlier* (Schonlau & Zou, 2020).



Gambar 2. 1 Ilustrasi cara kerja *Random Forest*

Berdasarkan gambar 2.1, *random Forest* bekerja dengan menggunakan teknik yang bernama *bootstrap sampling*, yaitu teknik *resampling* di mana setiap subset diambil secara acak dengan pengembalian (*with replacement*). Ini berarti satu data point dapat muncul lebih dari sekali di satu subset, sementara data lain mungkin tidak terpilih. Proses dilanjutkan dengan pembentukan *decision tree* dengan menggunakan *features randomness*, dimana hanya subset acak dari fitur yang dipertimbangkan untuk menentukan split terbaik dengan tujuan untuk mengurangi korelasi antar *tree* dan meningkatkan keanekaragaman model. Kriteria yang digunakan untuk split terbaik pada *Random Forest* adalah *gini impurity*, yang menggunakan persamaan.

$$G = 1 - \sum_{i=1}^C p(i)^2 \quad (2.1)$$

Keterangan :

G = Nilai gini impurity

C = Jumlah kelas dalam *subset*

p(i) = Proporsi sample dari kelas i dalam *node*

Setelah nilai *Gini Impurity* dihitung untuk setiap kemungkinan split, *Random Forest* akan memilih split yang menghasilkan nilai *Gini Impurity* terkecil, yaitu split yang menghasilkan node paling "murni". Selanjutnya, untuk mengukur kontribusi masing-masing fitur dalam pemisahan ini, digunakan perhitungan *weighted gini*, yang mempertimbangkan bobot masing-masing *node* dalam menentukan seberapa signifikan fitur tersebut dalam mengurangi impurity secara keseluruhan (Disha & Waheed, 2022). Berikut merupakan persamaan *weighted gini*.

$$gini\ weight = \sum_{i=0}^{k-1} \frac{n_i}{n} \cdot gini\ impurity \quad (2.2)$$

Keterangan:

- n_i = jumlah sample pada cabang ke- i
- n = total sample pada *node* induk
- k = jumlah cabang (2)

Dengan menggabungkan teknik bootstrap sampling dan feature randomness, *Random Forest* mengurangi risiko *overfitting*, yang sering terjadi pada *decision tree* individu. Setelah semua *tree* dalam *ensemble* selesai dilatih, prediksi akhir untuk klasifikasi dilakukan dengan voting mayoritas dari semua *tree*,

2.4 *GridSearchCV*

GridSearchCV adalah teknik dalam *machine learning* yang digunakan untuk menemukan kombinasi parameter optimal untuk sebuah model dengan melakukan pencarian melalui berbagai set parameter yang mungkin (Ahmadi et al., 2023). Istilah "*grid*" merujuk pada kumpulan kombinasi parameter yang telah ditetapkan sebelumnya, sementara "*CV*" atau *cross-validation* mengacu pada

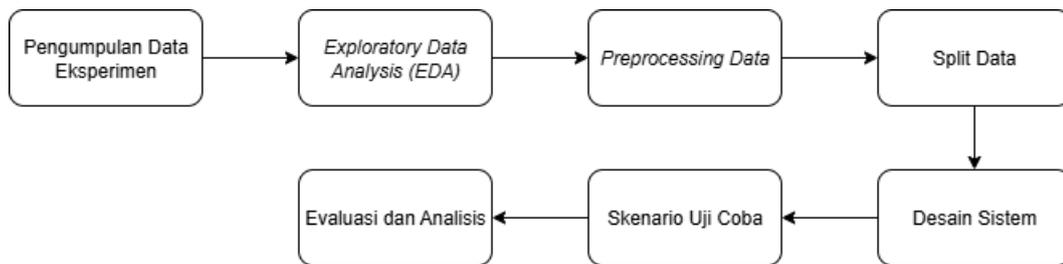
teknik validasi silang yang membagi dataset menjadi beberapa bagian, untuk memastikan model diuji secara menyeluruh pada berbagai subset data. Proses ini memungkinkan evaluasi performa model yang lebih objektif dan mengurangi risiko *overfitting* (Ahmad et al., 2022).

Dalam penelitian ini, *GridSearchCV* digunakan untuk mengoptimalkan performa algoritma *Random Forest*. Teknik ini diterapkan untuk menentukan kombinasi parameter terbaik, seperti jumlah estimasi *tree* (*n_estimators*), kedalaman maksimum *tree* (*max_depth*), dan lainnya. Dengan menggunakan *cross-validation*, yang mana dalam penelitian ini menggunakan *fold = 5*, *GridSearchCV* membantu memastikan bahwa parameter yang terpilih memberikan akurasi terbaik pada data validasi. Penggunaan *GridSearchCV* diharapkan dapat meningkatkan akurasi hasil klasifikasi dan memberikan model yang lebih andal untuk dataset yang digunakan dalam penelitian ini.

BAB III

METODE PENELITIAN

Metode penelitian merupakan bab yang membahas mengenai pendekatan yang digunakan untuk menganalisis data guna menjawab memecahkan masalah yang diteliti. Pada gambar 3.1 menjelaskan mengenai prosedur penelitian yang digunakan dalam penelitian. Proses penelitian dimulai dengan pengumpulan data eksperimen yang menjadi dasar dalam proses analisis dan pemodelan. Setelah data terkumpul, dilakukan tahapan *Exploratory Data Analysis* (EDA) untuk memahami karakteristik, distribusi, dan pola dalam data, serta mendeteksi potensi masalah seperti *missing value*, *outlier*, atau ketidakseimbangan kelas. Selanjutnya, data dipersiapkan melalui proses *preprocessing*, yang mencakup transformasi data kategorikal menjadi numerikal dan penanganan data tidak valid, agar data siap digunakan untuk pelatihan model. Proses dilanjutkan dengan membagi data menjadi *training* dan *testing*, lalu perancangan sistem, yaitu tahap mendesain alur dan struktur sistem yang akan dibangun. Sistem yang telah dirancang kemudian diuji melalui beberapa skenario uji coba untuk mengamati performa model yang dihasilkan. Tahap akhir adalah evaluasi dan analisis terhadap hasil uji coba, yang bertujuan untuk menilai efektivitas model dan mengetahui sejauh mana sistem mampu menjawab tujuan penelitian.



Gambar 3. 1 prosedur penelitian

3.1 Data Eksperimen yang Digunakan

Pada penelitian ini, peneliti menggunakan *dataset* “*Heart Failure Prediction Dataset*” yang diambil dari bank data *Kaggle* (fedesoriano, 2021) (<https://www.kaggle.com/datasets/fedoriano/heart-failure-prediction/data>).

Dataset ini dikumpulkan dari beberapa dataset terdahulu yang terdiri dari Cleveland, Hungarian, Switzerland, Long Beach VA, dan Stalog (Heart) Dataset. Pada dataset sendiri terdiri dari 918 *record* dengan 11 fitur dan 1 target yang berisi 2 kelas, yaitu 0 sebagai negatif penyakit gagal jantung dan 1 sebagai positif penyakit gagal jantung, yang dapat dilihat pada tabel 3.1.

Tabel 3. 1 Deskripsi Dataset

No	Nama Atribut	Deskripsi	Keterangan
1.	<i>Age</i>	Usia pasien	29 - 77 tahun
2.	<i>Sex</i>	Jenis kelamin pasien	M (Laki-laki), F (Perempuan)
3.	<i>ChestPainType</i>	Tipe nyeri dada yang dialami pasien	- TA (Typical Angina) - ATA (Atypical Angina) - NAP (non-Anginal Pain) - ASY (Asymptomatic)
4.	<i>RestingBP</i>	Tekanan darah pasien saat istirahat (sistolik).	0 - 200 mmHg
5.	<i>Cholesterol</i>	Kadar kolesterol dalam darah.	0 - 603 mg/dl
6.	<i>FastingBS</i>	Kadar gula darah saat puasa, dengan kategori normal atau tinggi	0 (≤ 120 mg/dl), 1 (> 120 mg/dl)
7.	<i>RestingECG</i>	Hasil rekaman listrik jantung pasien saat istirahat.	- Normal - ST (abnormalitas gelombang ST-T)

			- LVH (hipertrofi ventrikel kiri)
8.	<i>MaxHR</i>	Detak jantung maksimum yang dicapai saat aktivitas fisik	60 - 202 bpm
9.	<i>ExerciseAngina</i>	Nyeri dada yang terjadi saat berolahraga atau aktivitas fisik	Y (Ya), N (Tidak)
10.	<i>Oldpeak</i>	Nilai perubahan segmen ST pada hasil rekaman jantung.	0 - 6.2
11.	<i>ST_Slope</i>	Bentuk kemiringan segmen ST saat puncak latihan fisik.	- <i>Up</i> (menaik) - <i>Flat</i> (datar) - <i>Down</i> (menurun)
12.	<i>HeartDisease</i>	Status apakah pasien menderita penyakit jantung.	0 (Tidak), 1 (Ya)

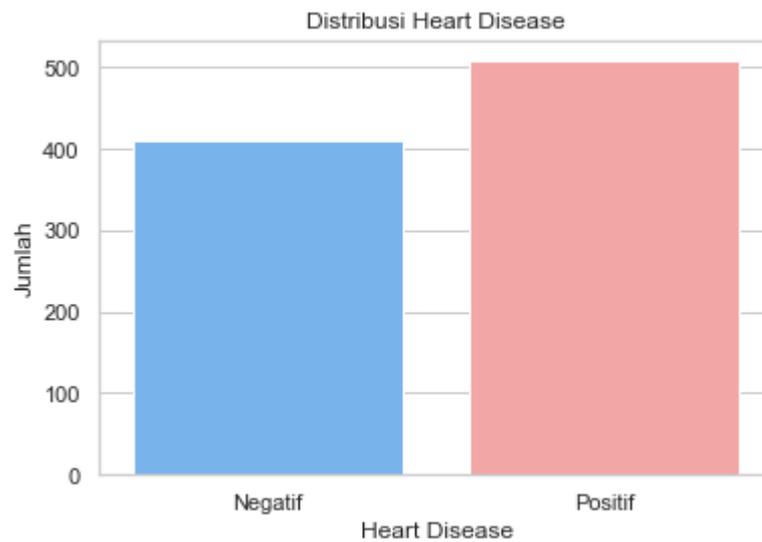
3.2 Exploratory Data Analysis (EDA)

Merupakan tahap penting dalam proses penelitian data sains yang bertujuan untuk memahami struktur, pola, hubungan, serta karakteristik umum dari data yang digunakan. Pada penelitian ini, EDA dilakukan untuk memperoleh wawasan awal terhadap dataset yang digunakan, serta untuk membantu dalam proses pemodelan selanjutnya. Analisis ini juga berguna untuk mendeteksi adanya *missing value*, *outliers*, atau *data imbalance*.

```
Age          0
Sex          0
ChestPainType  0
RestingBP    0
Cholesterol  0
FastingBS    0
RestingECG   0
MaxHR        0
ExerciseAngina  0
Oldpeak      0
ST_Slope     0
HeartDisease  0
dtype: int64
```

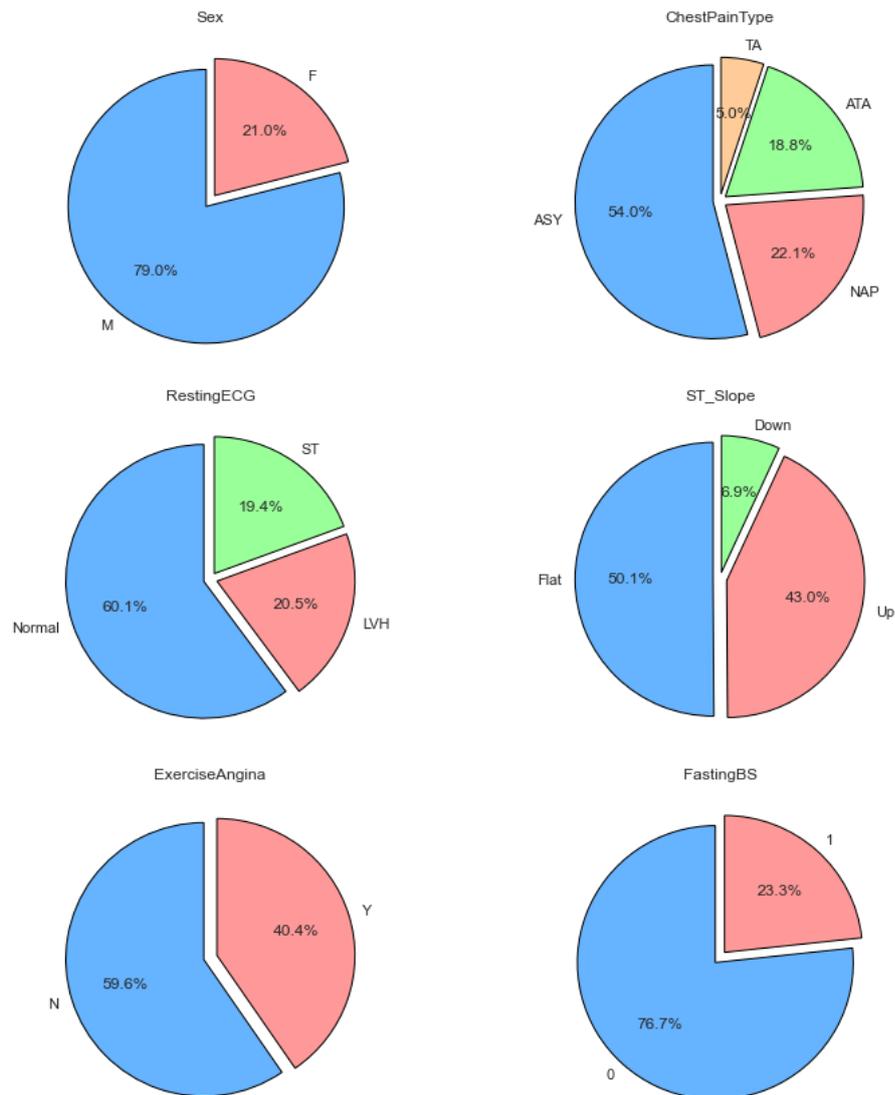
Gambar 3. 2 Pengecekan *Missing Value*

Berdasarkan gambar 3.2, dilakukan pengecekan *missing value* untuk mencari apakah ada data yang tidak lengkap, namun sesuai dengan gambar bahwa tidak ada data yang hilang sehingga data dapat diproses ke tahap selanjutnya.



Gambar 3. 3 Distribusi Target

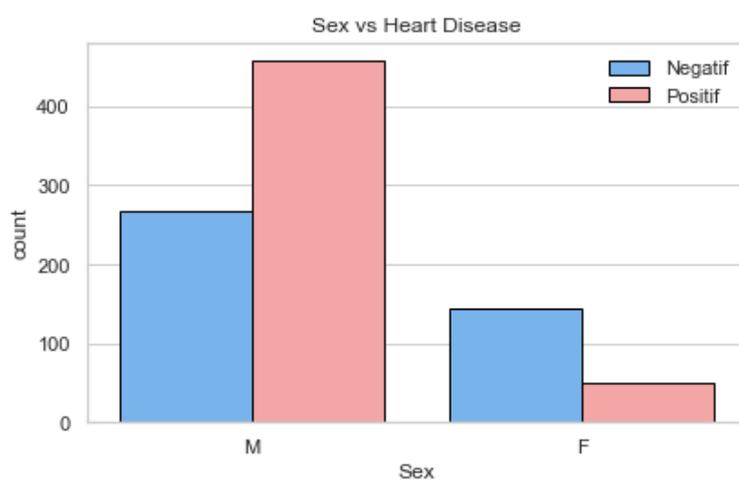
Berdasarkan gambar 3.3, dapat dilihat distribusi target mengenai jumlah pasien yang negatif penyakit gagal jantung adalah 410 pasien, dan yang positif penyakit gagal jantung terdapat 508 pasien. Hal ini menunjukkan bahwa distribusi kelas target tergolong cukup seimbang, meskipun jumlah pasien yang positif sedikit lebih banyak dibandingkan dengan yang negatif.



