KLASIFIKASI PENYAKIT SERANGAN JANTUNG MENGGUNAKAN ALGORITMA *SUPPORT VECTOR MACHINE* (SVM) DAN *RECURSIVE FEATURE ELIMINATION* (RFE)

SKRIPSI

Oleh :
MUHAMMAD PUTRA ZIDANNIZAR
NIM. 210605110143

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

KLASIFIKASI PENYAKIT SERANGAN JANTUNG MENGGUNAKAN ALGORITMA *SUPPORT VECTOR MACHINE* (SVM) DAN *RECURSIVE FEATURE ELIMINATION* (RFE)

SKRIPSI

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

Oleh :
MUHAMMAD PUTRA ZIDANNIZAR
NIM. 210605110143

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

HALAMAN PERSETUJUAN

KLASIFIKASI PENYAKIT SERANGAN JANTUNG MENGGUNAKAN ALGORITMA *SUPPORT VECROR MACHINE* (SVM) DAN *RECURSIVE FEATURE ELIMINATION* (RFE)

SKRIPSI

Oleh :
MUHAMMAD PUTRA ZIDANNIZAR
NIM. 210605110143

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 17 April 2025

Pembimbing I,	Pembimbing II,
Prof. Dr. Muhammad Faisal, M.T	Dr. Yunifa Miftachul Arif, M.T
NIP. 19740510 200501 1 007	NIP. 19830616 201101 1 004

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

KLASIFIKASI PENYAKIT SERANGAN JANTUNG MENGGUNAKAN ALGORITMA *SUPPORT VECROR MACHINE* (SVM) DAN *RECURSIVE FEATURE ELIMINATION* (RFE)

SKRIPSI

Oleh :
MUHAMMAD PUTRA ZIDANNIZAR
NIM. 210605110143

Telah Dipertahankan di Depan Dewan Penguji Skripsidan Dinyatakan Diterima
Sebagai Salah Satu PersyaratanUntuk Memperoleh
Gelar Sarjana Komputer (S.Kom)
Tanggal: 07 Mei 2025

Susunan Dewan Penguji

Ketua Penguji	: Dr. Irwan Budi Santoso, M.Kom NIP. 19770103 201101 1 004	( )
Anggota Penguji I	: Ajib Hanani, M.T NIP. 19840731 202321 1 013	( )
Anggota Penguji II	: Prof. Dr. Muhammad Faisal, M.T NIP. 19740510 20050 1 1007	( )
Anggota Penguji III	: Dr. Yunifa Miftachul Arif, M.T NIP. 19830616 201101 1 004	( )

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 Putra Zidannizar
NIM	: 210605110143
Fakultas / Jurusan	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Klasifikasi Penyakit Serangan Jantung Menggunakan Algoritma *Support Vector Machine* (SVM) dan *Recursive Feature Elimination* (RFE)

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, 17 April 2025
Yang membuat pernyataan,

Muhammad Putra Zidannizar
NIM. 210605110143

MOTTO

“Maka, sesungguhnya beserta kesulitan ada kemudahan.”

(Qs. Al-Insyirah · Ayat 5)

“Orang Lain Gak Akan Bisa Paham Struggle Dan Masa Sulitnya Kita Yang Mereka Ingintau Hanya Bagian *Success Stories.* Berjuanglah Untuk Dirimu Sendiri Walaupun Gaada Yang Tepuk Tangan. Kelak Diri Kita Di Masa Depan Akan Sangat Bangga Dengan Apa Yang Kita Perjuangkan Hari Ini.”

“Jalur Langit Pancen Ora Ketoro Tapi Insyallah Ketoto”

“Tidak Ada Kata Terlambat Untuk Menciptakan Kehidupan Yang Kamu Inginkan”

-zidannizar

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT atas

segala rahmat dan kemudahan-Nya, sehingga skripsi ini dapat

terselesaikan dengan baik. Karya ini penulis persembahkan kepada:

Mama Sinta dan Papa Dadik,

Yang selalu mengalirkan kasih sayang, usaha terbaik, do'a-do'a tulus, dukungan

baik materi maupun non materi, dan nasehat yang tiada henti

Mas Febri dan Adik Dinda

Yang menjdadi salah satu motovasi dan dorongan untuk menyelesaikan studi di perantauan ini.

Segenap keluarga besar,

Yang selalu mengiringi perjalanan penulis dengan do'a dan dukungan.

Aster Teknik Informatika angkatan 2021,

Yang telah berjuang bersama dari awal hingga akhir

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillahirabbilalamin, segala puji dan syukur senantiasa penulis panjatkan ke hadirat Allah subhanahu wa ta'ala atas berkat Rahmat, serta hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Klasifikasi Penyakit Serangan Jantung Menggunakan Algoritma *Support Vecror Machine* (SVM) Dan *Recursive Feature Elimination* (RFE)" dengan baik dan lancar. Shalawat serta salam tetap tercurahkan kepada Nabi Muhammad SAW, yang telah membawa kita dari zaman kebodohan menuju zaman kebenaran yakni Islam dan zaman yang penuh dengan ilmu pengetahuan sebagaimana yang di rasakan pada saat ini. Dan semoga kita semua mendapat syafaatnya di hari akhir kelak, Aamiin.

Penulis mengucapkan rasa terima kasih yang begitu besar kepada seluruh pihak yang memberikan dukungan dan motivasi kepada penulis sehingga dapat menyelesaikan skripsi ini. Ucapan terima kasih ini penulis disampaikan kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang
2. Prof. Dr. Hj. Sri Harini, 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.
4. Dr. Muhammad Faisal, M.T., selaku pembimbing I, yang telah dengan penuh dedikasi meluangkan waktu untuk memberikan bimbingan, arahan,

dan motivasi yang berharga selama proses penyusunan skripsi ini hingga dapat diselesaikan dengan baik.

5. Dr. Yunifa Miftachul Arif, M.T selaku pembimbing II, yang dengan tulus telah memberikan arahan, dan dengan cermat meluruskan kesalahan skripsi ini.
6. Dr. Irwan Budi Santoso, M.Kom selaku Ketua Penguji yang telah memberikan banyak saran dan arahan dari seminar proposal hingga sidang skripsi penulis.
7. Ajib Hanani, M.T selaku Anggota Penguji 1 yang telah memberikan saran dan masukan berharga sepanjang proses, mulai dari seminar proposal hingga sidang skripsi.
8. Okta Qomaruddin aziz, M.Kom selaku dosen yang mau membimbing penulis disaat penulis bingung akan tentang metode yang diterapkan di judul dan dosen yang selalu sabar membimbing di kela kesibukan beliau
9. Nia Faricha, S.Si selaku admin Program Studi Teknik Informatika yang selalu sabar memberikan informasi, membantu, dan memberikan arahan selama perkuliahan dan proses penulisan skripsi ini.
10. Segenap dosen, laboran, dan jajaran staff Program Studi Teknik Informatika yang telah memberikan ilmu, pengetahuan, dan dukungan selama penulis menjalani studi.
11. Kepada sosok panutan sepanjang hidup penulis, Papa Dadik Satriyo Harsono, Amd.Kep, yang dengan penuh kasih tak pernah lelah mengingatkan penulis untuk selalu mendekatkan diri kepada Allah,

khususnya dalam menjaga salat lima waktu dan berjamaah di masjid. Sosok ayah yang rela mengorbankan tenaga, waktu, bahkan kenyamanan dirinya demi kebahagiaan dan masa depan anak-anaknya. Seorang ayah yang tak hanya menjadi penyemangat, tetapi juga tempat berlabuh doa dan harapan, yang selalu percaya dan mendukung sepenuh hati agar anaknya tumbuh menjadi pribadi yang sukses, beriman, dan membanggakan

12. Kepada pintu syurga saya, Mama Furaidana Dewi Nasinta, sosok ibu penuh kasih yang tak pernah lupa menanyakan kabar di tengah kesibukannya, yang selalu mengingatkan untuk tidak melupakan sedekah walau sekecil apa pun, serta menanamkan nilai untuk senantiasa berbuat baik kepada siapa pun. Doa, perhatian, dan nasihat beliau menjadi cahaya penuntun dalam setiap langkah penulis.
13. Kepada kakak penulis, Yudhistira Satriyo Febriyanto, S.Kep., Ns., yang selalu hadir di saat penulis membutuhkan—baik dalam hal semangat, dukungan moral, hingga bantuan akomodasi dan finansial. Terima kasih atas segala pengorbanan, kepedulian, dan kasih sayang yang tak terbalas. Semoga Allah membalas semua kebaikanmu dengan limpahan rezeki, kesehatan, dan kebahagiaan yang tak pernah putus. Aamiin.
14. Kepada adik penulis, Dinda Setia Rini, yang dengan sabar dan penuh rindu selalu menantikan kepulangan penulis dari perantauan menempuh kuliah. Sosok adik yang menjadi pelipur lara di tengah lelah dan letih, serta menjadi sumber semangat tersendiri bagi penulis untuk terus berjuang. Engkau akan selalu penulis jaga dan sayangi sepenuh hati. Semoga kelak tumbuh menjadi

pribadi yang kuat, cerdas, dan bahagia, serta selalu dalam lindungan Allah SWT. Aamiin.

15. Kepada keluarga besar Mbah Suroyo dan Mbah Sri Mirantini dan Mbah Efendi dan Mbah Ainun, khususnya kepada anak-anak beliau yang juga merupakan adik-adik dari kedua orang tua penulis, yang telah menjadi bagian penting dalam perjalanan hidup penulis. Terima kasih atas perhatian, doa, dan dukungan yang selalu diberikan, baik secara langsung maupun tidak langsung. Semoga silaturahmi dan kasih sayang di antara kita selalu terjaga, dan Allah senantiasa melimpahkan keberkahan untuk keluarga besar ini

16. Kepada sepupu-sepupu penulis yang selalu menghadirkan canda tawa, semangat, dan rasa kebersamaan di setiap pertemuan. Terima kasih atas dukungan, kebersamaan, dan doa yang tak pernah luput, baik dalam suka maupun duka. Semoga ikatan persaudaraan ini selalu erat, dan kita semua senantiasa diberi kesehatan, rezeki yang berkah, serta kesuksesan dalam jalan masing-masing.

17. Kepada Tahira Khuwalidia Shabirah Ma’ruf, sosok yang selalu hadir di setiap fase perjuangan penulis. Terima kasih atas waktu, perhatian, dan dukungan yang tak pernah putus di saat senang maupun sulit. Kehadiranmu menjadi penyemangat tersendiri, pengingat agar tetap kuat dan fokus pada tujuan. Semoga segala kebaikan yang telah diberikan dibalas dengan keberkahan dan kebahagiaan yang melimpah.

18. Kepada teman-teman dekat penulis di kampung halaman, yang telah menjadi bagian dari perjalanan hidup sejak kecil, tempat berbagai cerita sederhana namun penuh makna. Terutama kepada Fatih, Nadia, Dimas dan Destia, yang selalu ada dalam suka dan duka, serta menjadi teman berbagi, bercanda, dan saling menguatkan. Terima kasih atas kebersamaan dan persahabatan yang tulus. Semoga ikatan ini tetap terjaga, dan kita semua dimudahkan dalam setiap langkah menuju masa depan.
19. Teman seperjuangan Suci, Adila, Chesaa, Hamidah dan Nurul yang sudah setia menemani, memberikan semangat, saran, dan dukungan selama proses penyelesaian skripsi ini. Kehadiran kalian, baik melalui diskusi, canda tawa, maupun dukungan moral, menjadi salah satu penguat dalam perjalanan ini.
20. Teman dekat penulis, Niya, Aulia, putri, Dika yang sudah menemani serta memberikan segala bantuan dari awal perkuliahan hingga saat ini.
21. Kepada teman kelas E, Fauzan, yudistira, abdil, rafi, zakin, furkon, faisal, aghata, parana, dsb. terima kasih atas kebersamaan selama ini—dari belajar, bukber, main badminton, hingga futsal bareng. Semoga silaturahmi ini tetap terjaga ke depannya.
22. Seluruh warga Teknik Informatika khususnya angkatan 2021 “Aster” yang telah memberikan kehangatan, motivasi, semangat, dan dukungan kepada penulis.
23. Teman-teman Himatif Encoder, Divisi Seni dan Olahraga ’22, yang telah memberikan semangat, dukungan, dan menjadi tempat penulis untuk mengembangkan *hardskill* maupun *softskill* selama masa perkuliahan.

24. Untuk diri sendiri, Muhammad Putra Zidannizar, yang telah melalui banyak proses jatuh, bangkit, lelah, namun tetap berjalan. Terima kasih telah bertahan sejauh ini, terus belajar, dan tidak menyerah meski kadang ingin berhenti. Semoga tetap konsisten dalam hal baik, terus berproses, dan tidak lupa tujuan awal. Ingat, segala sesuatu butuh waktu, dan selama terus melangkah, tidak ada usaha yang sia-sia. Percayalah, kerja keras dan doa tidak pernah mengkhianati hasil. Tidak ada kata terlambat untuk menciptakan kehidupan yang kamu inginkan.

Malang, 17 April 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Zidannizar, Muhammad Putra. 2025. **Klasifikasi Penyakit Serangan Jantung Menggunakan Algoritma *Support Vector Machine* (SVM) dan *Recursive Feature Elimination* (RFE)**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Muhammad Faisal, M.T (II) Dr. Yunifa Miftachul Arif, M.T.

Kata Kunci: *Normalisasi*, *Recursive Feature Elimination* (RFE), Serangan Jantung, *Synthetic Minority Oversampling Technique* (SMOTE), *Support Vector Machine* (SVM)

Penelitian ini membahas pentingnya analisis data medis yang kompleks dalam mendeteksi risiko serangan jantung. Tantangan utama dalam analisis data medis adalah ketidakseimbangan data dan kompleksitas atribut yang sering kali menyulitkan metode tradisional, sehingga berpotensi menyebabkan keterlambatan pengambilan keputusan medis. Untuk itu, penelitian ini mengusulkan pendekatan berbasis pembelajaran mesin menggunakan algoritma Support Vector Machine (SVM) yang dipadukan dengan teknik Recursive Feature Elimination (RFE), normalisasi, dan SMOTE untuk meningkatkan akurasi prediksi. Serangan jantung (infark miokard) merupakan kondisi serius yang sering disebabkan oleh aterosklerosis dan menjadi penyebab utama kematian global. Dengan pendekatan ini, diharapkan model prediksi yang dihasilkan dapat membantu tenaga medis dalam melakukan pencegahan dini secara lebih akurat dan efisien. Berdasarkan hasil pengujian, model terbaik diperoleh dengan kombinasi *kernel* RBF, Cost sebesar 10, dan Gamma sebesar 0.5 pada skenario data 90:10, menghasilkan akurasi 84.84%, presisi 82%, recall 88%, dan F1-Score 85%.

ABSTRACT

Zidannizar, Muhammad Putra. 2025. **Classification of Heart Attack Disease Using Support Vector Machine (SVM) and Recursive Feature Elimination (RFE) Algorithms.** Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University Malang. Advisor: (I) Dr. Muhammad Faisal, M.T (II) Dr. Yunifa Miftachul Arif, M.T.

Keywords: Normalization, Recursive Feature Elimination (RFE), Heart Attack, Synthetic Minority Oversampling Technique (SMOTE), Support Vector Machine (SVM).

This research discusses the importance of complex medical data analysis in detecting heart attack risk. The main challenges in medical data analysis are data imbalance and attribute complexity that often complicate traditional methods, potentially causing delays in medical decision-making. Therefore, this research proposes a machine learning-based approach using Support Vector Machine (SVM) algorithm combined with Recursive Feature Elimination (RFE), normalization, and SMOTE techniques to improve prediction accuracy. Heart attack (myocardial infarction) is a serious condition often caused by atherosclerosis and is the leading cause of global mortality. With this approach, it is expected that the resulting prediction model can help medical personnel in conducting early prevention more accurately and efficiently. Based on the test results, the best model is obtained with a combination of RBF *kernel*s.

الملخص

زيدا نزار، محمد بوترا. 2025. تصنيف أمراض نوبة القلب باستخدام خوارزمية آلة الدعم المتجه (SVM) وإزالة الميزات التكرارية (RFE). البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. محمد فيصل، الماجستير؛ المشرف الثاني: د. يونيفا ميفتاح العارف، الماجستير.

الكلمات المفتاحية: تطبيع، إزالة ميزات تكرارية، نوبة قلبية، تقنية زيادة عدد عينات أقلية اصطناعية، آلة الدعم متجه

تناول هذا البحث أهمية تحليل البيانات الطبية المعقدة في الكشف عن مخاطر نوبة القلب. التحدي الرئيسي في تحليل البيانات الطبية هو عدم توازن البيانات وتعقيد الخصائص الذي غالباً ما يصعب الطرق التقليدية، مما قد يؤدي إلى تأخير اتخاذ القرارات الطبية. ولذلك، اقترح هذا البحث منهجاً قائماً على التعلم الآلي باستخدام خوارزمية آلة الدعم المتجه (SVM) مدمجة بتقنية وإزالة الميزات التكرارية (RFE)، والتطبيع، وتقنية زيادة عدد عينات الأقلية الاصطناعية (SMOTE) لزيادة دقة التنبؤ. نوبة القلب (احتشاء العضلة القلبية) هي حالة خطيرة غالباً ما تكون ناتجة عن تصلب الشرايين وتسبب الوفاة على مستوى العالم. من المتوقع أنه من خلال هذا المنهج، يمكن أن يساعد النموذج التنبؤي الناتج في تمكين العاملين في مجال الطب من القيام بالوقاية المبكرة بدقة وكفاءة أكبر. استنادًا إلى نتائج الاختبار، تم الحصول على أفضل نموذج من خلال مزيج من نواة RBF، و Cost قدرها 10، و Gammaقدرها 0.5 في سيناريو البيانات 90:10، مما أدى إلى دقة 84.84%، وضبط 82%، واسترجاع 88%، وقيمة ف1 85%.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penyakit jantung merupakan istilah umum yang digunakan untuk menggambarkan gangguan terhadap fungsi kerja jantung. Penyakit jantung terdiri dari berbagai jenis, di antaranya penyakit kardiovaskular, jantung koroner, dan serangan jantung. Serangan jantung atau dalam medis bernama *Myocardial Infarction* atau *infark miokard* adalah gangguan jantung yang sangat serius(Amrullah et al., 2022). Yang dimana kondisi penurunan fungsi jantung yang menyebabkan terganggunya aliran darah dikarenakan pembuluh arteri mengalami penyempitan sehingga distribusi oksigen ke jaringan-jaringan tubuh tidak sesuai. Sebagaimana firman Allah pada surah *Al Haqqah* ayat 46:

ثُمَّ لَقَطَعْنَا مِنْهُ الْوَتِيْنَ

"Kemudian benar-benar kami potong urat tali jantungnya" (Al-Haqqah 46).

Pembuluh nadi, dalam hal ini merupakan pembuluh darah besar atau dalam dunia medis disebut aorta yang tekanannya langsung dari kontraksi jantung dan volume darah yang besar, tersumbat atau terpotongnya aorta dapat mengakibatkan syok bahkan kematian (Katsir, 2015). Ayat di atas menunjukkan pentingnya pemahaman tentang fungsi dan kerentanan jantung dalam konteks risiko serangan jantung. Dalam penelitian ini, ayat di atas dihubungkan dengan pemahaman akan akibat yang cukup serius dari gangguan pada jantung.

أَلَا وَإِنَّ فِي الْجَسَدِ مُضْغَةً، إِذَا صَلَحَتْ صَلَحَ الْجَسَدُ كُلُّهُ، وَإِذَا فَسَدَتْ فَسَدَ الْجَسَدُ كُلُّهُ، أَلَا وَهِيَ الْقَلْبُ

"Ingatlah, di dalam jasad terdapat segumpal daging. Apabila ia baik, maka baiklah seluruh jasadnya, dan apabila ia rusak, maka rusaklah seluruh jasadnya. Ketahuilah, segumpal daging itu adalah qalbu (hati/jantung)." (HR Bukhari No 52 dan Muslim No 1599).

Menurut Ibnu Katsir hubungan antara surah *Al-Haqqah* ayat 46, dan hadits riwayat Bukhari dan Muslim diatas dengan penelitian ini terletak pada penekanan terhadap pentingnya Kesehatan jantung. Surah *Al-Haqqah* ayat 46 menggambarkan konsekuensi fatal dari kerusakan jantung. Dalam Bahasa Arab jantung disebut sebagai '*Al Qalbu*' seperti pada hadist di atas yang menekankan peran sentral jantung dalam kesehatan tubuh secara keseluruhan.

Di seluruh dunia, penyakit jantung menjadi penyebab utama kematian, menyerang siapa saja tanpa memandang usia, jenis kelamin, atau latar belakang. Menurut data dari Organisasi Kesehatan Dunia (WHO), sekitar 17 juta kematian terjadi setiap tahunnya yang disebabkan oleh penyakit serangan jantung, dengan penyakit jantung dan stroke berkontribusi pada sekitar 80% dari jumlah tersebut. Dan serangan jantung menjadi penyebab fatal bagi 7,3 juta nyawa di berbagai belahan dunia (Fais et al., 2023). Pergeseran pola penyakit dibuktikan dengan data statistik laporan *World Health Statistic* 2010, (Saktiningtyastuti & Astuti, 2017). Penyakit tidak menular merupakan penyebab utama 58 juta kematian di dunia dan *presentase* penyakit jantung dan pembuluh darah sebanyak 30% dari total penyakit tidak menular lainnya.

