KLASIFIKASI INFEKSI HIV MENGGUNAKAN INTEGRASI *PRINCIPAL COMPONENT ANALYSIS* DAN *SUPPORT VECTOR MACHINE*

SKRIPSI

Oleh :
GIGIH AGUNG PRASETYO
NIM. 210605110138

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

KLASIFIKASI INFEKSI HIV MENGGUNAKAN INTEGRASI *PRINCIPAL COMPONENT ANALYSIS* DAN *SUPPORT VECTOR MACHINE*

SKRIPSI

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

Oleh :
GIGIH AGUNG PRASETYO
NIM. 210605110138

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

HALAMAN PERSETUJUAN

KLASIFIKASI INFEKSI HIV MENGGUNAKAN INTEGRASI *PRINCIPAL COMPONENT ANALYSIS* DAN *SUPPORT VECTOR MACHINE*

SKRIPSI

Oleh :
GIGIH AGUNG PRASETYO
NIM. 210605110138

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 16 Mei 2025

Pembimbing I,	Pembimbing II,
Dr. Agung Teguh Wibowo Almais, M.T	Dr. Totok Chamidy, M. Kom
NIP. 19860301 202321 1 016	NIP. 19691222 200604 1 001

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 INFEKSI HIV MENGGUNAKAN INTEGRASI *PRINCIPAL COMPONENT ANALYSIS* DAN *SUPPORT VECTOR MACHINE*

SKRIPSI

Oleh :
GIGIH AGUNG PRASETYO
NIM. 210605110138

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 16 Mei 2025

Susunan Dewan Penguji

Ketua Penguji	: A'la Syauqi, M.Kom NIP. 19771201 200801 1 007	( )
Anggota Penguji I	: Supriyono, M. Kom NIP. 19841010 201903 1 012	( )
Anggota Penguji II	: Dr. Agung Teguh Wibowo Almais, M.T NIP. 19860301 202321 1 016	( )
Anggota Penguji III	: Dr. Totok Chamidy, M. Kom NIP. 19691222 200604 1 001	( )

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	:	Gigih Agung Prasetyo
NIM	:	210605110138
Fakultas / Program Studi	:	Sains dan Teknologi / Teknik Informatika
Judul Skripsi	:	Klasifikasi Infeksi HIV Menggunakan Integrasi *Principal Component Analysis* Dan *Support Vector Machine*

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, 16 Mei 2025
Yang membuat pernyataan

Gigih Agung Prasetyo
NIM. 210605110138

MOTTO

... Hidup Seperti Larry ...

HALAMAN PERSEMBAHAN

Dengan hati yang penuh syukur ke hadirat Sang Maha Pencipta, atas limpahan rahmat, kesehatan, serta kekuatan yang tak terhingga, sehingga karya sederhana ini dapat terselesaikan dengan baik. Dengan segala kerendahan hati, penulis mempersembahkan skripsi ini kepada:

Ayah dan Ibu tercinta,

Bunari dan Sudarmi

Cahaya yang tak pernah padam dalam setiap langkah, penopang doa di setiap rintangan. Terima kasih atas cinta tanpa syarat, ketulusan tanpa pamrih, dan pengorbanan yang tak mungkin terbalas oleh apa pun di dunia ini.

Kakak-kakakku tersayang,

Nanang Bayu Ariyanto dan Deri Krisdianto

Sumber semangat dan inspirasi, tempat berpulang di tengah hiruk-pikuk dunia. Terima kasih atas dukungan dan kebersamaan yang menguatkan di setiap waktu.

Teman-teman seperjuangan,

Keluarga besar Teknik Informatika Angkatan 2021 “ASTER”

Terima kasih atas tawa yang menenangkan, diskusi yang mencerahkan, dan perjuangan yang kita lalui bersama. Semoga tali silaturahmi yang terjalin tetap kuat, dan kesuksesan senantiasa menyertai langkah kita ke depan.

KATA PENGANTAR

Bismillahirrahmaanirrahiim, Assalamu'alaikum wr. wb.

Segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu wa ta'ala atas limpahan rahmat, taufik, dan hidayah-Nya sehingga penulisan skripsi yang berjudul "Klasifikasi Infeksi HIV Menggunakan Integrasi *Principal Component Analysis* dan *Support Vector Machine*" ini dapat diselesaikan dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Penulis menyadari bahwa pencapaian ini tidak lepas dari bantuan, bimbingan, doa, serta dukungan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang, atas kepemimpinan dan kebijakan yang terus mendorong peningkatan mutu pendidikan dan riset di lingkungan universitas.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi, yang telah memberikan dukungan penuh dalam proses akademik dan non-akademik mahasiswa.
3. Dr. Fachrul Kurniawan, M.MT., IPM., selaku Ketua Program Studi Teknik Informatika, atas arahan dan semangatnya dalam membina mahasiswa menjadi insan akademik yang kompeten dan berakhlak.

4. Dr. Agung Teguh Wibowo Almais, M.T., selaku Dosen Pembimbing I, atas segala bimbingan, kesabaran, dan ilmu yang telah diberikan selama proses penyusunan skripsi ini.
5. Dr. Totok Chamidy, M.Kom., selaku Dosen Pembimbing II, atas masukan berharga dan dukungan yang sangat membantu dalam penyelesaian karya tulis ini.
6. A'la Syauqi, M.Kom., dan Supriyono, M.Kom., selaku dosen penguji, atas waktu, perhatian, dan evaluasi objektif yang sangat membantu penulis dalam melihat kelemahan serta kelebihan penelitian ini secara jernih.
7. Seluruh staf dan dosen Program Studi Teknik Informatika, yang telah menanamkan ilmu, semangat, dan nilai-nilai akademik selama proses perkuliahan.
8. Kedua orang tua dan kakak tercinta: Bapak Bunari, Ibu Sudarmi, Nanang Bayu Ariyanto, dan Deri Krisdianto, doa dan cinta kalian tidak pernah putus, bahkan saat penulis sendiri mulai kehilangan arah.
9. Pasangan penulis, Oktaviana Wahyu Setyoningtyas, yang dengan penuh kesabaran, kasih sayang, dan dukungan tanpa syarat selalu mendampingi di saat-saat tersulit.
10. Teman-teman seperjuangan di Kelas E Teknik Informatika, atas tawa, kerja kelompok, diskusi panjang, dan semangat belajar bersama. Kalian membuat perjalanan kuliah menjadi penuh kenangan yang tak tergantikan.

11. Seluruh anggota Divisi Intelektual HIMATIF Encoder, tempat penulis bertumbuh dalam ide, gagasan, dan kerja intelektual yang menyenangkan namun menantang.
12. Keluarga besar Teknik Informatika angkatan 2021 “ASTER”, atas solidaritas, semangat kolaborasi, dan kebersamaan yang tidak akan pernah terlupakan.
13. Teman-teman KKM 119 yang telah berbagi suka duka selama pengabdian masyarakat di Desa Ngingit, Tumpang, Malang. Dari kalian, penulis belajar tentang realitas, empati, dan arti berbagi ilmu dengan masyarakat.
14. Sahabat terbaik penulis, Muhammad Zakin Nada Raya dan Gianda Atthariq, rekan sevisi dalam pendirian GIG-IT, yang telah menjadi sumber inspirasi dalam kolaborasi dan pengembangan karya bersama.

Penulis menyadari bahwa karya ini masih jauh dari kata sempurna. Oleh karena itu, segala bentuk kritik dan saran yang membangun sangat penulis harapkan demi peningkatan kualitas di masa yang akan datang. Semoga skripsi ini dapat memberikan manfaat nyata, menjadi referensi bagi penelitian selanjutnya, dan berkontribusi dalam pengembangan teknologi berbasis nilai-nilai keislaman.

Malang, Mei 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Prasetyo, Gigih Agung. 2025. **Klasifikasi Infeksi HIV Menggunakan Integrasi *Principal Component Analysis* dan *Support Vector Machine*.** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Agung Teguh Wibowo Almais, M.T (II) Dr. Totok Chamidy, M.Kom.

.

Kata kunci: HIV, Klasifikasi, PCA, SVM.

Penelitian ini membahas klasifikasi infeksi HIV dengan pendekatan integratif antara *Principal Component Analysis* (PCA) dan *Support Vector Machine* (SVM). Permasalahan utama dalam klasifikasi infeksi HIV terletak pada tingginya kompleksitas data medis, yang sering kali terdiri dari banyak fitur dengan skala yang berbeda dan tingkat korelasi tinggi. Tujuan dari penelitian ini adalah untuk mereduksi dimensi data menggunakan PCA guna mengekstraksi fitur paling informatif, lalu mengklasifikasikannya menggunakan SVM agar diperoleh hasil klasifikasi yang akurat dan efisien. Penelitian ini menggunakan dataset "*AIDS Virus Infection*" dari *AIDS Clinical Trials Group Study 175* yang terdiri dari 2.139 data pasien dan 23 fitur. Tahapan penelitian meliputi normalisasi data, reduksi dimensi dengan PCA, pelatihan dan pengujian model dengan SVM menggunakan kernel RBF, serta evaluasi hasil menggunakan *confusion matrix*. Evaluasi dilakukan dengan tiga rasio pembagian data (90:10, 80:20, 70:30) dan menghasilkan nilai akurasi tertinggi pada skenario 1 dengan rata-rata akurasi 80.92%, *precision* 83.24%, *recall* 77.82%, dan *F1-score* 80.32%. Hasil penelitian menunjukkan bahwa kombinasi PCA dan SVM dapat mengklasifikasikan infeksi HIV secara efektif dan efisien, serta dapat dijadikan sebagai referensi dalam pengembangan sistem deteksi penyakit berbasis *machine learning*.

ABSTRACT

Prasetyo, Gigih Agung. 2025. **Classification of HIV Infections Using the Integration of Principal Component Analysis and Support Vector Machine.** Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Dr. Agung Teguh Wibowo Almais, M.T (II) Dr. Totok Chamidy, M.Kom.

Key words: HIV, Classification, PCA, SVM.