Gambar 3. 4 Distribusi Data *Categorical*

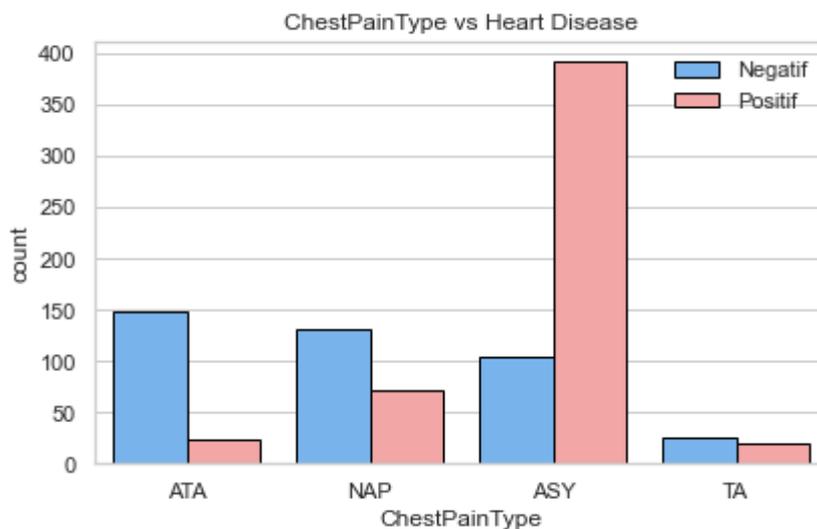
Berdasarkan gambar 3.4, kita bisa mengetahui detail dari data *categorical*. Dimulai dari *Sex*, dengan mayoritas pasien adalah *male* dengan 79% dan *female* 21%. Untuk *ChestPainType* sendiri, Sebagian besar pasien mengalami nyeri dada yang tidak memiliki gejala khas (*asymptomatic*), yang merupakan indikasi umum pada penderita jantung serius dengan 54%, diikuti oleh *NAP* dengan 22.1%, *ATA* dengan 18.8%, dan *TA* dengan 5%. *Resting ECG* sendiri, *normal* menjadi mayoritas

dengan 60.1%, *LVH* dengan 20.5%, dan *ST* dengan 19.4%. Untuk *ST_Slope*, Kemiringan segmen *ST Flat* dan *Up* lebih umum ditemukan dengan 50.1% dan 43%, dan *Flat* sering dikaitkan dengan iskemia jantung dengan 6.9%. Pada *ExerciseAngina* terbagi dua, dengan Sebagian besar pasien tidak mengalami angina saat berolahraga (59.6%), namun proporsi yang mengalami tetap cukup tinggi dan relevan untuk deteksi risiko (40.4%). Pada *FastingBS*, Mayoritas pasien memiliki kadar gula darah normal saat puasa (76.7%), namun ada hampir seperempat yang melebihi batas ambang, berpotensi mengarah ke komplikasi metabolik (23.3%).



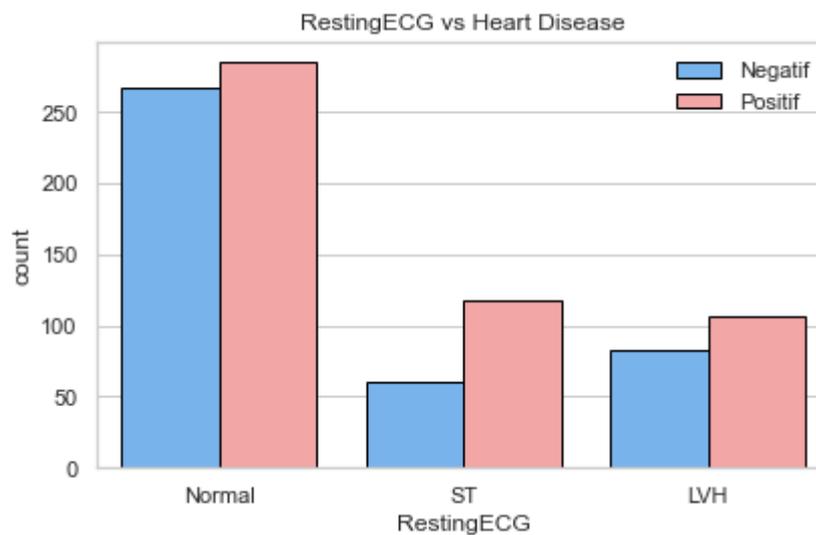
Gambar 3. 5 distribusi Pasien berdasarkan Sex

Pada gambar 3.5, diketahui bahwa untuk data baik negatif ataupun positif penyakit gagal jantung didominasi oleh *male*, dengan jumlah data untuk positif *male* = 458, negatif *male* = 267, positif *female* = 50, dan negatif *female* = 143.



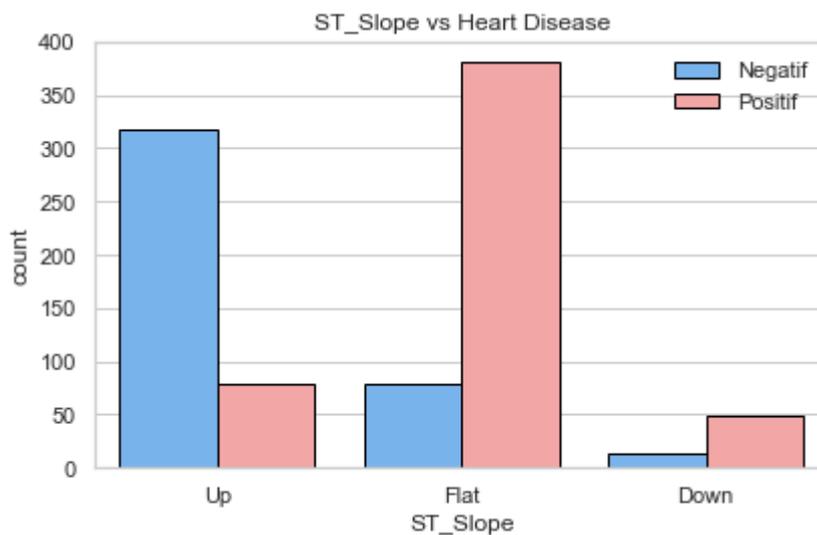
Gambar 3. 6 distribusi Pasien berdasarkan *ChestPainType*

Berdasarkan gambar 3.6, distribusi *ChestPainType* menunjukkan bahwa mayoritas pasien yang terdeteksi positif penyakit gagal jantung berada pada kategori ASY (*Asymptomatic*), yakni sebanyak 392 pasien. Hal ini mengindikasikan bahwa banyak pasien yang menderita penyakit jantung tidak menunjukkan gejala nyeri dada yang khas. Sebaliknya, jenis nyeri dada ATA (*Atypical Angina*) justru didominasi oleh pasien yang tidak mengalami penyakit gagal jantung, yaitu sebanyak 149 pasien dibandingkan hanya 24 pasien yang positif. Sementara pada kategori NAP (*Non-Anginal Pain*) dan TA (*Typical Angina*), meskipun terdapat kasus positif, jumlah pasien yang tidak mengalami penyakit lebih tinggi



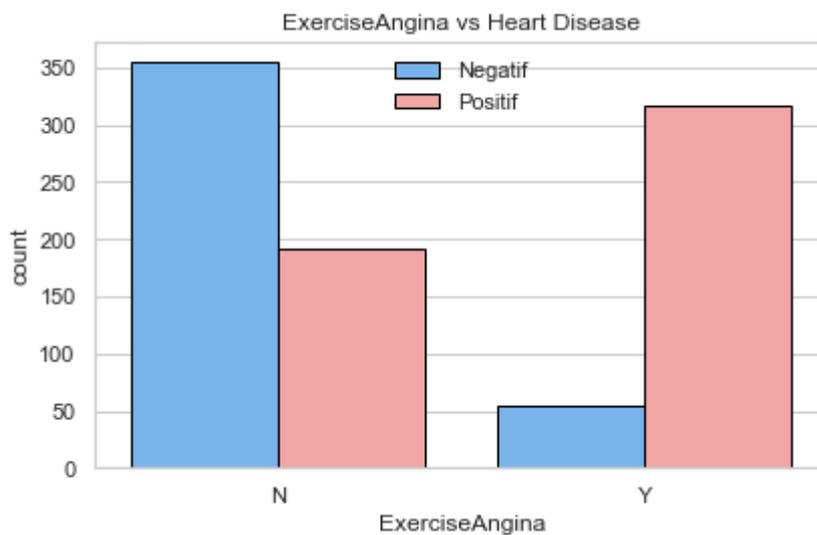
Gambar 3. 7 distribusi Pasien berdasarkan *RestingECG*

Berdasarkan gambar 3.7, distribusi fitur *RestingECG*, dapat dilihat bahwa kategori Normal merupakan jenis hasil elektrokardiogram yang paling umum ditemukan, baik pada pasien yang negatif (267 pasien) maupun positif (285 pasien) penyakit gagal jantung. Namun demikian, terdapat pola pada kategori ST (*abnormalitas segmen ST-T*) dan LVH (*Left Ventricular Hypertrophy*). Pada kategori ST, jumlah pasien yang mengalami penyakit gagal jantung (117 pasien) lebih tinggi dibandingkan yang tidak (61 pasien), menunjukkan potensi keterkaitan abnormalitas ST dengan risiko penyakit jantung. Sementara itu, pada kategori *LVH*, distribusi pasien relatif seimbang dengan sedikit dominasi pada pasien positif. Secara keseluruhan, hasil ini mengindikasikan bahwa kelainan pada hasil EKG, khususnya abnormalitas ST-T, dapat menjadi salah satu sinyal awal terhadap kemungkinan adanya gangguan jantung.



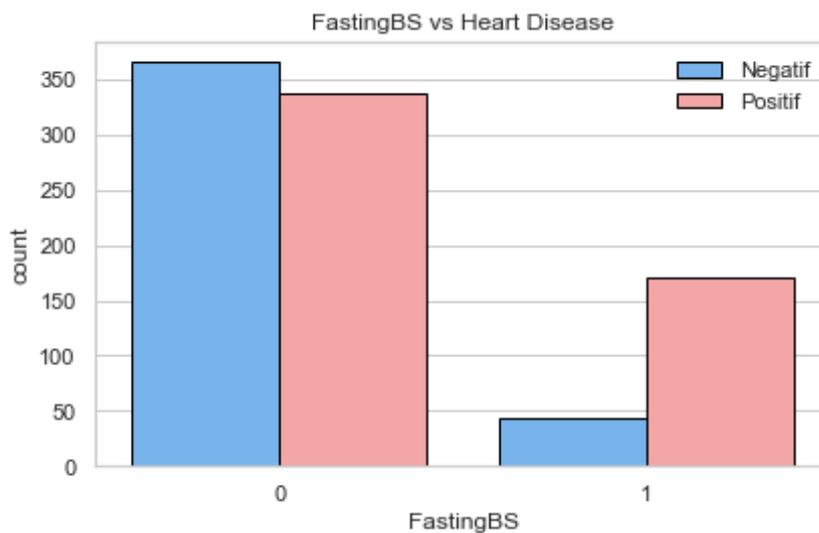
Gambar 3. 8 distribusi Pasien berdasarkan *ST_Slope*

Berdasarkan gambar 3.8, distribusi data pada fitur *ST_Slope* terlihat bahwa mayoritas pasien yang positif penyakit gagal jantung berada pada kategori *Flat*, yaitu sebanyak 381 pasien, dibandingkan hanya 79 pasien yang negatif. Hal ini menunjukkan bahwa kemiringan segmen ST yang datar (*Flat*) sangat dominan di antara pasien dengan kondisi jantung bermasalah. Sebaliknya, kategori *Up* didominasi oleh pasien yang tidak mengalami penyakit gagal jantung, yakni sebanyak 317 pasien, dan hanya 78 pasien yang positif. Pada kategori *Down*, meskipun jumlahnya tidak sebanyak kategori lainnya, tetap terlihat kecenderungan lebih tinggi pada pasien yang positif (49) dibandingkan negatif (14). Temuan ini menunjukkan bahwa bentuk kemiringan segmen ST, khususnya tipe *Flat*, berpotensi menjadi indikator penting dalam mengidentifikasi risiko penyakit jantung pada pasien.



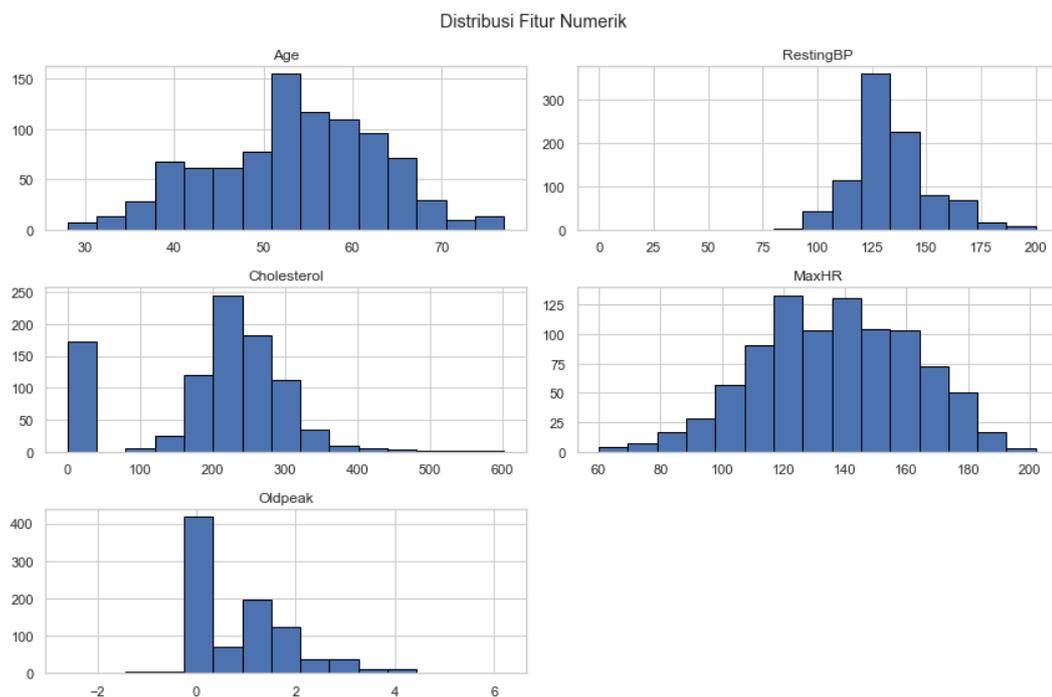
Gambar 3. 9 distribusi Pasien berdasarkan *ExerciseAngina*

Berdasarkan gambar 3.9, terlihat adanya perbedaan yang cukup mencolok antara kedua kelas. Pasien yang tidak mengalami angina saat berolahraga (N) sebagian besar berada dalam kelompok negatif penyakit gagal jantung, yaitu sebanyak 355 pasien, dibandingkan dengan 192 pasien yang positif. Sebaliknya, pada pasien yang mengalami angina saat berolahraga (Y), jumlah yang positif penyakit gagal jantung jauh lebih tinggi, yaitu 316 pasien, dibandingkan hanya 55 pasien yang negatif. Pola ini mengindikasikan bahwa munculnya angina saat aktivitas fisik merupakan faktor klinis yang kuat terkait dengan risiko penyakit jantung, dan dapat dijadikan sebagai salah satu indikator penting dalam proses deteksi dini.



Gambar 3. 10 distribusi Pasien berdasarkan *FastingBS*

Berdasarkan gambar 3.10, diketahui bahwa mayoritas pasien memiliki nilai *FastingBS* = 0, yaitu sebanyak 366 pasien negatif dan 338 pasien positif penyakit gagal jantung. Meskipun jumlahnya lebih banyak pada kedua kelas, perlu diperhatikan bahwa pada pasien dengan *FastingBS* = 1 (kadar gula darah puasa melebihi 120 mg/dL), terdapat 170 pasien yang positif penyakit gagal jantung dibandingkan hanya 44 pasien yang negatif. Hal ini menunjukkan bahwa kadar gula darah puasa yang tinggi cenderung berkorelasi dengan peningkatan risiko penyakit gagal jantung. Dengan demikian, fitur ini dapat menjadi salah satu penanda klinis yang relevan dalam analisis risiko, terutama pada pasien dengan riwayat atau gejala terkait metabolik.



Gambar 3. 11 Distribusi Fitur *Numerical*

Gambar 3.11 memberikan gambaran mengenai karakteristik pasien pada dataset. Pada fitur *Age*, mayoritas pasien berada dalam rentang usia 50 hingga 60 tahun, menunjukkan bahwa penyakit gagal jantung cenderung lebih banyak dialami oleh individu paruh baya hingga lanjut usia. Fitur *RestingBP* (tekanan darah saat istirahat) sebagian besar berkisar antara 120 hingga 140 mmHg, yang masih tergolong normal hingga pra-hipertensi, meskipun terdapat beberapa nilai ekstrem. Sementara itu, fitur *Cholesterol* menunjukkan puncak distribusi di angka 200 hingga 250 mg/dL, namun keberadaan nilai nol yang cukup signifikan mengindikasikan kemungkinan adanya data hilang yang perlu ditangani lebih lanjut.

Fitur *MaxHR* (denyut jantung maksimum saat aktivitas) memiliki distribusi yang cukup simetris dengan puncak pada 120 hingga 160 bpm, menandakan kisaran

umum kapasitas jantung pasien. Sedangkan pada fitur *Oldpeak*, yang merepresentasikan depresi segmen ST pada hasil EKG, distribusi terlihat miring ke kanan dengan sebagian besar pasien memiliki nilai mendekati nol, namun terdapat nilai-nilai ekstrem hingga di atas 4. Pola distribusi ini penting untuk dipertimbangkan dalam tahap praproses data, khususnya untuk penanganan outlier dan normalisasi, guna memastikan performa model yang optimal pada proses klasifikasi selanjutnya.