Berdasarkan laporan *World Health Statistic* 2018, tercatat 17,1 juta orang meninggal dunia akibat penyakit jantung coroner dan diperkirakan angka ini akan meningkat terus hingga tahun 2030 ,menjadi 23,4 juta kematian di dunia(Sartika &

Pujiastuti, 2020). Berdasarkan data Riset Kesehatan Dasar (Riskesdas) tahun 2018, angka kejadian penyakit jantung semakin meningkat dari tahun ke tahun. Penyakit ini berkontribusi pada sepertiga dari semua kematian di dunia, dan dapat dicegah jika dideteksi dan ditangani secara dini. Faktor risiko utama yang memicu serangan jantung meliputi gaya hidup yang tidak sehat seperti merokok, pola makan yang buruk, obesitas, kurangnya aktivitas fisik, dan konsumsi alkohol yang berlebihan (Rahayu et al., 2020).

Kebiasaan tersebut telah menjadi bagian dari gaya hidup sebagian masyarakat seiring dengan perubahan pola hidup. Karena itu, tidak mengherankan jika banyak pasien dengan penyakit kardiovaskular mengalami serangan jantung berulang. Penelitian terbaru oleh Triyanta (2011) menyatakan bahwa kualitas tidur yang buruk dapat menjadi pemicu terjadinya serangan jantung(Saktiningtyastuti & Astuti,2017).

Dalam Islam, pentingnya menjaga kesehatan sangat ditekankan. Salah satu cara yang diajarkan dalam Al-Qur'an adalah dengan menjaga kualitas tidur Hal ini tercermin dalam Surah *Al-Furqan* (25;47), yang menyebutkan:

وَهُوَ الَّذِي جَعَلَ لَكُمُ اللَّيْلَ لِبَاسًا وَالنَّوْمَ سُبَاتًا وَجَعَلَ النَّهَارَ نُشُورًا

"Dan Dialah yang menjadikan malam bagimu (sebagai) pakaian, dan tidur untuk istirahat, dan Dia menjadikan siang untuk bangun berusaha." (Q.S Al Furqon:47).

Dari ayat di atas dapat dipahami bahwa Allah telah menciptakan malam untuk tidur sebagai bentuk pemulihan tubuh, yang penting untuk menjaga kesehatan jantung. Tidur yang berkualitas menjadi bagian penting dalam menjaga kesehatan jantung dan mencegah serangan jantung. Mengabaikan pentingnya istirahat dan keseimbangan hidup, seperti kurang tidur, dapat meningkatkan risiko

serangan jantung, sehingga ayat ini dapat menjadi pengingat akan pentingnya menjaga keseimbangan hidup untuk kesehatan jantung kita.

Kementerian Kesehatan Republik Indonesia melakukan berbagai upaya untuk mencegah penyakit jantung, seperti penguatan layanan primer melalui edukasi masyarakat, perluasan cakupan imunisasi rutin menjadi 14 antigen, serta skrining 14 penyakit penyebab kematian tertinggi, termasuk penyakit jantung, pada tiap kelompok usia. Tercatat peningkatan prevalensi serangan jantung sebesar 2% per tahun pada usia di bawah 40 tahun sejak 2000 hingga 2016, sehingga peningkatan kapasitas dan kapabilitas layanan primer juga terus didorong(Kemenkes, 2022a).

Meskipun berbagai upaya telah dilakukan untuk mencegah dan menangani serangan jantung, salah satu masalah utama yang dihadapi adalah kurangnya metode prediksi yang akurat dan efisien untuk mendeteksi risiko serangan jantung secara dini. Data medis yang tersedia sering kali bersifat kompleks dan tidak seimbang, sehingga metode tradisional sulit memberikan hasil prediksi yang optimal. Hal ini dapat mengakibatkan keterlambatan dalam pengambilan keputusan medis dan meningkatnya risiko fatal bagi pasien. Oleh karena itu, diperlukan pendekatan baru yang mampu menganalisis data secara efisien, memberikan prediksi yang akurat, dan membantu tenaga medis dalam upaya pencegahan dini serangan jantung(Dimsyiar M Al Hafiz et al., 2021).

Dalam beberapa dekade terakhir, kemajuan teknologi, terutama dalam bidang kecerdasan buatan (*Artificial Intelligence*) dan pembelajaran mesin (*Machine Learning*), telah membuka peluang baru untuk meningkatkan akurasi dan

kecepatan dalam diagnosis penyakit, termasuk serangan jantung. *Machine learning* telah terbukti menjadi alat yang sangat efisien untuk menangani berbagai tugas klasifikasi medis yang kompleks. Salah satu algoritma pembelajaran mesin yang banyak digunakan dalam klasifikasi adalah svm (Maisat et al., 2023). Klasifikasi adalah proses menemukan sekumpulan pola atau fungsi-fungsi yang mendeskripsikan dan memisahkan kelas data tertentu lainnya, dan digunakan untuk memprediksi data(Setio et al., 2020).

SVM merupakan metode klasifikasi yang mampu memisahkan data ke dalam dua kelas berdasarkan pemilihan *hyperplane* terbaik di antara dua set data. SVM memiliki keunggulan dalam hal kecepatan komputasi dan kemampuannya untuk menangani data dengan dimensi yang tinggi, menjadikannya alat yang efektif dalam memecahkan masalah klasifikasi yang sulit. Dalam konteks penyakit jantung, penerapan SVM dapat membantu dalam menganalisis data kesehatan pasien untuk mengklasifikasi kemungkinan serangan jantung secara akurat.

Penelitian sebelumnya telah menunjukkan bahwa SVM unggul dalam melakukan klasifikasi, terutama pada data yang memiliki banyak fitur dan atribut. Misalnya, Penelitian lain menggunakan SVM digunakan untuk memprediksi bencana banjir(Dwiasnati & Devianto, 2021), tingkat akurasi yang dihasilkan dengan menggunakan *alogaritma* SVM dengan optimasi *Particle Swarm Optimization* (PSO) sebesar 97,62%.

Namun, performa SVM dalam tugas klasifikasi sangat bergantung pada pemilihan *kernel* dan parameter yang tepat. Beberapa jenis *kernel* yang umum digunakan dalam SVM adalah linear, *Radial Basis Function* (RBF), *polynomial*,

dan *sigmoid*. Selain itu, parameter seperti nilai C (*regularization*) dan gamma juga harus dioptimalkan untuk menghasilkan model prediksi yang akurat. Oleh karena itu, penting untuk melakukan analisis mendalam terhadap *kernel* dan parameter SVM yang digunakan untuk menentukan kombinasi terbaik.

Selain itu, keseimbangan data dalam dataset medis sering kali menjadi tantangan. Jumlah kasus positif (misalnya, pasien dengan serangan jantung) biasanya lebih sedikit dibandingkan kasus negatif. Ketidakseimbangan ini dapat mengurangi akurasi model prediksi. Untuk mengatasi masalah ini, teknik SMOTE (*Synthetic Minority Over-sampling Technique*) sering digunakan untuk menyeimbangkan distribusi data sebelum diolah oleh model SVM.

Penelitian ini berfokus pada pengembangan model klasifikasi penyakit serangan jantung menggunakan metode SVM dengan tujuan untuk menentukan model dan performa terbaik. Dataset yang digunakan dalam penelitian ini diambil dari *platform Kaggle*, yang berisi data pasien dengan berbagai atribut kesehatan terkait risiko serangan jantung. Dataset ini akan dibagi menjadi data latih dan data uji untuk menguji performa model dengan *kernel* dan parameter. Penelitian ini diharapkan dapat membantu dalam pengembangan model prediksi serangan jantung yang lebih akurat dan efisien, dengan hasil prediksi yang diharapkan mencapai tingkat akurasi yang optimal.

Dengan adanya penelitian ini, diharapkan metode SVM yang dipadukan dengan teknik EDA, RFE, SMOTE dan normalisasi dapat menghasilkan model yang mampu memprediksi serangan jantung secara akurat, sehingga dapat

berkontribusi dalam upaya pencegahan dini penyakit serangan jantung serta memberikan solusi praktis dalam diagnosis penyakit serangan jantung.

1.2 Rumusan Masalah

Bagaimana performa metode *Support Vector Machine* dalam mengaklasifikasi populasi yang berisiko serangan jantung?

1.3 Batasan Masalah

a. Menggunakan *Kernel* Liniear, *Kernel* RBF dan parameter C dan Gamma.
b. Memanfaatkan data dari kaggle yaitu *Heart Attack Analysis dan Prediction Dataset.*(https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset)
c. Menggunaakan seluruh atribut pada *Heart Attack Analysis dan Prediction Dataset.*
d. Data dibagi menjadi 3 model untuk pengujian.
e. Menggunakan bahasa pemrograman *phyton.*
f. Mengggunakan *software Google Colab*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini untuk mengetahui model serta peforma yang terbaik dalam melakukan klasifikasi berdasarkan data *Heart Attack Analysis dan Prediction Dataset*.

1.5 Manfaat Penelitian

a. Penelitian ini diharapkan dapat memberikan manfaat untuk mengklasifikasikan risiko serangan jantung. Hal ini dapat membantu upaya mencegah risiko serangan jantung.
b. Memberikan wawasan tentang metode *Support Vector Machine*.
c. Hasil dari penelitian ini dapat digunakan sebagai landasan pada penelitian selanjutnya.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terdahulu

Dalam penelitian (Alhabib, 2022) membahas tentang komparasi metode *Deep Learning, Naïve Bayes,* dan *Random Forest*(RB) dalam prediksi serangan jantung. Data yang digunakan pada penelitian ini adalah dataset publik dari Kaggle dengan judul *"Heart Attack Analysis & Prediction Dataset"*, yang terdiri dari 303 data dengan 14 atribut seperti umur, jenis kelamin, tekanan darah, kolesterol, dan atribut lainnya. Dataset tersebut dibagi menjadi data latih dan data uji, dengan teknik validasi yang menggunakan cross-validation. Hasil penelitian menunjukkan bahwa algoritma Deep Learning memberikan akurasi tertinggi sebesar 83,49%, diikuti oleh Naïve Bayes sebesar 81,84%, dan Random Forest sebesar 80,18%.

Dalam penelitian yang dilakukan oleh (Marthin Luter Laia & Yudi Setyawan, 2020) membahas tentang perbandingan dua metode, yaitu *Support Vector Machine* (SVM) dan *Naïve Bayes Classifier* (NBC), dalam memprediksi curah hujan. Tahapan preprocessing data meliputi penanganan missing value menggunakan metode interpolasi linier. Selanjutnya, dataset dibagi menjadi 90% untuk data training dan 10% untuk data *testing*. Pada metode SVM, *kernel* yang digunakan adalah *kernel* linier dengan berbagai nilai parameter C, yaitu 0,01; 0,1; 1; 10; dan 100. Berdasarkan hasil analisis, metode SVM menghasilkan akurasi sebesar 79,45%, sedangkan metode NBC menghasilkan akurasi sebesar 65,75%. Hal ini menunjukkan bahwa metode SVM memiliki kinerja yang lebih baik

dibandingkan dengan NBC dalam memprediksi curah hujan pada dataset yang digunakan.

Penelitian (Naufal et al., 2023) membahas perbandingan algoritma SVM, RF, dan NB untuk klasifikasi teks *cyberbullying* di media sosial. Tahapan preprocessing meliputi case folding, eliminasi stopword dan tanda baca, penghilangan karakter berulang, tokenisasi, *stemming*, serta pembobotan menggunakan TF-IDF. Pengujian akurasi menggunakan metode 10-*fold cross-validation*. Hasil penelitian menunjukkan bahwa algoritma SVM mencapai akurasi tertinggi sebesar 83%, diikuti *Random Forest* (RF) dengan akurasi 82%, dan *Naïve Bayes* (NB) dengan akurasi 74%. Dari hasil ini, dapat disimpulkan bahwa SVM memiliki performa terbaik dalam klasifikasi *cyberbullying* dibandingkan algoritma lainnya.

Dalam penelitian (Abdurrahman, 2023) menunjukkan bahwa penerapan metode SVM dengan tiga *kernel* berbeda untuk klasifikasi kanker payudara memberikan hasil yang bervariasi, di mana *kernel* Linear menghasilkan akurasi sebesar 99% (terbaik), diikuti *kernel* RBF dengan akurasi 92%, dan *kernel* Sigmoid dengan akurasi 41%. Dalam tahap *preprocessing* meliputi imputasi data yang hilang menggunakan *mean*, *imputing* dan *encoding* data kategorikal menjadi numerik dengan *label encoder*.

Dalam penelitian (Oryza Habibie Rahman et al., 2021) menunjukkan bahwa penerapan metode SVM untuk klasifikasi ujaran kebencian memberikan hasil terbaik dengan penggunaan *kernel* RBF menghasilkan akurasi sebesar 93%, diikuti *kernel* Linear dengan akurasi 92%, dan *kernel* Sigmoid dengan akurasi 92%. Dalam

tahap *prapocessing* yang meliputi *case folding*, *filtering*, *tokenizing*, *stemming*, dan pembobotan menggunakan TF-IDF.

Penelitian (Arif et al., 2024) membahas perbandingan algoritma SVM dan KNN untuk klasifikasi penyakit serangan jantung. Dataset yang digunakan adalah *"Heart Attack Analysis & Prediction"* dari Kaggle, yang terdiri dari 303 baris dengan 14 fitur. Tahapan preprocessing meliputi pengecekan *missing value, normalisasi* menggunakan *Z-Score*, serta seleksi fitur menggunakan PCA untuk mengurangi dimensi data. Pengujian akurasi dilakukan menggunakan *confusion matrix* dan *Receiver Operating Characteristic* (ROC). Hasil penelitian menunjukkan bahwa algoritma KNN mencapai akurasi tertinggi sebesar 90,16%, diikuti SVM dengan akurasi 86,88%. Dari hasil ini, dapat disimpulkan bahwa KNN memiliki performa terbaik dalam klasifikasi penyakit serangan jantung dibandingkan SVM.

Penelitian (Pradana et al., 2022) membahas perbandingan algoritma SVM dan NB untuk klasifikasi peluang penyakit serangan jantung. Dataset yang digunakan adalah *"Heart Attack Analysis & Prediction"* dari Kaggle. Tahapan preprocessing meliputi pembagian data latih dan uji dengan dua skenario: 20% data untuk pelatihan pada skenario pertama dan 40% pada skenario kedua. Hasil penelitian menunjukkan bahwa algoritma SVM mencapai akurasi terbaik sebesar 87% pada skenario pertama, sedangkan algoritma *Naïve Bayes* memperoleh akurasi terbaik sebesar 84% pada skenario yang sama. Dari hasil ini, dapat disimpulkan bahwa SVM memiliki performa lebih baik dibandingkan dengan Naïve Bayes.

Penelitian (Maisat et al., 2023) membahas penerapan metode SVM dengan optimasi hyperparameter menggunakan *GridSearchCV* untuk memprediksi kemungkinan serangan jantung. Dataset yang digunakan adalah *Heart Attack Analysis & Prediction Dataset*. Tahapan preprocessing meliputi pemeriksaan data untuk mendeteksi baris duplikat atau nilai kosong (missing values), normalisasi data untuk mengurangi redundansi dan memastikan kualitas data, serta pemisahan data menjadi tiga bagian: training data (70%), test data (15%), dan validation data (15%). Hasil penelitian menunjukkan bahwa model yang dioptimalkan menghasilkan akurasi sebesar 86,0%, *precision* sebesar 84,0%, *recall* sebesar 91,0%, dan *F1-score* sebesar 87,0%. Dari hasil ini, dapat disimpulkan bahwa SVM dengan optimkoppasi *Hyperparameter* memberikan performa yang cukup baik dalam klasifikasi serangan jantung.

Penelitian (Putranto et al., 2023) membahas sistem prediksi penyakit jantung berbasis web menggunakan algoritma SVM. Dataset yang digunakan terdiri dari 918 data dengan 12 atribut, diambil dari UCI Machine Learning Repository. Tahapan preprocessing meliputi handling outlier menggunakan Z-score, pengecekan korelasi antar variabel, dan handling imbalanced data dengan metode SMOTE.

Model SVM dikembangkan dengan *hyperparameter* tuning menggunakan GridSearchCV, menghasilkan nilai parameter optimal berupa gamma sebesar 0,01, C sebesar 0,1, dan *kernel* polynomial. Hasil pengujian menunjukkan akurasi model sebesar 85%, precision 78%, dan recall 89%. Model ini kemudian

diimplementasikan ke dalam sistem berbasis web menggunakan *framework Streamlit*.

Tabel 2. 1 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian
1	(Alhabib, 2022)	Komparasi Metode Deep Learning, Naïve Bayes dan Random Forest untuk Prediksi Penyakit Jantung	*Deep Learning, Naïve Bayes dan Random Forest)*	algoritma Deep Learning memberikan akurasi tertinggi sebesar 83,49%, Naïve Bayes sebesar 81,84%, dan Random Forest sebesar 80,18%.
2	(Marthin Luter Laia & Yudi Setyawan, 2020)	PERBANDINGAN HASIL KLASIFIKASI CURAH HUJAN MENGGUNAKAN METODE SVM DAN NBC	*Support Vector Machine Dan Naïve Bayes*	metode SVM menghasilkan akurasi sebesar 79,45%, sedangkan metode Naïve Bayes Classifier (NBC) menghasilkan akurasi sebesar 65,75%.
3	(Naufal et al., 2023)	Analisis Perbandingan Tingkat Performa Algoritma SVM, Random Forest, dan Naïve Bayes untuk Klasifikasi Cyberbullying pada Media Sosial	*SVM, Random Forest, dan Naïve Bayes*	algoritma SVM mencapai akurasi tertinggi sebesar 83%, diikuti Random Forest (RF) dengan akurasi 82%, dan Naïve Bayes (NB) dengan akurasi 74%.
4	(Abdurrahman, 2023)	Klasifikasi Kanker Payudara Menggunakan Algoritma SVM dengan *Kernel* RBF, Linier, dan Sigmoid	*Support Vector Machine*	*kernel* Linear menghasilkan akurasi sebesar 99% (terbaik), *kernel* RBF dengan akurasi 92%, dan *kernel* Sigmoid dengan akurasi 41%;
5	(Oryza Habibie Rahman et al., 2021)	Klasifikasi Ujaran Kebencian pada Media Sosial Twitter Menggunakan Support Vector Machine	*Support Vector Machine*	*kernel* RBF, menghasilkan akurasi sebesar 93%, *kernel* Linear dengan akurasi 92%, dan *kernel* Sigmoid dengan akurasi 92%.
6	Arif et al., 2024	Klasifikasi Penyakit Serangan Jantung Menggunakan Metode Machine Learning *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM)	*K-Nearest Neighbors* (KNN) *dan Support Vector Machine* (SVM)	KNN mencapai akurasi tertinggi sebesar 90,16%, diikuti SVM dengan akurasi 86,88%
7	Pradana et al., 2022	KOMPARASI METODE *SUPPORT VECTOR MACHINE* DAN *NAÏVE BAYES* DALAM	*SUPPORT VECTOR MACHINE*	SVM mencapai akurasi terbaik sebesar 87% pada skenario pertama, sedangkan

No	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian
		KLASIFIKASI PELUANG PENAKIT SERANGAN JANTUNG	*DAN NAÏVE BAYES*	algoritma *Naïve Bayes* memperoleh akurasi terbaik sebesar 84% pada skenario yang sama
8	Maisat et al., 2023	Implementasi Optimasi Hyperparameter GridSearchCV Pada Sistem Prediksi Serangan Jantung Menggunakan SVM	*SUPPORT VECTOR MACHINE*	Hasil penelitian menunjukkan bahwa model yang dioptimalkan menghasilkan akurasi sebesar 86,0%, precision sebesar 84,0%, recall sebesar 91,0%, dan F1-score sebesar 87,0%.
9	Putranto et al., 2023	Sistem Prediksi Penyakit Jantung Berbasis Web Menggunakan Metode SVM dan Framework Streamlit	*SUPPORT VECTOR MACHINE*	Hasil pengujian karnel polymonial menunjukkan akurasi model sebesar 85%, precision 78%, dan recall 89%.

2.2 Klasifikasi

Klasifikasi adalah metode analisis data yang bertujuan untuk membangun model yang dapat mengelompokkan data ke dalam kategori tertentu. Klasifikasi adalah proses membangun model atau fungsi yang merepresentasikan dan membedakan kelas-kelas data, dengan tujuan untuk memprediksi kelas dari objek yang belum diketahui labelnya (Annur, 2018). Terdapat dua pendekatan utama dalam klasifikasi, yaitu *supervised* dan *unsupervised*. Dalam klasifikasi *supervised*, seperti yang diterapkan dalam penelitian ini, data sudah memiliki label atau kategori yang diketahui, sehingga model dapat dilatih menggunakan informasi tersebut (Darujati et al., 2012). Pendekatan ini sangat cocok untuk klasifikasi penyakit seperti penyakit serangan jantung, di mana kategori atau kondisi kesehatan pasien sudah ditentukan sebelumnya. Proses klasifikasi secara umum terdiri dari beberapa tahapan, yakni:

1. ***Pre-processing***: Tahapan ini bertujuan untuk membersihkan data sehingga dapat diolah lebih lanjut. Misalnya, dalam klasifikasi penyakit, data yang digunakan dapat berupa data pasien yang perlu dinormalisasi atau diubah ke dalam bentuk yang lebih mudah diolah oleh model.
2. ***Feature Extraction/Selection***: Proses ini penting untuk memilih fitur-fitur yang relevan dari data yang akan digunakan oleh model klasifikasi. Dalam kasus penyakit serangan jantung ini semua fiture digunakan dalam penelitian ini.
3. **Pemilihan Model**: Algoritma pembelajaran mesin, *Support Vector Machine* (SVM), dipilih untuk membangun model klasifikasi. SVM dikenal sangat efektif dalam memisahkan kelas data yang bersifat kompleks dan berfungsi dengan baik untuk masalah klasifikasi biner maupun multi-kelas.
4. ***Training*** **dan** ***Testing***: Model dilatih menggunakan data yang telah disiapkan, dan kemudian diuji untuk menilai performanya. Pada penelitian ini, *Support Vector Machine* (SVM) menggunakan dua jenis *kernel*, yaitu Linear *Kernel* dan RBF (*Radial Basis Function*) *Kernel*, yang dipilih untuk meningkatkan kemampuan model dalam memprediksi penyakit jantung. Linear *Kernel* efektif untuk data yang dapat dipisahkan secara linier, sementara RBF *Kernel* cocok untuk data dengan pola yang lebih kompleks dan non-linear. Kedua *kernel* ini digunakan untuk mendapatkan hasil prediksi yang lebih akurat.