This research addresses the classification of HIV infection using an integrative approach between Principal Component Analysis (PCA) and Support Vector Machine (SVM). The main problem in HIV infection classification lies in the high complexity of medical data, which often consists of many features with different scales and high levels of correlation. The purpose of this study is to reduce the dimensionality of the data using PCA to extract the most informative features, and then classify them using SVM to obtain accurate and efficient classification results. This study uses the "AIDS Virus Infection" dataset from AIDS Clinical Trials Group Study 175 which consists of 2,139 patient data and 23 features. The research stages include data normalization, dimension reduction with PCA, training and model testing with SVM using RBF kernel, and evaluation of results using confusion matrix. The evaluation was conducted with three data sharing ratios (90:10, 80:20, 70:30) and resulted in the highest accuracy value in scenario 1 with an average accuracy of 80.92%, precision 83.24%, recall 77.82%, and F1-score 80.32%. The results show that the combination of PCA and SVM can classify HIV infection effectively and efficiently, and can be used as a reference in the development of machine learning-based disease detection systems.

مستخلص البحث

براسيتيو، جيجيه أجونج. 2025. تصنيف عدوى فيروس نقص المناعة البشرية باستخدام تكامل تحليل المكونات الرئيسية وآلة دعم المتجهات. أُطرُوحَة. برنامج دراسة هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانج. المشرف: (أ) الدكتور الدكتور توتوك شاميدي، م.كوم.

الكلمات الرئيسية: فيروس نقص المناعة البشرية، التصنيف، تحليل المكونات الرئيسية، آلية الدعم المتجهي.

تناقش هذه الدراسة تصنيف عدوى فيروس نقص المناعة البشرية باستخدام نهج تكاملي بين تحليل المكونات الرئيسية (PCA) وآلة الدعم المتجه (SVM). تكمن المشكلة الرئيسية في تصنيف الإصابة بفيروس نقص المناعة البشرية في التعقيد الكبير للبيانات الطبية، والتي تتكون في كثير من الأحيان من العديد من الميزات ذات المقاييس المختلفة ومستويات الارتباط العالية. الهدف للحصول SVM لاستخراج الميزات الأكثر إفادة، ثم تصنيفها باستخدام PCA من هذه الدراسة هو تقليل أبعاد البيانات باستخدام على نتائج تصنيف دقيقة وفعالة. استخدمت هذه الدراسة مجموعة بيانات "عدوى فيروس الإيدز" من دراسة مجموعة التجارب السريرية للإيدز رقم 175 والتي تتكون من 2139 بيانات مريض و23 سمة. تتضمن مراحل البحث تطبيع البيانات، وتقليل الأبعاد باستخدام PCA، وتقييم النتائج باستخدام مصفوفة الارتباك. تم إجراء RBF، باستخدام نواة SVM وتدريب النموذج واختباره باستخدام التقييم باستخدام ثلاث نسب لمشاركة البيانات (90:10، 80:20، 70:30) وأنتج أعلى قيمة دقة في السيناريو 1 بمتوسط دقة 80.32٪ F1 ودقة 83.24٪، واسترجاع 77.82٪، ونتيجة 80.92٪. وتظهر نتائج الدراسة أن الجمع بين تحليل المكونات يمكن أن يصنف عدوى فيروس نقص المناعة البشرية بشكل فعال وكفء، ويمكن استخدامه كمرجع في تطوير SVM الرئيسية وتقنية نظام الكشف عن الأمراض القائم على التعلم الآلي.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era modern yang terus berkembang, teknologi dan ilmu pengetahuan memiliki peran yang sangat penting dalam memecahkan berbagai permasalahan global, salah satunya adalah penyebaran infeksi HIV. HIV adalah singkatan dari *Human Immununodeficiency Virus* yang merusak sistem kekebalan tubuh manusia. AIDS adalah singkatan dari *Aquired Immunodeficiency Syndrome* yang merupakan kumpulan gejala dan tanda penyakit yang terjadi karena ketidakmampuan sistem pertahanan tubuh untuk berfungsi dengan baik (Sonawane & Barkade, 2023). Menurut WHO, HIV sudah membunuh 40.4 juta (32.9-51.3 juta) orang dan masih menjadi masalah utama bagi kesehatan masyarakat di seluruh dunia. Penularan terus terjadi di semua negara, dengan tren peningkatan infeksi baru di beberapa negara meskipun sebelumnya terjadi penurunan (Parums, 2024).

Klasifikasi dalam konteks infeksi HIV memiliki peran yang sangat penting, terutama dalam mendukung proses deteksi dini dan pengambilan keputusan medis yang tepat. Dengan klasifikasi yang akurat, tenaga medis dapat membedakan antara pasien yang terinfeksi dan tidak terinfeksi secara lebih cepat, sehingga memungkinkan intervensi yang lebih dini, efisien, dan tepat sasaran. Hal ini sangat krusial mengingat HIV merupakan penyakit yang progresif dan membutuhkan penanganan sejak dini untuk menekan laju penyebarannya serta meningkatkan kualitas hidup pasien. Selain itu, klasifikasi berbasis data juga mendukung

pengembangan sistem pendukung keputusan dalam layanan kesehatan, khususnya dalam menghadapi volume data medis yang besar dan kompleks. Oleh karena itu, pengembangan metode klasifikasi yang andal dan efektif menjadi kebutuhan mendesak dalam upaya penanggulangan HIV secara lebih sistematis dan berbasis teknologi (Faisah et al., 2022).

Dalam hal ini ilmu pengetahuan sangat dibutuhkan untuk memperoleh wawasan yang luas. Segala urusan dari semua bidang tidak terlepas dari ilmu yang harus dipelajari dan telah diberikan Allah Subhanahu wa ta'ala seperti yang tercantum dalam Al Qur'an surah At-Talaq ayat 12 sebagai berikut

اللَّهُ الَّذِي خَلَقَ سَبْعَ سَمَاوَاتٍ وَمِنَ الْأَرْضِ مِثْلَهُنَّ يَتَنَزَّلُ الْأَمْرُ بَيْنَهُنَّ لِتَعْلَمُوا أَنَّ اللَّهَ عَلَىٰ كُلِّ شَيْءٍ قَدِيرٌ وَأَنَّ اللَّهَ قَدْ أَحَاطَ بِكُلِّ شَيْءٍ عِلْمًا

"Allah yang menciptakan tujuh langit dan dari (penciptaan) bumi juga serupa. Perintah Allah berlaku padanya, agar kamu mengetahui bahwa Allah Mahakuasa atas segala sesuatu, dan ilmu Allah benar-benar meliputi segala sesuatu"(QS. At-Talaq:12)

Dalam Tafsir Al-Misbah, M. Quraish Shihab (Shihab, 2021) menjelaskan bahwa ayat ini menekankan betapa luas dan mendalamnya kekuasaan serta ilmu Allah Subhanahu wa ta'ala yang meliputi seluruh alam semesta, dari langit hingga bumi, termasuk segala aspek kehidupan manusia. Pengetahuan yang dimiliki oleh umat manusia, baik itu dalam bidang teknologi, kesehatan, maupun ilmu pengetahuan lainnya, adalah bagian dari anugerah Allah Subhanahu wa ta'ala yang harus dimanfaatkan dengan sebaik-baiknya. Ayat ini mengingatkan bahwa ilmu pengetahuan merupakan sarana untuk memahami dan mengelola dunia, termasuk

dalam menghadapi masalah-masalah besar seperti penyakit-penyakit menular, misalnya HIV/AIDS.

Lebih lanjut, Quraish Shihab menambahkan bahwa ilmu yang diberikan oleh Allah Subhanahu wa ta'ala adalah amanah yang harus dijaga dan digunakan dengan penuh tanggung jawab. Dalam konteks ini, manusia diajak untuk menggunakan ilmu dan teknologi demi kebaikan umat, seperti dalam penanggulangan penyakit atau pengembangan ilmu kesehatan. Namun, meskipun manusia berusaha dengan keras, penting untuk selalu diingat bahwa segala solusi dan pengetahuan yang ada berasal dari kekuasaan dan kehendak Allah Subhanahu wa ta'ala. Oleh karena itu, usaha manusia dalam memanfaatkan ilmu pengetahuan harus senantiasa disertai dengan kesadaran akan keterbatasannya dan kebergantungannya pada Allah Subhanahu wa ta'ala.

Sejalan dengan itu, sebagai umat Muslim, kita juga diajarkan untuk senantiasa mencari dan mengembangkan ilmu pengetahuan. Sebagaimana yang diajarkan dalam sebuah hadits Nabi yang diriwayatkan oleh Ibnu Majah No. 224 yang berbunyi :

طَلَبُ الْعِلْمِ فَرِيْضَةٌ عَلَى كُلِّ مُسْلِمٍ

"*Menuntut ilmu itu wajib atas setiap Muslim*" (*H.R. Ibnu Majah. Darani, 2021*)

Kemajuan teknologi informasi, khususnya dalam analisis data, telah membuka peluang baru untuk mengklasifikasikan dan mengantisipasi berbagai masalah kesehatan (Setiaji & Pramudho, 2022). Salah satu metode yang berkembang pesat adalah PCA dan SVM. PCA digunakan untuk mereduksi dimensi

data, mempermudah analisis, dan mengurangi kompleksitas model sambil tetap mempertahankan informasi penting. Hal ini sangat relevan dalam klasifikasi infeksi HIV, di mana data medis seringkali memiliki banyak variabel. Dengan mereduksi dimensi, PCA membantu fokus pada fitur-fitur yang paling relevan (Eliyanto & Sugiyarto, 2020). Namun, keterbatasan utama dalam penerapan PCA adalah bahwa teknik ini hanya efektif jika data memiliki fitur yang cukup banyak untuk diproses, dan menghasilkan informasi yang tereduksi tanpa mengurangi nilai prediktifnya (Hediyati & Suartana, 2021). Sementara itu, SVM dikenal efektif dalam klasifikasi, terutama dengan dataset yang kompleks dan jumlah data terbatas (Abdillah, 2019), namun tantangan utamanya adalah risiko *overfitting*, terutama jika dataset yang digunakan tidak cukup representatif atau memiliki *noise* yang tinggi (Firmansyach et al., 2023).

Integrasi teknik PCA dan SVM dalam konteks ini juga sangat terkait dengan sistem informasi, yang memainkan peran penting dalam pengelolaan dan pengolahan data yang besar dan kompleks. Data yang tersedia dalam bentuk besar dan kompleks, seperti dataset yang diambil dari *Kaggle*, memerlukan sistem informasi yang efisien untuk mengelola, memproses, dan menganalisisnya. Sistem informasi yang baik akan memungkinkan pengolahan data yang lebih cepat dan akurat, serta mendukung pengambilan keputusan yang berbasis data. Dalam pengolahan dan analisis data ini, peran sistem informasi sangat vital dalam menyediakan infrastruktur yang memungkinkan teknik-teknik seperti PCA dan SVM untuk diterapkan secara optimal, sehingga menghasilkan klasifikasi yang lebih akurat dan dapat diandalkan (Ritonga & Muhandhis, 2021).