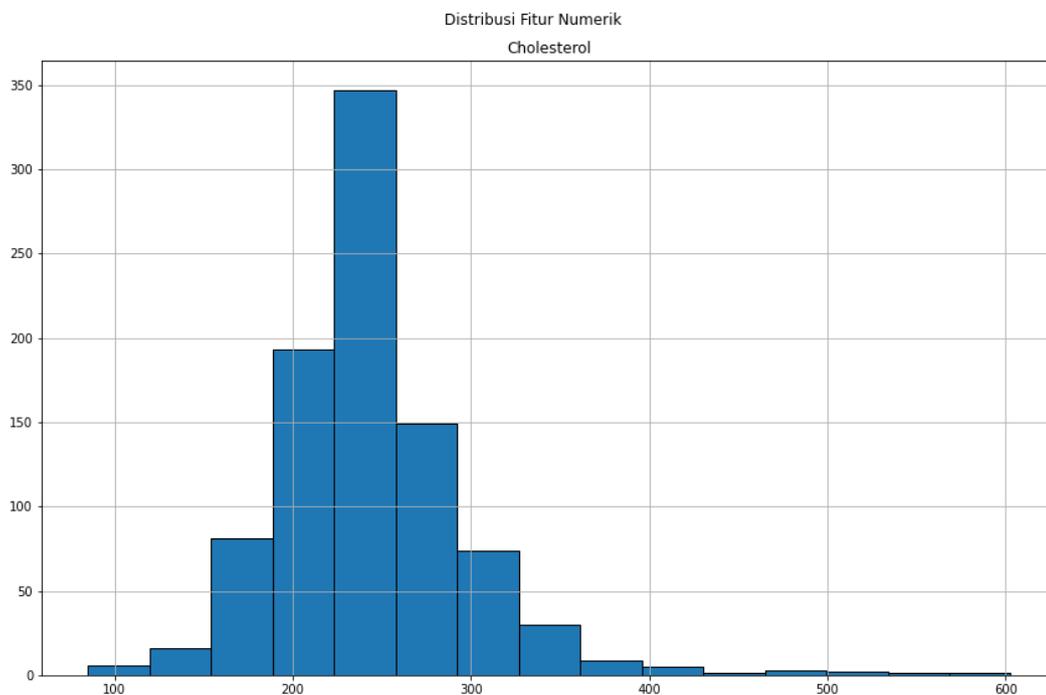
3.3 Preprocessing

Preprocessing adalah langkah awal yang penting dalam pengolahan data sebelum model machine learning diterapkan. Pada tahap ini, data mentah yang diperoleh dari dataset akan dipersiapkan melalui serangkaian prosedur untuk meningkatkan kualitas dan relevansinya bagi analisis. Pada penelitian ini, peneliti menerapkan *encoding* untuk merubah data *categorical* menjadi *numerical*. Metode *encoding* seperti ini sudah digunakan untuk *random forest* pada data medis oleh beberapa peneliti sebelumnya, seperti (Alwakid et al., 2025) yang menggunakannya pada data *cardiovascular disease* (CVD), dan penelitian oleh (Mansoori et al., 2023) yang menggunakan *encoding* sebagai *preprocessing* untuk memprediksi seberapa lama seorang pasien rawat inap di rumah sakit.

3.3.1 Penanganan *Outlier*

Berdasarkan penjelasan pada EDA sebelumnya, terdapat *outlier* pada fitur *cholesterol*, yang harus ditangani. Penanganan dilakukan dengan mengganti nilai 0

pada fitur menjadi *mean* dari fitur itu sendiri. Hasil dari penanganan ini dapat dilihat pada gambar 3.12.



Gambar 3. 12 Penanganan *Outlier* pada *Cholesterol*

3.3.2 *One-Hot Encoding*

One-hot encoding merupakan salah satu teknik *preprocessing* untuk merubah data *categorical* menjadi *numerical*. Teknik ini dilakukan dengan mengonversi setiap nilai kategori menjadi vektor biner, di mana hanya satu elemen yang bernilai 1 (aktif) dan sisanya 0. Peneliti menggunakan *function* *pd.get_dummies()* yang berfungsi sama dengan *one-hot encoding* dengan merubah datanya menjadi *true* dan *false*. Pada penelitian, fitur yang menggunakan *one-hot encoding* adalah *Sex*, *ChestPainType*, *RestingECG*, dan *ST_Slope*. Fitur-fitur tersebut dipilih karena bersifat nominal, atau tidak memiliki urutan. Hasil data setelah menggunakan *one-hot encoding* dapat dilihat pada tabel 3.2.

Tabel 3. 2 Sampel Data setelah *One-Hot Encoding*

<i>Sex_F</i>	<i>Sex_M</i>	<i>ChestPainType_ASY</i>	...	<i>ST_Slope_Flat</i>	<i>ST_Slope_Up</i>
<i>False</i>	<i>True</i>	<i>False</i>	...	<i>False</i>	<i>True</i>
<i>True</i>	<i>False</i>	<i>False</i>	...	<i>True</i>	<i>False</i>
<i>False</i>	<i>True</i>	<i>False</i>	...	<i>False</i>	<i>True</i>
<i>True</i>	<i>False</i>	<i>True</i>	...	<i>True</i>	<i>False</i>
<i>False</i>	<i>True</i>	<i>False</i>	...	<i>False</i>	<i>True</i>

3.3.3 Label Encoding

Selain *one-hot encoding*, *label encoding* juga salah satu teknik untuk merubah data *categorical* menjadi *numerical*. Teknik ini bekerja dengan mengonversi setiap kategori unik dalam sebuah kolom menjadi integer. Dalam penelitian, fitur yang menggunakan *label encoding* adalah *ExerciseAngina*. Fitur ini dipilih karena bersifat ordinal, atau memiliki urutan, yang mana pada fitur ini berisi data N dan Y, yang bersifat sama seperti 0 dan 1. Hasil perubahan data pada fitur dapat dilihat pada gambar 3.13.

ExerciseAngina
0
0
0
1
0
0
0
0
0
1
0

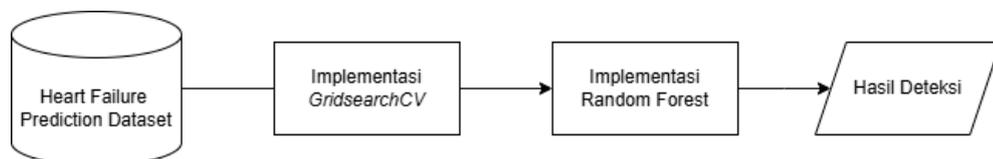
Gambar 3. 13 Fitur yang telah dirubah dengan *Label Encoding*

3.4 Split Data

Split data adalah proses pembagian dataset menjadi beberapa bagian, umumnya *training set* dan *testing set*. Tujuan dari pemisahan ini adalah untuk melatih model menggunakan sebagian data (*training set*) dan menguji performa model pada data yang belum pernah dilihat (*testing set*). Pada langkah ini, peneliti akan membagi data yang akan digunakan menjadi beberapa skenario, yaitu skenario A dengan rasio perbandingan 60:40, skenario B dengan rasio perbandingan 70:30, dan skenario C dengan perbandingan 80:20 (Sephya et al., 2023).

3.5 Desain Sistem

Tahap desain sistem merupakan tahap yang digunakan peneliti untuk menggambarkan secara umum bagaimana sistem melakukan deteksi penyakit gagal jantung. Desain sistem untuk penelitian ini ditunjukkan oleh gambar 3.3.

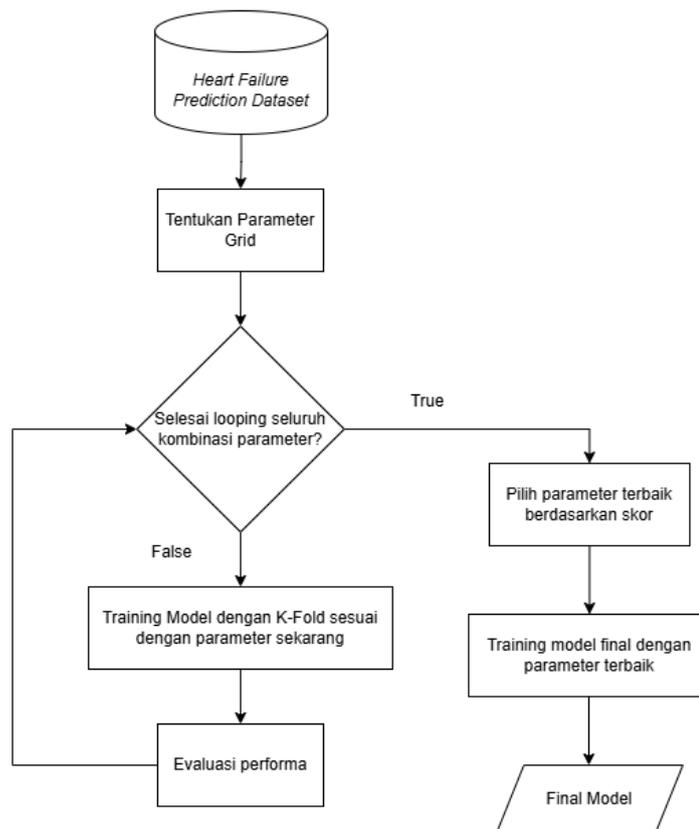


Gambar 3. 14 Desain Sistem

Berdasarkan gambar 3.14, sistem penelitian dimulai dengan melakukan penginputan dataset yang akan digunakan, yang mana pada penelitian ini adalah “*Heart Failure Prediction Dataset*”. Proses selanjutnya adalah implementasi dari *GridsearchCV* dan *Random Forest*, dimana *GridsearchCV* akan mencari *hyperparameter* optimal untuk model *Random Forest* sesuai dengan grid yang sudah ditetapkan sebelumnya. Hasil dari *hyperparameter* optimal dari *GridsearchCV* kemudian akan digunakan untuk melakukan training pada final

model. Hasil dari model kemudian digunakan untuk melakukan deteksi apakah seseorang terkena penyakit gagal jantung atau tidak.

3.6 Implementasi *GridsearchCV*



Gambar 3. 15 Flowchart Algoritma *GridsearchCV*

Berdasarkan Gambar 3.15 dan Subbab 2.4, *GridSearchCV* merupakan teknik optimasi *hyperparameter* yang digunakan untuk meningkatkan performa model, dalam hal ini algoritma *Random Forest*. *GridSearchCV* bekerja dengan mencari kombinasi parameter terbaik melalui eksplorasi semua kemungkinan nilai dari parameter grid yang telah ditentukan. Setiap kombinasi parameter diuji dengan proses *k-fold cross-validation*, yaitu data pelatihan dibagi menjadi *k* bagian (*fold*),

dan model dilatih serta *divalidasi* sebanyak k kali, dengan bagian data yang berbeda sebagai data validasi di setiap iterasi. Dalam penelitian ini digunakan nilai $k = 5$.

Untuk setiap kombinasi parameter, didapatkan 5 skor evaluasi dari hasil validasi pada tiap *fold*. Skor ini kemudian dihitung rata-ratanya menggunakan fungsi *np.mean(scores)*, dan hasil tersebut disimpan sebagai *avg_score*. Nilai rata-rata skor ini mencerminkan performa umum dari kombinasi parameter tersebut terhadap berbagai pembagian data. Setelah seluruh kombinasi diuji, *GridSearchCV* akan:

1. Membandingkan seluruh *avg_score*
2. Memilih kombinasi parameter dengan nilai *avg_score* tertinggi sebagai parameter terbaik.

Setelah parameter terbaik diperoleh, model akan dilatih ulang menggunakan seluruh data pelatihan dengan kombinasi parameter tersebut. Model hasil pelatihan inilah yang disebut *final model*, dan digunakan untuk pengujian akhir atau prediksi. Tahapan algoritma *GridSearchCV* yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Menentukan ruang pencarian *hyperparameter*
2. Melakukan *K-Fold Cross Validation* untuk setiap kombinasi parameter.
3. Menghitung rata-rata skor evaluasi (*avg_score*)
4. Memilih kombinasi parameter terbaik
5. Melatih ulang model menggunakan parameter terbaik

Tabel 3.3 berikut menunjukkan kombinasi *hyperparameter* yang diuji dalam proses *GridSearchCV*.

Tabel 3. 3 *Tuning hyperparameter*

Parameter	Keterangan
$n_estimators = [50, 100]$	Jumlah <i>decision trees</i> dalam <i>forest</i> .
$max_depth = [None, 10, 20]$	Kedalaman maksimum untuk setiap <i>decision tree</i> .
$max_features = [3, 4, 7, 9]$	Jumlah maksimum fitur yang akan dipertimbangkan untuk mencari <i>split</i> terbaik di setiap <i>node</i> .
$min_samples_split = [2, 5, 10]$	Jumlah sampel minimum yang diperlukan untuk <i>split</i> sebuah <i>node</i>

3.7 Implementasi Algoritma *Random forest*

Berdasarkan subbab 2.3, *Random Forest* merupakan metode *ensemble learning* yang bekerja dengan membangun sejumlah *decision tree* secara acak, yang mana tiap *decision tree* berisi *subset data* dan fitur yang berbeda (*bootstrap sampling* dan *random subspace*). Kemudian menggabungkan hasil prediksi dari setiap *tree* untuk menghasilkan klasifikasi akhir yang lebih akurat dan stabil (*majority voting*). Algoritma *random forest* sebagai berikut.

1. Menentukan *Hyperparameter Random Forest*

Dalam membangun model *Random Forest*, penentuan *hyperparameter* merupakan tahap penting yang sangat memengaruhi performa akhir model. *Hyperparameter* adalah parameter yang nilainya ditentukan sebelum proses pelatihan model. Oleh karena itu, pemilihan kombinasi *hyperparameter* yang tepat bertujuan untuk mengoptimalkan kemampuan model dalam melakukan klasifikasi. Dalam penelitian ini, ada 4 *hyperparameter* yang akan digunakan, yaitu $n_estimator$, max_depth , $max_features$, dan min_sample .

2. *Bootstrap Sampling*

Bootstrap sampling adalah pengambilan sampel secara acak dengan pengembalian (*with replacement*) dari dataset asli, digunakan untuk membangun data pelatihan bagi setiap *tree* di dalam *Random Forest*.

3. Membuat *decision tree*

Tahap ini dimulai dengan menentukan fitur terbaik yang akan digunakan sebagai root node pada *decision tree*. Penentuan fitur dilakukan melalui proses pemilihan subset fitur secara acak menggunakan metode *random subspace*. Dalam metode ini, hanya sebagian fitur (*subset*) dari seluruh fitur yang tersedia yang dipilih secara acak untuk dipertimbangkan pada setiap node, termasuk root node.

Jika dataset memiliki total F fitur, maka pada setiap *node* hanya dipilih m fitur secara acak, di mana nilai m ditentukan berdasarkan parameter *max_features*. Nilai ini dapat disesuaikan secara manual atau mengikuti nilai default. Untuk kasus klasifikasi, nilai default yang umum digunakan adalah:

$$m = \sqrt{F} \quad (3.1)$$

Keterangan :

m = jumlah fitur yang digunakan dalam bootstrap sampling
 F = jumlah total fitur

Setelah subset fitur ditentukan, langkah selanjutnya adalah memilih fitur dan nilai *threshold* terbaik yang menghasilkan *Gini impurity* terkecil. Data dengan format *kategorikal* disini dapat diproses semana mestinya. Namun untuk data *numerical* sendiri, menurut (Probst et al., 2019), belum ada metode penentuan split yang terbukti unggul secara umum, karena performa sangat bergantung pada

karakteristik dan distribusi data. Oleh karena itu, dalam penelitian ini digunakan metode populer, yaitu mengambil nilai tengah (*midpoint*) antara dua nilai unik yang telah diurutkan sebagai kandidat *threshold*, dengan rumus sebagai berikut.

$$threshold = \frac{(v_i + v_{i+1})}{2} \quad (3.2)$$

Keterangan :

v = data unik (tidak ada duplikasi)
i = indeks

Untuk menilai kualitas split, digunakan ukuran *Gini impurity* yang dihitung dengan rumus.

$$Gini = 1 - \sum_{i=1}^c p(i)^2 \quad (3.3)$$

Keterangan :

C = Jumlah kelas dalam *subset*
p(i) = Proporsi sample dari kelas i dalam *node*

Selanjutnya, *Gini impurity* dari node kiri dan kanan digunakan untuk menghitung *Gini weight*, yaitu skor gabungan dari kedua *node* hasil split. Rumus *Gini weight* adalah.

$$gini\ weight = \sum_{i=0}^{k-1} \frac{n_i}{n} \cdot gini\ impurity \quad (3.4)$$

Keterangan:

n_i = jumlah sampel pada cabang ke-i
n = total sampel pada *node* induk
k = jumlah cabang (2)

Split terbaik dipilih berdasarkan nilai *Gini weight* terkecil. Proses pembangunan *decision tree* dilakukan secara rekursif hingga salah satu dari kondisi penghentian (*stopping criteria*) terpenuhi, yaitu:

- a. Kedalaman *tree* mencapai batas maksimum (*max_depth*)
- b. Jumlah data dalam *node* kurang dari *min_sample*
- c. Seluruh data dalam *node* homogen (*impurity = 0*)

4. Prediksi Menggunakan 1 *tree*

Setelah semua *tree* dalam *Random Forest* selesai dibangun, setiap *tree* akan digunakan untuk melakukan prediksi terhadap data uji. Proses prediksi ini dilakukan secara independen oleh masing-masing *tree*. Setiap *tree* menghasilkan satu output kelas.