2.3 Penyakit Serangan Jantung

Penyumbatan arteri koroner terjadi ketika aliran darah yang menuju otot jantung terhambat, sehingga dapat memicu serangan jantung. Serangan jantung dikenal secara ilmiah sebagai *myocardial infarction*. Penyebab utama dari kondisi ini sering kali adalah *aterosklerosis*, yaitu *Aterosklerosis* merupakan penyempitan pembuluh darah yang diakibatkan oleh penumpukan plak kolesterol pada dinding pembuluh darah tersebut(Kemenkes, 2022b). serangan jantung, karena terjadi *iskemia miokard* atau kekurangan oksigen pada otot jantung, yaitu jika mengeluhkan adanya nyeri dada atau nyeri hebat di ulu hati (*epigastrium*) yang bukan disebabkan oleh trauma, terjadi pada laki-laki berusia 35 tahun atau perempuan berusia di atas 40 tahun(Supriyono, 2008).

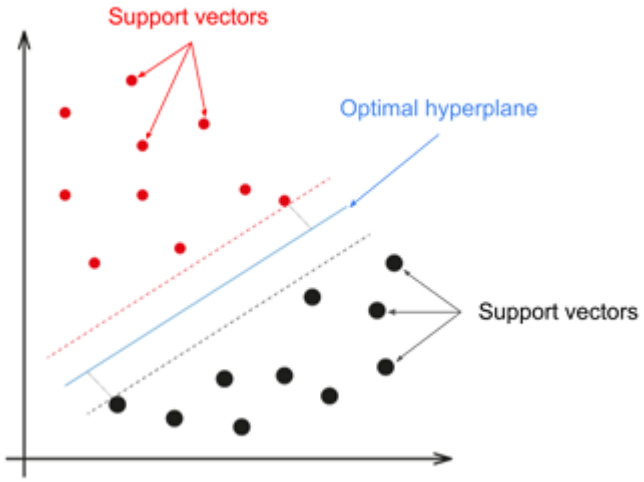
Beberapa faktor risiko dapat meningkatkan peluang terjadinya serangan jantung, antara lain riwayat keluarga dengan penyakit jantung, kebiasaan merokok, hipertensi, diabetes, kadar kolesterol yang tinggi, obesitas, kurangnya aktivitas fisik, serta pola makan yang tidak sehat (Tamlander et al., 2022). Selain itu, faktor stres juga dapat memperburuk kondisi kesehatan kardiovaskular, sehingga meningkatkan risiko serangan jantung. Deteksi dini sangat penting dalam menangani serangan jantung, dan beberapa tindakan medis dapat dilakukan untuk menyelamatkan nyawa, seperti *angioplasti*, yaitu prosedur untuk membuka arteri yang tersumbat menggunakan balon atau stent untuk memulihkan aliran darah (Rohayati, 2020). Selain itu, terapi farmakologis dengan pemberian obat-obatan seperti aspirin atau pengencer darah sering digunakan untuk mengurangi dampak serangan jantung.

Pencegahan serangan jantung melibatkan perubahan gaya hidup yang lebih sehat, termasuk berhenti merokok, menjaga pola makan yang seimbang, rutin berolahraga, dan mengelola stres. Pengendalian faktor risiko melalui pengobatan juga penting untuk mencegah serangan jantung di masa depan. Upaya-upaya ini dapat mengurangi kemungkinan terjadinya serangan jantung berulang serta meningkatkan kualitas hidup pasien.

2.4 *Support Vector Machine* (SVM)

Metode *Support Vector Machine* (SVM) pertama kali dipublikasikan pada tahun 1992 oleh *Vapnik* bersama *Bernhard Boser* dan *Isabelle Guyon*. Penggunaan SVM telah terbukti mampu meningkatkan akurasi dalam proses klasifikasi (Santoso et al., 2022). SVM merupakan metode yang digunakan untuk klasifikasi dan regresi, dan berdasarkan pada teori pembelajaran mesin yang bertujuan untuk meningkatkan akurasi prediksi. SVM juga secara otomatis mencegah *overfitting* atau kecocokan data yang berlebihan.

SVM bekerja dengan prinsip *Structural Risk Minimization* (SRM), yang bertujuan untuk menemukan fungsi pemisah (*classifier hyperplane*) terbaik yang dapat memisahkan dua kelas pada ruang input (Medyanti & Faisal, 2023). *Hyperplane* ini berfungsi sebagai garis pembatas antara kelas-kelas yang berbeda, yang dihasilkan sebagai *output* dari proses klasifikasi. Dengan bantuan *hyperplane*, SVM mengelompokkan data ke dalam kelas yang sesuai berdasarkan karakteristiknya (Rahmi et al., 2024). Algoritma ini sangat efektif dalam meningkatkan akurasi klasifikasi melalui pemisahan data yang optimal di ruang *input*.



Gambar 2. 1 SVM Hyperplane

Dalam konteks ini, dengan menerapkan ukuran baru, SVM dapat memperoleh *hyperplane optimal* yang berfungsi untuk mengklasifikasikan data dari dua kelas, baik secara *linier* maupun melalui pemetaan *nonlinier* ke dimensi yang lebih tinggi. Dengan cara ini, data dari kedua kelas dapat dipisahkan menggunakan *hyperplane* SVM berusaha untuk menemukan *hyperplane* terbaik, yaitu *hyperplane* yang terletak di antara dua kumpulan objek yang masing-masing berasal dari kelas yang berbeda (Ritonga & Purwaningsih, 2018).

Proses untuk menemukan *hyperplane* menggunakan SVM, seperti yang diuraikan oleh Campbell (Sirait et al., 2019) sebagai berikut:

1. Terdapat data $\vec{X}_1 \in (\vec{X}_1, \vec{X}_2, \dots \vec{X}_n)$ dengan $\vec{X}_n$ merupakan data yang terdiri dari *M* atribut dan kelas target $y_i \in +1, -1$
2. Asumsikan bahwa kelas +1 dan -1 dapat dipisahkan oleh hyperplane, didefinisikan sebagai persamaan (2.1) :

$$\vec{w}.\vec{x} + b = 0 \qquad (2.1)$$

Dari persamaan (2.1) diperoleh :

$$\vec{w}.\vec{x} + b \geq 1, \text{ untuk kelas +1} \quad (2.2)$$

$$\vec{w}.\vec{x} + b \leq 1, \text{ untuk kelas -1} \quad (2.3)$$

3. SVM bertujuan untuk menemukan hyperplane pemisah dengan memaksimalkan jarak antara dua kelas. Memaksimalkan jarak dengan mencari titik minimal persamaan:

$$min_w \frac{1}{2}\left|\left|\vec{w}\right|\right|^2 \quad (2.4)$$

Dengan kendala

$$y_i(\vec{x}_i.\vec{w} + b) - 1 \geq 0 \quad (2.5)$$

Keterangan:
y_i : target ke-*i*
$\vec{w}$: bidang normal
$\vec{x}_i$: data input
b : posisi bidang relatif terhadap pusat koordinat

4. Membuat fungsi keputusan untuk persamaan (2.6) dan (2.7):

$$\text{Linear : } f\left(\vec{X}_d\right) = sign(\sum_{i=1}^{n} \alpha_i\, y_i(\vec{x}_i\vec{x}_d) + b) \quad (2.6)$$

$$\text{Non Linear : } f\left(\vec{X}_d\right) = sign(\sum_{i=1}^{n} \alpha_i\, y_i K(\vec{x}_i\vec{x}_d) + b) \quad (2.7)$$

Keterangan:
n : jumlah support vector
$\vec{x}_i$: support vector ke-*i*
$\vec{x}_d$: data uji
α : bobot dari setiap vector, yang dihitung saat optimasi
y_i : label ke-i
b : bias, yang dihitung saat optimasi

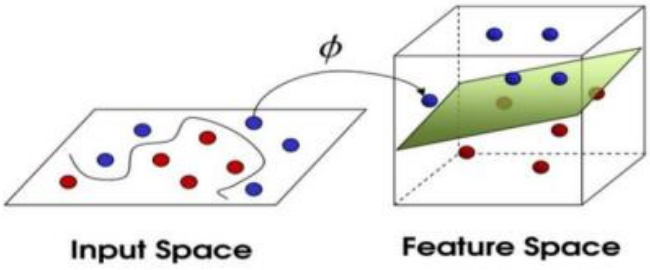
2.5 *Synthetic Minority Oversampling Technique* (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) adalah metode untuk menyeimbangkan distribusi data pada kelas minoritas dengan cara menyeleksi jumlah data sampel pada kelas hingga seimbang dengan sampel pada kelas mayoritas (Kasanah et al., 2019). Teknik ini bekerja dengan mengidentifikasi

data mayoritas yang paling dekat dengan data minoritas, lalu menghasilkan data baru berupa sampel sintetis dari kelas minoritas. Teknik ini bermanfaat dalam mengatasi tidakseimbangan dalam klasifikasi data dan memperbaiki performa model klasifikasi.

2.6 Fungsi *Kernel*

Saat terdapat masalah data yang tidak dapat dipisahkan secara linear dalam ruang input, soft margin SVM tidak dapat menemukan *hyperplane* yang kuat untuk meminimalkan misklasifikasi data dan menghasilkan generalisasi yang baik. Oleh karena itu, *kernel* berfungsi untuk mentransformasikan data ke dalam ruang berdimensi lebih tinggi, yang dikenal sebagai ruang *kernel*. Keberhasilan SVM tidak hanya bergantung pada struktur dasar algoritma, tetapi juga pada kemampuan untuk mengatasi berbagai jenis data melalui penggunaan *kernel*.



Gambar 2. 2 Fungsi *Kernel*

Dalam pendekatan ini, data direpresentasikan dalam bentuk *kernel* yang mengukur tingkat kesamaan atau perbedaan antar objek data. *Kernel* dapat dirancang untuk berbagai jenis data, baik data kontinu maupun diskrit, serta digunakan dalam struktur data seperti urutan dan grafik. Konsep substitusi *kernel* juga diterapkan dalam metode lain untuk analisis data, namun SVM menjadi yang

paling terkenal di antara metode berbasis *kernel* dengan aplikasi yang luas. SVM menggunakan *kernel* untuk merepresentasikan data, sehingga disebut sebagai metode berbasis *kernel*.

2.6.1 *Kernel* Linear

Linear *kernel* adalah fungsi *kernel* yang paling sederhana. Linear *kernel* digunakan ketika data yang di analisis sudah dapat dipisahkan secara linear. *Kernel* ini sangat cocok digunakan saat terdapat banyak fitur, karena pemetaan ke ruang berdimensi lebih tinggi tidak signifikan meningkatkan kinerja, seperti yang terjadi dalam klasifikasi teks. Pada klasifikasi teks, baik jumlah instance (dokumen) maupun jumlah fitur (kata) biasanya sangat besar. Ini adalah salah satu *kernel* yang paling umum digunakan dan terbukti menjadi fungsi terbaik ketika ada banyak fitur (Hamzah & Othman, 2021). Secara matematis, *kernel* linear didefinisikan dengan persamaan:

$$k(x, y) = x.y \tag{2.9}$$

Keterangan:
x dan y : vektor data

2.6.2 *Kernel Radial Basis Function* (RBF)

RBF *kernel* adalah fungsi *kernel* yang umum digunakan saat data tidak dapat dipisahkan secara linear. RBF *kernel* memiliki dua parameter utama, yaitu *Gamma* dan *Cost* (C). Parameter *Cost* (C) berfungsi sebagai mekanisme optimasi SVM untuk mengurangi kesalahan klasifikasi di setiap sampel pada dataset pelatihan. Sementara itu, parameter *Gamma* mengatur seberapa jauh pengaruh satu sampel data pelatihan terhadap sampel lain. Nilai gamma yang rendah menunjukkan bahwa

pengaruh sampel meluas atau "jauh", sedangkan nilai gamma yang tinggi menunjukkan pengaruh yang lebih terbatas atau "dekat." Dengan gamma rendah, titik-titik yang jauh dari garis pemisah masih akan dipertimbangkan dalam penentuan posisi garis tersebut. Sebaliknya, dengan gamma tinggi, hanya titik-titik yang dekat dengan garis pemisah yang akan diperhitungkan dalam pembuatan model. Persamaan 2.10 yang digunakan dalam RBF *kernel*.

$$\mathrm{K}(x, y) = exp[-\gamma \|x - \gamma\|]^2 \quad (2.10)$$

Keterangan:
x dan y : vektor data
exp : fungsi eksponensial
γ : parameter skalar

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Pengumpulan Data

Pada tahapan ini peneliti melakukan pengumpulan data menggunakan data sekunder. Data yang digunakan dalam penelitian ini yaitu dataset "Heart Attack Analysis & Prediction Dataset" yang merupakan open access data web dari Kaggle. Total data yang didapat berjumlah 303 data dengan 14 atribut seperti pada tabel penjelasan dibawah. Pada penelitian ini digunakan 13 fitur karena 1 atributnya yaitu Num yang merupakan variabel output.

Tabel 3. 1 Detail Atribut Dataset

No.	Fitur	Keterangan
1.	*Age*	Usia pasien
2.	*Sex*	Jenis kelamin pasien
3.	CP – *chest pain*	Jenis nyeri dada yang diderita pasien. Dimana terdapat 4 nilai yaitu: Tipe 1: *typical angina* Tipe 2: *atypical angina* Tipe 3: *non-anginal pain* Tipe 4: *asymptomatic*
4.	Trestbps – *resting blood pressure*	Tekanan darah pasien Ketika dalam keadaan istirahat menggunakan satuan mg/dl
5.	*chol*	Kadar kolesterol dalam darah pasien menggunakan satuan mg/dl
6.	FBS – *fasting blood sugar*	Kadar gula darah pasien, dengan 2 nilai, dimana 1 jika kadar gula darah pasien lebih dari 120 mg/dl, dan 0 jika gula darah pasien kurang dari atau sama dengan 120 mg/dl.
7.	*Restecg*	Kondisi EKG pasien Ketika dalam keadaan istirahat dengan 3 nilai, dimana 0 untuk keadaan normal, 1 untuk ST-T *wave abnormality* yaitu keadaan gelombang inversions T atau ST meningkat maupun menurun lebih dari 0,5 mV, dan nilai 2 untuk keadaan ventricular kiri mengalami hipertropi.
8.	*Thalach*	Rata-rata detak jantung pasien.
9.	*Exang*	Keadaan dimana pasien akan mengalami nyeri dada apabila berolahraga, terdapat 2 nilai, dimana 0 tidak nyeri, dan 1 apabila menyebabkan nyeri.
10.	*Oldpeak*	Penurunan ST akibat olahraga
11.	*Sloap*	Slope dari puncak ST setelah berolahraga dengan 3 nilai, dimana 0 untuk *downsloping*, 1 untuk *flat*, dan 2 untuk *upsloping*.
12.	*Ca*	Banyaknya pembuluh darah yang terdeteksi melalui proses pewarnaan flourosopy

12.	*Thal*	Detak jantung pasien dengan 3 nilai, dimana 1 berarti fixed defect, 2 berarti normal, dan 3 yang berarti *reversable defect*
14.	*Num*	Hasil diagnosa penyakit jantung dengan 2 nilai, dimana 0 jika penyempitan diameter kurang dari atau sama dengan 50%, dan 1 jika penyempitan diameter lebih dari 50%.

Tabel 3. 2 Sample 10 Dataset

age	sex	cp	trtbps	chol	Fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
63	1	3	145	233	1	0	150	0	02.03	0	0	1	1
37	1	2	130	250	0	1	187	0	03.05	0	0	2	1
41	0	1	130	204	0	0	172	0	01.04	2	0	2	1
56	1	1	120	236	0	1	178	0	00.08	2	0	2	1
57	0	0	120	354	0	1	163	1	00.06	2	0	2	1
57	1	0	140	192	0	1	148	0	00.04	1	0	1	1
56	0	1	140	294	0	0	153	0	01.03	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	00.05	2	0	3	1
57	1	2	150	168	0	1	174	0	01.06	2	0	2	1

3.1.1 *Exploratory Data Analysis* (EDA)

Exploratory Data Analysis (EDA adalah tahap penting dalam analisis data yang bertujuan untuk memahami karakteristik dataset, mengidentifikasi pola atau tren, serta memvisualisasikan informasi penting sebelum melanjutkan ke tahap analisis lebih lanjut. Pada penelitian ini, EDA difokuskan untuk memvisualisasikan distribusi variabel numerik, serta mengidentifikasi keberadaan outlier dalam dataset.

Proses EDA melibatkan visualisasi distribusi variabel numerik menggunakan *boxplot* dan *histogram*. Langkah pertama adalah mengidentifikasi variabel numerik yang akan divisualisasikan, seperti *'age', 'trtbps', 'chol', 'thalachh', dan 'oldpeak'*. Variabel-variabel ini mencerminkan berbagai faktor klinis yang dapat mempengaruhi risiko serangan jantung.

Tahap awal dalam eksplorasi data dilakukan untuk memahami karakteristik dataset melalui visualisasi. Fokus utama dalam penelitian ini adalah mendeteksi outlier dan memahami distribusi data menggunakan teknik berikut:

1. **Memeriksa Dimensi Data**: Dataset memiliki sejumlah baris (sampel) dan kolom (fitur). Informasi ini membantu memahami skala dataset sebelum dilakukan analisis lebih lanjut.
2. **Mengidentifikasi Distribusi Data:** Distribusi data dianalisis untuk memahami pola persebaran setiap atribut. Visualisasi seperti histogram dan boxplot digunakan untuk variabel numerik.

3. **Mendeteksi Outlier:** Boxplot digunakan untuk mengidentifikasi outlier. Secara matematis, nilai dianggap outlier jika berada di luar rentang berikut: di mana adalah kuartil pertama, kuartil kedua dan kuartil ketiga, dengan:

    1. Menentukan *Interquartile Range* (IQR), dengan menggunakan persamaan 3.1:

$$\text{Interquartile Range} = Q_3 - Q_1 \quad (3.1)$$

Keterangan:
Q3 adalah quartil ketiga, yaitu nilai yang membagi data menjadi 75% bagian terbawah dan 25% bagian teratas.
Q1 adalah quartil pertama, yaitu nilai yang membagi data menjadi 25% bagian terbawah dan 75% bagian teratas.