Salah satu tantangan dalam klasifikasi infeksi HIV adalah menangani kompleksitas data yang tinggi agar data tersebut tetap dapat dianalisis secara efektif. Data medis sering kali memiliki dimensi yang besar, sehingga peneliti memerlukan metode yang dapat mereduksi jumlah fitur tanpa kehilangan informasi penting. Metode PCA dapat menyederhanakan data, sedangkan SVM dikenal memiliki kemampuan yang baik dalam melakukan klasifikasi. Oleh karena itu, penelitian ini berfokus pada pemanfaatan integrasi PCA dan SVM untuk menangani klasifikasi infeksi HIV secara lebih optimal. Melalui pendekatan ini, peneliti berharap dapat memperoleh strategi pemrosesan data yang lebih efisien, sehingga kecepatan dan akurasi analisis data pasien dapat meningkat. Mengingat angka infeksi HIV yang masih tinggi serta pentingnya metode analisis yang tepat, maka penelitian ini bertujuan untuk mengembangkan model klasifikasi yang dapat mendukung upaya deteksi dan pencegahan HIV secara lebih optimal.

1.2 Rumusan Masalah

Bagaimana pengaruh penerapan teknik PCA terhadap peningkatan efisiensi dan kinerja algoritma SVM dalam melakukan proses klasifikasi dan prediksi terhadap data infeksi HIV?

1.3 Batasan Masalah

Berikut adalah batasan masalah yang digunakan dalam penelitian ini:

a) Data diambil dari *AIDS Clinical Trials Group Study* 175 yang diambil dari *Kaggle* (*Kaggle*: https://www.kaggle.com/datasets/aadarshvelu/aids-virus-infection-prediction/data).

b) Jumlah fitur yang digunakan dikurangi menggunakan pendekatan *variance* 95%.

1.4 Tujuan Penelitian

Adapun, tujuan dari penelitian ini mencakup hal-hal berikut:

a) Mengimplementasikan teknik PCA untuk mereduksi dimensi data dalam klasifikasi infeksi HIV.
b) Menggunakan metode SVM untuk melakukan klasifikasi infeksi HIV berdasarkan data yang telah melalui proses reduksi dimensi.
c) Mengevaluasi kinerja model yang dihasilkan berdasarkan metrik akurasi, *precision*, *recall*, dan *F1-score* untuk memahami efektivitas pendekatan yang diterapkan.

1.5 Manfaat Penelitian

Manfaat penelitian ini mencakup beberapa aspek yang diharapkan:

a) Menyediakan pendekatan klasifikasi infeksi HIV yang lebih terstruktur dengan memanfaatkan reduksi dimensi untuk meningkatkan efisiensi analisis data.
b) Memberikan pemahaman yang lebih baik tentang bagaimana integrasi PCA dan SVM dapat diterapkan dalam pengolahan data medis yang kompleks.
c) Menjadi referensi bagi pengembangan metode klasifikasi berbasis *machine learning* dalam analisis data kesehatan.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Sebelum mendalami penelitian ini, penting untuk memahami penelitian terdahulu terkait klasifikasi data kesehatan menggunakan *machine learning*. Penelitian oleh Aden Wahyu P. Samudra dkk. (2022), membahas klasifikasi HIV/AIDS menggunakan aplikasi *Rapid Miner* dengan algoritma *K-Means Clustering*. Data diperoleh dari *MoleculeNet* dan diolah menggunakan fitur senyawa kimia *SMILES* dengan *library* RDKIT, lalu dibagi menjadi dua set (70:30 dan 80:20) untuk normalisasi dan pemilihan atribut. Hasilnya, penelitian ini berhasil mengelompokkan data pasien menjadi beberapa klaster dan mengidentifikasi status HIV sebagai positif (1) atau negatif (0). Temuan ini menunjukkan kemampuan model dalam mengklasifikasikan status kesehatan pasien secara akurat (Samudra et al., 2022).

Selanjutnya, penelitian oleh Gusti Eka Yuliastuti dkk. (2022), bertujuan untuk mengklasifikasi penyakit menular seksual (PMS) menggunakan algoritma *Naïve Bayes*. Penelitian ini dilatarbelakangi oleh meningkatnya kasus PMS di Indonesia, khususnya di Kota Malang, dengan data yang diperoleh dari salah satu puskesmas di daerah tersebut. Sistem pakar yang dikembangkan memanfaatkan algoritma *Naïve Bayes* untuk mengklasifikasi PMS berdasarkan gejala pasien melalui tahapan seperti pengumpulan data, *pre-processing*, klasifikasi, dan evaluasi. Hasil penelitian menunjukkan akurasi sebesar 76.67%, menandakan

potensi algoritma ini dalam mendeteksi PMS secara dini. Temuan ini diharapkan mendukung penanganan PMS yang lebih cepat dan efektif melalui pemanfaatan teknologi informasi (Yuliastuti et al., 2022).

Penelitian lain yang dilakukan oleh Indra Maulana dkk. (2024), bertujuan mengoptimalkan akurasi model pembelajaran mesin dalam klasifikasi tumor otak dengan menggunakan PCA sebagai teknik reduksi dimensi. Penelitian ini melibatkan algoritma seperti *Logistic Regression*, *Random Forest*, SVM, KNN, dan *Naïve Bayes* untuk menganalisis dataset berisi 3.264 citra MRI tumor otak. Proses penelitian mencakup *pre-processing* gambar, penskalaan fitur, penerapan PCA, dan implementasi model. Hasilnya menunjukkan bahwa PCA meningkatkan akurasi SVM dari 81% menjadi 83% dan KNN dari 68% menjadi 71%, meskipun menurunkan akurasi *Logistic Regression* dan *Naïve Bayes*. Studi ini menyoroti pentingnya memilih algoritma dan metode *pre-processing* yang tepat dalam klasifikasi citra medis serta perlunya eksplorasi lebih lanjut terhadap teknik reduksi dimensi. (Maulana et al., 2024).

Kemudian, penelitian oleh Rian Oktafiani dan Enny Itje Sela (2024), bertujuan meningkatkan akurasi klasifikasi kanker payudara dengan menggabungkan PCA dan *Synthetic Minority Oversampling Technique* (SMOTE). Dataset yang digunakan adalah *Wisconsin Breast Cancer Diagnostic* (WBCD), dengan algoritma *Random Forest* (RF) dan SVM sebagai model klasifikasi. PCA digunakan untuk reduksi dimensi, sementara SMOTE menyeimbangkan kelas mayoritas dan minoritas. Hasil penelitian menunjukkan bahwa SVM dengan kernel RBF dan PCA (n-komponen = 6) mencapai akurasi terbaik sebesar 99.07%,

sedangkan RF memperoleh akurasi 98.32%. Penelitian ini menyimpulkan bahwa kombinasi PCA dan SMOTE lebih efektif dalam meningkatkan kinerja SVM dibandingkan RF untuk klasifikasi kanker payudara (Oktafiani, 2024).

Selain itu, penelitian oleh Kasmawati dan Rokhana Dwi Bekti (2019), membahas klasifikasi status HIV di Rumah Sakit Tiom, Kabupaten Lanny Jaya, Papua, menggunakan metode SVM dan *Regresi Logistik Biner*. Berdasarkan data tahun 2018, dari 150 pasien, 25% positif HIV dan 75% negatif, dengan variabel independen seperti jenis kelamin, umur, status pernikahan, pendidikan, daerah tempat tinggal, dan pendapatan. SVM dengan kernel linier dan parameter *cost* (C) 0,1 menghasilkan error klasifikasi sebesar 26.87%, sementara regresi logistik menemukan hanya tingkat pendidikan yang signifikan memengaruhi status HIV. SVM lebih unggul dibandingkan regresi logistik, dengan nilai CCR 93.75% pada *data testing*. Penelitian ini menyimpulkan bahwa SVM adalah metode terbaik untuk klasifikasi status HIV dalam kasus ini (Bekti, 2019).

Tabel 2.1 menyajikan rangkuman penelitian terdahulu yang membahas penerapan berbagai metode klasifikasi, seperti *Rapid Miner*, *Naive Bayes*, PCA, dan SVM, untuk mengklasifikasikan data terkait HIV/AIDS, tumor otak, kanker, dan status HIV. Penelitian-penelitian ini bertujuan untuk meningkatkan akurasi model klasifikasi, dengan hasil yang bervariasi, mulai dari akurasi 42% hingga 99%. Usulan penelitian ini berfokus pada klasifikasi infeksi HIV menggunakan metode PCA dan SVM.

Tabel 2.1 Penelitian Terdahulu

No.	Referensi	Topik	Metode	Tujuan	Hasil
1.	(Samudra et al., 2022)	Klasifikasi HIV/AIDS	*Rapid Miner*	Penggunaan aplikasi *RapidMiner* untuk klasifikasi HIV/AIDS	Model klasifikasi berhasil mengidentifikasi data terkait aktivitas HIV, di mana status HIV diinterpretasikan sebagai positif (1) atau negatif (0).
2.	(Yuliastuti et al., 2022)	Klasifikasi Penyakit Menular Seksual	*Naïve Bayes*	Penggunaan algoritma *Naïve Bayes* untuk klasifikasi penyakit menular seksual	Hasil akurasinya mendapatkan akurasi sebesar 76.67%
3.	(Maulana et al., 2024)	Klasifikasi Tumor Otak	PCA	Optimisasi akurasi model *machine learning* dalam klasifikasi tumor otak dengan PCA	Hasil akurasinya mendapatkan akurasi dari SVM dari 81% menjadi 83%, KNN dari 68% menjadi 71%, LR dari 77% menjadi 69%, dan *Naïve Bayes* dari 49% mejadi 42%
4.	(Oktafiani, 2024)	Klasifikasi Kanker Payudara	PCA, SMOTE, *Random Forest*, SVM	Klasifikasi kanker payudara menggunakan PCA, SMOTE, *Random Forest*, dan SVM	Hasil akurasinya mendapatkan akurasi dari SVM 99.07%, dan *Random Forest* 98.32%
5.	(Bekti, 2019)	Klasifikasi Status HIV	SVM, *Regresi Logistik Biner*	Klasifikasi status HIV dengan metode SVM dan *regresi logistik biner*	Klasifikasi *support vector machine* dengan menggunakan kernel *linier* dengan parameter *cost* (C) sebesar 0.1 dan *error* sebesar 0.268657.
Usulan Penelitian					
6.	(Prasetyo, 2024)	Klasifikasi Infeksi HIV	PCA dan SVM	Klasifikasi infeksi HIV dengan metode PCA dan SVM	

Keterbaruan dalam penelitian ini terletak pada penggunaan PCA yang dipadukan dengan SVM untuk mereduksi dimensi data serta meningkatkan akurasi dan efisiensi model, di mana pada penelitian terdahulu belum ada yang menggunakan kombinasi metode PCA dan SVM dalam klasifikasi infeksi HIV. Dengan pendekatan ini, diharapkan dapat diperoleh model yang lebih akurat dan efisien, sehingga dapat memberikan kontribusi signifikan dalam pengembangan sistem prediksi infeksi HIV.