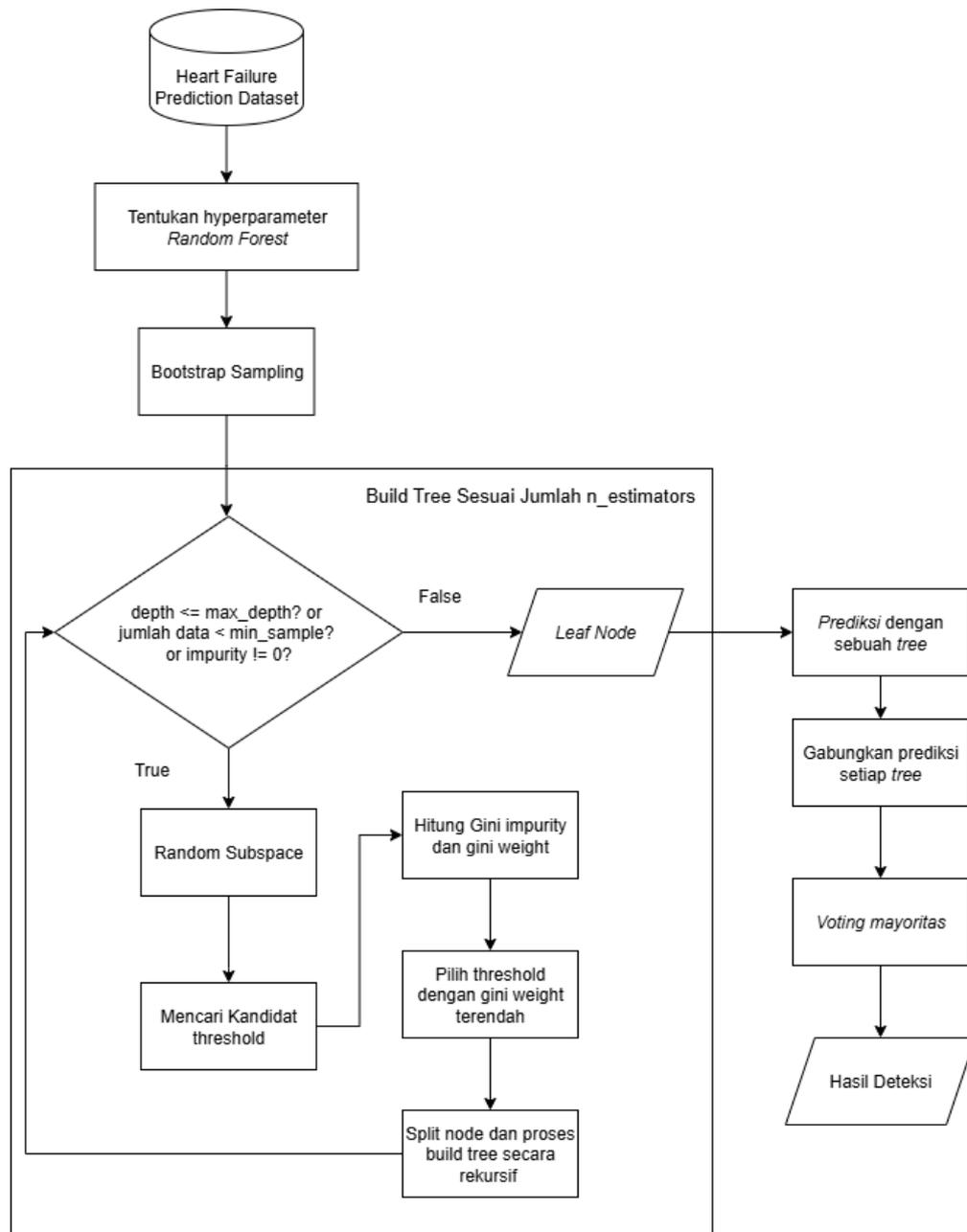
5. Penggabungan Prediksi dari Seluruh *tree*

Seluruh hasil prediksi dari setiap *tree* kemudian dikumpulkan menjadi satu kumpulan hasil. Tujuannya adalah untuk melihat seberapa banyak *tree* yang memprediksi masing-masing kelas terhadap satu data uji.

6. *Majority Voting*

Setelah semua prediksi diperoleh, *Random Forest* akan melakukan voting mayoritas. Kelas yang paling sering muncul dari hasil prediksi seluruh *tree* akan dipilih sebagai prediksi akhir. Dengan cara ini, *Random Forest* dapat menghasilkan prediksi yang lebih stabil dan tahan terhadap kesalahan prediksi dari *tree* individual.

Berdasarkan penjelasan tersebut, berikut merupakan flowchart dari algoritma *random forest* pada gambar 3.16 berikut.

Gambar 3. 16 Flowchart *Random Forest*

3.8 Skenario Uji Coba

Skenario uji coba berfungsi sebagai pengujian sistem penelitian yang telah dibuat dengan menghitung nilai akurasi, presisi, *recall* dan *F1-score* menggunakan

confusion matrix. Pada penelitian ini, pengujian akan dilakukan dengan 3 model split data yang sebagaimana terdapat pada tabel 3.4.

Tabel 3. 4 Split Data

Skenario	Data Training	Data Testing
A	60	40
B	70	30
C	80	20

Dimana pada setiap model akan diuji coba sebelum dan sesudah menggunakan *GridsearchCV*, dan *encoding* sehingga total uji coba adalah 12 skenario, dengan rincian masing-masing pengujian sebagai berikut.

1. Uji coba skenario A1 - NE
2. Uji coba skenario A1 - E
3. Uji coba skenario A2 - NE
4. Uji coba skenario A2 - E
5. Uji coba skenario B1 - NE
6. Uji coba skenario B1 - E
7. Uji coba skenario B2 - NE
8. Uji coba skenario B2 - E
9. Uji coba skenario C1 - NE
10. Uji coba skenario C1 - E
11. Uji coba skenario C2 - Ne
12. Uji coba skenario C2 – E

Keterangan :

- 1 = Tidak menggunakan *GridsearchCV*
- 2 = Menggunakan *GridsearchCV*
- NE = *No-Encoding*
- E = *Encoding*

3.9 Evaluasi

Confusion Matrix adalah sebuah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dalam *machine learning*. *Confusion matrix* pada tabel 3.5 memberikan gambaran mengenai hasil prediksi model dengan membandingkan nilai sebenarnya (*true labels*) dengan nilai yang diprediksi oleh model. Dalam klasifikasi biner, *confusion matrix* biasanya berbentuk tabel 2x2 yang terdiri dari empat elemen utama (Luque et al., 2019).

Tabel 3. 5 *Confusion Matrix*

<i>Actual Class</i>	<i>Predicted Class</i>	
	Positif Gagal Jantung (P)	Negatif Gagal Jantung (N)
Positif Gagal Jantung(P)	True Positive (TP)	False Negative (FN)
Negatif Gagal Jantung (N)	False Positive (FP)	True Negative (TN)

- a. *True Positive (TP)*: Jumlah data yang diklasifikasikan dengan benar sebagai positif penyakit gagal jantung.
- b. *True Negative (TN)*: Jumlah data yang diklasifikasikan dengan benar sebagai negatif penyakit gagal jantung.
- c. *False Positive (FP)*: Jumlah data yang secara salah diprediksi sebagai positif penyakit gagal jantung, namun aktualnya negatif.
- d. *False Negative (FN)*: Jumlah data yang secara salah dipredikti sebagai negatif penyakit gagal jantung, namun aktualnya positif.

Dengan menggunakan *confusion matrix*, beberapa metrik evaluasi dapat dihitung untuk menilai performa model, seperti akurasi, presisi, *recall*, dan *F1-score*. Metrik-metrik ini membantu dalam memahami seberapa baik model menangani klasifikasi data yang benar dan kesalahan prediksi yang dibuat.

Akurasi (*Accuracy*) adalah ukuran seberapa sering model membuat prediksi yang benar, dihitung sebagai perbandingan antara jumlah prediksi benar (*True Positive* dan *True Negative*) terhadap total jumlah prediksi.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.5)$$

Keterangan:

TP = *True Positive* (jumlah sampel positif yang terklasifikasi dengan benar)
 FN = *False Negative* (jumlah sampel positif yang terklasifikasi salah)
 FP = *False Positive* (jumlah sampel yang terklasifikasi salah sebagai positif)
 TN = *True Negative* (jumlah sampel yang terklasifikasi benar sebagai negatif)

Presisi (*Precision*) adalah metrik yang digunakan untuk menilai ketepatan model dalam memprediksi kelas positif, atau berapa proporsi dari prediksi positif yang benar-benar termasuk dalam kategori positif. Presisi dihitung dengan membandingkan jumlah prediksi positif yang benar dengan total prediksi positif yang dibuat oleh model. Metrik ini membantu mengevaluasi seberapa akurat model dalam mengidentifikasi data yang sebenarnya positif di antara semua data yang diprediksi sebagai positif.

$$Presisi = \frac{TP}{TP + FP} \quad (3.6)$$

Keterangan:

TP = *True Positive* (jumlah sampel positif yang terklasifikasi dengan benar)
 FP = *False Positive* (jumlah sampel yang terklasifikasi salah sebagai positif)

Recall, atau dikenal juga sebagai *Sensitivity* atau *True Positive Rate*, mengukur seberapa baik model menangkap semua kasus positif, atau berapa banyak dari total data positif yang diklasifikasikan dengan benar.

$$Recall = \frac{TP}{TP + FN} \quad (3.7)$$

Keterangan:

TP = *True Positive* (jumlah sampel positif yang terklasifikasi dengan benar)

FN = *False Negative* (jumlah sampel positif yang terklasifikasi salah)

F1-score adalah rata-rata harmonis dari *precision* dan *recall*. *F1-score* memberikan keseimbangan antara *precision* dan *recall*, terutama jika ada ketidakseimbangan kelas dalam dataset.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3.8)$$

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian

Berdasarkan pada subbab 3.7, uji coba dilakukan dengan menggunakan 12 skenario, dengan yang tidak menggunakan *encoding* dan yang menggunakan, lalu yang tidak menggunakan *GridsearchCV* dan yang menggunakannya. Hasil dari uji coba akan dievaluasi menggunakan *confusion matrix* untuk mencari tahu model prediksi terbaik.

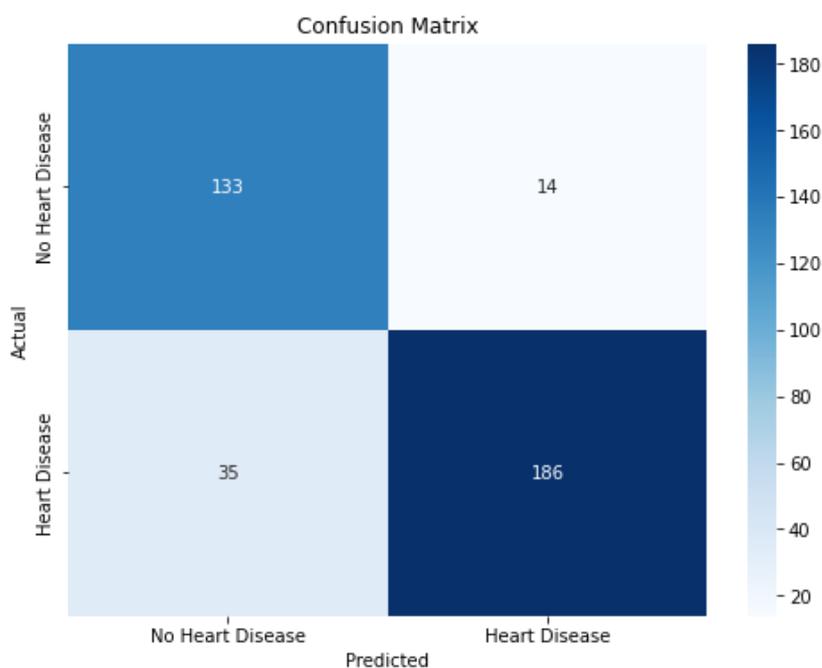
4.1.1 Pengujian tanpa *GridsearchCV*

Dalam pengujian ini, tidak digunakan optimasi *hyperparameter* menggunakan *GridsearchCV* sehingga *hyperparameter* yang digunakan dalam training merupakan default, yaitu $n_estimator = 100$, $max_depth = None$, $max_features = 5$, $min_sample = 2$.

4.1.1.1 Model A1 – NE

Pada pengujian model A1 - NE, dilakukan dengan pembagian data *training* dan data *testing* sebesar 60% : 40%, yang mana sebesar 550 dan 368 record. *confusion matrix* dari pengujian ini dapat dilihat pada gambar 4.1 dan tabel 4.1. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 186 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 35 record, untuk FP (*False Positive*) yaitu

prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 14 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 133 record.



Gambar 4. 1 *confusion matrix* model A1 - NE tanpa *GridsearchCV*

Tabel 4. 1 Hasil *confusion matrix* Model A1 - NE tanpa *GridsearchCV*

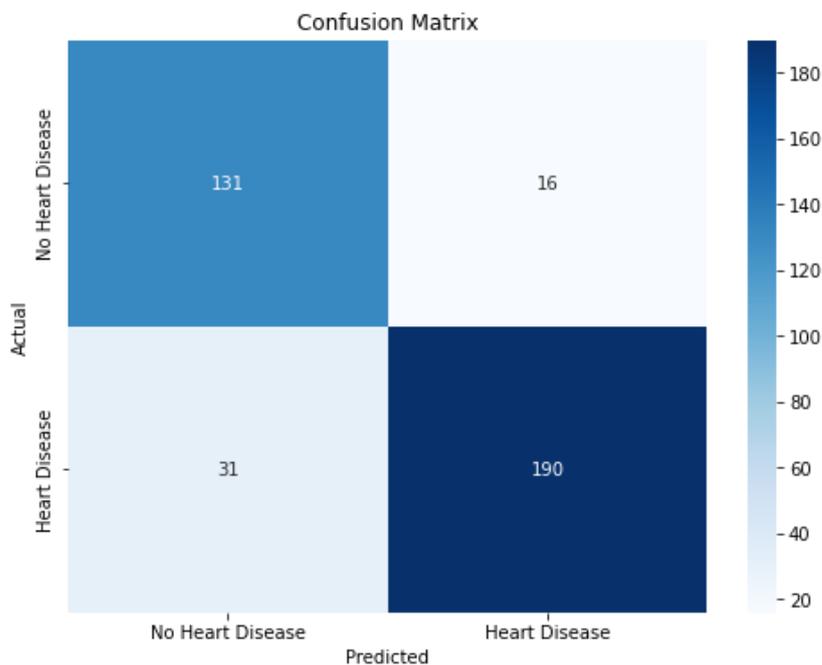
Akurasi	Presisi	Recall	F1-score
87%	86%	87%	86%

Dari hasil *confusion matrix* diperoleh perhitungan akurasi, presisi, *recall* dan *F1-score* yang dapat dilihat pada tabel 4.1, diperoleh kinerja model A1 – NE, menunjukkan performa yang cukup baik dalam melakukan klasifikasi terhadap data penyakit gagal jantung. Model ini memperoleh akurasi sebesar 87%, yang mencerminkan bahwa secara keseluruhan model mampu mengklasifikasikan data dengan tingkat ketepatan yang tinggi.

Dari sisi presisi, model mencatatkan nilai 86%, menunjukkan bahwa sebagian besar prediksi positif (terkena penyakit gagal jantung) adalah benar. Pada *recall* sendiri, model mendapatkan nilai 87% menunjukkan bahwa model cukup sensitif dalam mengenali pasien yang benar-benar memiliki penyakit gagal jantung. *F1-score* pada model ini mendapatkan 86%, mencerminkan model dapat menyeimbangkan *trade-off* antara presisi dan *recall*.

4.1.1.2 Model A1 - E

Pada pengujian model A1 – E, dilakukan dengan pembagian data *training* dan data *testing* sebesar 60% : 40%, yang mana sebesar 550 dan 368 record. *confusion matrix* dari pengujian ini dapat dilihat pada gambar 4.2 dan tabel 4.2. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 190 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 31 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 16 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 131 record.