    2. Menentukan batas outliers berdasarkan IQR dengan menggunakan persamaan berikut:

$$Batas\ Bawah = Q_1 - 1{,}5 \cdot Interquartile\ Range \quad (3.2)$$

$$Batas\ Atas = Q_3 + 1{,}5 \cdot Interquartile\ Range \quad (3.3)$$

3.1.2 Synthetic Minority Oversampling Technique (SMOTE)

Pada tahapan preprocessing data, dilakukan Synthetic Minority Oversampling Technique (SMOTE) untuk menyeimbangkan imbalanced data atau data yang tidak seimbang. Terdapat imbalanced data pada dataset yang digunakan untuk penelitian ini, sehingga diperlukan Teknik SMOTE untuk menanganinya sehingga distribusinya seimbang. Kelas 0 (tidak berisiko serangan jantung) memiliki 138 data, sedangkan kelas 1 (berisiko serangan jantung) memiliki 165 data. Meskipun memiliki perbedaan yang tidak begitu signifikan, namun peneliti tetap melakukan teknik SMOTE untuk menangani ketidakseimbangan data pada

dataset dengan tujuan agar dapat menghasilkan klasifikasi yang lebih baik. Penggunaan SMOTE juga dapat meningkatkan risiko overvitting. Berikut adalah tahapan dalam penerapan SMOTE:

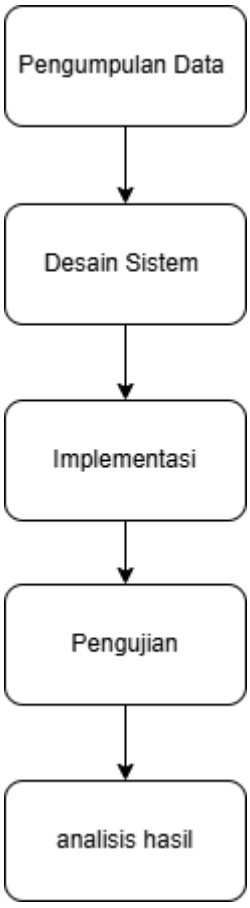
1. Menghitung jarak antar data pada data minoritas
2. Menetukan nilai presentase SMOTE yang akan digunakan untuk menciptakan data sintesis
3. Menciptakan data sintesis dengan persamaan sebagai berikut:

$$x_{new} = x_i + (x_{nearest} - x_i) \cdot z \quad (3.4)$$

Keterangan:
x_{new} : data sintesis baru.
x_i : data ke- i dari kelas minor.
$x_{nearest}$: data dari kelas minor yang memiliki jarak terdekat dari x_i.
z : nilai random antara 0 dan 1

3.2 Desain Penelitian

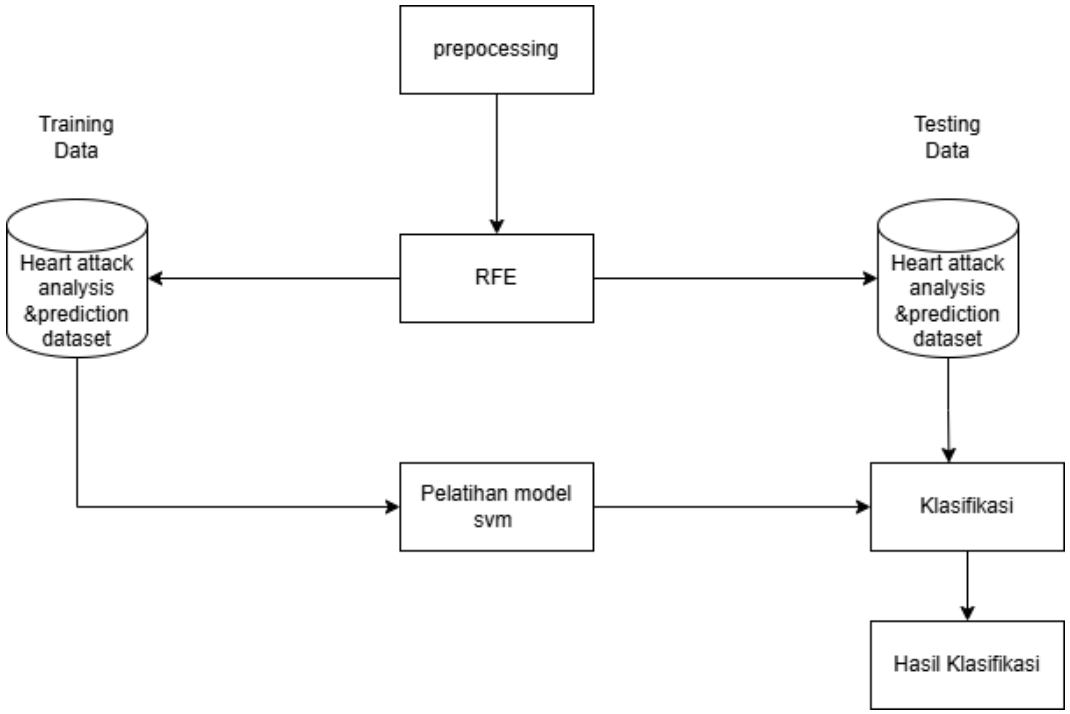
Penelitian ini memerlukan tahapan – tahapan yang akan dilakukan secara sistematis supaya target dari penelitian ini dapat tercapai dengan baik. Dengan adanya Desain sistem ini dapat mempermudah dalam Menyusun tahapan yang akan dilakukan. Tahapan ini meliputi pengumpulan data, kemudian pembuatan desain sistem dari proses yang di lakukan, Kemudian implementasi desain sistem yang telah didesain meliputi proses input, proses pengolahan data, dan proses output setelah melakukan tahap processing dan implementasi. Hal ini berguna untuk mendapatkan hasil yang optimal di butuhkan skenario uji. Alur proses desain penelitian dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Desain penelitian

3.3 Desain Sistem

Desain sistem ini dimulai dengan tahapan pengumpulan data dan *preprocessing* untuk mengimplementasikan analisis data dan pengembangan model klasifikasi untuk klasifikasi risiko serangan jantung. Dalam proses membangun model untuk klasifikasi risiko serangan jantung dilakukan pengumpulan data, pemisahan data, EDA, RFE, SMOTE, Normalisasi, split data, membangun model SVM, dan Evaluasi.



Gambar 3. 2 Desain Sistem

3.4 *Preprocessing*

3.4.1 Normalisasi Data

Normalisasi adalah proses penskalaan atau pemetaan data yang merupakan bagian dari tahap pra-pemrosesan. Terdapat beberapa teknik normalisasi, seperti min-max, Z-score, dan feature scaling. Dalam penelitian ini, digunakan metode normalisasi min-max. Min-Max Normalization merupakan salah satu teknik normalisasi yang umum dipakai untuk mengatasi perbedaan nilai antar fitur yang memiliki rentang yang terlalu besar(Ramadhan & Khoirunnisa, 2021). Tahap awalanya yaitu:

1. Mengidentifikasi nilai maksimum dan minimum
2. Menghitung nilai maksimum dan minimum Dengan rumus Berikut

$$Normalisasi = \frac{data - \min(data)}{\max(data) - \min(data)} \quad (3.5)$$

Keterangan:
Data : nilai asli yag ingin di normalisasi
Min(data) : nilai minimum dalam fitur
Max(data) : nilai maksimum

3.4.2 *Recursive Feature Elimination*

Recursive Feature Elimination (RFE) adalah algoritma seleksi fitur yang secara bertahap menyingkirkan fitur yang tidak penting. RFE efektif dalam menangani dataset berdimensi tinggi, meningkatkan akurasi, dan efisiensi model pembelajaran mesin (Priyatno & Widiyaningtyas, 2024).

Dalam penelitian ini, *Recursive Feature Elimination* (RFE) digunakan untuk memilih fitur yang paling relevan secara iteratif, dengan menghapus fitur yang kontribusinya kecil terhadap model prediksi. Proses ini bertujuan untuk meningkatkan efisiensi model dan mengurangi kompleksitas data.

Dalam penelitian ini ada beberapa tahapan yang harus dilakukan dalam *Recursive Feature Elimination:*

1. **Mendefinisikan Model Awal:** Model awal digunakan untuk menilai kepentingan atribut. Dalam penelitian ini, model awal yang digunakan adalah *Support Vector Machine* (SVM) dengan *kernel* linear.

$$w = \sum \propto_i y_i x_i \tag{3.6}$$

Keterangan:
w : vektor bobot akhir setelah semua alpha di perbarui
$\propto_i$: pengalih lagrang yang dioptimalkan untuk data latih ke i
y_i : label kelas untuk dayta latih ke i
x_i : vektor fitur untuk data latih ke i

2. **Menilai kepentingan fitur :** Nilai kepentingan fitur dihitung berdasarkan bobot hasil training, dengan mengkuadratkan bobot masing-masing fitur, sebagaimana ditunjukkan oleh :

$$C_i = w_i^2,\ i = 1,2 \ldots., |S| \quad (3.7)$$

Keterangan:
C_i : Nilai pentingnya fitur ke-i
$|S|$: Jumlah seluruh fitur

3. **Menjalankan Iterasi Eliminasi:** Pada setiap iterasi, atribut dengan kepentingan (importance) terendah dihapus berdasarkan cek bobot fitur dari model. Proses ini dilakukan secara bertahap hingga jumlah fitur yang diinginkan tercapai.

Setelah fitur-fitur terbaik diperoleh melalui proses RFE berbasis SVM *kernel* linear, model pembelajaran akhir dilatih menggunakan SVM dengan *kernel Radial Basis Function* (RBF). Penggunaan *kernel* RBF bertujuan untuk meningkatkan performa model dalam mengklasifikasikan data yang bersifat non-linear. Pada tahap ini, karena SVM RBF tidak memiliki bobot fitur eksplisit, evaluasi pentingnya fitur dilakukan menggunakan metode *Permutation Importance* untuk mengukur dampak masing-masing fitur terhadap hasil prediksi.

Normalisasi adalah proses penskalaan atau pemetaan data yang merupakan bagian dari tahap pra-pemrosesan. Terdapat beberapa teknik normalisasi, seperti min-max, Z-score, dan feature scaling. Dalam penelitian ini, digunakan metode normalisasi min-max.

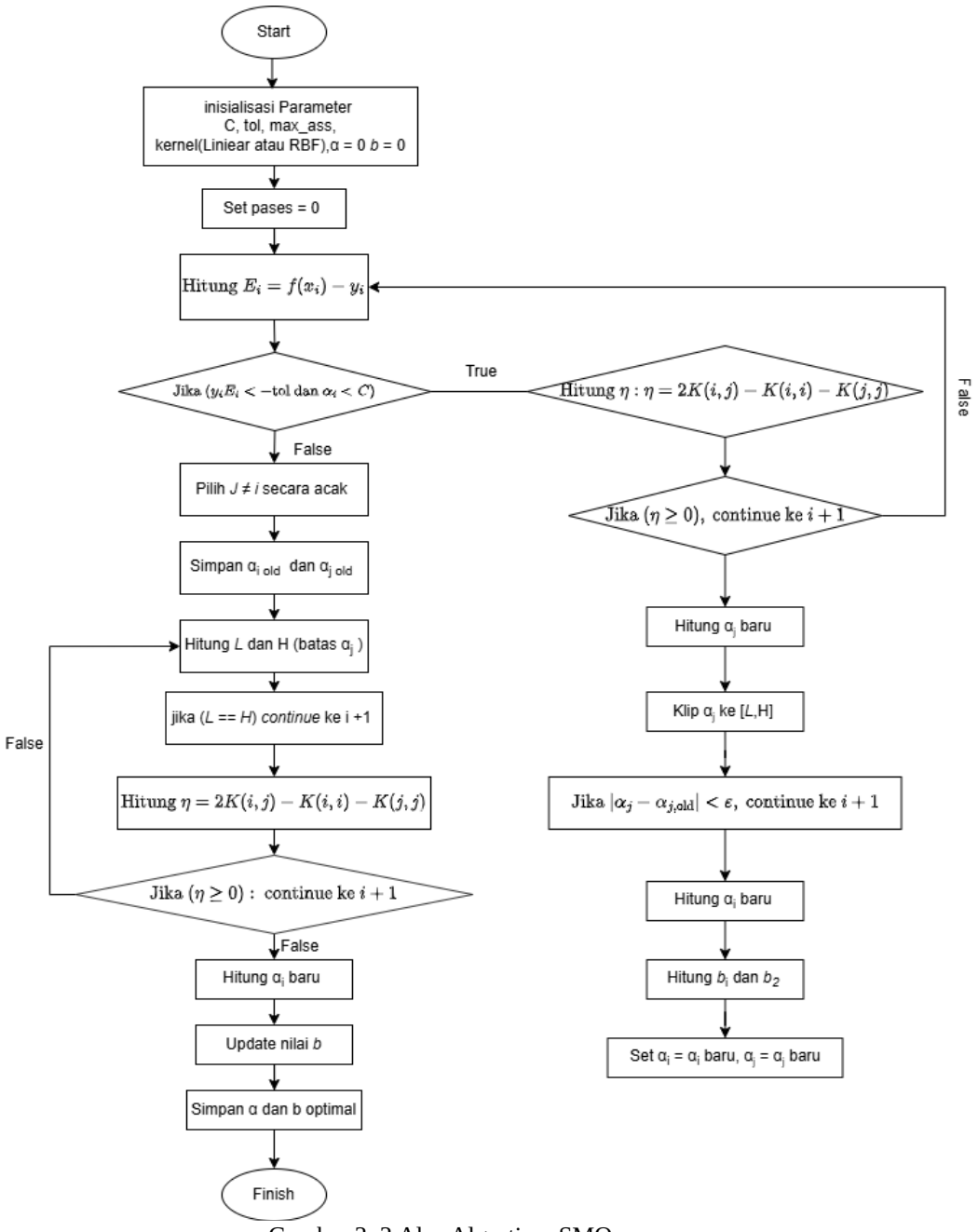
3.5 Implementasi Metode *Support Vector Machine*

Proses implementasi metode SVM dalam penelitian ini dilakukan dengan langsung menghitung *kernel* dari data yang diambil secara acak. Pasien yang terdiagnosis positif diberikan label 1, sedangkan pasien yang terdiagnosis negatif diberi label 0. Pada tahap ini, data yang telah dinormalisasi digunakan untuk menghitung *kernel* tanpa membahas lebih lanjut mengenai vektor bobot atau bias. Perhitungan *kernel* bertujuan untuk memproyeksikan data ke dimensi yang lebih tinggi, sehingga model SVM dapat memisahkan data dengan lebih baik. Simulasi ini menunjukkan bagaimana perhitungan *kernel* dilakukan secara langsung menggunakan data sampel yang telah diambil.

Untuk melatih model *Support Vector Machine* yang bertujuan untuk mencari hyperplane optimal yang dapat memisahkan data dari dua kelas secara maksimal. Ini dilakukan dengan memaksimalkan margin dan meminimalkan kesalahan klasifikasi. Untuk menyelesaikan masalah optimasi ini, digunakan metode *Sequential Minimal Optimization* (SMO) karena lebih efisien dibanding metode optimasi kuadratik konvensional, terutama untuk dataset besar.

Untuk mempermudah pemahaman mengenai langkah-langkah pelatihan SVM menggunakan metode *Sequential Minimal Optimization* (SMO), Gambar 3.3 berikut ini menyajikan diagram alir *Flowchart* dari proses algoritma tersebut. *Flowchart* ini menunjukkan urutan inisialisasi parameter, pemilihan pasangan data yang belum optimal, perhitungan galat dan batas optimasi, hingga pembaruan nilai parameter α (alpha) dan bias (b). Proses ini dilakukan secara iteratif hingga

mencapai jumlah iterasi maksimum yang telah ditentukan atau kondisi konvergen tercapai.



Gambar 3. 3 Alur Algortima SMO

Flowchart pada Gambar 3.3 menggambarkan alur kerja algoritma *Sequential Minimal Optimization* (SMO) yang digunakan untuk menyelesaikan

permasalahan optimasi pada algoritma *Support Vector Machine* (SVM). Proses dimulai dengan inisialisasi parameter, yaitu nilai konstanta C, toleransi kesalahan (tol), jumlah maksimum iterasi (max_ass), jenis *kernel* yang digunakan (Linear atau RBF), serta inisialisasi nilai awal parameter α dan *b* yang diset ke nol. Setelah itu, variabel passes diset ke nol untuk menandai jumlah iterasi tanpa pembaruan parameter α.

Pada setiap iterasi, algoritma menghitung nilai galat $E_i = f(x_i) - y_i$ untuk setiap data. Kemudian dilakukan pengecekan terhadap kondisi KKT (*Karush-Kuhn-Tucker*). Jika kondisi $y_i E_i < -tol$ dan $\alpha_i < C$ terpenuhi, maka data tersebut layak untuk dioptimasi. Selanjutnya, algoritma memilih secara acak indeks data ke-j yang berbeda dari i, kemudian menyimpan nilai lama dari α_i dan α_j. Setelah itu, dihitung batas bawah L dan batas atas H untuk nilai α_j yang baru. Jika nilai L dan H sama, maka iterasi akan dilanjutkan ke data berikutnya karena tidak ada ruang untuk pembaruan.

Langkah selanjutnya adalah menghitung nilai η menggunakan formula $\eta = 2K(i,j) - K(i,i) - K(j,j)$. Jika $\eta \geq 0$, maka proses dilanjutkan ke iterasi berikutnya karena pembaruan dianggap tidak menguntungkan. Jika tidak, maka α_j diperbarui dan diklip agar tetap berada dalam rentang [L, H]. Apabila selisih absolut antara α_j baru dan lama lebih kecil dari nilai ambang (epsilon), maka iterasi juga dilanjutkan karena perubahan dianggap tidak signifikan. Jika perubahan signifikan terjadi, maka nilai α_i turut diperbarui, diikuti dengan perhitungan nilai bias baru. Nilai bias dihitung dengan mempertimbangkan dua kandidat bias, yaitu b_1 dan b_2, tergantung dari posisi α_i dan α_j.

Setelah nilai α dan *b* diperbarui, keduanya disimpan sebagai parameter model yang optimal. Proses ini terus diulang hingga model mencapai konvergensi, yaitu ketika tidak ada pembaruan signifikan terhadap parameter α selama sejumlah iterasi berturut-turut sesuai batas maksimum max_ass. Dengan strategi ini, algoritma SMO dapat menyelesaikan optimasi SVM secara efisien dengan hanya memproses dua parameter pada setiap langkah iterasi.

Langkah-langkah dalam flowchart tersebut dapat direpresentasikan juga dalam bentuk formulasi matematis sebagai berikut:

1. Fungsi objektif SVM yang dioptimasi oleh SMO

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \, k(x_i, x_j) \tag{3.8}$$

Dengan batasan :

$0 \leq \alpha_i \leq C$

$\sum_{i=1}^{n} \alpha_i y_i = 0$

2. Proses optimasi menggunakan SMO dengan persamaan :

Menghitung galat (eror)

$$E_i = f(x_i) - y_i \tag{3.9}$$

Fungsi Keputusan (fungsional margin) *kernel* liniear

$$f(x_i) = \sum_{i=1}^{n} \alpha_i \, y_i K(x_i, x_j) + b \tag{3.10}$$

Untuk *kernel* RBF, fungsi *kernel* $K(x_i, x_j)$ didefinisikan sebagai:

$$K(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right) \tag{3.11}$$

Dimana:
$\|x_i - x_j\|^2$: Euclidean distanc kuadrat antara dua vaktor x_i dan x_j
γ : parameter *kernel* RBF, menentukan pengaruh jarak antar titik

Sehingga fungsi dengan *kernel* RBF menjadi:

$$f(x_i) = \sum_{i=1}^{n} \alpha_i \, y_j \exp\left(-\gamma \|x_i - x_j\|^2 \right) + b \qquad (3.12)$$

3. Update parameter svm

   Menghitung nilai η dengan persamaan:

$$\eta = 2(x_i, x_j) - (x_i, x_i) - (x_j, x_j) \qquad (3.13)$$

   Memperbarui nilai α_j

$$\alpha_j^{baru} = \alpha_j + \frac{y_j(E_i - E_j)}{\eta} \qquad (3.14)$$

   Memperbarui α_i

$$\alpha_i^{baru} = \alpha_i + y_i \, y_j(\alpha_j^{lama} - \alpha_j^{baru}) \qquad (3.15)$$

   Memberbarui bias (b) dengan persamaan:

$$b_{baru} = b_{lama} - E_i - y_i \left(\alpha_i^{baru} - \alpha_i^{lama}\right) K(x_i, x_i) \qquad (3.16)$$
$$- y_j \left(\alpha_j^{baru} - \alpha_j^{lama}\right) K(x_i, x_j)$$

4. Penghitugan vektor bobot w (hanya untuk *kernel* liniear)

$$w = \sum \propto_i y_i x_i \qquad (3.17)$$

5. Degan w dan b yang baru, dapat menghitung prediksi. Rumus persamaan untuk prediksi sebagai berikut :

   Untuk *kernel* liniear

$$prediksi = sign\,(w.x + b) \qquad (3.18)$$

   Untuk *kernel* RBF

$$prediksi = sign\big(\, f(x)\big)$$

$$= sign\left(\sum_{i=1}^{n} \alpha_i y_i \cdot \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right) + b\right) \qquad (3.19)$$

3.6 Skenario Pengujian

Peneliti akan menggunakan beberapa skenario pengujian untuk mengevaluasi sistem. Salah satu hal penting dalam proses ini adalah melakukan pembagian dataset, karena pembagian data sangat berpengaruh terhadap performa model. Setelah dilakukan *preprocessing* data, langkah berikutnya adalah membagi dataset atau yang dikenal sebagai *split data*. Proses ini memisahkan dataset menjadi dua bagian, yaitu data pelatihan dan data pengujian. Data pelatihan dimanfaatkan untuk membangun model machine learning agar mampu mengenali pola atau hubungan antara fitur dan target dalam data, sedangkan data uji digunakan untuk mengevaluasi performa model dalam mengklasifikasikan atau memprediksi data baru yang belum pernah dilihat sebelumnya. Untuk menemukan model dengan akurasi terbaik, peneliti membagi dataset ke dalam berbagai proporsi atau skenario berbeda untuk pelatihan dan pengujian, yaitu dengan menggunakan persentase data latih dan data uji sebesar 90:10, 80:20, dan 70:30, sebagaimana ditampilkan pada Tabel 3.3.

Tabel 3. 3 Skenario Pengujian

Skenario	Presentase Data Latih	Presentase Data Uji
1	90%	10%
2	80%	20%
3	70%	30%

Selain skenario pembagian dataset, terdapat juga sub-skenario tuning parameter. Tuning parameter bertujuan untuk menemukan nilai parameter yang optimal agar SVM dapat mencapai keseimbangan yang baik antara kesalahan dalam proses klasifikasi dan margin maksimum, sesuai dengan karakteristik data yang ada. Setiap skenario akan diuji dengan sub-skenario seperti pada Tabel 3.4.

Tabel 3. 4 Tuning Parameter

Fungsi *Kernel*	*Hyperparameter*	Nilai Hyperparameter
Kernel Linear	C (Cost)	0.1, 1, 10
Kernel RBF	C (Cost)	0.1, 1, 10
	γ (Gamma)	0.1, 0.5, 1

Confusion matrix adalah sebuah tabel yang sering digunakan untuk mengevaluasi kinerja model klasifikasi dalam *machine learning*. Tabel ini memberikan gambaran yang lebih rinci tentang jumlah data yang diklasifikasikan dengan benar dan salah, dengan cara membandingkan hasil prediksi model dengan nilai aktual. *Confusion matrix* merupakan salah satu alat analisis prediktif yang berguna untuk memvisualisasikan dan membandingkan nilai hasil prediksi model dengan nilai aktual.