2.2 HIV dan AIDS

HIV adalah virus yang menyerang sistem kekebalan tubuh, khususnya sel CD4 yang memiliki peran penting dalam melawan infeksi. HIV melemahkan sistem kekebalan dengan merusak sel CD4, sehingga tubuh menjadi rentan terhadap infeksi dan penyakit. Ketika jumlah sel CD4 dalam tubuh turun secara signifikan, infeksi HIV bisa berkembang menjadi AIDS, yaitu tahap akhir dari infeksi HIV. AIDS ditandai dengan rusaknya sistem kekebalan tubuh sehingga tubuh lebih rentan terhadap berbagai infeksi oportunistik dan kanker yang jarang terjadi pada orang dengan sistem kekebalan tubuh yang sehat (Lela et al., 2022).

Menurut laporan terbaru dari WHO pada tahun 2024, HIV/AIDS telah menyebabkan kematian lebih dari 40 juta orang di seluruh dunia, dengan sekitar 37 juta orang yang masih hidup dengan HIV. Data di Indonesia menunjukkan tren peningkatan infeksi HIV, terutama di kalangan populasi kunci, seperti pekerja seks, pengguna narkoba suntik, dan pasangan mereka (Parums, 2024). Untuk menangani HIV, upaya pencegahan, deteksi dini, dan perawatan berkelanjutan sangat penting. Teknologi *machine learning* dalam menganalisis data medis untuk deteksi dini infeksi HIV dapat mendukung upaya ini, khususnya dalam klasifikasi data yang relevan dengan status infeksi pasien, yang penting bagi pengembangan kebijakan kesehatan publik dan strategi pengendalian (Balzer et al., 2020).

2.3 *Split Data*

Dalam penerapan *machine learning*, pembagian dataset menjadi da*ta training* dan *data testing* merupakan tahap penting untuk memastikan model dapat belajar dengan baik sekaligus dievaluasi secara objektif. Proses ini dikenal sebagai

split data, di mana dataset dibagi ke dalam dua atau lebih subset untuk memisahkan data yang digunakan dalam proses pelatihan dan pengujian model (Uçar et al., 2020). Tidak ada aturan baku dalam menentukan rasio pembagian data, tetapi beberapa rasio umum yang digunakan dalam penelitian *machine learning* adalah sebagai berikut:

a) 90:10 → 90% data digunakan untuk pelatihan, 10% untuk pengujian. Biasanya digunakan ketika jumlah data terbatas, sehingga model dapat belajar lebih banyak. Namun, proporsi *data testing* yang kecil bisa berisiko dalam generalisasi model.
b) 80:20 → 80% data digunakan untuk pelatihan, 20% untuk pengujian. Ini merupakan rasio yang sering digunakan dalam banyak studi karena memberikan keseimbangan antara pelatihan dan evaluasi model.
c) 70:30 → 70% data digunakan untuk pelatihan, 30% untuk pengujian. Digunakan ketika ingin memiliki lebih banyak *data testing* agar evaluasi model lebih akurat.

Pemilihan rasio *split data* dapat bergantung pada ukuran dataset, kompleksitas model, serta kebutuhan eksperimen. Dalam eksperimen skenario penelitian ini, tiga rasio *split data* digunakan untuk menganalisis bagaimana perbedaan proporsi data pelatihan dan pengujian dapat mempengaruhi performa klasifikasi infeksi HIV menggunakan PCA dan SVM.

2.4 *Principal Component Analysis* (PCA)

PCA adalah metode aktual yang digunakan untuk mereduksi dimensi data kompleks dengan mempertahankan informasi yang paling penting. Teknik ini

bermanfaat terutama untuk dataset berukuran besar dan berdimensi tinggi yang sulit dianalisis secara langsung. PCA bekerja dengan mentransformasikan variabel-variabel yang berkorelasi menjadi sejumlah variabel baru yang tidak berkorelasi, yang dikenal sebagai komponen utama. Komponen ini diurutkan berdasarkan seberapa besar variasi data yang bisa dijelaskan oleh setiap komponen, dengan komponen pertama menjelaskan variasi terbesar, diikuti oleh komponen-komponen berikutnya (Diba et al., 2023). Berikut adalah langkah-langkah dari PCA:

A. Standarisasi data

Standarisasi data adalah proses untuk mengubah nilai-nilai data agar memiliki skala yang sama, dengan rata-rata 0 dan deviasi standar 1. Proses ini penting untuk memastikan bahwa semua variabel memiliki kontribusi yang setara dalam model, sehingga dapat meningkatkan akurasi dan efisiensi perhitungan, terutama saat menghitung korelasi antar data. Standarisasi juga membantu menghindari bias yang disebabkan oleh perbedaan skala antar variabel.

B. Menghitung matriks kovarians

Sebelum melakukan perhitungan matriks kovarians, perlu dipahami terlebih dahulu konsep dasar mengenai korelasi (*covariance*) antara dua variabel. Korelasi ini mengukur hubungan antara dua atribut, yang kemudian dapat digunakan untuk mendapatkan nilai *eigen* dan vektor *eigen*, yang penting dalam proses analisis data, seperti pada metode PCA.

1) Varian atribut

$$var(A_1) = \frac{\sum_{i-1}^{n}(x_i - \bar{x})^2}{n-1} \quad (2.1)$$

$$var(A_2) = \frac{\sum_{i-1}^{n}(y_i - \bar{y})^2}{n-1} \quad (2.2)$$

2) Kovarian dua atribut

$$cov(A_1, A_2) = \frac{\sum_{i-1}^{n}(x_i - \bar{x}) - (y_i - \bar{y})}{n-1} \quad (2.3)$$

Keterangan:

a) x_i : data ke-i
b) $\bar{x}$: nilai rata-rata dari seluruh nilai x
c) y_i : data ke-i
d) $\bar{y}$: nilai rata-rata dari seluruh nilai y
e) n : banyaknya data

3) Matriks kovarian

$$C = \begin{pmatrix} cov(A_1, A_1) & cov(A_1, A_2) \\ cov(A_2, A_1) & cov(A_2, A_2) \end{pmatrix} \quad (2.4)$$

C. Menghitung nilai *eigen*

Eigenvalue dalam PCA berperan untuk menyatakan seberapa besar keragaman yang mampu dijelaskan oleh suatu variabel PC. Semakin besar nilai *eigenvalue*, semakin banyak varians data yang dapat dijelaskan oleh komponen tersebut, yang berarti komponen tersebut lebih penting dalam representasi data. Jika M adalah matriks $m \times m$, maka setiap λ memenuhi persamaan:

$$Mv = \lambda v \quad (2.5)$$

Sehingga setiap nilai *eigenvalue* harus memenuhi persamaan determinan dibawah ini:

$$|M - \lambda \mathrm{I}| = 0 \qquad (2.6)$$

Keterangan:

1) M : matriks kovarians
2) v : *eigenvector*
3) λ : *eigenvalue*
4) I : matriks identitas

D. Menghitung *principal component*

Setelah nilai eigenvalue dan eigenvector sudah diketahui, maka PC dapat dihitung dengan cara mengurutkan nilai *eigen* dari yang terbesar ke terkecil. Nilai *eigen* terbesar akan memiliki kontribusi paling besar terhadap varians data, sehingga komponen utama ini menjadi representasi yang paling signifikan dari data yang ada. Selanjutnya, komponen utama ini digunakan untuk mereduksi dimensi data sambil mempertahankan informasi yang paling penting.

E. Reduksi dimensi

Setelah nilai *eigen* diurutkan, tidak semua variabel PC akan dipilih. Hanya PC yang mempunyai nilai persentase > 80% atau boleh membuat asumsi untuk digunakan dalam proses mereduksi dimensi data. Reduksi dimensi bertujuan untuk menyederhanakan data yang kompleks tanpa mengurangi informasi yang signifikan, sehingga mempermudah proses analisis dan pemodelan lebih lanjut.. Rumus reduksi dimensi dapat dilihat pada persamaan 2.7 dibawah ini :

$$Transformed\ Data = Row\ Data \times Row\ Feature\ Vector \quad (2.7)$$

2.5 *Support Vector Machine* (SVM)

SVM adalah salah satu metode *machine learning* yang digunakan untuk klasifikasi dan regresi. SVM bekerja dengan mencari *hyperplane* yang optimal yang memisahkan data ke dalam dua atau lebih kelas. *Hyperplane* ini adalah batas yang memaksimalkan margin antara dua kelas, sehingga memberikan pembagian yang paling jelas antara kelompok data. SVM dikenal sangat efektif dalam menangani masalah klasifikasi dengan dataset berdimensi tinggi, dan juga dapat diaplikasikan pada dataset yang kompleks dan berukuran terbatas (Tommy Rustandi et al., 2023). *Hyperplane* didefinisikan sebagai sebuah fungsi linear yang dapat dituliskan dalam persamaan berikut:

$$f(x) = w^T \times x + b \quad (2.8)$$

Pada persamaan ini:

a) w adalah vektor bobot yang menentukan orientasi *hyperplane*,
b) x adalah vektor fitur data,
c) b adalah bias (*offset*) yang menentukan posisi *hyperplane* terhadap titik asal dalam ruang fitur.

Untuk menentukan *hyperplane* yang optimal, SVM menggunakan prinsip margin maksimum, yaitu jarak terdekat antara *hyperplane* dan titik data dari kedua kelas. Titik data yang berada pada batas margin disebut *support vectors*, yang memainkan peran kunci dalam menentukan posisi *hyperplane*. Persamaan margin maksimum dirumuskan sebagai berikut:

$$y_i(w^T \times x_i + b) \geq 1 \quad (2.9)$$

dengan:

a) y_i sebagai label kelas (+1 atau -1),
b) x_i sebagai titik data,
c) w dan b sebagai parameter *hyperplane* yang dioptimalkan.