Gambar 4. 2 *confusion matrix* model A1 - E tanpa *GridsearchCV*

Tabel 4. 2 Hasil *confusion matrix* Model A1 - E tanpa *GridsearchCV*

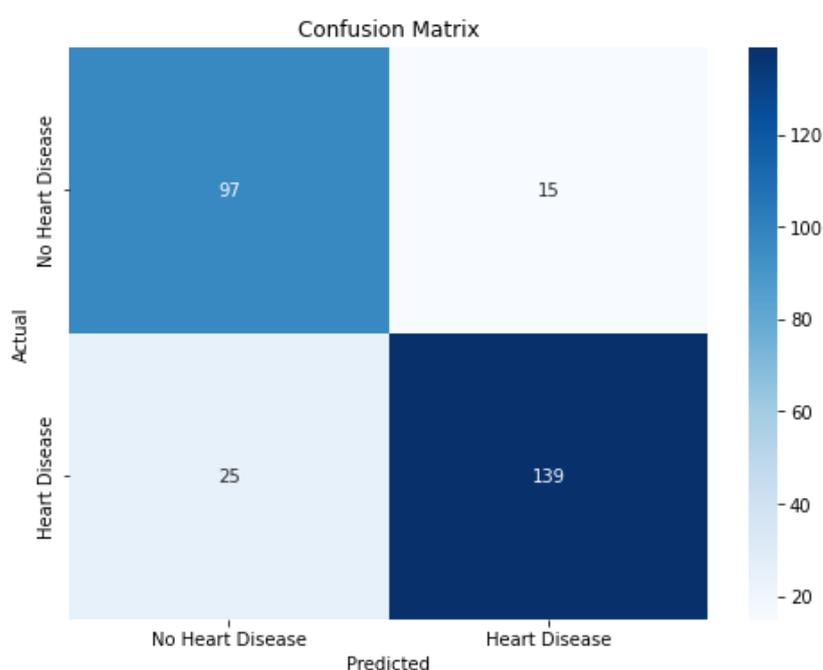
Akurasi	Presisi	Recall	F1-score
87%	86.5%	87.5%	87%

Dari hasil *confusion matrix* diperoleh perhitungan akurasi, presisi, *recall* dan *F1-score* yang dapat dilihat pada tabel 4.2, diperoleh kinerja model A1 – E dengan akurasi sebesar 87%. Secara keseluruhan, model memiliki rata-rata presisi sebesar 86.5%, *recall* 87.5%, dan *F1-score* 87%, mencerminkan performa klasifikasi yang seimbang namun masih dapat ditingkatkan.

4.1.1.3 Model B1 – NE

Pada pengujian model B1 – NE, dilakukan dengan pembagian data *training* dan data *testing* sebesar 70% : 30%, yang mana sebesar 642 dan 276 record. *confusion matrix* dari pengujian ini dapat dilihat pada gambar 4.3 dan tabel 4.3. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung

dan benar terkena penyakit gagal jantung ada 139 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 25 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 15 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 97 record.



Gambar 4.3 *confusion matrix* model B1 - NE tanpa *GridsearchCV*

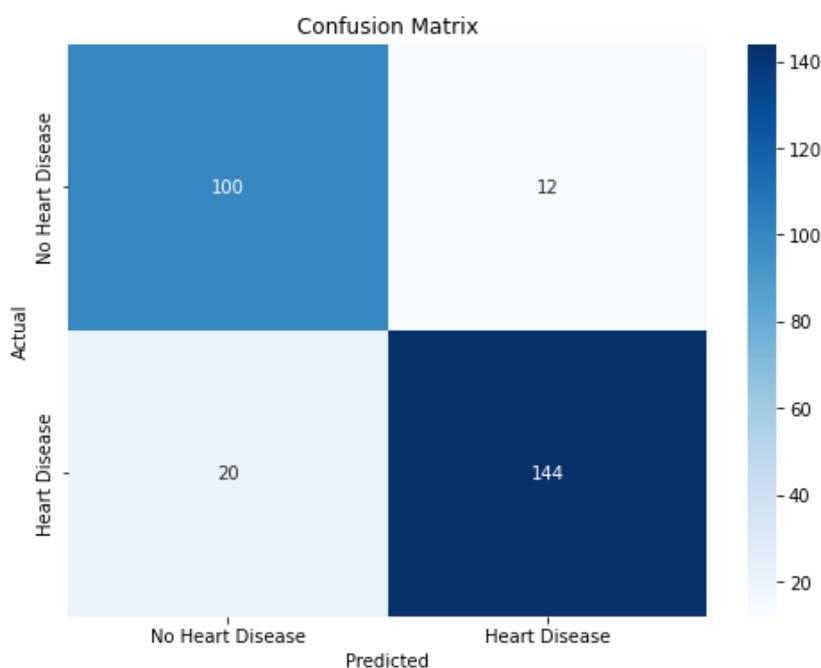
Tabel 4.3 Hasil *confusion matrix* Model B1 - NE tanpa *GridsearchCV*

Akurasi	Presisi	Recall	F1-score
86%	85%	86%	85%

Berdasarkan tabel 4.3, didapat hasil *confusion matrix* untuk model B1 – NE, dengan tidak menggunakan *encoding*, model mendapatkan hasil akurasi 86%, dengan presisi 85%, *recall* 86% dan *F1-score* 85%.

4.1.1.4 Model B1 - E

Pada pengujian model B1 - E, dilakukan dengan pembagian data *training* dan data *testing* sebesar 70% : 30%, yang mana sebesar 642 dan 276 record. *confusion matrix* dari pengujian ini dapat dilihat pada gambar 4.4 dan tabel 4.4. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 144 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 20 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 12 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 100 record.



Gambar 4. 4 *Confusion matrix* model B1 - E tanpa *GridsearchCV*

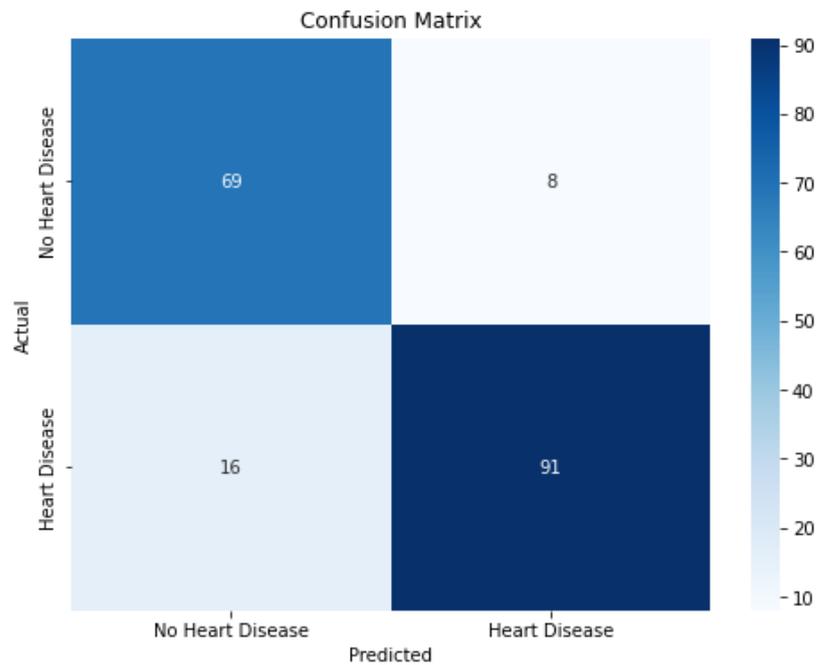
Tabel 4. 4 Hasil *confusion matrix* Model B1 - E tanpa *GridsearchCV*

Akurasi	Presisi	Recall	F1-score
88%	87.5%	88.5%	88%

Berdasarkan hasil *confusion matrix* yang ditampilkan pada Tabel 4.4, model B1 – E Model B1 berhasil mencapai akurasi sebesar 88%, yang mencerminkan kemampuan model dalam mengklasifikasikan data secara keseluruhan dengan lebih tepat. Untuk metrik lainnya seperti presisi, *recall* dan *F1-score*, model ini mampu mendapatkan hasil yang tinggi yaitu sekitar 87.5% untuk presisi, 88.5% untuk *recall* dan 88% untuk *F1-score*.

4.1.1.5 Model C1 – NE

Pada pengujian model C1 - NE, dilakukan dengan pembagian data *training* dan data *testing* sebesar 80% : 20%, yang mana sebesar 734 dan 184 record. *confusion matrix* dari pengujian ini dapat dilihat pada gambar 4.5 dan tabel 4.5. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 91 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 16 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 8 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 69 record.



Gambar 4. 5 *confusion matrix* model C1 - NE tanpa *GridsearchCV*

Tabel 4. 5 Hasil *confusion matrix* Model C1 - NE tanpa *GridsearchCV*

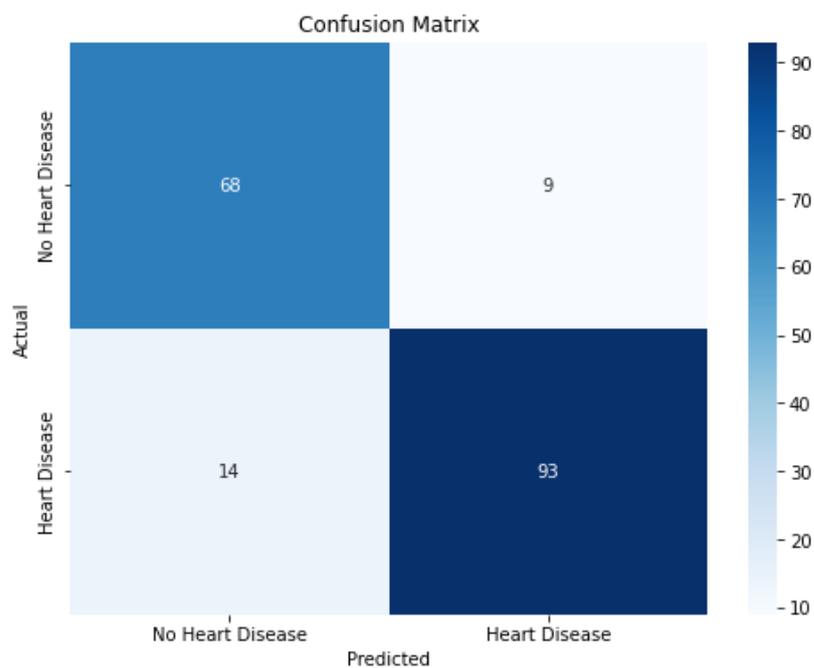
Akurasi	Presisi	Recall	F1-score
87%	86.5%	87.5%	87.5%

Berdasarkan tabel 4.5, model C1 – NE, mampu mendapatkan hasil akurasi sebesar 87%. Untuk presisi, model ini mendapatkan hasil 86.5%, sedangkan untuk hasil *recall* dan *F1-score*, model ini mendapatkan hasil yang sama, yaitu sebesar 87.5%.

4.1.1.6 Model C1 - E

Pada pengujian model C1 – E, dilakukan dengan pembagian data *training* dan data *testing* sebesar 80% : 20%, yang mana sebesar 734 dan 184 record. *confusion matrix* dari pengujian ini dapat dilihat pada gambar 4.6 dan tabel 4.6. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 93 record, untuk FN (*False Negative*)

yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 14 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 9 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 68 record.



Gambar 4. 6 *confusion matrix* model C1 – E tanpa *GridsearchCV*

Tabel 4. 6 Hasil *confusion matrix* Model C1 – E tanpa *GridsearchCV*

Akurasi	Presisi	Recall	F1-score
88%	87%	87.5%	87.5%

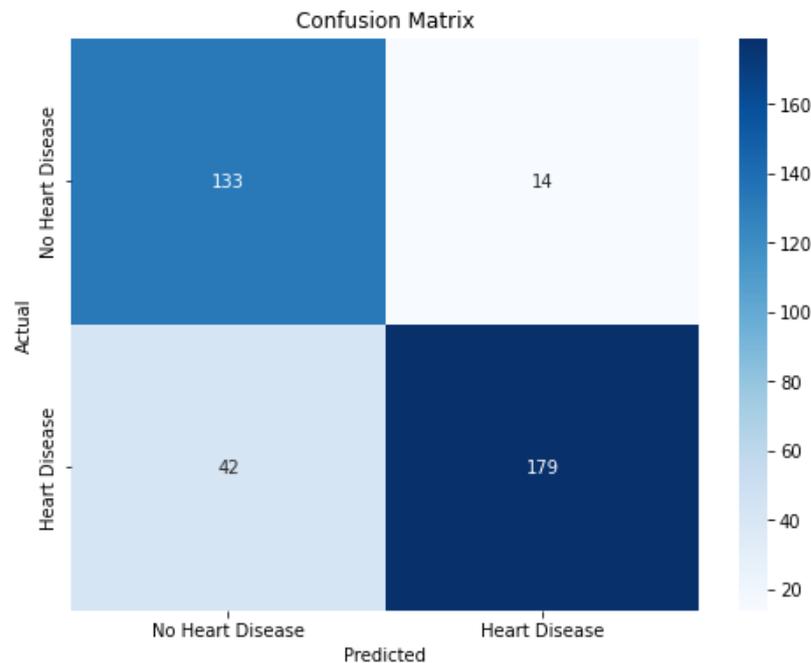
Dari hasil *confusion matrix* pada tabel 4.3, model C1 – E menunjukkan performa yang sebanding dengan model B1 dengan akurasi 88%. Secara keseluruhan, model ini mendapatkan presisi 87%, *recall* 87.5%, dan *F1-score* 87.5%.

4.1.2 Pengujian dengan *GridsearchCV*

Pada pengujian ini menggunakan optimasi parameter menggunakan *GridsearchCV*, dengan *hyperparameter* sesuai yang telah dijelaskan pada tabel 3.2.

4.1.2.1 Model A2 – NE

Pada pengujian model A2 – NE, dilakukan dengan pembagian data *training* dan data *testing* sebesar 60% : 40%, yang mana sebesar 550 dan 368 record. Berdasarkan hasil *GridsearchCV*, didapat hasil parameter terbaik dengan $n_estimator = 100$, $max_depth = None$, $max_feature = 3$, dan $min_samples = 10$. *confusion matrix* dari pengujian ini dapat dilihat pada gambar 4.7 dan tabel 4.7. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 179 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 42 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 14 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 133 record.



Gambar 4. 7 *confusion matrix* model A2 - NE dengan *GridsearchCV*

Tabel 4. 7 Hasil *confusion matrix* Model A2 - NE dengan *GridsearchCV*

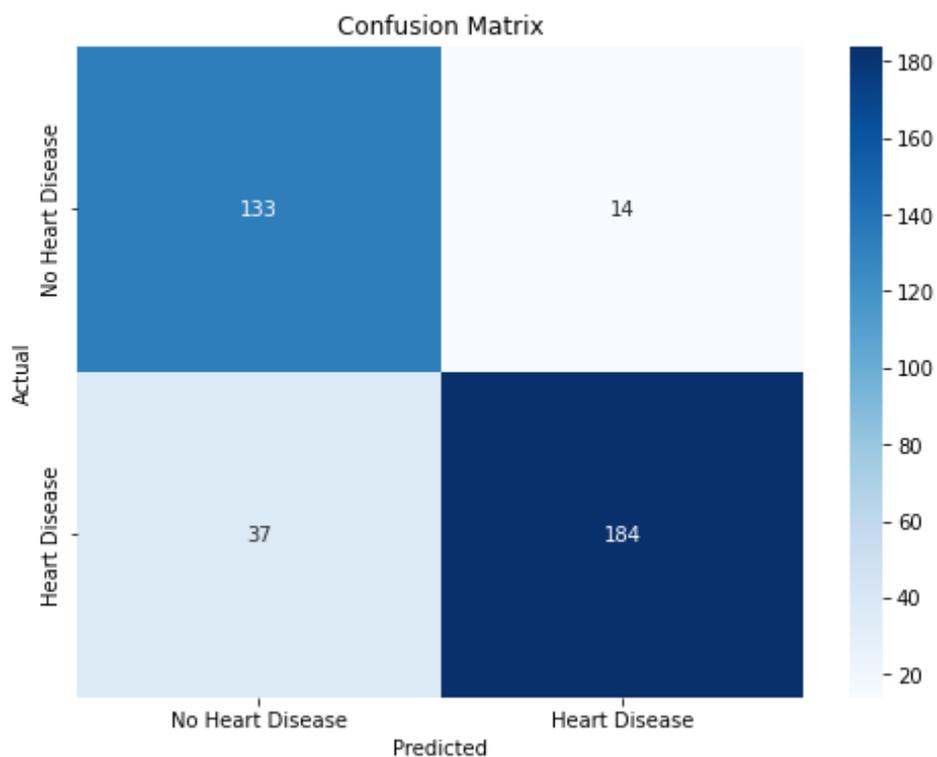
Akurasi	Presisi	Recall	F1-score
85%	84.5%	85.5%	84.5%

Berdasarkan tabel 4.7, model A2 – Ne, yang telah menggunakan *GridsearchCV* menunjukkan performa yang sedikit lebih menurun dibandingkan model A1 – Ne, yang juga sama-sama tidak menggunakan *encoding*. Hasilnya, akurasi model ini hanya mendapat 85%, presisi 84.5%, *recall* 85.5%, dan *F1-score* 84.5%.

4.1.2.2 Model A2 - E

Pada pengujian model A2 – E, dilakukan dengan pembagian data *training* dan data *testing* sebesar 60% : 40%, yang mana sebesar 550 dan 368 record. Berdasarkan hasil *GridsearchCV*, didapat hasil parameter terbaik dengan $n_estimator = 100$, $max_depth = None$, $max_feature = 3$, dan $min_samples = 10$.

confusion matrix dari pengujian ini dapat dilihat pada gambar 4.8 dan tabel 4.8. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 184 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 37 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 14 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 133 record.