Dalam konteks evaluasi model, *confusion matrix* biasanya dihitung menggunakan data uji (test data), bukan data latih (*training data*). Ini karena data uji digunakan untuk mengukur kemampuan model dalam menggeneralisasi, atau seberapa baik model dapat bekerja pada data baru yang belum pernah dilihat selama pelatihan. Sementara itu, pada data latih, model belajar untuk mengenali pola dari data tersebut. Menghitung *confusion matrix* pada data latih bisa menunjukkan performa yang optimis, namun tidak selalu mencerminkan kinerja yang

sesungguhnya pada data baru. Oleh karena itu, *confusion matrix* pada data uji memberikan gambaran yang lebih obyektif tentang kemampuan model.

Berikut adalah perbedaan antara *training* data dan *test* data dalam konteks *confusion matrix*:

1. *Training* Data (Data Latih): Digunakan untuk melatih model. *Confusion matrix* jarang dihitung pada data latih, karena hasil yang diperoleh bisa terlalu optimis dan tidak mencerminkan performa model pada data yang belum pernah dilihat.
2. *Test* Data (Data Uji): Digunakan untuk mengevaluasi model setelah proses pelatihan selesai. *Confusion matrix* pada data uji mencerminkan Kemampuan model mengklasifikasikan data yang belum dilihat menunjukkan seberapa baik model melakukan generalisasi selama pelatihan, sehingga memberikan gambaran kinerja yang lebih realistis.

Di dalam confusion matrix, terdapat empat nilai utama yang menjadi acuan, yaitu:

1. *True Positive* (TP): Jumlah data pasien yang benar-benar menderita penyakit serangan jantung dan berhasil terdeteksi positif oleh algoritma.
2. *False Positive* (FP): Jumlah data pasien yang tidak terkena penyakit serangan jantung tetapi ada kesalahan klasifikasi pada algoritma yang terdeteksi positif.
3. *False Negative* (FN): Jumlah data pasien yang yang seharusnya tidak terkena penyakit jantung tetapi ada kesalahan klasifikasi dan tidak menunjukkan negatif.

4. *True Negative* (TN): Jumlah data pasien yang tidak terkena penyakit serangan jantung dan teridentifikasi benar tidak terkena serangan jantung oleh algoritma.

Dalam hal ini dapat di artikan bahwa TP dan TN menunjukkan hasil klasifikasi yang benar, sedangkan pada FP dan FN menujukkan adanya kesalahan yang dilakukan oleh algoritma dalam mengidentifikasi yang dapat menyebabkan efektifitas model pada penyakit serangan jantung. Yang dapat digambarkan seperti tabel 3.5 di bawah ini menunjukkan struktur *confusion matrix*.

Tabel 3. 5 Confusion Matrix

		Hasil Prediksi Algoritma	
		Positif	**Negatif**
Kondisi pasien sebenarnya	**Positif**	True Positive (TP) (Pasien sakit yang terdeteksi dengan benar)	False Negative (FN) (Pasien sakit yang salah terdeteksi sehat)
	Negatif	False Positive (FP) (Pasien sehat yang salah terdeteksi sakit)	True Negative (TN) True Negative (Pasien sehat yang terdeteksi dengan benar)

Berdasarkan *confusion matrix*, berbagai metrik evaluasi seperti Akurasi (*Accuracy*), Presisi (*Precision*), Recall, dan F1-Score yang dapat di jelaskan seperti di bawah ini.

1. Akurasi

Akurasi adalah metrik yang mengukur seberapa sering model melakukan prediksi yang benar, baik untuk kelas positif maupun negatif. Ini adalah metrik evaluasi yang paling umum dan memberikan gambaran umum tentang kinerja model secara keseluruhan.

$$\text{Akurasi (}Accuracy\text{)} = \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100\% \qquad (3.20)$$

2. Presisi

Presisi mengukur seberapa akurat prediksi positif yang dibuat oleh model. Artinya, dari seluruh data yang diprediksi sebagai positif, berapa banyak yang benar-benar positif.

$$\text{Presisi } (\textit{Precision}) = \frac{TP}{TP+FP} \text{ x } 100\% \quad (3.21)$$

3. Recall

Recall mengukur kemampuan model dalam menemukan semua data positif. Dari seluruh data yang sebenarnya positif, berapa banyak yang berhasil diprediksi dengan benar.

$$\text{Recall } (\textit{Sensitivity}) = \frac{TP}{TP+FN} \text{ x } 100\% \quad (3.22)$$

4. F1-Score

F1-Score adalah rata-rata harmoni antara presisi dan recall. Metrik ini digunakan ketika dibutuhkan keseimbangan antara presisi dan recall, terutama pada data yang tidak seimbang.

$$\text{F1-Score} = \frac{2xPrecisionxRecall}{Precision+Recall} \text{ x } 100\% \quad (3.23)$$

BAB IV

HASIL DAN PEMBAHASAN

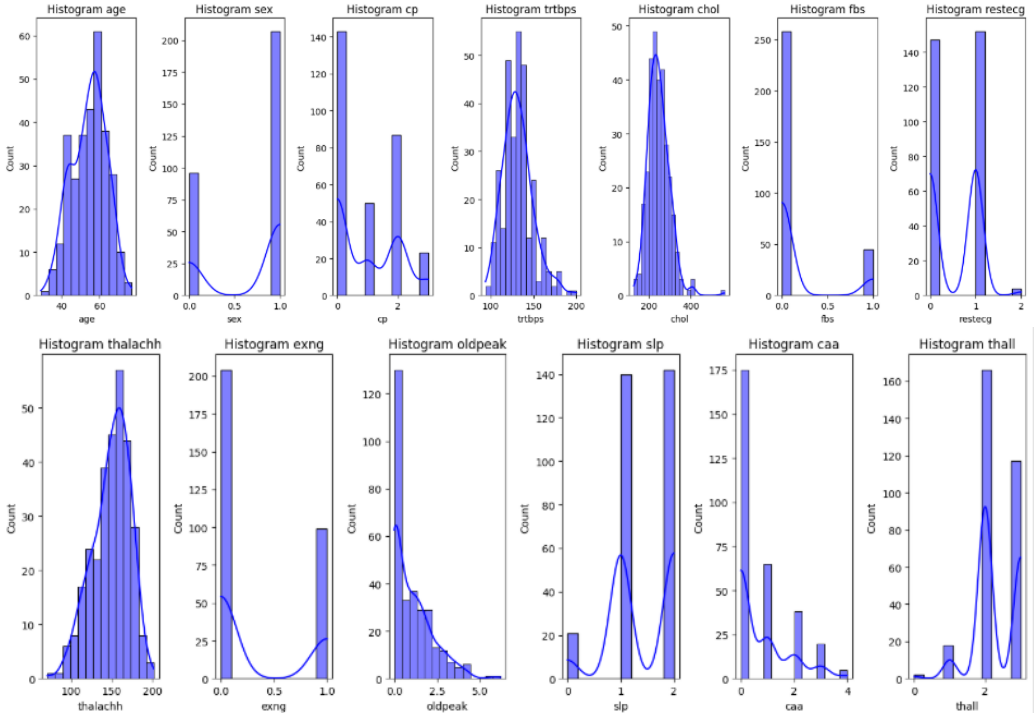
4.1 Hasil Preprocessing Data

Penelitian ini menggunakan data sekunder yang berasal dari https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset . Tahapan Preprocessing data dapat meliputi pemisahan data utuk memisahkan antar fiture dan targe, EDA sebagai visualisi data dengan histogram dan boxplot. Selanjutnya Smote untuk menyeimbangkan data dengan oversampling, dan normalisasi data untuk menyemaratakan skala data, dan terakhir menggunakan RFE.

Proses pelatihan model SVM pada penelitian ini dilakukan dengan bantuan pustaka scikit-learn untuk efisiensi dan kepraktisan dalam implementasi. Namun demikian, untuk mendukung validasi teori dan memperkuat pemahaman terhadap algoritma *Sequential Minimal Optimization* (SMO) yang digunakan dalam pelatihan SVM, penulis juga menyertakan kode manual perhitungan SMO pada bagian lampiran.

4.1.1 *Exploratory Data Analysis* (EDA)

Pada tahap EDA peneliti hanya memvisualisasikan penyebaran data dengan histogram dan boxplot untuk mengetahui outlier.

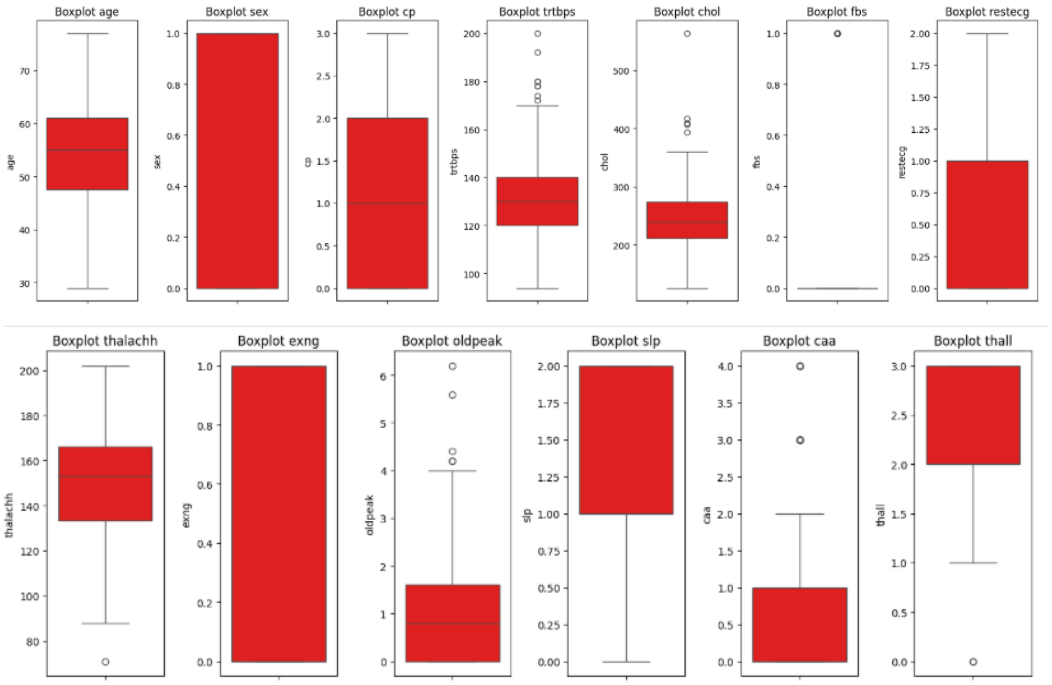


Gambar 4. 1 Histogram

Histogram digunakan untuk menganalisis distribusi setiap fitur dalam dataset. Berdasarkan visualisasi, fitur *age* dan *trtbps* menunjukkan distribusi yang mendekati normal dengan bentuk yang simetris di sekitar nilai rata-rata. Sementara itu, fitur *chol* dan *thalachh* memiliki pola distribusi yang hampir normal tetapi sedikit condong ke kanan.

Beberapa fitur menunjukkan distribusi yang tidak simetris. Fitur *oldpeak* memiliki distribusi skewed positif (condong ke kanan) karena terdapat ekor panjang pada sisi kanan histogramnya. Sebaliknya, fitur *thall* menunjukkan distribusi skewed negatif (condong ke kiri) dengan ekor panjang di sisi kiri.

Selain itu, beberapa fitur bersifat kategorikal, *seperti sex*, *cp*, *fbs*, *restecg*, *exng*, *slp*, dan *caa*, yang memiliki nilai diskrit dengan distribusi yang lebih terpisah dibandingkan fitur kontinu. Pemahaman terhadap distribusi data ini penting untuk mengidentifikasi kemungkinan adanya outlier serta menentukan teknik pemrosesan yang sesuai.



Gambar 4. 2 *Boxplot* Diagram

Boxplot digunakan untuk mengidentifikasi *outlier* dalam dataset. Berdasarkan visualisasi, fitur *trtbps*, *chol*, *thalachh*, *oldpeak*, dan *caa* memiliki *outlier*, yang ditandai dengan titik-titik di luar *whisker*. Misalnya, fitur *chol* dan *trtbps* memiliki nilai ekstrem di atas batas atas, sedangkan *thalachh* menunjukkan *outlier* di sisi bawah.

Sebaliknya, fitur *age*, *sex*, *cp*, *fbs*, *restecg*, *exng*, *slp*, dan *thall* tidak memiliki *outlier* karena seluruh nilai berada dalam rentang *whisker*, menunjukkan distribusi yang lebih stabil. Dalam penelitian ini, *outlier* tidak dihapus, karena dataset medis

mengandung informasi penting yang bisa berpengaruh signifikan terhadap prediksi penyakit jantung. Menghilangkan *outlier* dapat menyebabkan kehilangan data yang relevan, sehingga semua data tetap digunakan untuk memastikan model *Support Vector Machine* (SVM) dapat menangani berbagai kondisi pasien dengan optimal.

4.1.2 SMOTE

Dataset awal tidak seimbang, dengan kelas 1 sebanyak 165 sampel dan kelas 0 hanya 138 sampel. Ketidakseimbangan ini dapat menyebabkan model cenderung lebih akurat dalam memprediksi kelas mayoritas.

Untuk mengatasi hal ini, digunakan SMOTE (Synthetic Minority Over-sampling Technique), yang merupakan metode oversampling untuk menambah jumlah sampel kelas minoritas. Setelah SMOTE diterapkan, jumlah kelas 0 meningkat menjadi 165 sampel, sehingga distribusi menjadi seimbang (1:165, 0:165). Dengan data yang seimbang, model *Support Vector Machine* (SVM) dapat belajar lebih optimal tanpa bias terhadap kelas tertentu.

4.1.3 Normalisasi

Normalisasi fitur adalah proses penskalaan data agar berada dalam rentang tertentu, yang dalam hal ini antara 0 dan 1. Tujuan dari normalisasi adalah memastikan setiap fitur memiliki skala yang seragam, sehingga algoritma *machine learning* dapat bekerja lebih optimal dan tidak bias terhadap fitur dengan nilai skala yang lebih besar.

Dalam penelitian ini, normalisasi dilakukan menggunakan *MinMaxScaler* dari *library Scikit-learn*. *MinMaxScaler* bekerja dengan mentransformasikan data

berdasarkan nilai minimum dan maksimum dari setiap fitur, sehingga nilai data berada dalam rentang [0,1]. Dengan rumus Persamaan (4.1):

$$Normalisasi = \frac{data - \min(data)}{\max(data) - \min(data)} \quad (4.1)$$

Keterangan:
Normalisasi : nilai yang sudah di normalisasi
N : nilai asli yang ingin di normalisasi
Min(data) : nilai minimum dalam fitur
Max(data) : nilai maksimum

Tabel 4. 1 Sebelum Normalisasi

age	*sex*	*cp*	*trtbps*	*chol*	*Fbps*	*Output*
63	1	3	145	233	1	1
37	1	2	130	250	0	1
41	0	1	130	204	0	1
56	1	1	120	236	0	1

Contoh simulasi dari sampel data pada *feature age* sebagai berikut:

$$Normalisasi = \frac{63 - 29}{77 - 29} = \frac{34}{48} = 0.70$$

Tabel 4.2 menunjukkan contoh data yang telah normalisasi.

Tabel 4. 2 Hasil Normalisasi

age	*sex*	*cp*	*trtbps*	*chol*	*Fbps*	*Output*
0.70	1	1	0.48	0.24	1	1
0.16	1	0.66	0.33	0.28	0	1
0.25	0	0.33	0.33	0.17	0	1
0.56	1	0.33	0.24	0.25	0	1

4.1.4 Hasil RFE

Berdasarkan hasil RFE, fitur yang paling berpengaruh dalam klasifikasi risiko serangan jantung adalah *oldpeak*, jumlah pembuluh darah berwarna (*caa*), jenis nyeri dada (*cp*), hasil tes thalium (*thall*), detak jantung maksimum (*thalachh*) seperti tabel 4.3. Fitur-fitur ini memiliki importance tertinggi, menunjukkan kontribusi besar dalam membedakan individu berisiko dan tidak berisiko.

Tabel 4. 3 menampilkan *ranking* dan nilai *importance* dari setiap fitur berdasarkan penggunaan *kernel* linear dan RBF pada metode *Support Vector Machine* (SVM).

Tabel 4. 3 Hasil Cek Pembobotan RFE

No	Fitur	*Impotance* (*Linear*)	*Importance* (RBF)
1	*oldpeak*	1.810298	0.005758
2	*Caa*	1.765270	0.058182
3	*Cp*	1.732684	0.076667
4	*Thall*	1.561357	0.019394
5	*Thalachh*	1.271840	0.002121
6	*Sex*	0.990420	0.031818
7	*Trtbps*	0.840017	0.003030
8	*Chol*	0.786630	-0.006061
9	*Slp*	0.739050	0.011818
10	*exng*	0.689603	0.018788
11	*Age*	0.000000	0.000000
12	*Fbs*	0.000000	0.000000
13	*Restecg*	0.000000	0.000000

Nilai *importance* dalam RFE menunjukkan seberapa besar pengaruh setiap fitur dalam membantu model SVM melakukan klasifikasi. Dalam SVM dengan *kernel* linear, *importance* dihitung berdasarkan koefisien fitur, sedangkan dalam *kernel* RBF, *importance* lebih kompleks karena model mendeteksi pola non-linier.

Beberapa fitur seperti kolesterol *(chol)* dan tekanan darah saat istirahat *(trtbps)* memiliki nilai absolut tinggi dalam dataset, tetapi *importance*-nya rendah karena korelasi dengan fitur lain yang lebih dominan atau kontribusinya kecil dalam membedakan kelas. Selain itu, beberapa fitur memiliki *importance* mendekati nol, yang berarti model menganggapnya kurang relevan dalam klasifikasi.

Dengan demikian, fitur yang memiliki *importance* tertinggi lebih diutamakan karena memberikan informasi yang lebih relevan dan meningkatkan akurasi model dalam mendeteksi risiko serangan jantung. Sementara itu, fitur-fitur yang dianggap kurang relevan seperti *age*, *fbs*, dan *restecg* tidak dilanjutkan ke

tahap pengujian karena kontribusinya yang rendah terhadap performa model dan 1 atribut tidak dilakukan pemrosesan karena merupakan *output*.

4.1.5 Hasil Data Spliting

Proses pembagian data menghasilkan dua bagian, yaitu data *training* dan data testing. Tabel 4. 4 hasil pembagian data dengan berbagai rasio yang berbeda. Pembagian data dilakukan menggunakan metode *random state*.

Tabel 4. 4 Hasil Rasio Pembagian Data

Skenario	Presentase Data Latih	Presentase Data Uji
90:10	297	33
80:20	264	66
70:30	231	99

4.2 Hasil Uji Coba

Pada bagian ini, akan membahas hasil evaluasi performa model dengan menggunakan metode SVM berdasarkan *kernel* Linear dan RBF beserta *tuning* parameternya. Hasil evaluasi performa model didapatkan berdasarkan perbandingan antara nilai *y_test* (data aktual) dan *y_pred* (data hasil prediksi) sebagai tolak ukur tingkat keberhasilan metode. Evaluasi performa model berfokus pada *confusion matrix* untuk menunjukkan tingkat akurasi model berdasarkan klasifikasi.

4.2.1 Hasil Uji Coba Skenario 1

Pengujian pada skenario 1 menggunakan rasio pembagian data 90:10 untuk mengevaluasi performa model SVM setelah seleksi fitur menggunakan RFE.

Distribusi kelas pada data *training* adalah 148 sampel kelas negatif (0) dan 149 sampel kelas positif (1) atau 297 data, sedangkan pada data testing terdapat 17 sampel kelas negatif (0) dan 16 sampel kelas positif (1) atau 33 data. Distribusi ini menunjukkan bahwa data relatif seimbang, sehingga model dapat belajar dari kedua kelas tanpa bias yang signifikan.

Pengujian dilakukan untuk menilai kemampuan model dalam mengklasifikasikan risiko serangan jantung berdasarkan fitur yang telah terpilih. Evaluasi performa model akan menggunakan *confusion matrix*, yang mencakup metrik akurasi, presisi, recall, dan F1-score, guna memahami seberapa baik model dapat membedakan individu berisiko dan tidak berisiko. Untuk mendapatkan performa terbaik dari model SVM, dilakukan *tuning hyperparameter* pada dua jenis *kernel*, yaitu Linear dan RBF. Nilai *hyperparameter* yang digunakan dalam *tuning* terdapat pada tabel 4. 5:

Tabel 4. 5 *Kernel* Dan *Hyperparameter* Skenario 1

Percobaan	*Kernel*	*Hyperparameter*	
		Cost (C)	*Gamma* (γ)
1	Linear	0.1	
2		1	
3		10	
4	RBF	0.1	0.1
5			0.1
6			1
7		1	0.1
8			0.5
9			1
10		10	0.1
11			0.5
12			1

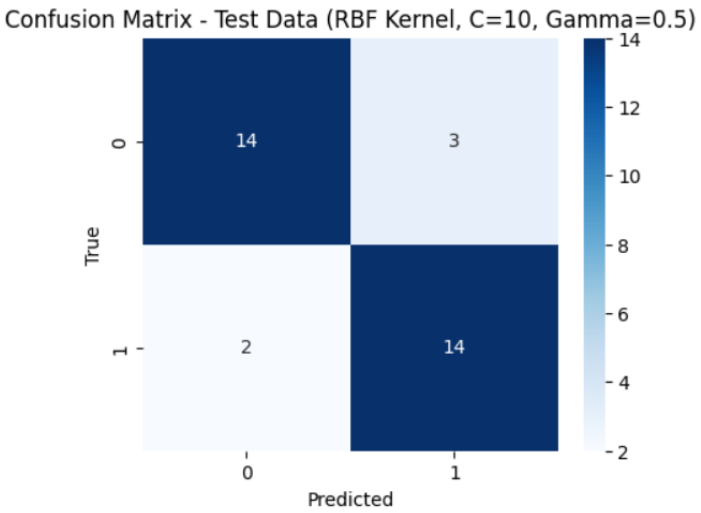
Pengujian skenario 1 memakan waktu sekitar 6 detik. Evaluasi pada model SVM ini menggunakan *confusion matirix,* lalu dilakukan perhitungan Akurasi, *Precision, Recall*, F1-score berdasarkan *True Positive* (TP), *False Positive* (FP),

True Negative (TN), dan *False Negative* (FN) menggunakan data *testing* sebanyak 33. Tabel 4. 6 adalah hasil evaluasi pengujian skenario 1.