Dalam kasus di mana data dapat dipisahkan secara linear, SVM memastikan bahwa semua data dari kelas berbeda berada di sisi *hyperplane* yang benar. Namun, untuk dataset yang tidak dapat dipisahkan secara linear, SVM menggunakan pendekatan kernel *trick*. Teknik ini memetakan data ke ruang dimensi yang lebih tinggi menggunakan fungsi kernel seperti RBF atau *polynomial* kernel. Dengan cara ini, data nonlinear di ruang asli dapat dipisahkan secara linear di ruang fitur yang baru (Tommy Rustandi et al., 2023).

2.6 Fungsi Kernel dan Kernel *Radial Basis Function* (RBF)

Pada sub-bab ini, akan dijelaskan dua konsep penting dalam SVM, yaitu fungsi kernel dan salah satu jenis kernel yang aktual, yaitu Kernel RBF. Penjelasan ini dimulai dengan pemahaman dasar mengenai fungsi kernel secara umum, yang berfungsi untuk mengubah data yang tidak terpisah secara linier menjadi data yang dapat dipisahkan dalam ruang dimensi yang lebih tinggi. Kernel RBF, sebagai salah satu jenis kernel yang paling populer, digunakan untuk menangani masalah klasifikasi dengan mempertimbangkan kedekatan antara titik data, sehingga memberikan fleksibilitas dalam membangun model yang lebih akurat.

2.6.1 Fungsi Kernel

Fungsi kernel adalah komponen utama dalam SVM yang digunakan untuk menangani data yang tidak dapat dipisahkan secara linear di ruang asli. Fungsi ini

memetakan data ke ruang dimensi lebih tinggi tanpa perlu menghitung representasi eksplisit data di ruang tersebut. Pendekatan ini dikenal dengan istilah kernel *trick*, yang memungkinkan SVM memisahkan data dengan pola kompleks secara efisien (Rabbani et al., 2023). Fungsi kernel menghitung kesamaan antara dua titik data x dan x' menggunakan persamaan:

$$K(x, x') = \emptyset(x) \times \emptyset(x') \quad (2.10)$$

Di mana:

a) $K(x, x')$ adalah hasil fungsi kernel,
b) $\emptyset(x)$ dan $\emptyset(x')$ adalah representasi data x dan x' di ruang dimensi lebih tinggi.

Dengan menggunakan kernel, SVM dapat mengatasi data yang tidak dapat dipisahkan secara linear di ruang asli, memungkinkan algoritma untuk membuat keputusan yang lebih akurat dalam tugas klasifikasi. Teknik kernel ini bekerja dengan memetakan data ke ruang dimensi yang lebih tinggi, di mana pemisahan antara kelas-kelas menjadi lebih mudah dilakukan. Hal ini meningkatkan kemampuan model untuk menangani masalah klasifikasi yang lebih kompleks dan non-linear.

2.6.2 Kernel RBF

Kernel RBF adalah salah satu fungsi kernel yang paling sering digunakan dalam SVM. Kernel ini menggunakan fungsi *Gaussian* untuk menghitung kesamaan antara dua titik data, yang sangat efektif untuk dataset dengan pola yang tidak linear. Dengan menggunakan kernel RBF, SVM dapat mengatasi tantangan dalam memisahkan data yang memiliki hubungan yang kompleks dan non-linear,

memberikan fleksibilitas yang lebih besar dalam pengklasifikasian (Rabbani et al., 2023). Persamaan kernel RBF adalah:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (2.11)$$

Di mana:

a) $\|x - x'\|^2$ adalah jarak *Euclidean* antara dua titik data x dan x'.
b) $\gamma > 0$ adalah parameter kernel yang menentukan pengaruh suatu titik data terhadap lingkungan sekitarnya.

Kernel RBF memiliki beberapa keunggulan yang membuatnya sangat efektif dalam berbagai aplikasi, termasuk dalam pengolahan data medis. Salah satu keunggulannya adalah kemampuan untuk menangani pola non-linear, yang menjadikannya sangat berguna untuk dataset yang tidak dapat dipisahkan secara linear. Selain itu, kernel RBF memiliki kemampuan generalisasi yang baik, terutama jika parameter γ dioptimalkan dengan tepat, yang memungkinkan model untuk menghindari masalah overfitting maupun underfitting. Keunggulan lainnya adalah kompatibilitasnya dengan data berdimensi tinggi, yang membuatnya bekerja dengan baik pada dataset yang memiliki banyak fitur, termasuk dalam konteks klasifikasi infeksi HIV, yang melibatkan data medis kompleks.

2.7 *Confusion Matrix*

Confusion Matrix adalah alat evaluasi yang digunakan dalam analisis kinerja model klasifikasi. Matriks ini menyediakan gambaran terperinci mengenai jumlah prediksi yang benar dan salah berdasarkan kelas actual dan kelas prediksi (Suryadewiansyah & Tju, 2022). Selanjutnya, untuk memperoleh pemahaman yang lebih mendalam mengenai kinerja model, penting untuk memeriksa komponen-komponen yang terdapat dalam *confusion matrix*.

A. Komponen *Confusion Matrix*:

Untuk memahami lebih dalam mengenai bagaimana Con*fusion Matrix* bekerja dalam mengevaluasi kinerja model klasifikasi, penting untuk mengetahui komponen-komponen yang membentuknya. Setiap komponen ini memberikan informasi penting tentang sejauh mana model mampu memprediksi data dengan benar atau salah. Berikut adalah penjelasan tentang masing-masing komponen dalam *Confusion Matrix* yang digunakan untuk mengukur akurasi dan kesalahan dalam prediksi.

1) ***True Positive* (TP):** Jumlah data actual positif yang diprediksi dengan benar sebagai positif.
2) ***True Negative* (TN):** Jumlah data actual negatif yang diprediksi dengan benar sebagai negatif.
3) ***False Positive* (FP):** Jumlah data actual negatif yang salah diprediksi sebagai positif (kesalahan tipe I).
4) ***False Negative* (FN):** Jumlah data actual positif yang salah diprediksi sebagai negatif (kesalahan tipe II).

B. Metrik Evaluasi Berdasarkan *Confusion Matrix*

Setelah memahami komponen *Confusion Matrix*, langkah selanjutnya adalah menghitung metrik evaluasi untuk menilai kinerja model klasifikasi. Metrik-metrik ini memberikan gambaran tentang seberapa efektif model dalam melakukan prediksi. Berdasarkan nilai dari *Confusion Matrix*, metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score* dapat

dihitung untuk mengevaluasi kinerja model. Berikut adalah penjelasan mengenai masing-masing metrik evaluasi yang umum digunakan.

1) ***Accuracy*:** Mengukur proporsi total prediksi yang benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.12)$$

2) ***Precision*:** Mengukur proporsi prediksi positif yang benar.

$$Precision = \frac{TP}{TP + FP} \quad (2.13)$$

3) ***Recall*:** Mengukur kemampuan model dalam mendeteksi semua data positif.

$$Recall = \frac{TP}{TP + FN} \quad (2.14)$$

4) **F1-*Score*:** Rata-rata harmonis antara *precision* dan *recall*.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.15)$$

Matrix yang digunakan meliputi akurasi, *precision*, *recall*, dan *F-Measure*. Akurasi mengukur sejauh mana prediksi yang tepat (TP dan TN) sesuai dengan hasil keseluruhan. *Precision* menunjukkan seberapa sering prediksi positif sesuai dengan *Alert* (TP) dibandingkan dengan semua prediksi positif. *Recall* mengukur seberapa sering prediksi positif cocok dengan *Alert* positif (TP) dibandingkan dengan semua *Alert* positif. *F-Measure*, yang juga dikenal sebagai *F-score* atau *F-*

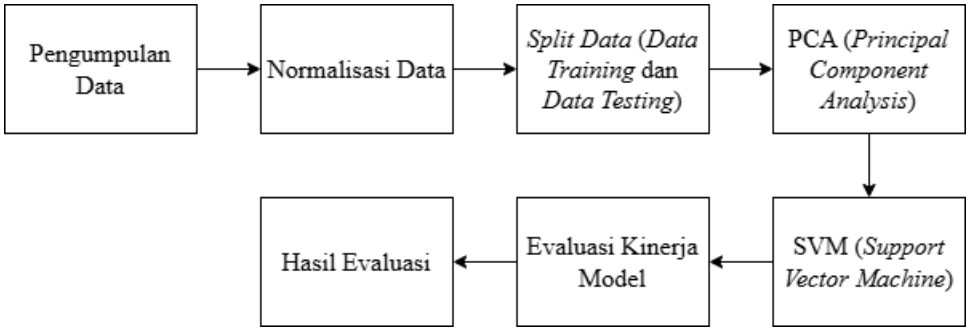
1 score, adalah rata-rata harmonis dari *precision* dan *recall*, dengan nilai terbaik 1 (artinya *precision* dan *recall* sempurna) dan nilai terburuk 0 (Suryadewiansyah & Tju, 2022).

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Desain Sistem

Gambar 3.1 menunjukkan desain sistem yang digunakan dalam penelitian ini. Penelitian ini mengintegrasikan metode PCA dan SVM untuk tujuan klasifikasi infeksi HIV. Dataset yang digunakan diperoleh dari *AIDS Clinical Trials Group Study 175*, yang mencakup data pasien dengan berbagai atribut relevan. Setelah data dikumpulkan, langkah pertama yang dilakukan adalah normalisasi data untuk menyamakan skala nilai pada setiap fitur dalam dataset. Proses normalisasi ini penting agar setiap fitur memiliki kontribusi yang seimbang selama analisis, sehingga model dapat memberikan hasil yang lebih akurat.



Gambar 3.1 Desain Sistem

Langkah berikutnya adalah pembagian dataset menjadi *data training* dan *data testing*. *Data training* digunakan untuk melatih model, sementara *data testing* digunakan untuk mengevaluasi performa model. Pembagian ini dilakukan sebelum penerapan PCA, untuk memastikan tidak ada kebocoran data antara *data training* dan *data testing*. Setelah pembagian data, proses PCA diterapkan untuk mereduksi

dimensi data dengan memilih komponen utama yang mencakup informasi penting. Setelah reduksi dimensi, data yang telah diproses digunakan untuk pelatihan dan pengujian model menggunakan algoritma SVM. Tahap akhir adalah evaluasi kinerja model menggunakan *Confusion Matrix* untuk mengukur akurasi, *precision*, *recall*, dan *F1-score* model.