Gambar 4. 8 *confusion matrix* model A2 – E dengan *GridsearchCV*

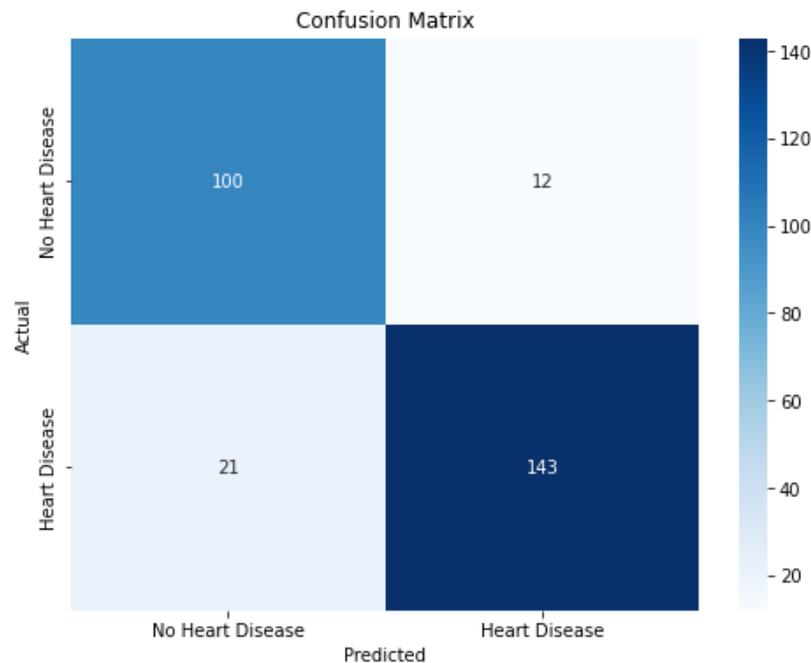
Tabel 4. 8 Hasil *confusion matrix* Model A2 – E dengan *GridsearchCV*

Akurasi	Presi	Recall	F1-score
86%	85.5%	86.5%	86%

Tabel 4.8 menunjukkan hasil evaluasi performa Model A2 – E yang dilatih menggunakan teknik *GridSearchCV* untuk pencarian kombinasi *hyperparameter* terbaik. Model ini menghasilkan akurasi sebesar 86%, sedikit lebih rendah dibandingkan Model A1 – E yang memiliki akurasi 87%. Untuk metrik lainnya, model ini mendapatkan hasil presisi sebesar 85.5%, *recall* 86.5%, dan *F1-score* 86%.

4.1.2.3 Model B2 – NE

Pada pengujian model B2 - NE, dilakukan dengan pembagian data *training* dan data *testing* sebesar 70% : 30%, yang mana sebesar 642 dan 276 record. Berdasarkan hasil *GridsearchCV*, didapat hasil parameter terbaik dengan $n_estimator = 100$, $max_depth = None$, $max_feature = 4$, dan $min_samples = 5$. *confusion matrix* dari pengujian ini dapat dilihat pada gambar 4.9 dan tabel 4.9. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 143 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 21 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 12 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 100 record.



Gambar 4. 9 *confusion matrix* model B2 - NE dengan *GridsearchCV*

Tabel 4. 9 Hasil *confusion matrix* Model B2 - NE dengan *GridsearchCV*

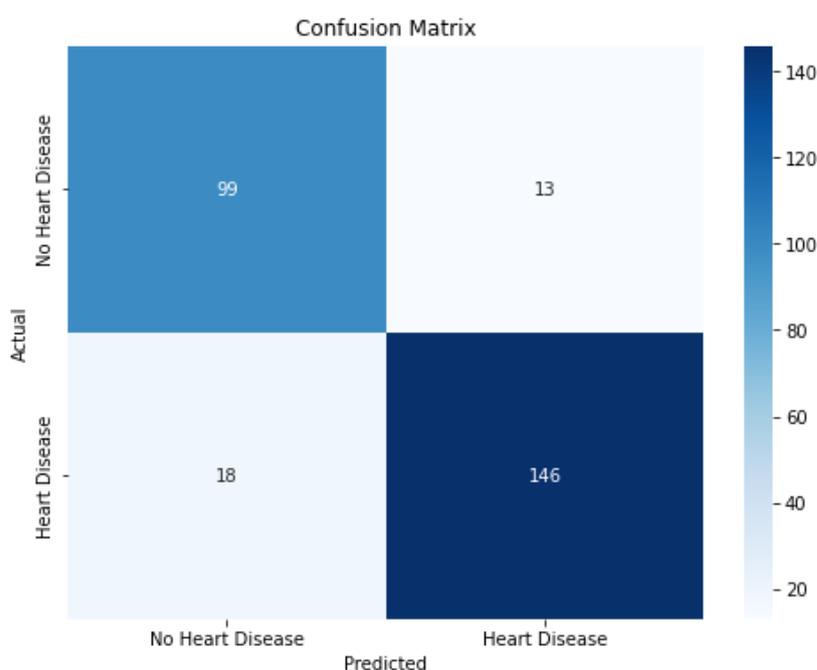
Akurasi	Presisi	Recall	F1-score
88%	87.5%	88%	88%

Berdasarkan tabel 4.9, model B2 – NE, berhasil menaikkan performa model dibandingkan dengan model B1 – NE dengan kenaikan 2% pada akurasi, dimana model B2 – NE mendapat akurasi 88%. Untuk metrik lainnya sendiri, model ini mendapatkan presisi 87.5%, *recall* 88%, dan *F1-score* 88%.

4.1.2.4 Model B2 - E

Pada pengujian model B2 – E, dilakukan dengan pembagian data *training* dan data *testing* sebesar 70% : 30%, yang mana sebesar 642 dan 276 record. Berdasarkan hasil *GridsearchCV*, didapat hasil parameter terbaik dengan $n_estimator = 100$, $max_depth = 10$, $max_feature = 4$, dan $min_samples = 10$. *confusion matrix* dari pengujian ini dapat dilihat pada gambar 4.10 dan tabel 4.10.

Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 146 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 18 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 13 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 99 record.



Gambar 4. 10 *confusion matrix* model B2 – E dengan *GridsearchCV*

Tabel 4. 10 Hasil *confusion matrix* Model B2 – E dengan *GridsearchCV*

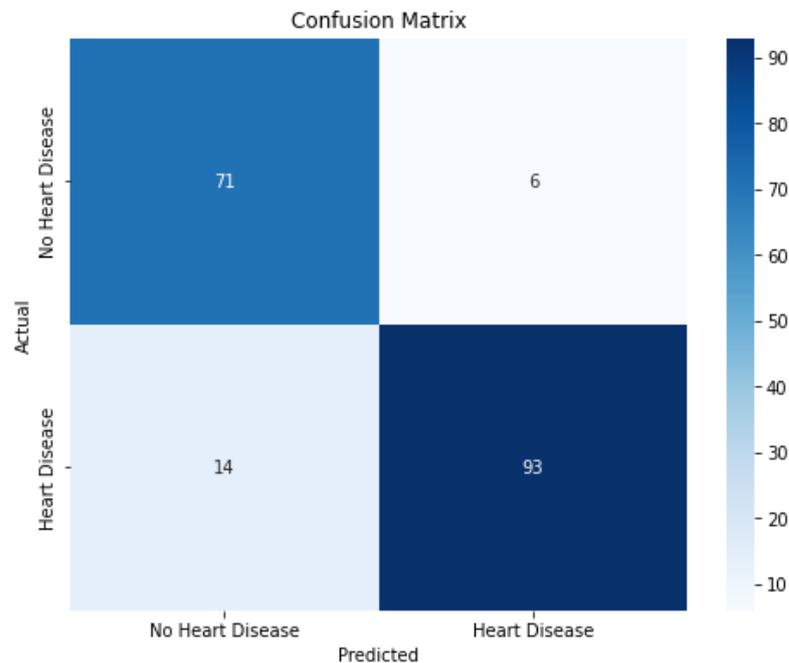
Akurasi	Presisi	Recall	F1-score
88%	87%	88%	87.5%

Berdasarkan hasil confusion matrix pada Tabel 4.10, model B2 – E menunjukkan performa yang cukup baik dengan akurasi sebesar 88%. Secara keseluruhan, model ini mendapat presisi sebesar 87%, *recall* 88%, dan *F1-score*

87.5%. Performa dari model ini sedikit menurun dibandingkan model B1 – E, terutama pada bagian presisi dan *recall*.

4.1.2.5 Model C2 – NE

Pada pengujian model C2 - NE, dilakukan dengan pembagian data *training* dan data *testing* sebesar 80% : 20%, yang mana sebesar 734 dan 184 record. Berdasarkan hasil *GridsearchCV*, didapat hasil parameter terbaik dengan $n_estimator = 100$, $max_depth = 10$, $max_feature = 3$, dan $min_samples = 2$. *confusion matrix* dari pengujian ini dapat dilihat pada gambar 4.11 dan tabel 4.11. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 93 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 14 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 6 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 71 record.



Gambar 4. 11 *confusion matrix* model C2 - NE dengan *GridsearchCV*

Tabel 4. 11 Hasil *confusion matrix* Model C2 - NE dengan *GridsearchCV*

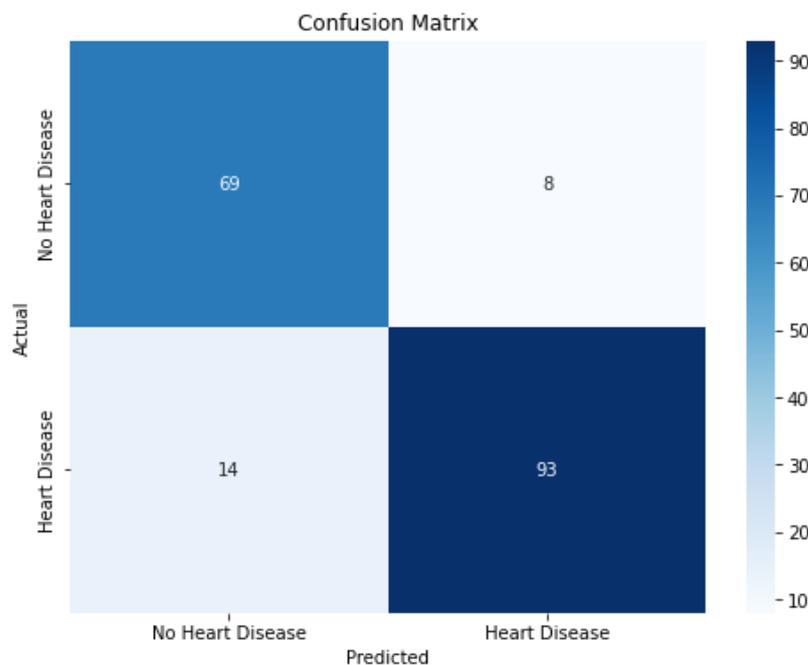
Akurasi	Presisi	Recall	F1-score
89%	89%	89.5%	89%

Berdasarkan tabel 4.11, dapat dilihat performa dari model C2 – Ne, dengan keseluruhannya mendapat akurasi 89%, presisi 89%, *recall* 89.5%, dan *F1-score* 89%. Model ini mengalami peningkatan dibandingkan model C1 – NE, yang akurasinya hanya 87%. Model ini dapat dibilang model terbaik dibandingkan model lainnya.

4.1.2.6 Model C2 - E

Pada pengujian model C2 – E, dilakukan dengan pembagian data *training* dan data *testing* sebesar 80% : 20%, yang mana sebesar 734 dan 184 record. Berdasarkan hasil *GridsearchCV*, didapat hasil parameter terbaik dengan $n_estimator = 100$, $max_depth = None$, $max_feature = 9$, dan $min_samples = 2$.

confusion matrix dari pengujian ini dapat dilihat pada gambar 4.12 dan tabel 4.12. Hasilnya adalah TP (*True Positive*) yaitu prediksi terkena penyakit gagal jantung dan benar terkena penyakit gagal jantung ada 93 record, untuk FN (*False Negative*) yaitu prediksi tidak terkena penyakit gagal jantung, namun aktualnya terkena penyakit gagal jantung ada 14 record, untuk FP (*False Positive*) yaitu prediksi terkena penyakit gagal jantung, namun aktualnya tidak terkena penyakit gagal jantung ada 8 record, dan terakhir TN (*True Negative*) yaitu prediksi dan aktualnya sama-sama tidak terkena penyakit gagal jantung ada 69 record.



Gambar 4. 12 *confusion matrix* model C2 dengan *GridsearchCV*

Tabel 4. 12 Hasil *confusion matrix* Model C2 dengan *GridsearchCV*

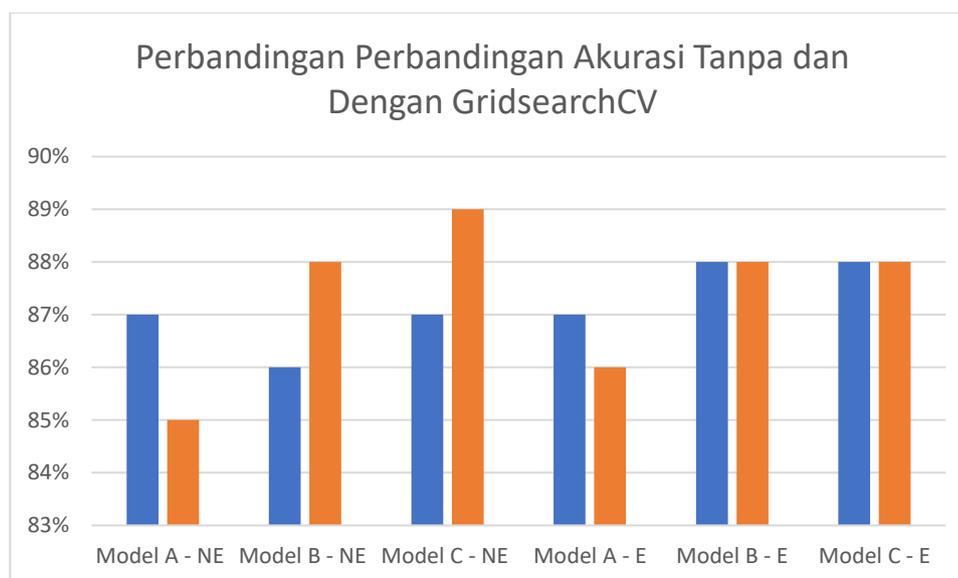
Akurasi	Presisi	Recall	F1-score
88%	87.5%	88.5%	87.5%

Berdasarkan hasil *confusion matrix* pada Tabel 4.6, model C2 – E menunjukkan performa yang sangat baik secara keseluruhan, dengan akurasi

sebesar 88%. Jika dibandingkan dengan model C1 – E, performa model C2 – E dapat dikatakan hampir setara, dengan peningkatan kecil pada beberapa metrik, seperti pada presisi model ini yang mendapatkan 87.5%, *recall* 88.5%, dan *F1-score* 87.5%.

4.1.3 Pembahasan

Hasil dari skenario uji coba yang telah dilakukan untuk menentukan skenario mana yang menghasilkan nilai paling optimal dalam melatih model *Random Forest*. Dimulai dari perbandingan akurasi Gambar 4.13.



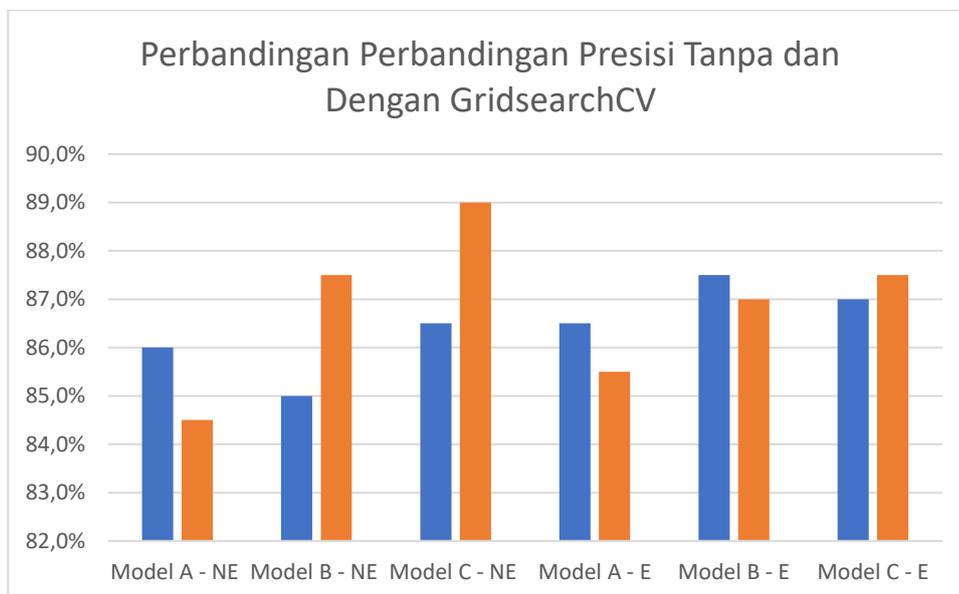
Gambar 4. 13 Perbandingan Akurasi Tanpa dan Dengan *GridsearchCV*

Gambar 4.13 memperlihatkan perbandingan akurasi antara model yang tidak menggunakan *GridsearchCV* dan dengan yang menggunakannya. Dimulai dengan model yang tidak menggunakan *encoding*, model A – NE yang tidak menggunakan *GridsearchCV* mendapatkan akurasi 87%, dimana penggunaan *GridsearchCV* pada model ini menurunkan performanya menjadi 85%. Untuk model B – NE yang tidak menggunakan *GridsearchCV* memiliki performa 86% dan optimasi *GridsearchCV*

menaikkan hasil akurasi menjadi 88%. Hal yang sama juga berlaku pada model C – NE, dimana penggunaan *GridsearchCV* menaikkan performa dari 87% menjadi 89%.