Tabel 4. 6 Hasil Evaluasi Pengujian Skenario 1

NO	*Kernel*	*Hyperparameter*		Evaluasi			
		Cost	*Gamma*	Akurasi	*Precision*	*Recall*	F1-score
1	Linear	0.1		78.78%	76.47%	81.25%	78.78%
2		1		78.78%	73.68%	87.50%	80.00%
3		10		78.78%	73.68%	87.50%	80.00%
4	RBF	0.1	0.1	78.78%	73.68%	87.50%	80.00%
5			0.5	78.78%	76.47%	81.25%	78.78%
6			1	78.78%	76.47%	81.25%	78.78%
7		1	0.1	78.78%	76.47%	81.25%	78.78%
8			0.5	75.75%	72.22%	81.25%	76.47%
9			1	81.81%	81.25%	81.25%	81.25%
10		10	0.1	78.78%	73.68%	87.50%	80.00%
11			0.5	84.84%	82.35%	87.50%	84.84%
12			1	81.81%	81.25%	81.25%	81.25%

Dari percobaan sebanyak 12 kali pada skenario 1 dengan 2 *kernel* dan *tuning hyperparameter*, Nilai akurasi tertinggi terdapat pada *kernel* RBF dengan konfigurasi *hyperparameter* C = 10 dan γ = 0.5, akurasi yaitu sebesar 84.84% beserta dengan *precision* 82%, *recall* 88%, dan F1-Score 85%. Gambar 4.3 menunjukkan nilai *confusion matrix* pada *kernel* RBF dengan konfigurasi *hyperparameter* C = 10 dan γ = 0.5.



Gambar 4. 3 *Confusion Matrix Kernel* RBF C = 10 dan γ = 0.5

4.2.2 Hasil Uji coba Skenario 2

Pada skenario ini, dataset dibagi menjadi 80% data latih (*training*) dan 20% data uji (testing) untuk mengevaluasi performa model SVM setelah seleksi fitur menggunakan RFE. Distribusi kelas dalam data *training* adalah 132 sampel kelas negatif (0) dan 132 sampel kelas positif (1), sedangkan dalam data testing terdapat 33 sampel kelas negatif (0) dan 33 sampel kelas positif (1). Dengan distribusi yang seimbang antara kedua kelas, model diharapkan dapat belajar secara optimal tanpa bias terhadap salah satu kelas.

Pengujian ini bertujuan untuk mengevaluasi kemampuan model dalam mengklasifikasikan individu yang berisiko dan tidak berisiko mengalami serangan jantung berdasarkan fitur yang telah dipilih melalui metode RFE. Evaluasi dilakukan menggunakan *confusion matrix* serta metrik akurasi, presisi, *recall*, dan F1-score, untuk mengukur performa model dalam membedakan kedua kelas dengan tepat. Selain itu, dilakukan tuning *hyperparameter* untuk memperoleh kombinasi

parameter terbaik yang dapat meningkatkan akurasi model, dengan nilai yang telah ditentukan pada tabel 4.7:

Tabel 4. 7 *Kernel* dan *Hyperparameter* Skenario 2

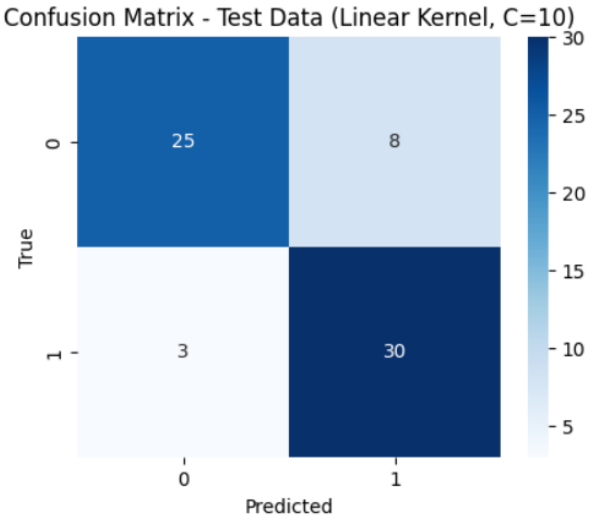
Percobaan	*Kernel*	*Hyperparameter*	
		Cost (C)	*Gamma* (γ)
1	Linear	0.1	
2		1	
3		10	
4	RBF	0.1	0.1
5			0.1
6			1
7		1	0.1
8			0.5
9			1
10		10	0.1
11			0.5
12			1

Pengujian skenario 2 memakan waktu sekitar 6 detik. Evaluasi pada model SVM ini menggunakan *confusion matrix*, lalu dilakukan perhitungan Akurasi, *Precision*, *Recall*, F1-score berdasarkan *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) menggunakan data *testing* sebanyak 66. Tabel 4. 8 adalah hasil evaluasi pengujian skenario 2.

Tabel 4. 8 Hasil Evaluasi Pengujian Skenario 2

NO	*Kernel*	*Hyperparameter*		Evaluasi			
		Cost	*Gamma*	Akurasi	*Precision*	*Recall*	F1-score
1	Linear	0.1		81.81%	78.37%	87.87%	82.85%
2		1		81.81%	76.92%	90.90%	83.33%
3		10		83.33%	78.94%	90.90%	84.50%
4	RBF	0.1	0.1	81.81%	76.92%	90.90%	83.33%
5			0.5	80.30%	77.77%	84.84%	81.15%
6			1	78.78%	77.14%	81.81%	79.41%
7		1	0.1	80.30%	77.77%	84.84%	81.15%
8			0.5	80.30%	76.31%	87.87%	81.69%
9			1	81.81%	78.37%	87.87%	82.85%
10		10	0.1	81.81%	76.92%	90.90%	83.33%
11			0.5	83.33%	82.35%	84.84%	83.58%
12			1	80.30%	79.41%	81.81%	80.59%

Dari percobaan sebanyak 12 kali pada skenario 1 dengan 2 *kernel* dan *tuning hyperparameter*, Nilai akurasi tertinggi terdapat pada *kernel* Linear dengan konfigurasi *hyperparameter* C = 10, yaitu sebesar 83.33% beserta dengan precision 79%, *recall* 91%, dan F1-Score 85%. Gambar 4. 4 menunjukkan nilai *confusion matrix* pada *kernel* Linear dengan konfigurasi *hyperparameter* C =10.



Gambar 4. 4 *Confusion Matrix Kernel* Linear C = 10

4.2.3 Hasil uji coba Skenario 3

Pada skenario ini, dataset dibagi menjadi 70% data latih (*training*) dan 30% data uji (t*esting*) untuk mengevaluasi performa model SVM setelah seleksi fitur menggunakan RFE. Distribusi kelas dalam data *training* adalah 115 sampel kelas negatif (0) dan 116 sampel kelas positif (1), sedangkan dalam data testing terdapat 50 sampel kelas negatif (0) dan 49 sampel kelas positif (1). Dengan distribusi yang tetap seimbang, model diharapkan dapat belajar tanpa bias signifikan terhadap salah satu kelas.

Pengujian ini bertujuan untuk menilai kemampuan model dalam mengklasifikasikan individu yang berisiko dan tidak berisiko mengalami serangan jantung berdasarkan fitur yang telah dipilih. *Evaluasi performa model* dilakukan menggunakan *confusion matrix* serta metrik akurasi, presisi, *recall*, dan F1-score untuk mengetahui seberapa baik model dalam membedakan kedua kelas. Selain itu, *tuning hyperparameter* dilakukan untuk mencari kombinasi parameter terbaik yang dapat meningkatkan akurasi model. Nilai *hyperparameter* yang digunakan pada tabel 4. 9:

Tabel 4. 9 *Kernel* dan *Hyperparameter* skenario 3

Percobaan	*Kernel*	*Hyperparameter*	
		Cost (*C*)	*Gamma* (γ)
1	Linear	0.1	
2		1	
3		10	
4	RBF	0.1	0.1
5			0.1
6			1
7		1	0.1
8			0.5
9			1
10		10	0.1
11			0.5
12			1

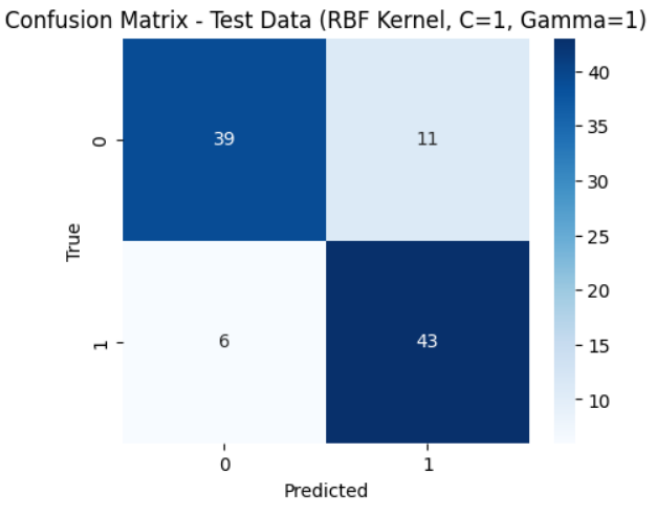
Pengujian skenario 3 memakan waktu sekitar 6 detik. Evaluasi pada model SVM ini menggunakan *confusion matrix*, lalu dilakukan perhitungan Akurasi, *Precision*, *Recall*, F1-score berdasarkan *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) menggunakan data *testing* sebanyak 99. Tabel 4. 10 adalah hasil evaluasi pengujian skenario 3.

Tabel 4. 10 Hasil Evaluasi Pengujian Skenario 3

NO	*Kernel*	*Hyperparameter*		Evaluasi			
		Cost	*Gamma*	Akurasi	*Precision*	*Recall*	F1-score
1	Linear	0.1		79.79%	74.57%	89.79%	81.48%
2		1		80.80%	76.78%	87.75%	81.90%

3		10		79.79%	77.35%	83.67%	80.39%
4	RBF	0.1	0.1	78.78%	72.58%	91.83%	81.08%
5			0.5	78.78%	73.33%	89.79%	80.73%
6			1	76.76%	74.07%	81.63%	77.66%
7		1	0.1	79.79%	75.43%	87.75%	81.13%
8			0.5	79.79%	76.36%	85.71%	80.76%
9			1	82.82%	79.62%	87.75%	83.49%
10		10	0.1	80.80%	76.78%	87.75%	81.90%
11			0.5	79.79%	77.35%	83.67%	80.39%
12			1	79.79%	77.35%	83.67%	80.39%

Dari percobaan sebanyak 12 kali pada skenario 3 dengan 2 *kernel* dan *tuning hyperparameter*, Nilai akurasi tertinggi terdapat pada *kernel* RBF dengan konfigurasi *hyperparameter* C = 1 dan γ = 1, yaitu sebesar 82.82% beserta dengan *precision* 80%, *recall* 88%, dan F1-Score 83%. Gambar 4. 5 menunjukkan nilai *confusion matrix* pada *kernel* RBF dengan konfigurasi *hyperparameter* C = 1 dan γ = 1.



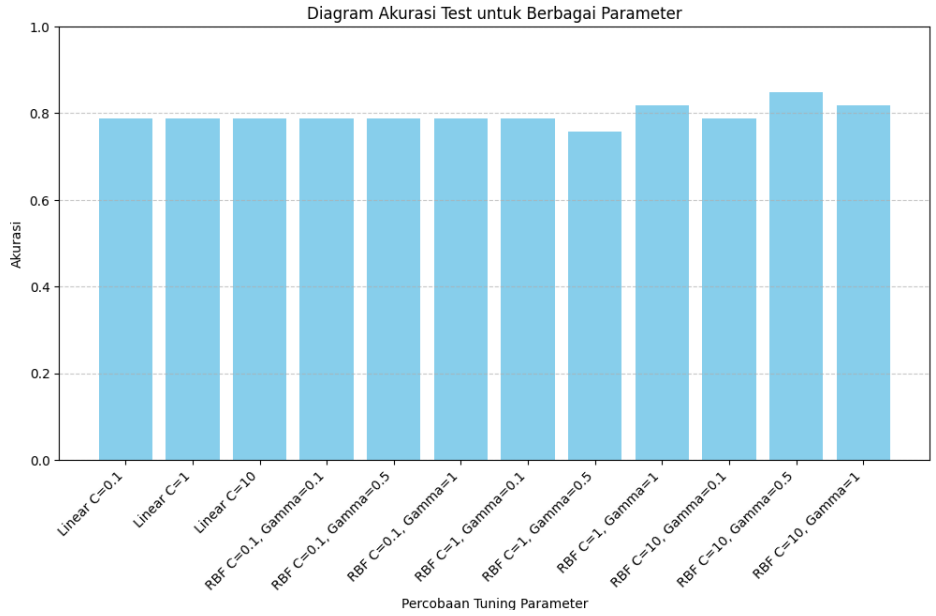
Gambar 4. 5 *Confusion Matrix Kernel* Linear C = 1 γ = 1

4.3 Pembahasan

Penelitian ini melakukan tiga skenario utama dengan membagi data menggunakan rasio yang berbeda untuk menentukan skenario yang menghasilkan

performa paling optimal dalam penggunaan metode *Support Vector Machine* untuk mengklasifikasi penyakit serangan jantung. Pada setiap skenario, peneliti menguji berbagai kombinasi parameter dengan menerapkan *kernel* linear serta variasi parameter *cost* dan gamma untuk menemukan konfigurasi yang memberikan hasil terbaik.

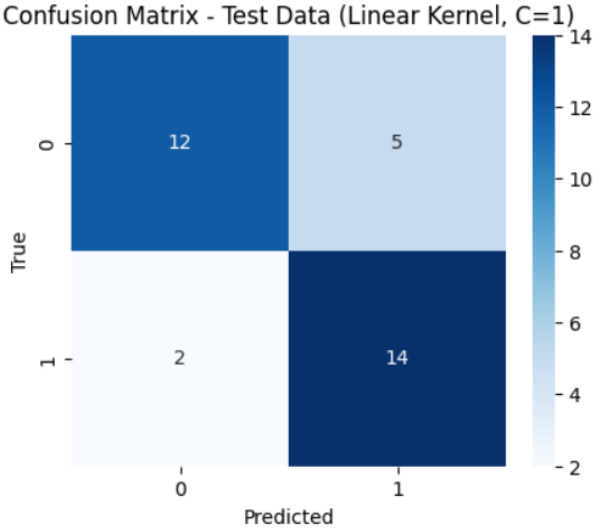
Pengujian pada skenario 1 menggunakan rasio pembagian data 90:10, dengan 297 data untuk *training* dan 33 data untuk testing. Beberapa sub-skenario diuji dengan berbagai nilai parameter untuk memperoleh hasil klasifikasi yang paling akurat. Pengujian skenario 1 memakan waktu sekitar 1 detik. Berikut adalah grafik hasil evaluasi pengujian skenario 1:



Gambar 4. 6 Diagram Akurasi Skenario 1

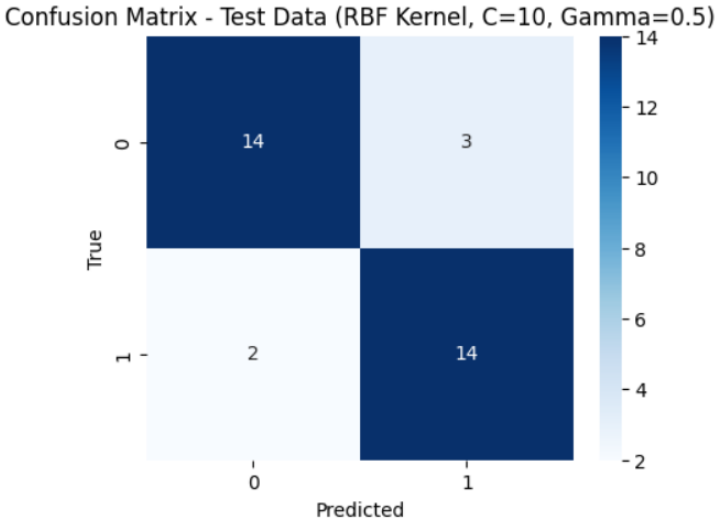
Pada Gambar 4. 6 menunjukkan hasil akurasi model pada pengujian skenario 1 menggunakan *confusion matrix*. Nilai akurasi tertinggi terdapat pada *kernel* Linear dengan konfigurasi *hyperparameter* C = 1, yaitu sebesar 78.78% beserta dengan *precision* 74%, *recall* 88%, dan F1-Score 80%. Gambar 4. 7

menunjukkan nilai *confusion matrix* pada *kernel* Linear dengan konfigurasi *hyperparameter* C = 1. Pada penelitian ini, nilai C = 1 dipilih karena memberikan keseimbangan antara generalisasi dan akurasi. Sebenarnya, hasil C=1 dan C=10 sama, tetapi C yang lebih rendah lebih baik dalam menghindari *overfitting*, sehingga model lebih stabil dan dapat bekerja lebih baik pada data baru.



Gambar 4. 7 *Confusion Matrix Kernel* Linear C = 1

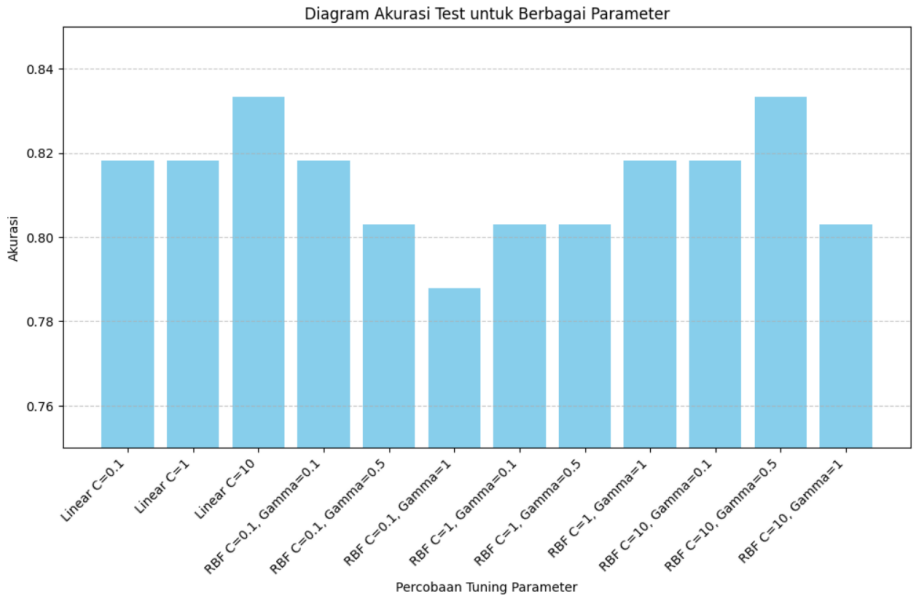
Sedangkan, akurasi tertinggi *kernel* RBF di pengujian skenario 1 terdapat pada konfigurasi *hyperparameter* C = 10 dan γ = 0.5 dengan nilai akurasi yaitu sebesar 84.84% beserta dengan *precision* 82%, *recall* 88%, dan F1-Score 85%., yang berasal dari nilai *confusion matrix* seperti yang tertera pada Gambar 4. 8.



Gambar 4. 8 *Confusion Matrix Kernel* rbf C = 10 dan γ = 0.5

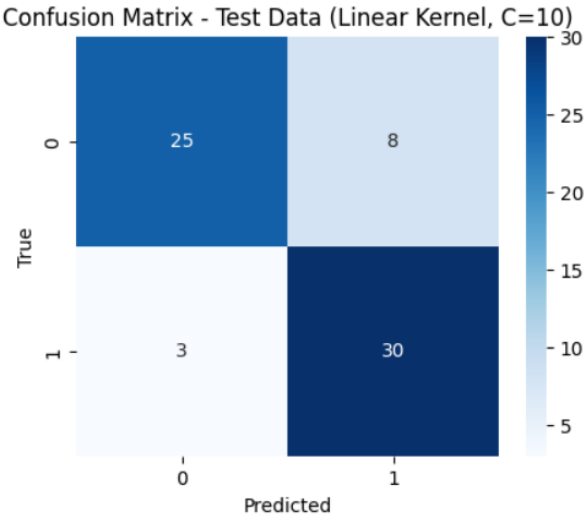
Kernel RBF dengan konfigurasi *hyperparameter* C = 10 dan γ = 0.5 menghasilkan akurasi sebesar 84.84%. Model ini juga mencatat nilai *precision* sebesar 82%, *recall* 88%, dan F1-Score 85%, yang menunjukkan keseimbangan yang baik dalam mengklasifikasikan data positif serta meminimalkan kesalahan negatif. *Hyperparameter* C mengontrol tingkat regulasi model, di mana nilai C yang lebih tinggi seperti 10 memungkinkan model menangkap pola lebih kompleks dalam data, meskipun berisiko *overfitting*. Selain itu, γ pada *kernel* RBF berfungsi untuk mengontrol kompleksitas batas keputusan, dengan nilai γ = 0.5 yang memberikan keseimbangan antara menangkap pola non-linear dalam data tanpa terlalu sensitif terhadap *noise*.