3.2 Pengumpulan Data

Tabel 3.1 menyajikan daftar fitur dataset yang digunakan dalam penelitian ini beserta penjelasan untuk setiap atributnya. Setiap atribut memberikan informasi penting yang berkaitan dengan kondisi pasien, seperti waktu pengamatan, jenis pengobatan, riwayat kesehatan, dan hasil pemeriksaan medis yang dilakukan pada pasien. Fitur-fitur ini menjadi dasar dalam membangun model klasifikasi yang bertujuan untuk memprediksi status infeksi HIV pada pasien berdasarkan data yang tersedia.

Tabel 3.1 Fitur Dataset

No.	Atribut	Penjelasan
1.	*Time*	Waktu sampai kegagalan atau akhir pengamatan.
2.	*Trt*	Jenis pengobatan yang diterima (0 = ZDV saja; 1 = ZDV + ddI, 2 = ZDV + Zal, 3 = hanya ddI).
3.	*Age*	Usia (tahun) pada awal pengamatan.
4.	*Wtkg*	Berat badan (kg) pada awal pengamatan.
5.	*Hemo*	Apakah pasien memiliki *hemofilia* (0 = tidak, 1 = ya).
6.	*Homo*	Apakah pasien melakukan aktivitas homoseksual (0 = tidak, 1 = ya).
7.	*Drugs*	Riwayat penggunaan obat suntik (0 = tidak, 1 = ya).
8.	*Karnof*	Skor *Karnofsky* (skala 0-100 untuk menilai kondisi kesehatan umum pasien).
9.	*Oprior*	Apakah pasien telah menerima terapi *antiretroviral non-ZDV* sebelum waktu pengamatan (0 = tidak, 1 = ya).

No.	Atribut	Penjelasan
10.	*Z30*	Apakah pasien menerima ZDV dalam 30 hari sebelum waktu pengamatan (0 = tidak, 1 = ya).
11.	*Preanti*	Jumlah hari terapi *antiretroviral* sebelum waktu pengamatan.
12.	*Race*	Ras pasien (0 = kulit putih, 1 = non-kulit putih).
13.	*Gender*	Jenis kelamin pasien (0 = perempuan, 1 = laki-laki).
14.	*Str2*	Riwayat penggunaan *antiretroviral* (0 = belum pernah, 1 = sudah pernah).
15.	*Strat*	Kategori terapi *antiretroviral* (1 = belum pernah, 2 = lebih dari 1 tapi kurang dari 52 minggu, 3 = lebih dari 52 minggu).
16.	*Symptom*	Apakah pasien menunjukkan gejala (0 = tanpa gejala, 1 = dengan gejala).
17.	*Treat*	Jenis pengobatan yang diterima (0 = hanya ZDV, 1 = lainnya).
18.	*Offtrt*	Apakah pasien berhenti pengobatan sebelum 96±5 minggu (0 = tidak, 1 = ya).
19.	*Cd40*	Jumlah sel CD4 pada awal pengamatan.
20.	*Cd420*	Jumlah sel CD4 pada minggu ke-20±5.
21.	*Cd80*	Jumlah sel CD8 pada awal pengamatan.
22.	*Cd820*	Jumlah sel CD8 pada minggu ke-20±5.
23.	*Infected*	Status infeksi AIDS (0 = tidak terinfeksi, 1 = terinfeksi).

Setelah memahami fitur-fitur yang tercantum dalam tabel, langkah berikutnya adalah memproses dan menganalisis data untuk melatih model klasifikasi. Proses ini melibatkan penggunaan berbagai teknik seperti normalisasi data, pemilihan atribut yang relevan, serta penerapan metode analisis yang tepat untuk mendapatkan hasil yang akurat. Dengan mengandalkan fitur-fitur yang telah dijelaskan, model akan dilatih untuk mengenali pola dalam data dan memberikan prediksi yang bermanfaat bagi diagnosis dan pengobatan pasien.

3.3 Normalisasi Data

Normalisasi diperlukan untuk memastikan bahwa setiap fitur dalam dataset memiliki skala yang sama. Tanpa normalisasi, fitur dengan rentang nilai yang lebih

besar dapat mendominasi analisis. Normalisasi yang digunakan dalam penelitian ini adalah *Standard Scaler*, yang mengubah data sehingga memiliki rata-rata nol dan varians satu. Rumus untuk normalisasi menggunakan *Standard Scaler* adalah sebagai berikut:

$$z = \frac{x - \mu}{\sigma} \tag{3.1}$$

Keterangan:

a) x: nilai asli fitur,
b) μ: rata-rata dari fitur,
c) σ: standar deviasi dari fitur.

Dengan metode *Standard Scaler*, semua fitur memiliki rata-rata nol dan standar deviasi satu. Hal ini memastikan bahwa analisis tidak dipengaruhi oleh perbedaan skala antar fitur. Selain itu, teknik ini membantu model untuk belajar lebih efisien, karena algoritma tidak akan lebih fokus pada fitur dengan skala yang lebih besar.

3.4 *Split Data*

Tabel 3.2 menunjukkan pembagian dataset menjadi dua bagian utama, yaitu *data training* dan *data testing*. Proses ini bertujuan untuk memisahkan data yang digunakan untuk melatih model dan data yang digunakan untuk mengevaluasi performa model. Tiga rasio pembagian data yang digunakan adalah 90:10, 80:20, dan 70:30. Pemilihan rasio ini bertujuan untuk menguji kinerja model pada berbagai skenario jumlah data training, sehingga dapat dianalisis efeknya terhadap kemampuan generalisasi model.

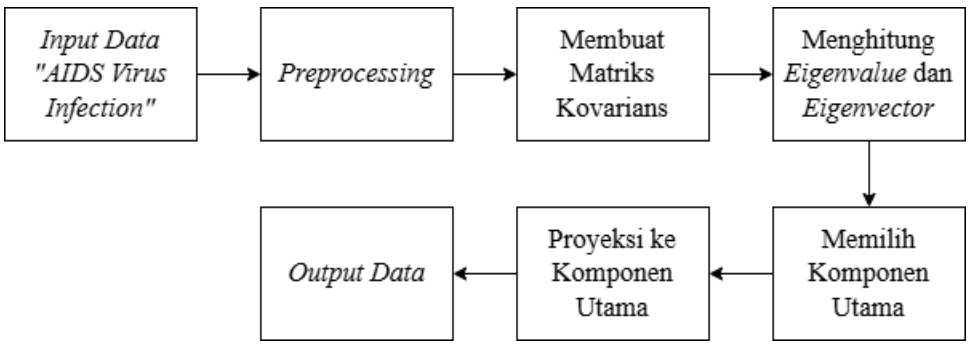
Tabel 3.2 Contoh *Split Data*

No.	Rasio	*Data Training*	*Data Testing*
1.	90:10	937	105
2.	80:20	833	209
3.	70:30	729	313

Dari Tabel tersebut, terlihat bahwa rasio 90:10, model memiliki lebih banyak data untuk pelatihan, yang dapat meningkatkan akurasi pada *data training*. Namun, rasio ini mungkin kurang representatif untuk menguji data baru karena proporsi *data testing* yang kecil. Sebaliknya, rasio 70:30 memberikan lebih banyak data untuk evaluasi, sehingga dapat menunjukkan kemampuan model dalam menangani data yang tidak pernah dilihat sebelumnya, meskipun potensi *overfitting* pada *data training* mungkin meningkat.

3.5 Implementasi Metode *Principal Component Analysis* (PCA)

Gambar 3.2 menggambarkan proses analisis menggunakan PCA pada dataset "*AIDS Virus Infection*". Tujuan utama dari penerapan PCA ini adalah untuk mereduksi dimensi data, menyederhanakan jumlah variabel yang dianalisis, namun tetap mempertahankan informasi penting yang ada dalam data asli. Dengan demikian, klasifikasi infeksi HIV menggunakan SVM dapat dilakukan dengan lebih efisien dan akurat.



Gambar 3.2 Proses Analisis Menggunakan PCA

Implementasi PCA dalam penelitian ini dilakukan melalui serangkaian tahapan yang terstruktur, dimulai dari persiapan data hingga menghasilkan dataset dengan dimensi yang lebih rendah. Setiap tahapan dirancang untuk memastikan proses analisis berjalan secara sistematis dan mendukung tujuan reduksi dimensi. Tahapan-tahapan tersebut mencakup *preprocessing data*, pembuatan matriks kovarians, perhitungan *eigenvalue* dan *eigenvector*, serta proyeksi data ke komponen utama yang dipilih berdasarkan kontribusinya terhadap varians data.

3.5.1 *Input Data*

Dataset yang digunakan dalam penelitian ini adalah dataset "*AIDS Virus Infection*" yang diperoleh dari *AIDS Clinical Trials Group Study 175*. Dataset ini terdiri dari 23 fitur dan 2.139 baris data yang mencakup informasi demografis pasien, riwayat penggunaan obat, serta hasil tes klinis yang relevan untuk mendeteksi infeksi HIV. Dataset ini dipilih karena menyediakan data yang kaya dan mendalam, sehingga mendukung analisis PCA untuk mereduksi dimensi sekaligus membantu proses klasifikasi menggunakan SVM secara lebih efisien. Selain itu, dataset ini merepresentasikan tantangan dalam pengolahan data dengan dimensi yang tinggi. Sebelum dilakukan analisis menggunakan PCA, dataset ini telah melewati tahapan *preprocessing* seperti yang dijelaskan pada sub bab 3.3, guna memastikan data siap untuk direduksi dimensinya tanpa kehilangan informasi penting.

3.5.2 *Preprocessing*

Pada tahap *preprocessing*, langkah pertama yang dilakukan adalah normalisasi data. Proses normalisasi ini bertujuan untuk memastikan bahwa semua fitur memiliki skala yang setara, sehingga tidak ada fitur yang mendominasi analisis hanya karena memiliki nilai yang lebih besar atau lebih kecil dibandingkan fitur lainnya. Seperti yang telah dijelaskan pada sub bab 3.3, metode *Z-Score Normalization* digunakan untuk melakukan normalisasi ini. Normalisasi memastikan bahwa setiap fitur memiliki rata-rata nol ($\mu = 0$) dan standar deviasi satu ($\sigma = 1$), sehingga PCA dapat bekerja secara optimal.