Pada model yang menggunakan *encoding* sendiri, model A – E mengalami hal yang mirip dengan model A – Ne, dimana penggunaan *GridsearchCV* menurunkan performanya, dari 87% sebelum optimasi, dan 86% setelah optimasi. Untuk model B – E dan C – E, mendapatkan hasil yang sama baik sebelum optimasi dan sesudah optimasi, dimana sama-sama mendapatkan hasil akurasi 88%.

Berdasarkan penjelasan diatas, rata-rata optimasi *GridsearchCV* berhasil menaikkan atau setidaknya mempertahankan hasil akurasi dibandingkan model yang tidak menggunakannya, kecuali untuk model A baik yang tidak menggunakan *encoding* ataupun yang menggunakannya, dimana performa optimasi *GridsearchCV* mendapatkan hasil yang lebih rendah dibandingkan yang tidak menggunakannya.

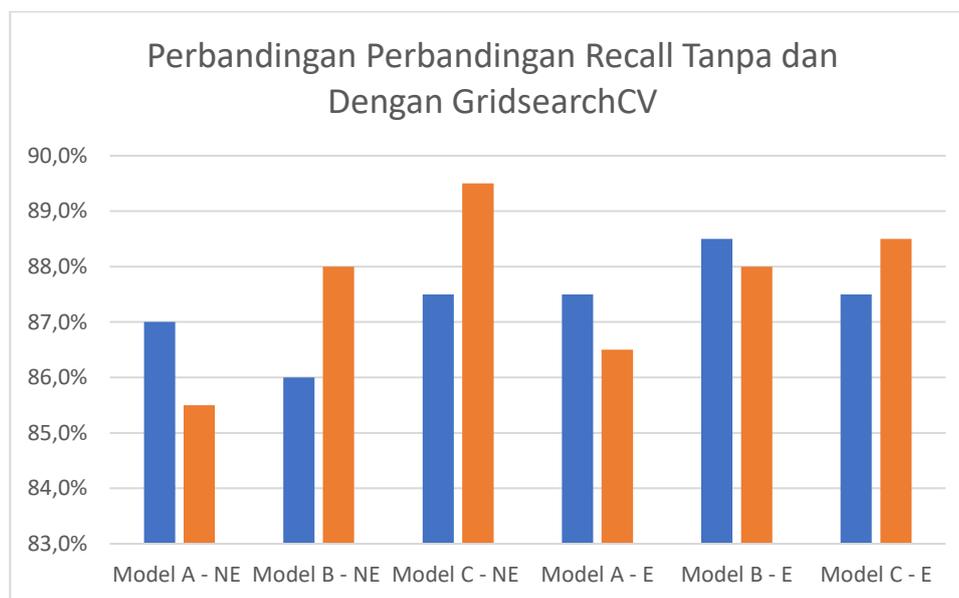


Gambar 4. 14 Perbandingan Presisi Tanpa dan Dengan *GridsearchCV*

Gambar 4.14 memperlihatkan perbandingan presisi untuk model yang tidak menggunakan *GridsearchCV* dan yang menggunakannya. Terlihat pada gambar bahwa hasil optimasi *GridsearchCV* beragam tergantung modelnya. Model A – NE dengan optimasi menurunkan hasil presisi dari 86% menjadi sekitar 84.5%. Untuk model B – Ne dan C – Ne, hasil optimasi *GridsearchCV* berhasil menaikkan performa presisi. Untuk model B – NE, presisi hasil optimasi naik dari 85% menjadi 87.5%. Untuk model C – Ne, performa berhasil ditingkatkan dari 86.5% menjadi 89%.

Pada model yang menggunakan *encoding*, optimasi *GridsearchCV* mayoritas memiliki performa yang sedikit lebih rendah dibandingkan yang tidak menggunakan. Untuk model A – E, presisi menurun dari 86.5% menjadi 85.5%. Untuk model B – E, presisi menurun dari 87.5% menjadi 87%. Untuk model C – E, presisi performa meningkat dari 87% menjadi 87.5%.

Berdasarkan penjelasan sebelumnya, hasil optimasi *GridsearchCV* memberikan hasil yang beragam dimana untuk model yang tidak menggunakan *encoding* mayoritas mengalami kenaikan performa presisi, namun untuk yang menggunakan *encoding*, performa presisi mayoritas sedikit menurun.



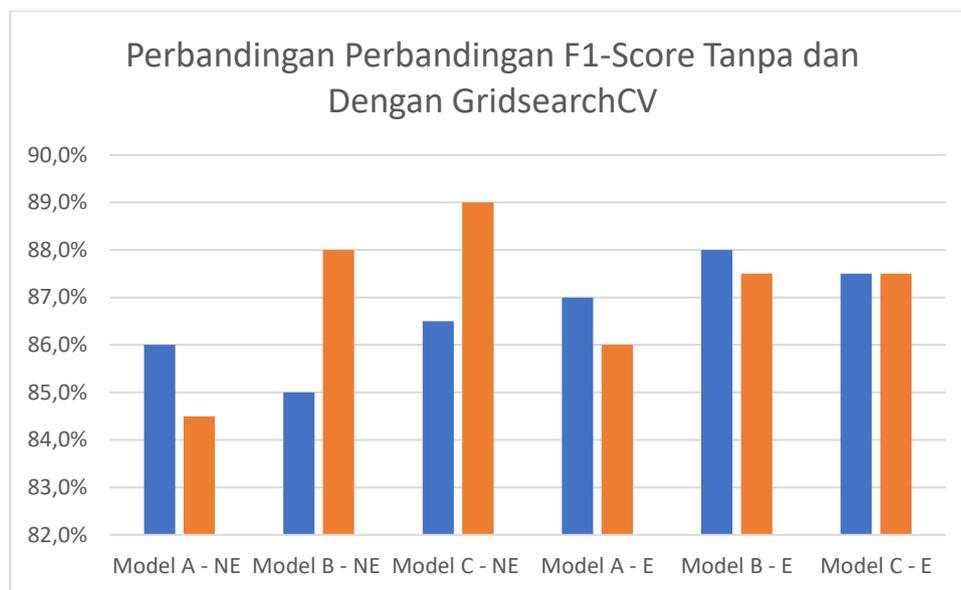
Gambar 4. 15 Perbandingan *Recall* Tanpa dan Dengan *GridsearchCV*

Berdasarkan gambar 4.15, hasil perbandingan *recall* antara model yang tidak menggunakan optimasi dan yang menggunakannya tidak jauh berbeda dari presisi. Model A – Ne, mengalami penurunan setelah menggunakan optimasi *GridsearchCV* dari 87% menjadi 85.5%. Pada model B – Ne, optimasi berhasil menaikkan hasil *recall* dari 86% menjadi 88%. Sama seperti model B – Ne, optimasi pada model C – NE berhasil menaikkan performa *recall* dari 87.5% menjadi 89.5%.

Untuk model yang menggunakan *encoding*, optimasi *GridsearchCV* pada metrik *recall* hanya berhasil menaikkan performa untuk model C – E. Untuk model A – E, performa *recall* menurun dari 87.5% menjadi 86.5%. pada model B – E, performa *recall* juga menurun sedikit dari 88.5% menjadi 88%. Sedangkan pada model C – E, performa *recall* meningkat dari 87.5% menjadi 88.5%.

Berdasarkan penjelasan diatas, hasil optimasi *GridsearchCV* pada metrik *recall* tidak jauh berbeda dengan metrik presisi sebelumnya. Dengan model yang

tidak menggunakan *encoding* rata-rata mengalami peningkatan performa, yang berbanding terbalik dengan model yang menggunakan *encoding* dengan mayoritas mengalami penurunan performa.



Gambar 4. 16 Perbandingan *F1-score* Tanpa dan Dengan *GridsearchCV*

Gambar 4.16 menunjukkan perbandingan *F1-score* untuk model yang tidak menggunakan optimasi *GridsearchCV* dan yang menggunakannya. Pada model yang tidak menggunakan *encoding*, optimasi memberikan hasil yang memuaskan secara mayoritas. Pada model A – NE, performa setelah optimasi *GridsearchCV* mengalami penurunan dari 86% menjadi 84.5%. Berbanding terbalik untuk model B – NE dan C – NE yang mengalami peningkatan performa. Pada model B – NE, performa setelah optimasi meningkat pesat dari 85% menjadi 88%. Hasil pada model C – NE tidak jauh berbeda dengan model sebelumnya, dimana peningkatan terjadi dari 86.5% menjadi 89%.

Berdasarkan penjelasan sebelumnya, didapat bahwa hasil optimasi *hyperparameter* menggunakan *GridsearchCV* memberikan hasil yang berbeda-pada tiap skenario pengujian. Pada model yang tidak menggunakan *GridsearchCV* atau dapat dikatakan model yang menggunakan parameter default ($n_estimators = 100$, $max_depth = None$, $max_features = 5$, $min_sample = 2$), memberikan performa yang cukup baik secara umum. Meskipun belum melalui proses *tuning*, model-model ini mampu memberikan hasil evaluasi yang kompetitif, terutama dalam hal akurasi dan *recall*.

Pada model A2 – Ne, *hyperparameter* yang telah dioptimasi ($n_estimators=50$, $max_depth=10$, $max_features=4$, $min_samples=2$) menyebabkan penurunan pada metrik evaluasi. Hal ini menunjukkan bahwa model menjadi kurang kompleks (*underfitting*), akibat jumlah *tree* yang lebih sedikit dan kedalaman yang dibatasi. Kombinasi ini membuat model tidak cukup fleksibel untuk menangkap pola yang kompleks dalam data, sehingga kemampuan prediksi yang tepat menurun. Pada model A2 – E, *hyperparameter* yang dioptimasi ($n_estimators=100$, $max_depth=None$, $max_features=3$, $min_samples=10$) juga menyebabkan penurunan performa, meskipun tidak sejauh model A – NE. Pada model ini, menggunakan *max_features* dikurangi dari 5 menjadi 3, dan *min_sample* ditambah menjadi 10. Perubahan parameter ini membuat performa menurun dikarenakan kurangnya informasi dari tiap fitur saat proses pembangunan *tree* dan menambahnya jumlah minimum sampel yang menyebabkan model tidak dapat mempelajari pola yang lebih detail pada proses *training*.

Pada model B2 – NE, *hyperparameter* yang telah dioptimasi ($n_estimators=100$, $max_depth=None$, $max_features=4$, $min_samples=5$) berhasil memberikan performa yang lebih baik dibandingkan model B1 – Ne. Kenaikan performa terjadi dikarenakan pengurangan *max_features* menjadi 4 dan ditambahkan *min_sample* menjadi 5 menyebabkan model dapat meningkatkan keragaman antar *tree* dan mencegah kedalaman *tree* hingga terlalu dalam. Untuk model yang menggunakan *encoding* sendiri, model B2 – E dengan optimasi *hyperparameter* ($n_estimators = 100$, $max_depth = 10$, $max_features = 4$, $min_sample = 10$) memberikan performa yang stagnan pada akurasi, namun sedikit menurun untuk metrik lainnya. Dengan perbedaan *hyperparameter* seperti *max_depth*, *max_features*, dan *min_sample*, pembuatan *tree* dibatasi untuk kedalaman dan jumlah sampel minimal yang dibutuhkan. Selain itu, pengurangan *max_features* juga membuat *tree* kehilangan informasi penting yang seharusnya dibutuhkan pada proses *training* dan menyebabkan performa pada metrik presisi, *recall*, dan *F1-score* menurun.

Untuk model C2 – Ne sendiri, optimasi *hyperparameter* ($n_estimators = 100$, $max_depth = 10$, $max_features = 3$, $min_sample = 2$) dapat dikatakan sukses dengan model ini menjadi model terbaik diantara model lainnya. Untuk perubahan performa pada skenario ini terjadi karena membatasi kedalaman menjadi 10 dapat mencegah *tree* menjadi terlalu kompleks, namun tetap dalam karena penggunaan sampel minimal untuk split tetap 2. Untuk *max_features* sendiri, pengurangannya menjadi 3 meningkatkan keragaman antar *tree* yang dapat memperbaiki generalisasi sendiri. Untuk model C2 – E, penggunaan optimasi *hyperparameter*

(*n_estimators* = 100, *max_depth* = None, *max_features* = 9, *min_sample* = 2) memberikan efek stagnan untuk akurasi, namun meningkatkan performa untuk metrik lainnya. Hal ini sendiri terjadi karena pada skenario ini, perubahannya adalah penggunaan *max_feature* yang dinaikkan menjadi 9, hal ini menyebabkan setiap *tree* memiliki akses informasi yang lebih banyak dibandingkan *max_feature* pada model yang menggunakan *hyperparameter default* yaitu 5.

Berdasarkan hasil pengujian, dapat disimpulkan bahwa model terbaik dalam penelitian ini adalah model C2 – NE. Model ini menunjukkan performa paling optimal pada seluruh metrik evaluasi, dengan akurasi sebesar 89%, presisi 89%, *recall* 89.5%, dan *F1-score* 89%. Pencapaian ini tidak terlepas dari kualitas dataset yang digunakan, sebagaimana terungkap dalam tahapan *Exploratory Data Analysis* (EDA).

Hasil EDA menunjukkan bahwa dataset memiliki distribusi kelas yang cukup seimbang antara kelas negatif dan positif, sehingga model tidak mengalami bias terhadap salah satu kelas. Selain itu, distribusi fitur numerik seperti *Age*, *RestingBP*, *MaxHR*, dan *Oldpeak* tergolong normal, mendukung proses pembelajaran model secara efektif. Satu-satunya anomali ditemukan pada fitur *Cholesterol* yang mengandung nilai outlier berupa nol, namun hal ini telah ditangani dengan baik pada tahap *preprocessing*, sehingga tidak mengganggu kinerja model.

Dengan demikian, performa unggul dari model C2 – NE mencerminkan bahwa kombinasi antara kualitas data yang baik, pemilihan skenario split data yang tepat (80:20), serta optimasi *hyperparameter* menggunakan *GridSearchCV* berhasil

menghasilkan model yang andal dan akurat dalam mendeteksi penyakit gagal jantung. Pada penjelasan berikut akan diberikan detail pada model C2 – Ne, mulai dari penggunaan *GridsearchCV* hingga prediksi pada model.

Tabel 4. 13 Proses *GridsearchCV* pada Model C2

No.	<i>n_estimators</i>	<i>max_depth</i>	<i>max_features</i>	<i>min_samples</i>	<i>Avg Accuracy</i>
1	50	None	3	2	0.8664
2	50	None	3	5	0.8610
3	50	None	3	10	0.8692
4	50	None	4	2	0.8705
5	50	None	4	5	0.8637
...
49	100	10	3	2	0.8624
50	100	10	3	5	0.8542
51	100	10	3	10	0.8583
52	100	10	4	2	0.8515
53	100	10	4	5	0.8569
...
72	100	20	9	10	0.8419

Proses penggunaan *GridsearchCV* pada model C2 – NE, dimulai dengan mendefinisikan parameter grid sesuai dengan tabel 3.3, yang mana akan dilakukan *training* terhadap setiap kombinasi yang ada sesuai dengan tabel 4.13. Hasilnya adalah *best* parameter dengan skor terbaiknya. Pada model C2 – NE *hyperparameter* terbaiknya adalah *n_estimator* = 100, *max_depth* = 10, *max_feature* = 3, dan *min_samples* = 2, dengan skor 0.86239. Hasil ini akan dijadikan *hyperparameter* untuk model *random forest* pada *training final model*.

Gambar pada lampiran 1 memperlihatkan salah satu visualisasi dari *decision tree* yang terbentuk dari model C2 – NE saat proses *training* model. Visualisasi ini telah disederhanakan untuk memudahkan pemahaman alur pengambilan keputusan oleh model dalam mengklasifikasikan apakah seseorang menderita gagal jantung atau tidak. Proses klasifikasi dalam model dimulai dari *root node* dengan fitur

MaxHR. Jika nilai *MaxHR* kurang dari atau sama dengan 131.50, maka data akan diarahkan ke cabang kiri; sebaliknya, jika lebih dari 131.50, maka diarahkan ke cabang kanan.

Pada cabang kiri, model memeriksa fitur *Sex*. Jika pasien berjenis kelamin perempuan ($Sex == F$), maka model akan mengevaluasi fitur *RestingBP*. Jika $RestingBP \leq 134.00$, maka keputusan akan dilanjutkan ke fitur *ExerciseAngina*. Jika nilai *ExerciseAngina* adalah N, maka akan dilanjutkan ke fitur berikutnya yang tidak ditampilkan pada gambar. Namun jika hasilnya adalah *false*, maka prediksi akhir adalah kelas 0 yang berarti tidak terkena penyakit gagal jantung. Namun jika pada fitur *Resting BP* > 134.00 , maka akan dilanjutkan oleh fitur *cholesterol* ≤ 250.00 , jika hasil *true* akan dilanjutkan oleh fitur berikutnya, dan jika *false* akan memberikan prediksi akhir kelas 1 yang berarti terkena penyakit gagal jantung. Proses yang sama juga terjadi pada cabang kanan.