Pengujian pada skenario 2 menggunakan rasio pembagian data 80:20, dengan 264 data untuk *training* dan 66 data untuk testing. Beberapa sub-skenario diuji dengan berbagai nilai parameter untuk memperoleh hasil klasifikasi yang paling akurat. Pengujian skenario 1 memakan waktu sekitar 1 detik. Berikut adalah grafik hasil evaluasi pengujian skenario 2.



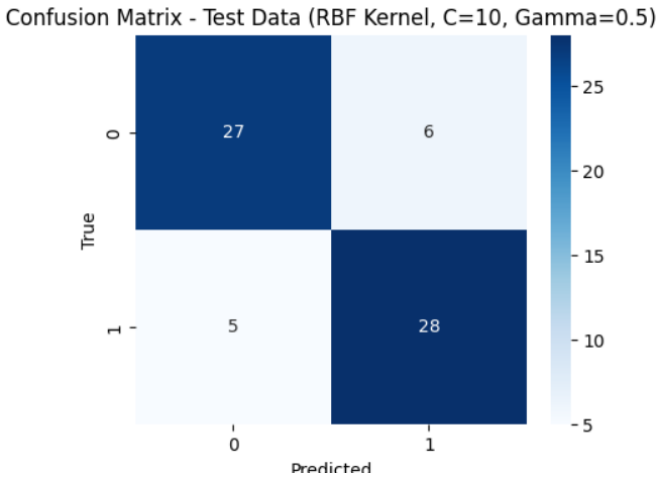
Gambar 4. 9 Diagram akurasi skenario 2

Pada Gambar 4. 9 menunjukkan hasil akurasi model pada pengujian skenario 1 menggunakan *confusion matrix*. Nilai akurasi tertinggi terdapat pada *kernel* Linear dengan konfigurasi *hyperparameter* C = 10, yaitu sebesar 83.33% beserta dengan *precision* 79%, *recall* 91%, dan F1-Score 85%. Yang berasal dari nilai *confusion matrix* seperti yang rpada Gambar 4. 10.



Gambar 4. 10 *Confusion Matrix Kernel* C = 10

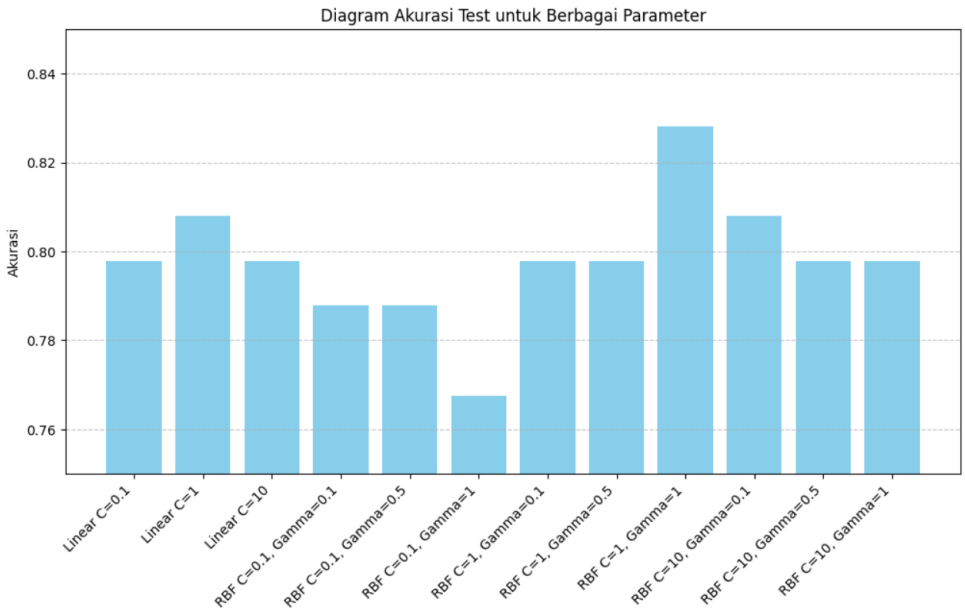
Sedangkan, akurasi tertinggi *kernel* RBF di pengujian skenario 1 terdapat pada konfigurasi *hyperparameter* C = 10 dan γ = 0.5 dengan nilai akurasi yaitu sebesar 83.33% beserta dengan *precision* 82%, *recall* 85%, dan F1-Score 84%. Yang berasal dari nilai *confusion matrix* seperti yang rpada Gambar 4. 11.



Gambar 4. 11 *Confusion Matrix Kernel* rbf C = 10 dan γ = 0.5

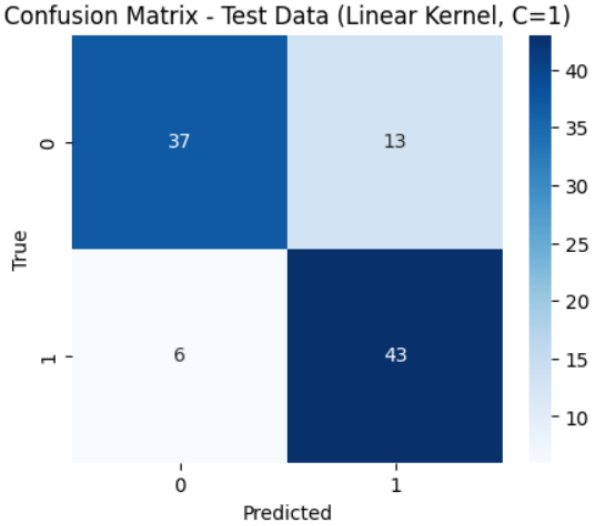
Pemilihan *kernel* Linear didasarkan pada keunggulannya dalam menghasilkan *recall* yang lebih tinggi (91% dibandingkan 85% pada *kernel* RBF), yang menunjukkan bahwa model lebih baik dalam mengidentifikasi kasus positif. Meskipun *precision*-nya sedikit lebih rendah, secara keseluruhan *kernel* Linear memberikan performa yang lebih stabil dan konsisten dibandingkan *kernel* RBF, sehingga menjadi pilihan yang lebih optimal dalam pengujian ini.

Pengujian pada skenario 3 menggunakan rasio pembagian data 70:30, dengan 231 data untuk *training* dan 99 data untuk testing. Beberapa sub-skenario diuji dengan berbagai nilai parameter untuk memperoleh hasil klasifikasi yang paling akurat. Pengujian skenario 1 memakan waktu sekitar 1 detik. Berikut adalah grafik hasil evaluasi pengujian skenario 3.



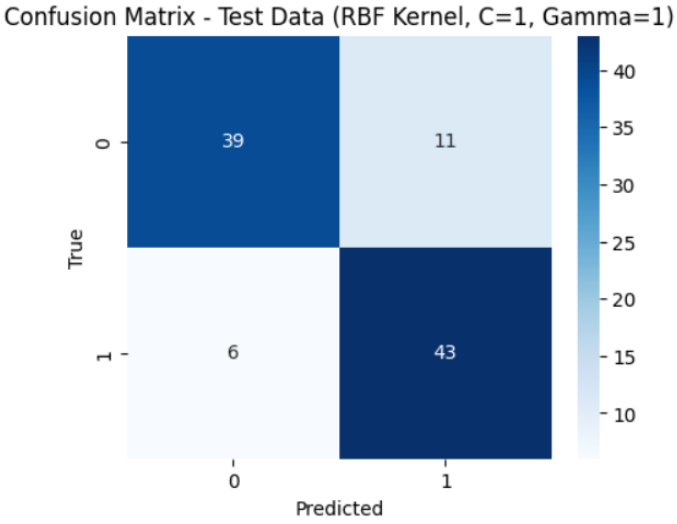
Gambar 4. 12 Diagram Akurasi Skenario 3

Pada Gambar 4. 12 menunjukkan hasil akurasi model pada pengujian skenario 1 menggunakan *confusion matrix*. Nilai akurasi tertinggi terdapat pada *kernel* Linear dengan konfigurasi *hyperparameter* C = 1, yaitu sebesar 80.80% beserta dengan *precision* 77%, *recall* 88%, dan F1-Score 82%. yang berasal dari nilai *confusion matrix* seperti yang tertera pada Gambar 4. 13.



Gambar 4. 13 *Confusion Matrix Kernel* rbf C = 1

Untuk, akurasi tertinggi *kernel* RBF di pengujian skenario 1 terdapat pada konfigurasi *hyperparameter* C = 1 dan γ = 1 dengan nilai akurasi yaitu sebesar 82.82% beserta dengan *precision* 80%, *recall* 88%, dan F1-Score 83%. yang berasal dari nilai *confusion matrix* seperti yang tertera pada Gambar 4. 14.



Gambar 4. 14 *Confusion Matrix Kernel* rbf C = 1 dan γ = 1

4.3.1 *Kernel* Pada Setiap Sekenario

Pemilihan *kernel* berpengaruh signifikan terhadap performa model dalam skenario 1 dengan pembagian data (90:10). *Kernel* Linear menunjukkan akurasi yang stabil di berbagai konfigurasi, mengindikasikan bahwa pemisahan data masih dapat dilakukan secara linear tanpa memerlukan transformasi kompleks. *Kernel* RBF juga memberikan hasil yang kompetitif, menunjukkan kemampuannya dalam menangkap pola non-linear dalam data.

Pada skenario ini, akurasi tertinggi diperoleh menggunakan *kernel* RBF, yang menandakan bahwa pendekatan non-linear lebih optimal dalam mengenali pola dalam data. Namun, perbedaan akurasi antara *kernel* RBF dan Linear tidak

terlalu signifikan, sehingga keduanya tetap menjadi pilihan yang kompetitif tergantung pada kebutuhan model.

Pengujian pada skenario 2 dengan pembagian data (80:20) juga menunjukkan bahwa pemilihan *kernel* memengaruhi performa model. *Kernel* Linear tetap menunjukkan akurasi yang stabil, mencerminkan bahwa pemisahan data dalam skenario ini masih cukup dapat dilakukan secara linear. Sementara itu, *kernel* RBF menunjukkan variasi akurasi yang lebih besar dibandingkan skenario sebelumnya, menandakan bahwa efektivitas transformasi non-linear bergantung pada konfigurasi parameter yang digunakan.

Dalam skenario ini, akurasi tertinggi diperoleh menggunakan *kernel* Linear, menunjukkan bahwa pendekatan linear lebih optimal dalam menangani data pada skenario ini. Hal ini mengindikasikan bahwa distribusi data masih memungkinkan pemisahan yang efektif menggunakan batas keputusan linear tanpa memerlukan transformasi non-linear tambahan.

Pada Pengujian skenario 3 dengan pembagian data (70:30) menunjukkan bahwa *kernel* RBF menghasilkan akurasi tertinggi dibandingkan *kernel* Linear. Hal ini mengindikasikan bahwa dalam skenario ini, pemisahan data lebih efektif menggunakan transformasi non-linear, memungkinkan model menangkap pola yang lebih kompleks dibandingkan pemisahan berbasis linear.\

Meskipun *kernel* Linear tetap menunjukkan performa yang stabil, hasil ini mengonfirmasi bahwa pada skenario ini, pendekatan non-linear lebih optimal dalam menangani distribusi data yang ada. Variasi akurasi pada *kernel* RBF juga

memperlihatkan bahwa efektivitasnya sangat bergantung pada konfigurasi parameter yang digunakan seperti tabel 4.11.

Tabel 4. 11 Rata-Rata Dari Setiap Skenario Pada *Kernel*

NO	*Kernel*	*Hyperparameter*		Evaluasi Akurasi			
		Cost	*Gamma*	Skenario 1	Skenario 2	Skenario 3	Rata-rata
1	Linear	0.1		78.78%	81.81%	79.79%	80.13%
2		1		78.78%	81.81%	80.80%	80.46%
3		10		78.78%	83.83%	79.79%	80.80%
4	RBF	0.1	0.1	78.78%	81.81%	78.78%	79.79%
5			0.5	78.78%	80.30%	78.78%	79.29%
6			1	78.78%	78.78%	76.76%	78.11%
7		1	0.1	78.78%	80.30%	79.79%	79.62%
8			0.5	75.75%	80.30%	79.79%	78.61%
9			1	81.81%	81.81%	82.82%	82.15%
10		10	0.1	78.78%	81.81%	80.80%	80.46%
11			0.5	84.84%	83.33%	79.79%	82.65%
12			1	81.81%	80.30%	79.79%	80.63%

Secara keseluruhan, hasil pengujian menunjukkan bahwa pemilihan *kernel* sangat bergantung pada karakteristik data dalam setiap skenario. *Kernel* Linear cenderung stabil dan optimal ketika data masih dapat dipisahkan secara linear, sementara *kernel* RBF lebih unggul dalam menangani pola yang lebih kompleks ketika distribusi data semakin beragam.

4.3.2 Cost Pada setiap Skenario

Pada skenario 1 dengan pembagian data (90:10). nilai *Cost* (C) mempengaruhi performa model. *kernel* Linear akurasi tertinggi diperoleh saat C = 1 dan C = 10, menunjukkan bahwa peningkatan C dapat membantu model lebih baik dalam memisahkan data. Sementara itu, pada *kernel* RBF, nilai C yang memberikan akurasi tertinggi juga berada pada C = 10, mengindikasikan bahwa

model dengan margin keputusan yang lebih ketat mampu menangkap pola dalam data dengan lebih baik.

Perbedaan ini menunjukkan bahwa pada skenario ini, pemilihan nilai C yang lebih besar lebih optimal untuk kedua *kernel*, terutama dalam meningkatkan akurasi klasifikasi. Namun, efeknya lebih terlihat pada *kernel* RBF, yang memiliki variasi akurasi lebih besar dibandingkan *kernel* Linear.

Pada skenario 2 dengan pembagian data (80:20), pengaruh nilai C terhadap performa model cukup terlihat. Pada *kernel* Linear, akurasi tertinggi diperoleh saat C = 10, menunjukkan bahwa peningkatan nilai C masih memberikan keuntungan dalam skenario ini.

Sementara itu, pada *kernel* RBF, akurasi tertinggi juga dicapai saat C = 10, menandakan bahwa model bekerja lebih optimal dengan nilai C yang lebih besar. Namun, variasi akurasi pada *kernel* RBF lebih beragam dibandingkan *kernel* Linear, menunjukkan bahwa sensitivitas terhadap C pada RBF lebih tinggi dibandingkan Linear.

Dan pada skenario 3 dengan pembagian data (70:30). Pengaruh nilai C terhadap performa model menunjukkan bahwa pada *kernel* Linear, akurasi tertinggi diperoleh saat C = 1. Hal ini menunjukkan bahwa dalam skenario ini, peningkatan nilai C pada *kernel* Linear tidak selalu berdampak positif terhadap akurasi model.

Pada *kernel* RBF, akurasi tertinggi juga diperoleh saat C = 1, yang mengindikasikan bahwa nilai C tersebut memberikan performa optimal dalam skenario ini. Hal ini menunjukkan bahwa model dapat menangkap pola data dengan

baik pada C = 1, sementara variasi akurasi tetap terlihat pada beberapa konfigurasi nilai C lainnya.

Secara umum, pada *kernel* Linear, performa model cenderung lebih stabil terhadap perubahan nilai C dibandingkan RBF. Hal ini terlihat dari akurasi yang relatif konsisten di berbagai skenario, tanpa adanya variasi besar seperti yang terjadi pada *kernel* RBF. Hal ini menunjukkan bahwa model dengan *kernel* Linear kurang sensitif terhadap perubahan nilai C dibandingkan dengan *kernel* RBF, yang lebih bergantung pada pemilihan *hyperparameter* yang tepat untuk mencapai performa optimal.

Sementara itu, pengaruh nilai C pada *kernel* RBF menunjukkan bahwa peningkatan nilai C dapat membantu model untuk lebih baik dalam memisahkan data, terutama pada skenario dengan proporsi data *training* yang lebih besar (90:10 dan 80:20). Namun, pada skenario dengan proporsi data *training* lebih kecil (70:30), nilai C yang lebih besar tidak selalu menghasilkan akurasi yang lebih tinggi, yang menunjukkan bahwa model bisa mengalami *overfitting* ketika nilai C terlalu besar. Dengan demikian, pemilihan nilai C yang optimal bergantung pada distribusi data dan proporsi pembagian data yang digunakan dalam skenario tertentu seperti pada tabel 4.12.

Tabel 4. 12 Pengaruh Cost

No.	*Kernel*	*Hyperparameter*		Evaluasi Akurasi			
		Cost	*Gamma*	Skenario 1	Skenario 2	Skenario 3	Rata-rata
1	Linear	0.1		78.78%	81.81%	79.79%	80.13%
2		1		78.78%	81.81%	80.80%	80.46%
3		10		78.78%	83.83%	79.79%	80.80%
4	RBF	0.1	0.1,0.5,1	78.78%	80.30%	78.11%	79.06%
5		1	0.1,0.5,1	78.78%	80.80%	80.80%	80.13%
6		10	0.1,0.5,1	81.05%	81.56%	80.30%	80.97%

4.3.3 Gamma Pada Setiap skenario

Pada skenario 1 dengan pembagian data (90:10). pemilihan nilai Gamma berpengaruh terhadap performa model serta Pada *kernel* RBF, kombinasi C = 10 dan Gamma = 0.5 menghasilkan akurasi tertinggi sebesar 84.84%, menunjukkan bahwa kombinasi ini memberikan pemisahan yang lebih optimal dalam menangkap pola data.

Sementara itu, pada C = 1, nilai Gamma = 1 menghasilkan akurasi lebih tinggi dibandingkan Gamma = 0.5, yang mengindikasikan bahwa dalam skenario ini, peningkatan Gamma dapat membantu model dalam menangkap pola yang lebih kompleks. Namun, saat C = 10, peningkatan Gamma dari 0.5 ke 1 justru menyebabkan penurunan akurasi. Hal ini menunjukkan bahwa meskipun Gamma yang lebih besar bisa meningkatkan performa dalam kondisi tertentu, ada batas optimal di mana nilai Gamma yang terlalu tinggi justru dapat mengurangi kemampuan model dalam melakukan generalisasi.

Pada skenario 2 dengan pembagian data (80:20), pengaruh nilai gamma terhadap performa model pada *kernel* RBF menunjukkan bahwa gamma 0.5 dengan C = 10 memberikan akurasi tertinggi. Sementara itu, pada C = 1, gamma 1 menghasilkan akurasi lebih tinggi dibandingkan gamma 0.5, yang menunjukkan bahwa pemilihan gamma mempengaruhi kemampuan model dalam menangkap pola data. Namun, saat C = 10, gamma 1 justru mengalami sedikit penurunan dibandingkan gamma 0.5, yang mengindikasikan bahwa peningkatan gamma tidak selalu berdampak positif terhadap performa model.

Pada skenario 3 dengan pembagian data (70:30), pengaruh nilai gamma terhadap performa model pada *kernel* RBF menunjukkan bahwa gamma 1 dengan C = 1 memberikan akurasi tertinggi. Selain itu, pada C = 1, gamma 1 memiliki performa lebih baik dibandingkan gamma 0.5 dan gamma 0.1, yang menunjukkan bahwa pemilihan gamma berpengaruh terhadap pola keputusan model.

Namun, saat C meningkat ke 10, akurasi pada gamma 1 mengalami sedikit penurunan dibandingkan gamma 1 pada C = 1, sementara gamma 0.1 justru memberikan akurasi lebih tinggi dibandingkan gamma lainnya pada C = 10. Hal ini mengindikasikan bahwa pada skenario ini, nilai C yang lebih tinggi tidak selalu meningkatkan akurasi, dan gamma 1 lebih efektif pada C yang lebih kecil, sedangkan gamma 0.1 lebih optimal ketika C lebih besar.

Secara umum pada tabel 4.13, pengaruh nilai Gamma menunjukkan bahwa pemilihan nilai Gamma yang tepat dapat meningkatkan kemampuan model dalam menangkap pola dalam data. Pada nilai C yang lebih kecil, peningkatan nilai Gamma dapat membantu meningkatkan akurasi, sedangkan pada nilai C yang lebih besar, pemilihan Gamma yang terlalu tinggi dapat menyebabkan model kehilangan kemampuan generalisasi. Oleh karena itu, keseimbangan antara nilai C dan Gamma sangat penting dalam menentukan performa optimal model SVM berbasis RBF.