Pada implementasi PCA, normalisasi data sangat penting karena PCA sensitif terhadap skala fitur. Jika data tidak dinormalisasi, hasil analisis PCA dapat terdistorsi, dan komponen utama yang dihasilkan mungkin tidak merepresentasikan pola dalam data dengan benar. Oleh karena itu, data yang digunakan dalam tahap ini telah melalui proses normalisasi seperti yang dijelaskan sebelumnya, untuk memastikan setiap fitur memiliki pengaruh yang sama dalam analisis.

3.5.3 Membuat Matriks Kovarians

Setelah data dinormalisasi, langkah berikutnya dalam proses PCA adalah membuat matriks kovarians. Matriks kovarians digunakan untuk memahami hubungan linier antara variabel-variabel dalam dataset. Hubungan ini penting untuk mengidentifikasi arah utama variansi data PC, yang nantinya akan digunakan untuk mereduksi dimensi dataset. Matriks kovarians dihitung menggunakan formula berikut:

$$Cov(X,Y) = \frac{\sum(X-\mu x)(Y-\mu y)}{n-1} \quad (3.2)$$

Di mana:

a) X dan Y : Dua fitur dalam dataset.
b) μx dan μy : Nilai rata-rata fitur X dan Y.
c) n : Jumlah sampel dalam dataset.

Proses ini menghasilkan sebuah matriks simetris yang mencerminkan hubungan antar fitur dalam dataset. Matriks ini menjadi dasar untuk menentukan *eigenvalue* dan *eigenvector* pada langkah berikutnya. *Eigenvalue* dan *eigenvector* akan digunakan untuk menentukan komponen utama PC dalam data.

3.5.4 Menghitung *EigenValue* dan *EigenVector*

Setelah matriks kovarians dihitung, langkah berikutnya dalam proses PCA adalah menghitung nilai *eigen* (*eigenvalue*) dan vektor *eigen* (*eigenvector*). Nilai *eigen* dan vektor *eigen* digunakan untuk menentukan komponen utama (PC) yang mewakili variansi terbesar dalam data. *Eigenvalue* menunjukkan jumlah variansi data yang dapat dijelaskan oleh setiap komponen utama. Semakin besar nilai *eigen*, semakin besar informasi (variansi) yang dijelaskan oleh komponen utama tersebut. Sementara itu *eigenvector* menunjukkan arah dari setiap komponen utama di dalam ruang fitur asli. Vektor ini digunakan untuk memproyeksikan data ke ruang komponen utama. Proses perhitungan *eigenvalue* dan *eigenvector* dilakukan dengan mendekompisi matriks kovarians. *Eigenvalue* dan *eigenvector* diperoleh dari penyelesaian persamaan karakteristik berikut:

$$|C - \lambda I| = 0 \quad (3.3)$$

Di mana:

a) C : Matriks kovarians
b) λ : *Eigenvalue*
c) I : Matriks identitas

Hasil dari perhitungan ini adalah pasangan nilai *eigen* dan vektor *eigen*, yang diurutkan dari nilai *eigen* terbesar hingga terkecil. Urutan ini memungkinkan identifikasi komponen utama yang paling signifikan berdasarkan kontribusi variansnya. Komponen utama dengan nilai *eigen* terbesar akan dipilih untuk analisis lebih lanjut.

3.5.5 Memilih Komponen Utama

Setelah *eigenvalue* dan *eigenvector* dihitung, langkah selanjutnya dalam proses PCA adalah memilih komponen utama PC yang akan digunakan untuk analisis. Pemilihan ini dilakukan berdasarkan nilai *eigen* yang telah diurutkan dari yang terbesar hingga terkecil. Komponen utama dengan nilai *eigen* terbesar dianggap memiliki kontribusi paling signifikan terhadap variansi data. Oleh karena itu, hanya sejumlah komponen utama dengan nilai *eigen* terbesar yang dipilih untuk melanjutkan analisis, sementara komponen lainnya diabaikan. Pemilihan ini bertujuan untuk mereduksi dimensi dataset tanpa kehilangan informasi yang signifikan.

Kriteria pemilihan komponen utama dalam penelitian ini didasarkan pada kontribusi varians kumulatif. Varians kumulatif dihitung sebagai persentase dari total varians yang dapat dijelaskan oleh komponen utama. Hanya komponen utama yang mampu menjelaskan hingga 95% dari total varians kumulatif yang dipilih

untuk analisis lebih lanjut. Langkah ini memastikan bahwa dataset hasil reduksi dimensi tetap mempertahankan informasi yang esensial dari data asli.

3.5.6 Proyeksi ke Komponen Utama

Setelah komponen utama dipilih berdasarkan *eigenvalue* terbesar, langkah berikutnya dalam proses PCA adalah memproyeksikan data asli ke ruang komponen utama tersebut. Proyeksi ini bertujuan untuk merepresentasikan data dalam dimensi yang lebih rendah, tetapi tetap mempertahankan informasi yang signifikan dari dataset asli. Proses proyeksi dilakukan dengan menggunakan matriks *eigenvector* yang telah dipilih. Formula proyeksi adalah sebagai berikut:

$$Y = X \cdot W \tag{3.4}$$

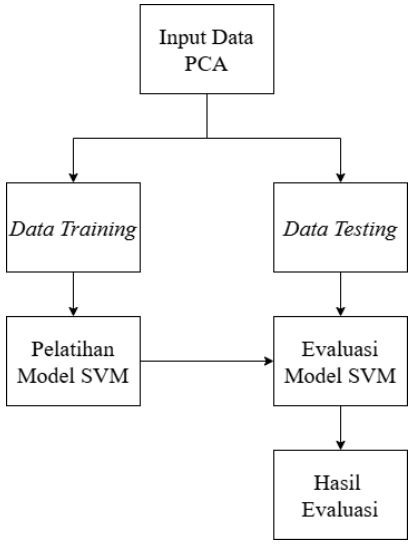
Di mana:

a) Y : Dataset hasil proyeksi dengan dimensi yang lebih rendah.
b) X : Dataset asli yang telah dinormalisasi.
c) W : Matriks *eigenvector* yang sesuai dengan komponen utama yang dipilih.

Dataset asli (X) diproyeksikan ke ruang baru (Y) yang terbentuk oleh komponen utama (W). Hasil dari proyeksi ini adalah dataset dengan dimensi yang lebih kecil, yang mengandung informasi paling penting dari data asli. Dengan menggunakan dataset yang lebih ringkas ini, proses analisis selanjutnya, seperti klasifikasi menggunakan SVM, dapat dilakukan dengan lebih efisien dan akurat. Proyeksi ke ruang komponen utama juga memungkinkan visualisasi data menjadi lebih sederhana, terutama jika dataset direduksi menjadi dua atau tiga dimensi. Namun, dalam penelitian ini, hasil proyeksi data digunakan sebagai input untuk klasifikasi.

3.6 Implementasi Metode *Support Vector Machine* (SVM)

Gambar 3.3 menunjukkan proses implementasi model SVM dalam klasifikasi infeksi HIV. Pada tahap ini, SVM diimplementasikan untuk memprediksi infeksi HIV berdasarkan dataset yang telah direduksi dimensinya menggunakan PCA. SVM dipilih karena efektif dalam menangani dataset kompleks dengan banyak fitur, yang berpotensi mengganggu kinerja model jika tidak dikelola dengan baik. PCA digunakan untuk mengurangi dimensi dataset, memperkecil risiko *overfitting*, dan meningkatkan akurasi model.



Gambar 3.3 Proses Implementasi SVM

Proses selanjutnya dimulai dengan pembagian data menjadi *data training dan testing*. *Data training* digunakan untuk melatih model, sedangkan *data testing* digunakan untuk mengevaluasi kinerja model. Setelah itu, model SVM dilatih menggunakan *data training* dan dievaluasi dengan *data testing* untuk memastikan akurasi dan efektivitas model dalam mengklasifikasikan infeksi HIV. Proses evaluasi ini sangat penting untuk memastikan model yang dibangun dapat

memberikan hasil yang akurat dan efisien pada data yang belum pernah dilihat sebelumnya.

3.6.1 Input Data PCA

Reduksi dimensi dilakukan menggunakan PCA, yang berfungsi untuk mengurangi jumlah fitur dataset dari 22 kolom menjadi beberapa komponen utama yang menyimpan sebagian besar informasi variansi data. Proses ini sangat penting mengingat dataset yang digunakan dalam penelitian ini sangat besar dan terdiri dari banyak fitur yang memiliki hubungan korelasi yang tinggi. Dengan mereduksi dimensi data, PCA membantu mengidentifikasi komponen utama yang relevan dan mengurangi kompleksitas analisis. Reduksi dimensi juga membantu menghilangkan *noise* dan informasi yang kurang relevan, sehingga proses pelatihan model dapat dilakukan dengan lebih efisien. Dengan data yang sudah direduksi dimensinya, model SVM dapat bekerja lebih optimal, tanpa mengorbankan kualitas prediksi.

3.6.2 *Data Training* dan *Data Testing* PCA

Setelah data melalui proses PCA, dataset dibagi menjadi dua bagian utama: *data training* dan *data testing*. *Data training* PCA digunakan untuk melatih model klasifikasi SVM, di mana model akan belajar mengenali pola yang ada dalam data dan mengklasifikasikan pasien yang terinfeksi HIV dan yang tidak. *Data testing* PCA kemudian digunakan untuk menguji seberapa baik model yang telah dilatih dalam mengklasifikasikan data baru yang belum pernah dilihat sebelumnya. Pembagian ini bertujuan untuk memastikan bahwa model tidak hanya menghafal

data yang ada, tetapi dapat menggeneralisasi dengan baik ke data yang tidak dikenal. Rasio pembagian data yang digunakan dalam eksperimen ini adalah 90:10, 80:20 dan 70:30.

3.6.3 Pelatihan Model SVM

Pada tahap ini, model SVM dilatih menggunakan *data training* yang telah dinormalisasi dan direduksi dimensinya menggunakan PCA. SVM adalah metode klasifikasi yang bekerja dengan mencari *hyperplane* yang optimal untuk memisahkan data ke dalam dua kelas, yaitu kelas "*infected*" dan "*non-infected*". Dalam penelitian ini, untuk mengatasi data yang tidak dapat dipisahkan secara linier di ruang asli, digunakan kernel RBF. Fungsi kernel RBF digunakan untuk menghitung kesamaan antara dua titik data berdasarkan jarak *Euclidean* dalam ruang fitur yang lebih tinggi, yang memungkinkan pemisahan data secara non-linear. Rumus kernel RBF adalah sebagai berikut:

$$K(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right) \tag{3.5}$$

Di mana:

a) $x_i \; dan \; x_j$ adalah vektor fitur data,
b) $\left\| x_i - x_j \right\|^2$ adalah jarak *Euclidian* antara dua vektor $x_i \; dan \; x_j$,
c) γ adalah parameter yang mengontrol bentuk fungsi kernel dan mempengaruhi seberapa jauh pengaruh titik data terhadap keputusan klasifikasi .