Tabel 4. 14 Contoh prediksi dari model C2

ID	Age	RestingBP	Cholesterol	MaxHR	...	Prediksi	Aktual
1	63	140	195	179	...	0	0
2	53	145	518	130	...	1	1
3	65	160	0	122	...	1	1
4	56	130	0	122	...	1	1
5	54	108	309	156	...	0	0
6	67	125	254	163	...	1	1
7	56	120	0	148	...	1	1
8	69	142	271	126	...	0	0
9	46	140	272	175	...	0	1
10	58	120	0	106	...	1	1
11	56	120	193	162	...	0	0
12	62	120	220	86	...	0	0
13	64	130	303	122	...	1	0
14	56	120	236	178	...	0	0
15	57	165	289	124	...	1	1
16	41	125	0	176	...	1	1
17	64	140	313	133	...	0	0
18	57	95	0	182	...	1	1
19	39	118	219	140	...	1	1
20	67	115	564	160	...	0	0

Pada Tabel 4.14 ditampilkan hasil prediksi dari Model C2 – NE terhadap sejumlah data uji. Berdasarkan hasil tersebut, secara keseluruhan model mampu melakukan prediksi dengan baik, di mana sebagian besar prediksi sesuai dengan label aslinya. Hal ini mencerminkan bahwa model telah mampu mengenali pola dalam data dengan tingkat akurasi yang cukup tinggi. Meskipun demikian, masih terdapat beberapa prediksi yang tidak sesuai, seperti pada data dengan ID 9 dan ID 13, yang menunjukkan bahwa model belum sepenuhnya sempurna dalam mengklasifikasikan seluruh data uji. Namun demikian, jumlah kesalahan yang relatif sedikit menunjukkan bahwa model memiliki performa yang stabil serta kemampuan generalisasi yang baik terhadap data yang belum pernah dilihat sebelumnya.

Untuk menjawab pernyataan masalah yang pertama, optimasi parameter menggunakan *GridSearchCV* menghasilkan dampak performa yang bervariasi tergantung pada skenario uji coba dan tipe pemrosesan data yang digunakan. Berdasarkan hasil pengujian yang telah dilakukan, terlihat bahwa pada model-model yang tidak menggunakan *encoding* (NE), penerapan *GridSearchCV* umumnya memberikan peningkatan performa yang signifikan. Hal ini ditunjukkan pada Model B – NE dan Model C – NE, yang mengalami kenaikan pada hampir seluruh metrik evaluasi, seperti akurasi, presisi, *recall*, dan *F1-score*. Sebaliknya, pada model-model yang menggunakan *encoding* (E), efek dari *GridSearchCV* cenderung stagnan atau hanya memberikan dampak sedikit secara performa, baik meningkatkan atau menurunkannya. Namun dalam beberapa model diketahui bahwa optimasi *GridsearchCV* memberikan performa jauh lebih rendah

dibandingkan model sebelum optimasi, yaitu model A – NE dan A – E. Hal ini kemungkinan disebabkan oleh pemilihan parameter hasil *tuning* yang kurang sesuai, seperti jumlah *tree* yang terlalu sedikit atau kedalaman *tree* yang terlalu dangkal, sehingga menyebabkan model mengalami *underfitting*.

Untuk menjawab pernyataan masalah kedua, hasil evaluasi menunjukkan bahwa Model C2 – NE, yaitu model tanpa *encoding* dan dengan optimasi *GridSearchCV*, memberikan performa terbaik di antara semua model yang diuji. Secara spesifik, model ini mencatatkan peningkatan sebesar 2% pada metrik akurasi dan presisi, serta peningkatan sebesar 2,5% pada metrik *recall* dan *F1-score*, jika dibandingkan dengan versi model sebelumnya yang tidak menggunakan *GridSearchCV*.

Visualisasi performa dan perbandingan antar model yang telah ditampilkan pada Gambar 4.13 hingga Gambar 4.16 mendukung temuan ini, dengan menunjukkan bahwa model C2 – NE konsisten memberikan hasil yang seimbang di semua metrik utama. Oleh karena itu, model C2 – NE dinyatakan sebagai model terbaik dalam penelitian ini. Hal ini sekaligus membuktikan bahwa pendekatan optimasi *hyperparameter* menggunakan *GridSearchCV* mampu menghasilkan model yang handal, seimbang, dan dapat digunakan sebagai alat bantu deteksi dini penyakit gagal jantung secara lebih akurat dan efisien.

Dari hasil penelitian ini diharapkan dapat berkontribusi dalam pengembangan teknologi medis yang mampu membantu deteksi dini penyakit gagal jantung, sehingga penanganannya dapat dilakukan lebih cepat dan tepat. Sebagai manusia yang tak luput dari ancaman masalah kesehatan, kita dituntut

untuk tidak pasrah begitu saja, melainkan terus berikhtiar mencari solusi. Hal ini sejalan dengan hadits riwayat Bukhari yang berbunyi :

لِكُلِّ دَاءٍ دَوَاءٌ، فَإِذَا أُصِيبَ دَوَاءُ الدَّاءِ بَرَأَ بِذُنِّ اللَّهِ

“Setiap penyakit ada obatnya. Apabila obat itu tepat untuk suatu penyakit, penyakit itu akan sembuh dengan seizin Allah.” (HR. Bukhari)

Hadits ini menunjukkan bahwa setiap penyakit yang Allah turunkan ke muka bumi juga telah disertai dengan penawarnya. Artinya, Allah tidak menciptakan suatu penyakit tanpa jalan penyembuhan (Muslim, n.d.). Namun, kesembuhan itu tidak serta-merta terjadi, melainkan membutuhkan proses pencarian dan penerapan obat yang sesuai. Hadits ini juga menegaskan bahwa kesembuhan tetap berada di tangan Allah, namun usaha manusia melalui diagnosis yang tepat dan pengobatan yang sesuai sangat dianjurkan sebagai bagian dari ikhtiar. Dalam konteks penelitian ini, pengembangan model *machine learning* untuk mendeteksi penyakit gagal jantung merupakan bentuk nyata dari upaya ilmiah dalam mencari "obat" atau solusi untuk membantu mengenali penyakit secara dini, agar dapat dilakukan penanganan medis yang lebih cepat dan efektif, sesuai dengan semangat yang diajarkan dalam hadits tersebut.

Upaya untuk membangun sistem pendeteksi tersebut diperlukan keseimbangan dan ketelitian dalam pemilihan metode agar hasilnya akurat dan bermanfaat. Hal tersebut sesuai dengan *QS Al-Furqan* ayat 2, yang menegaskan bahwa setiap ciptaan Allah SWT. telah ditetapkan dengan ukuran dan keseimbangan yang sempurna. Ayat tersebut berbunyi :

الَّذِي لَهُ مُلْكُ السَّمَاوَاتِ وَالْأَرْضِ وَلِي نَتَّخِذُ وَلَدًا وَلَيْ يَكُنْ لَهُ شَرِيكٌ فِي الْمُلْكِ وَخَلَقَ كُلَّ شَيْءٍ

فَ قَدَّرَهُ تَ قَدِيرًا ۝

“(Yaitu Zat) yang milik-Nyalah kerajaan langit dan bumi, (Dia) tidak mempunyai anak, dan tidak ada satu sekutu pun dalam kekuasaan(-Nya). Dia telah menciptakan segala sesuatu, lalu menetapkan ukuran-ukurannya dengan tepat.” (QS. Al-Furqan/ 25:2)

Menurut penjelasan dalam Tafsir Ibnu Katsir, ayat tersebut mengandung makna bahwa segala sesuatu selain Allah SWT adalah makhluk ciptaan-Nya yang tunduk sepenuhnya di bawah kekuasaan-Nya. Allah adalah satu-satunya Pencipta, Penguasa, dan Tuhan dari seluruh alam semesta. Seluruh ciptaan-Nya berada dalam kendali, pengaturan, dan ketetapan-Nya. Tafsir ini menggambarkan bahwa Allah menciptakan segala sesuatu dengan presisi dan menetapkan ukuran yang sesuai bagi setiap makhluk, sehingga tidak terdapat kekeliruan atau ketidakseimbangan dalam penciptaan-Nya. Segala sesuatu berjalan dalam kehendak dan ketentuan-Nya (Kathir, n.d.). Makna yang ada sesuai dengan usaha peneliti untuk mencari model yang lebih optimal berdasarkan *hyperparameter* yang diberikan demi mengurangi kesalahan yang terjadi dalam proses deteksi.

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa optimasi *hyperparameter* menggunakan *GridSearchCV* pada algoritma *Random Forest* untuk deteksi penyakit gagal jantung menghasilkan performa yang bervariasi pada setiap skenario uji coba. Pada skenario yang tidak menggunakan *encoding*, optimasi *GridsearchCV* memberikan hasil yang kurang optimal pada model A2 – NE, namun memberikan peningkatan performa pada model B2 – NE dan C2 – NE, dimana model C2 – NE dipilih sebagai model terbaik diantara model lainnya. Sedangkan pada model yang menggunakan *encoding*, optimasi *GridsearchCV* juga memberikan penurunan performa pada model A2 – E, dan memberikan performa yang cenderung stagnan untuk model B2 – E dan C2 – E.

Secara khusus, model terbaik dalam penelitian ini adalah C2 – NE, yang menunjukkan peningkatan performa yang signifikan dibandingkan dengan model sebelumnya, C1 – NE, yang belum dioptimasi menggunakan *GridSearchCV*. Model C2 – NE mengalami peningkatan pada seluruh metrik evaluasi, yaitu akurasi meningkat 2% dari 87% menjadi 89%, presisi 2% dari 87.5% menjadi 89.5%, serta *recall* dan *F1-score* 2.5% dari 86.5% menjadi 89%. Peningkatan ini mencerminkan efektivitas penggunaan *GridSearchCV* dalam mengoptimalkan *hyperparameter*, sehingga menghasilkan model yang lebih akurat, seimbang, dan andal dalam mendeteksi penyakit gagal jantung.

5.2 Saran

Berdasarkan analisa dan perbandingan performa model *Random Forest*, terdapat beberapa saran strategis yang dapat dijadikan rujukan untuk peningkatan performa deteksi penyakit gagal jantung di masa mendatang.

1. Perluasan ruang pencarian *Hyperparameter*, memperluas grid parameter dapat dicoba untuk mencari tahu parameter yang lebih optimal, atau metode optimasi lain seperti *RandomizedSearchCV* atau *Bayesian Optimization* agar hasil tuning lebih maksimal dan tidak terbatas pada kombinasi yang sempit.
2. Penggunaan dataset yang lebih besar dan seimbang, dataset yang digunakan oleh peneliti hanya mencakup 12 fitur dengan 918 jumlah data. Penelitian selanjutnya dapat menggunakan dataset yang lebih besar dan seimbang agar hasil model lebih general dan akurat dalam berbagai kondisi.
3. Penggunaan *library*, dalam penelitian ini, peneliti tidak menggunakan *library* untuk *random forest* ataupun *GridsearchCV*, hal ini membuat proses yang dapat dilakukan terbatas, terutama dalam penggunaan *GridsearchCV* yang proses komputasi menjadi lebih lama dibanding saat menggunakan *library*.
4. Menggunakan *split data* yang lain, dalam penelitian ini, peneliti hanya menggunakan 3 jenis split data, yaitu 60:40, 70:30, dan 80:20. Untuk penelitian selanjutnya dapat menggunakan *split* yang lebih banyak seperti 72:25 atau 90:10.

DAFTAR PUSTAKA

- Ahmad, G. N., Fatima, H., Ullah, S., Saidi, A. S., & others. (2022). Efficient medical diagnosis of human heart diseases using machine learning techniques with and without GridSearchCV. *IEEE Access*, *10*, 80151–80173.
- Ahmadi, Z., Abdullah, A., & Fakhruzi, I. (2023). Meningkatkan Kemampuan Model dalam Memprediksi Penyakit Jantung dengan Algoritma NCL dan GridSearchCV. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *7*(4), 1632–1633.
- Akther, K., Kohinoor, M. S. R., Priya, B. S., Rahaman, M. J., Rahman, M. M., & Shafiullah, M. (2024). Multi-Faceted Approach to Cardiovascular Risk Assessment by Utilizing Predictive Machine Learning and Clinical Data in a Unified Web Platform. *IEEE Access*.
- Alwakid, G., Ul Haq, F., Tariq, N., Humayun, M., Shaheen, M., & Alsadun, M. (2025). Optimized machine learning framework for cardiovascular disease diagnosis: A novel ethical perspective. *BMC Cardiovascular Disorders*, *25*(1), 123.
- Baghdadi, N. A., Farghaly Abdelaliem, S. M., Malki, A., Gad, I., Ewis, A., & Atlam, E. (2023). Advanced machine learning techniques for cardiovascular disease early detection and diagnosis. *Journal of Big Data*, *10*(1), 144.
- Dai, Q., Sherif, A. A., Jin, C., Chen, Y., Cai, P., & Li, P. (2022). Machine learning predicting mortality in sarcoidosis patients admitted for acute heart failure. *Cardiovascular Digital Health Journal*, *3*(6), 297–304.
- Darmawan, Z. M. E., & Dianta, A. F. (2023). Implementasi optimasi hyperparameter GridSearchCV pada sistem prediksi serangan jantung menggunakan SVM. *Teknologi: Jurnal Ilmiah Sistem Informasi*, *13*(1), 8–15.
- Disha, R. A., & Waheed, S. (2022). Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique. *Cybersecurity*, *5*(1), 1.
- fedesoriano. (2021). *Heart Failure Prediction Dataset* [Dataset]. <https://www.kaggle.com/fedesoriano/heart-failure-prediction>.
- García-Ordás, M. T., Bayón-Gutiérrez, M., Benavides, C., Avelaira-Mata, J., & Benítez-Andrades, J. A. (2023). Heart disease risk prediction using deep

learning techniques with feature augmentation. *Multimedia Tools and Applications*, 82(20), 31759–31773.

Genuer, R., Poggi, J.-M., Genuer, R., & Poggi, J.-M. (2020). *Random forests*. Springer.

Hermawan, K. A., Rizki, A., Sinaga, D. K., & Suwarman, H. R. (2024). Prediksi Gagal Jantung Berbasis Machine Learning Menggunakan Support Vector Machine dan Regresi Logistik. *Seminar Nasional Penelitian (SEMNAS CORISINDO 2024)*, 436–441.

Kathir, I. I. (n.d.). *Tafsir Ibn Kathir atas QS Al-Furqan (25):2*. Retrieved June 24, 2025, from <https://tafsirweb.com/6258-surat-al-furqan-ayat-2.html>

Luque, A., Carrasco, A., Martín, A., & de Las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91, 216–231.

Mansoori, A., Zeinalnezhad, M., & Nazarimanesh, L. (2023). Optimization of Tree-Based Machine Learning Models to Predict the Length of Hospital Stay Using Genetic Algorithm. *Journal of Healthcare Engineering*, 2023(1), 9673395.

Muhammad, Y., Tahir, M., Hayat, M., & Chong, K. T. (2020). Early and accurate detection and diagnosis of heart disease using intelligent computational model. *Scientific Reports*, 10(1), 19747.

Muslim, I. (n.d.). *Kitab al-Salam, Hadits No. 2204*. <https://hadits.id/hadits/muslim/2204>

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3), e1301.

Religia, Y., Rusdi, A., Romli, I., & Mazid, A. (2019). Feature Extraction Untuk Klasifikasi Pengenalan Wajah Menggunakan Support Vector Machine Dan K-Nearest Neighbor. *Pelita Teknologi*, 14(2), 85–92.

Schalock, R. L., Luckasson, R., & Tassé, M. J. (2021). An overview of intellectual disability: Definition, diagnosis, classification, and systems of supports. *American Journal on Intellectual and Developmental Disabilities*, 126(6), 439–442.

Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal*, 20(1), 3–29.

- Sepharni, A., Hendrawan, I. E., & Rozikin, C. (2022). Klasifikasi Penyakit Jantung dengan Menggunakan Algoritma C4. 5. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 7(2), 117–126.
- Septhya, D., Rahayu, K., Rabbani, S., Fitria, V., Rahmaddeni, R., Irawan, Y., & Hayami, R. (2023). Implementasi Algoritma Decision Tree dan Support Vector Machine untuk Klasifikasi Penyakit Kanker Paru: Implementation of Decision Tree Algorithm and Support Vector Machine for Lung Cancer Classification. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 3(1), 15–19.
- Speiser, J. L., Miller, M. E., Tooze, J., & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert Systems with Applications*, 134, 93–101.
- TafsirWeb.com. (n.d.). *Surat Al-Baqarah Ayat 57 Arab, Latin, Terjemah dan Tafsir – TafsirWeb*. <https://tafsirweb.com/362-surat-al-baqarah-ayat-57.html>
- Tamba, S. P. & others. (2022). PREDIKSI PENYAKIT GAGAL JANTUNG DENGAN MENGGUNAKAN RANDOM FOREST. *Jurnal Sistem Informasi Dan Ilmu Komputer Prima (JUSIKOM PRIMA)*, 5(2), 176–181.

LAMPIRAN