Tabel 4. 13 Pengaruh Gamma

NO	*Kernel*	*Hyperparameter*		Evaluasi Akurasi			
		Skenario	Gamma	0.1	1	10	Rata-rata
1	RBF	1	0.1	78.78%	78.78%	78.78%	78.78%
2			0.5	78.78%	75.75%	84.84%	79.79%
3			1	78.78%	81.81%	81.81%	80.80%
4		2	0.1	81.81%	80.30%	81.81%	81.31%
5			0.5	80.30%	80.30%	83.33%	81.31%
6			1	78.78%	81.81%	80.30%	80.30%
7		3	0.1	78.78%	79.79%	80.80%	79.79%

8			0.5	78.78%	79.79%	79.79%	79.45%
9			1	76.76%	82.82%	79.79%	79.79%

4.4 Integrasi Islam

Penelitian ini mengambil objek risiko serangan jantung yang dalam Al-Qur'an terdapat ayat yang menjelaskan tentang jantung, yaitu pada surah *Al-Haqqah* ayat 46.

ثُمَّ لَقَطَعْنَا مِنْهُ الْوَتِيْنَۚ

"Kemudian benar-benar kami potong urat tali jantungnya" (Al-Haqqah 46).

Menurut Ibnu Katsir dalam Tafsir Ibnu Katsir Pembuluh nadi, dalam hal ini merupakan pembuluh darah besar atau dalam dunia medis disebut aorta yang tekanannya langsung dari kontraksi jantung dan volume darah yang besar, tersumbat atau terpotongnya aorta dapat mengakibatkan syok bahkan kematian (Katsir, 2015). Ayat ini menjelaskan bagaimana fungsi jantung dalam kehidupan yang dimana ketika jantung berhenti maka dapa mengakibatkan kematian.

Untuk mendeteksi risiko penyakit jantung, penelitian ini menggunakan metode *Support Vector Machine* (SVM) dalam klasifikasi. SVM adalah salah satu teknik dalam *machine learning* yang mampu menganalisis dan mengklasifikasikan data medis untuk menentukan apakah seseorang berisiko mengalami serangan jantung atau tidak.

Dalam konteks klasifikasi, Islam juga memiliki sistem pengelompokan yang tidak didasarkan pada faktor biologis, melainkan pada ketakwaan. Hal ini dijelaskan dalam Surah *Al-Hujurat* ayat 13:

يٰٓاَيُّهَا النَّاسُ اِنَّا خَلَقْنٰكُمْ مِّنْ ذَكَرٍ وَّاُنْثٰى وَجَعَلْنٰكُمْ شُعُوْبًا وَّقَبَاۤىِٕلَ لِتَعَارَفُوْاۚ اِنَّ اَكْرَمَكُمْ عِنْدَ اللّٰهِ اَتْقٰىكُمْۗ اِنَّ اللّٰهَ عَلِيْمٌ خَبِيْرٌ

"Wahai manusia, sesungguhnya Kami telah menciptakan kamu dari seorang laki-laki dan perempuan. Kemudian, Kami menjadikan kamu berbangsa-bangsa dan bersuku-suku agar kamu saling mengenal. Sesungguhnya yang paling mulia di antara kamu di sisi Allah adalah orang yang paling bertakwa. Sesungguhnya Allah Maha Mengetahui lagi Mahateliti."

Hasil penelitian ini diharapkan dapat menjadi pengingat bagi masyarakat untuk senantiasa menjaga kesehatan. Kesehatan memiliki peran penting dalam menjalankan ibadah, karena seorang Muslim yang sehat akan lebih mampu melaksanakan kewajiban seperti shalat, puasa, serta berbagai amal kebaikan lainnya dengan lebih optimal. Terkait pentingnya kesehatan Rasulullah shallallahu ‘alaihi wa sallam bersabda:

نِعْمَتَانِ مَغْبُونٌ فِيهِمَا كَثِيرٌ مِنَ النَّاسِ الصِّحَّةُ وَالْفَرَاغُ

"Dua kenikmatan yang sering dilupakan oleh kebanyakan manusia adalah kesehatan dan waktu luang." (HR. Al-Bukhari: 6412, at-Tirmidzi: 2304, Ibnu Majah: 4170)".

Ibnu Bathal menjelaskan bahwa makna hadits ini adalah seseorang tidak dikatakan memiliki waktu luang hingga ia juga memiliki badan yang sehat. Barangsiapa yang memiliki hal tersebut (waktu luang dan badan yang sehat) hendaknya ia bersemangat agar jangan sampai ia tertipu dengan meninggalkan syukur kepada Allah atas segala nikmat yang diberikan-Nya. Termasuk bersyukur kepada Allah adalah dengan melaksanakan segala perintah-Nya dan menjauhi segala larangan-Nya. Barangsiapa yang tidak bersyukur seperti itu maka ialah orang yang tertipu.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil membangun sistem klasifikasi penyakit serangan jantung menggunakan metode SVM dan RFE, dengan evaluasi menggunakan metrik akurasi, presisi, recall, dan F1-Score. Pengujian difokuskan pada pencarian kombinasi *kernel* dan *hyperparameter* terbaik berdasarkan rata-rata hasil *confusion matrix*. Model terbaik diperoleh pada skenario data 90:10 dengan *kernel* RBF, *Cost* = 10, dan Gamma = 0.5, menghasilkan akurasi 84.84%, presisi 82%, recall 88%, dan F1-Score 85%. Rata-rata akurasi tertinggi diperoleh dari *kernel Linear* sebesar 82.48%, sementara pembagian data 80:20 menunjukkan performa paling stabil dengan rata-rata akurasi 81.35%. Penelitian ini menunjukkan bahwa variasi *kernel* dan *hyperparameter* sangat mempengaruhi performa klasifikasi model.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, masih terdapat banyak kekurangan dalam penelitian ini, sehingga diperlukannya kiritik dan saran yang membangun agar dapat meningkatkan pada penelitian selanjutnya. Berdasarkan hasil kesimpulan yang telah diperoleh dari skenario pengujian, beberapa saran yang dapat diberikan untuk pengembangan dan penelitian selanjutnya adalah sebagai berikut:

1. Penggunaan dataset yang lebih beragam dari sumber yang berbeda agar dapat menilai kemampuan model dalam mengklasifikasikan pada populasi yang berbeda.
2. Teknik *preprocessing* yang digunakan dapat diperluas, misalnya dengan menguji teknik normalisasi lain, seleksi fitur tambahan selain RFE.
3. Metode yang digunakan dalam penelitian ini adalah *Support Vector Machine*, penelitian selanjutnya dapat menggunakan metode lainnya yang diharapkan mampu memberikan hasil yang lebih baik.

DAFTAR PUSTAKA

Abdurrahman, G. (2023). Klasifikasi Kanker Payudara Menggunakan Algoritma SVM dengan *Kernel* RBF, Linier, dan Sigmoid. *JUSTIFY : Jurnal Sistem Informasi Ibrahimy*, *2*(1), 74–80. https://doi.org/10.35316/justify.v2i1.3370

Alhabib, I. (2022). Komparasi Metode Deep Learning, Naïve Bayes Dan Random Forest Untuk Prediksi Penyakit Jantung. *INFORMATICS FOR EDUCATORS AND PROFESSIONAL : Journal of Informatics*, *6*(2), 176. https://doi.org/10.51211/itbi.v6i2.1881

Amrullah, S., Rosjidi, C., Dhesa, D., Wurjatmiko, A., & H. et al. (2022). Faktor Resiko Penyakit Infark Miokard Akut di Rumah Sakit Umum Dewi Sartika Kota Kendari. *Jurnal Ilmiah Karya Kesehatan*, *02*(Mei), 1–7. https://stikesks-kendari.e-journal.id/JIKK/article/view/467

Annur, H. (2018). Klasifikasi Masyarakat Miskin Menggunakan Metode Naive Bayes. *ILKOM Jurnal Ilmiah*, *10*(2), 160–165. https://doi.org/10.33096/ilkom.v10i2.303.160-165

Arif, S. N. N., Siregar, A. M., Faisal, S., & Juwita, A. R. (2024). Klasifikasi Penyakit Serangan Jantung Menggunakan Metode Machine Learning K-Nearest Neighbors (KNN) dan Support Vector Machine (SVM). *Jurnal Media Informatika Budidarma*, *8*(3), 1617. https://doi.org/10.30865/mib.v8i3.7844

Darujati, C., Gumelar, A. B., Informasi, S., Komputer, F. I., Surabaya, U. N., Bayes, N., Teks, K., Indonesia, B., & Learning, S. (2012). Pemanfaatan Teknik Supervised Untuk. *Text*, *16*(1), 1–8.

Dimsyiar M Al Hafiz, Khoirul Amaly, Javen Jonathan, M Teranggono Rachmatullah, & Rosidi. (2021). Sistem Prediksi Penyakit Jantung Menggunakan Metode Naive Bayes. *Jurnal Rekayasa Elektro Sriwijaya*, *2*(2), 151–157. https://doi.org/10.36706/jres.v2i2.29

Dwiasnati, S., & Devianto, Y. (2021). Optimasi Prediksi Bencana Banjir menggunakan Algoritma SVM untuk penentuan Daerah Rawan Bencana Banjir. *Prosiding SISFOTEK*, 202–207. http://seminar.iaii.or.id/index.php/SISFOTEK/article/view/283

Fais, M. A., Revano, M., Lubis, A., Aulia, A., Syafitri, I., Komputer, I., Matematika, F., & Alam, I. P. (2023). *Implementasi Algoritma Decision Tree untuk Klasifikasi Serangan Jantung*. *1*(4), 207–212. https://doi.org/10.59581/jusiik-widyakarya.v1i4.1895

Fremmuzar, P., & Baita, A. (2023). Uji *Kernel* SVM dalam Analisis Sentimen Terhadap Layanan Telkomsel di Media Sosial Twitter. *Komputika : Jurnal Sistem Komputer*, *12*(2), 57–66. https://doi.org/10.34010/komputika.v12i2.9460

Hamzah, M. A., & Othman, S. H. (2021). A Review of Support Vector Machine-based Intrusion Detection System for Wireless Sensor Network with Different *Kernel* Functions. *International Journal of Innovative Computing*, *11*(1), 59–67. https://doi.org/10.11113/ijic.v11n1.303

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

Katsir, T. I. (2015). *Tafsir Surat Al-Haqqah, ayat 44-52*. Ibnukatsironline. http://www.ibnukatsironline.com/2015/10/tafsir-surat-al-haqqah-ayat-44-52.html

Kemenkes. (2022a). *Penyakit Jantung Penyebab Utama Kematian, Kemenkes Perkuat Layanan Primer*. https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20220929/0541166/penyakit-jantung-penyebab-utama-kematian-kemenkes-perkuat-layanan-primer/

Kemenkes. (2022b). *Plak Aterosklerosis: Mungkinkah Menyusut?* https://yankes.kemkes.go.id/view_artikel/1276/plak-aterosklerosis-mungkinkah-menyusut

Maisat, Z., Darmawan, E., & Fauzan, A. (2023). Implementasi Optimasi Hyperparameter GridSearchCV Pada Sistem Prediksi Serangan Jantung Menggunakan SVM Implementation of GridSearchCV Hyperparameter Optimization in Heart Attack Prediction System Using SVM. *Unipdu*, *13*(1), 8–15.

Marthin Luter Laia, & Yudi Setyawan. (2020). Perbandingan Hasil Klasifikasi Curah Hujan Menggunakan Metode SVM dan NBC. *Jurnal Statistika Industri Dan Komputasi*, *5*(2), 51–61.

Medyanti, W. A., & Faisal, M. (2023). Early Prediction System for Employee Attrition Company "XYZ" Using Support Vector Machine Algorithm. *CESS (Journal of Computer Engineering, System and Science)*, *8*(2), 429. https://doi.org/10.24114/cess.v8i2.46494

Naufal, M. F., Arifin, T., & Wirjawan, H. (2023). Analisis Perbandingan Tingkat Performa Algoritma SVM, Random Forest, dan Naïve Bayes untuk Klasifikasi Cyberbullying pada Media Sosial. *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, *8*, 82. https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

Oryza Habibie Rahman, Gunawan Abdillah, & Agus Komarudin. (2021). Klasifikasi Ujaran Kebencian pada Media Sosial Twitter Menggunakan Support Vector Machine. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(1), 17–23. https://doi.org/10.29207/resti.v5i1.2700

Pradana, M. G., Saputro, P. H., & Wijaya, D. P. (2022). Komparasi Metode Support

Vector Machine Dan Naïve Bayes Dalam Klasifikasi Peluang Penyakit Serangan Jantung. *Indonesian Journal of Business Intelligence (IJUBI)*, *5*(2), 87. https://doi.org/10.21927/ijubi.v5i2.2659

Priyatno, A. M., & Widiyaningtyas, T. (2024). a Systematic Literature Review: Recursive Feature Elimination Algorithms. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *9*(2), 196–207. https://doi.org/10.33480/jitk.v9i2.5015

Putranto, A., Azizah, N. L., & Ratna Ika, A. I. (2023). Sistem Prediksi Penyakit Jantung Berbasis Web Menggunakan Metode SVM dan Framework Streamlit. *Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, *4*(2), 442–452. https://archive.ics.uci.edu/ml/datasets/heart+disease

Rahayu, S., Subekhi, A., Astuti, D., Widaningsih, I., Sartika, I., Nurhayani, Pangestuti, & Rafidah. (2020). Upaya Mewaspadai Serangan Jantung Melalui Pendidikan Kesehatan. *JMM (Jurnal Masyarakat Mandiri)*, *4*(2), 163–171.

Rahmi, F., Ferra Yanuar, & Yudiantri Asdi. (2024). Penggunaan Metode Support Vector Machine (SVM) dalam Mengidentifikasi Tingkat Keparahan Pada Kecelakaan Lalu Lintas. *Statistika*, *24*(1), 1–10. https://doi.org/10.29313/statistika.v24i1.1690

Ramadhan, N. G., & Khoirunnisa, A. (2021). Klasifikasi Data Malaria Menggunakan Metode Support Vector Machine. *Jurnal Media Informatika Budidarma*, *5*(4), 1580. https://doi.org/10.30865/mib.v5i4.3347

Ritonga, A. S., & Purwaningsih, E. S. (2018). Penerapan Metode Support Vector Machine (SVM) Dalam Klasifikasi Kualitas Pengelasan Smaw (Shield Metal Arc Welding). *Ilmiah Edutic*, *5*(1), 17–25.

Rohayati, N. L. W. (2020). Analisis Faktor yang Berhubungan dengan Serangan Jantung Berulang pada Pasien Post Percutaneous Transluminal Coronary Angioplasty. *Jurnal Kesehatan*, *10*, 25–39.

Saktiningtyastuti, F., & Astuti, S. L. D. (2017). Faktor Yang Mempengaruhi Serangan Jantung Berulang Pada Pasien AMI Di Ruang ICVCU RSUD Dr. Moewardi Tahun 2016. *(Jkg) Jurnal Keperawatan Global*, *2*(1), 35–45. https://doi.org/10.37341/jkg.v2i1.31

Santoso, A., Nugroho, A., & Sunge, A. S. (2022). Analisis Sentimen Tentang Mobil Listrik Dengan Metode Support Vector Machine Dan Feature Selection Particle Swarm Optimization. *Journal of Practical Computer Science*, *2*(1), 24–31. https://doi.org/10.37366/jpcs.v2i1.1084

Sartika, M., & Pujiastuti, R. A. (2020). Analisis Tingkat Kecemasan Pasien Yang Akan Menjalani Tindakan Kateterisasi Jantung Di Rumah Sakit Omni Pulomas Jakarta Timur. *Jurnal Ilmu Kesehatan Indonesia (JIKSI)*, *1*(1), 1–9. https://doi.org/10.57084/jiksi.v1i1.377

Setio, P. B. N., Saputro, D. R. S., & Bowo Winarno. (2020). Klasifikasi Dengan

Pohon Keputusan Berbasis Algoritme C4.5. *PRISMA, Prosiding Seminar Nasional Matematika*, *3*, 64–71.

Sirait, D. T. C., Adiwijaya, A., & ... (2019). Analisis Perbandingan Reduksi Dimensi Principal Component Analysis (PCA) Dan Partial Least Square (PLS) Untuk Deteksi Kanker Menggunakan Data. *E-Proceeding of Engineering*, *6*(2), 8570–8581. https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/viewFile/9828/9689

Supriyono, M. (2008). Faktor-Faktor Risiko Yang Berpengaruh Terhadap Kejadian Penyakit Jantung Koroner Pada Kelompok Usia < 45 Tahun (Studi Kasus Di RSUP Dr. Kariadi Dan RS Telogorejo Semarang) Oleh. *Universitas Diponegoro*, *1*(2), 275–283.

Tamlander, M., Mars, N., Pirinen, M., Widén, E., & Ripatti, S. (2022). Integration of questionnaire-based risk factors improves polygenic risk scores for human coronary heart disease and type 2 diabetes. *Communications Biology*, *5*(1), 158. https://doi.org/10.1038/s42003-021-02996-0

LAMPIRAN

SMO LINIEAR

```
def train_svm_smo(X, y, C, tol=1e-4, max_passes=20):
    m, n = X.shape
    alphas = np.zeros(m)
    b = 0
    passes = 0
    K = X @ X.T  # Linear kernel

    while passes < max_passes:
        num_changed_alphas = 0
        for i in range(m):
            Ei = b + np.sum(alphas * y * K[:, i]) - y[i]
            if (y[i]*Ei < -tol and alphas[i] < C) or (y[i]*Ei > tol and alphas[i] > 0):
                j = np.random.choice([x for x in range(m) if x != i])
                Ej = b + np.sum(alphas * y * K[:, j]) - y[j]

                alpha_i_old = alphas[i]
                alpha_j_old = alphas[j]

                if y[i] != y[j]:
                    L = max(0, alphas[j] - alphas[i])
                    H = min(C, C + alphas[j] - alphas[i])
                else:
                    L = max(0, alphas[i] + alphas[j] - C)
                    H = min(C, alphas[i] + alphas[j])

                if L == H:
                    continue

                eta = 2 * K[i, j] - K[i, i] - K[j, j]
                if eta >= 0:
                    continue

                alphas[j] -= y[j] * (Ei - Ej) / eta
                alphas[j] = np.clip(alphas[j], L, H)

                if abs(alphas[j] - alpha_j_old) < 1e-5:
                    continue

                alphas[i] += y[i] * y[j] * (alpha_j_old - alphas[j])

                b1 = b - Ei - y[i] * (alphas[i] - alpha_i_old) * K[i, i] - y[j] * (alphas[j]
- alpha_j_old) * K[i, j]
```

```
            b2 = b - Ej - y[i] * (alphas[i] - alpha_i_old) * K[i, j] - y[j] * (alphas[j]
- alpha_j_old) * K[j, j]

            if 0 < alphas[i] < C:
                b = b1
            elif 0 < alphas[j] < C:
                b = b2
            else:
                b = (b1 + b2) / 2

            num_changed_alphas += 1

        if num_changed_alphas == 0:
            passes += 1
        else:
            passes = 0

    w = ((alphas * y)[:, None] * X).sum(axis=0)
    return w, b

def predict_svm(X, w, b):
    return np.sign(X @ w + b)
```

SMO RBF

```
def rbf_kernel(X1, X2, gamma):
    X1 = np.array(X1)
    X2 = np.array(X2)
    sq_dists = np.sum(X1**2, axis=1).reshape(-1, 1) + np.sum(X2**2, axis=1) -
2 * np.dot(X1, X2.T)
    return np.exp(-gamma * sq_dists)

def train_svm_smo_rbf(X, y, C, gamma=0.5, tol=1e-3, max_passes=5):
    m, n = X.shape
    alphas = np.zeros(m)
    b = 0
    passes = 0
    K = rbf_kernel(X, X, gamma)

    while passes < max_passes:
        num_changed_alphas = 0
        for i in range(m):
            Ei = np.dot((alphas * y), K[:, i]) + b - y[i]
            if (y[i]*Ei < -tol and alphas[i] < C) or (y[i]*Ei > tol and alphas[i] > 0):
                j = np.random.choice([x for x in range(m) if x != i])
```

```
            Ej = np.dot((alphas * y), K[:, j]) + b - y[j]

            alpha_i_old = alphas[i]
            alpha_j_old = alphas[j]

            if y[i] != y[j]:
               L = max(0, alphas[j] - alphas[i])
               H = min(C, C + alphas[j] - alphas[i])
            else:
               L = max(0, alphas[i] + alphas[j] - C)
               H = min(C, alphas[i] + alphas[j])
            if L == H:
               continue

            eta = 2.0 * K[i, j] - K[i, i] - K[j, j]
            if eta >= 0:
               continue

            alphas[j] -= y[j] * (Ei - Ej) / eta
            alphas[j] = np.clip(alphas[j], L, H)

            if abs(alphas[j] - alpha_j_old) < 1e-5:
               continue

            alphas[i] += y[i]*y[j]*(alpha_j_old - alphas[j])

            b1 = b - Ei - y[i]*(alphas[i]-alpha_i_old)*K[i,i] - y[j]*(alphas[j]-
alpha_j_old)*K[i,j]
            b2 = b - Ej - y[i]*(alphas[i]-alpha_i_old)*K[i,j] - y[j]*(alphas[j]-
alpha_j_old)*K[j,j]

            if 0 < alphas[i] < C:
               b = b1
            elif 0 < alphas[j] < C:
               b = b2
            else:
               b = (b1 + b2)/2.0

            num_changed_alphas += 1

      passes = passes + 1 if num_changed_alphas == 0 else 0

   return {'alphas': alphas, 'b': b, 'X': X, 'y': y, 'gamma': gamma}

def predict_svm_rbf(X_test, model):
   K = rbf_kernel(X_test, model['X'], model['gamma'])
```

```
    decision = (model['alphas'] * model['y']) @ K.T + model['b']
    return np.sign(decision)
```