Dalam pelatihan model SVM, terdapat dua *hyperparameter* utama yang harus diatur: C dan :. Kedua parameter ini sangat mempengaruhi kinerja model, di mana C mengontrol *trade-off* antara margin maksimum dan kesalahan klasifikasi, sedangkan : mengatur seberapa besar pengaruh titik data terhadap keputusan

klasifikasi. Pengaturan yang tepat dari keduanya penting untuk mencapai kinerja yang optimal.

1) C: Mengontrol *trade-off* antara margin maksimum dan kesalahan klasifikasi. Nilai C yang tinggi akan memaksa model untuk mengurangi kesalahan klasifikasi dengan meningkatkan margin, sedangkan nilai C yang lebih kecil akan memberikan margin yang lebih lebar dengan toleransi lebih besar terhadap kesalahan.
2) γ: Mengontrol seberapa besar pengaruh titik data individu terhadap keputusan klasifikasi. Nilai γ yang kecil membuat pengaruh lebih luas, sementara nilai γ yang besar membuat pengaruh lebih lokal.

3.7 Evaluasi Kinerja Model

Setelah model SVM dilatih, tahap selanjutnya adalah melakukan evaluasi untuk mengukur seberapa baik kinerja model menggunakan *data testing* yang terpisah. Evaluasi ini melibatkan beberapa langkah, dimulai dari prediksi menggunakan model yang telah dilatih, perhitungan berbagai metrik evaluasi, dan analisis mendalam terhadap hasil yang diperoleh. Metrik yang digunakan termasuk akurasi, *precision*, *recall*, dan *F1-Score*, yang memberikan gambaran tentang kemampuan model dalam mengklasifikasikan data dengan benar.

3.7.1 Prediksi *Data Testing*

Setelah *data testing* diproses melalui PCA, data tersebut dimasukkan ke dalam model SVM yang telah dilatih untuk menghasilkan prediksi label. Model ini memberikan label 1 untuk "*infected*" dan label 0 untuk "*non-infected*". Proses

prediksi dilakukan dengan membandingkan hasil prediksi model terhadap label aktual yang ada pada data testing. Perbandingan ini memungkinkan evaluasi terhadap kinerja model dalam mengklasifikasikan data yang belum pernah dilihat sebelumnya selama pelatihan. Meskipun model SVM umumnya memberikan hasil yang akurat, terkadang terdapat kesalahan klasifikasi, di mana model salah mengklasifikasikan data yang sebenarnya tidak terinfeksi sebagai terinfeksi, atau sebaliknya. Kesalahan seperti ini dapat mempengaruhi metrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1-Score,* yang digunakan untuk menilai performa model.

3.7.2 Perhitungan Matriks Evaluasi

Dalam penelitian ini, kinerja model dievaluasi menggunakan *Confusion Matrix* yang memberikan pemahaman mendalam tentang hasil klasifikasi model. Matriks ini menunjukkan bagaimana prediksi model berbanding dengan label sebenarnya. *Confusion Matrix* terdiri dari empat komponen utama:

a) TP: Jumlah data yang terinfeksi dan benar diprediksi sebagai terinfeksi.
b) TN: Jumlah data yang tidak terinfeksi dan benar diprediksi sebagai tidak terinfeksi.
c) FP: Jumlah data yang tidak terinfeksi namun diprediksi sebagai terinfeksi.
d) FN: Jumlah data yang terinfeksi namun diprediksi sebagai tidak terinfeksi.

Pada evaluasi model prediksi, beberapa metrik digunakan untuk mengukur kinerja model berdasarkan hasil prediksi yang diberikan, baik yang benar maupun yang salah. Metrik-metrik ini, yang dihitung dari nilai TP , TN , FP , dan FN,

memberikan gambaran tentang sejauh mana model dapat memprediksi dengan akurat, serta seberapa baik model menangani ketidakseimbangan data. Berikut adalah beberapa metrik evaluasi yang digunakan dalam analisis ini:

a) Akurasi mengukur proporsi total prediksi yang benar (baik TP maupun TN) terhadap semua prediksi yang dilakukan. Formula untuk menghitung akurasi adalah sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.6}$$

b) *Precision* mengukur proporsi prediksi positif yang benar-benar terinfeksi dari semua yang diprediksi sebagai positif. Formula *precision* adalah:

$$Presisi = \frac{TP}{TP + FP} \tag{3.7}$$

c) *Recall* mengukur seberapa banyak kasus terinfeksi yang berhasil diprediksi dengan benar. Formula recall adalah:

$$Recall = \frac{TP}{TP + FN} \tag{3.8}$$

d) *F1-Score* adalah rata-rata harmonis dari *precision* dan *recall*, yang memberikan gambaran seimbang tentang kinerja model, terutama ketika terdapat ketidakseimbangan antara jumlah kasus positif dan negatif. Formula *F1-Score* adalah:

$$F1 = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (3.9)$$

3.7.3 Interpretasi Hasil

Setelah melakukan perhitungan metrik evaluasi, hasil yang diperoleh memberikan gambaran yang jelas mengenai kinerja model dalam memprediksi data. Metrik-metrik ini membantu untuk mengevaluasi seberapa baik model dalam mengidentifikasi hasil yang benar dan seberapa efektif model dalam menangani berbagai jenis kesalahan. Berikut adalah interpretasi dari hasil perhitungan yang telah dilakukan:

a) Akurasi menunjukkan model dapat memprediksi dengan benar sebesar 66.6%.
b) *precision* model sangat tinggi (100%), berarti setiap prediksi positif oleh model selalu benar.
c) *Recall* model rendah (50%), berarti model masih melewatkan separuh dari data positif aktual.
d) *F1-Score* sebesar 66.6%, yang mengindikasikan keseimbangan antara *precision* dan *recall*.

3.8 Hasil Evaluasi

Setelah menghitung metrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1-Score*, hasil evaluasi menunjukkan bahwa meskipun model memiliki *precision* yang tinggi, ada beberapa area yang perlu diperbaiki, terutama dalam hal *recall*. Model SVM mampu memberikan hasil yang sangat akurat dalam memprediksi data

negatif (*non-infected*), namun kesulitan dalam mendeteksi semua kasus positif (*infected*). Oleh karena itu, meskipun *precision* model sangat baik, ada potensi untuk meningkatkan *recall*, yang akan mengurangi jumlah *false negatives* dan memberikan model yang lebih baik dalam mendeteksi infeksi HIV pada *data testing*.

Hasil ini memberikan wawasan penting bagi perbaikan lebih lanjut model SVM, dengan kemungkinan untuk mengoptimalkan parameter C dan γ dalam kernel RBF, atau mencoba pendekatan *undersampling* untuk menangani ketidakseimbangan kelas dalam data. Dalam hal ini, *undersampling* dapat digunakan untuk mengurangi jumlah data yang termasuk dalam kelas mayoritas (*non-infected*) untuk seimbang dengan jumlah data pada kelas minoritas (*infected*). Dengan mengurangi jumlah data *non-infected*, model akan fokus untuk mendeteksi infeksi dengan lebih baik. Dengan evaluasi yang lebih mendalam, model SVM dapat ditingkatkan untuk memberikan prediksi yang lebih akurat dalam konteks klasifikasi infeksi HIV.

3.9 Eksperimen Skenario

Eksperimen skenario dilakukan untuk mengevaluasi kinerja model klasifikasi infeksi HIV dengan pendekatan kombinasi PCA dan SVM. Tiga skenario utama digunakan dalam pembagian *data training* dan *testing*, yaitu 90:10, 80:20, dan 70:30, untuk mengamati bagaimana variasi pembagian data memengaruhi kinerja model. Eksperimen ini juga mencakup optimasi *hyperparameter* model menggunakan G*ridSearchCV*.

3.9.1 Langkah-Langkah Eksperimen

Eksperimen ini dilakukan untuk menguji kinerja model dalam berbagai skenario pembagian data dan teknik pra-pemrosesan. Tujuan dari eksperimen ini adalah untuk mengevaluasi bagaimana perubahan dalam pembagian data, reduksi dimensi, dan penanganan ketidakseimbangan kelas memengaruhi efektivitas model dalam melakukan prediksi. Berikut adalah langkah-langkah yang diambil dalam eksperimen ini:

A. *Split Data*

Pembagian data merupakan langkah awal yang penting dalam eksperimen ini, di mana data dibagi menjadi dua bagian, yaitu data untuk pelatihan (training) dan data untuk pengujian (testing). Pembagian ini dilakukan berdasarkan tiga skenario yang berbeda untuk menilai pengaruh proporsi data pelatihan dan pengujian terhadap kinerja model. Hal ini memungkinkan kita untuk mengevaluasi bagaimana variasi dalam jumlah data yang digunakan untuk pelatihan dapat mempengaruhi hasil prediksi yang dihasilkan oleh model.

B. Reduksi Dimensi dengan PCA

PCA diterapkan untuk mereduksi jumlah fitur dalam dataset, dengan mempertahankan 95% variansi data, sehingga hanya fitur-fitur yang paling signifikan yang dipertahankan. Langkah ini bertujuan untuk menyederhanakan data yang digunakan oleh model, meningkatkan efisiensi pemrosesan, mempercepat waktu pelatihan, serta mengurangi risiko

overfitting yang dapat terjadi akibat adanya fitur yang tidak relevan atau terlalu banyak informasi yang tidak penting.

C. *Undersampling* untuk Keseimbangan Kelas

Untuk menangani ketidakseimbangan data yang dapat menyebabkan model lebih memprioritaskan kelas mayoritas, dilakukan teknik *Random Undersampling*, di mana jumlah data pada kelas mayoritas (*non-infected*) dikurangi hingga sejajar dengan jumlah data pada kelas minoritas (*infected*). Dengan mengurangi jumlah data pada kelas mayoritas, langkah ini bertujuan untuk menciptakan keseimbangan yang lebih baik antara kedua kelas, sehingga model dapat belajar untuk memberikan perhatian yang lebih seimbang pada kedua kelas tanpa terdistorsi oleh dominasi kelas mayoritas, yang sering kali berisiko menyebabkan bias dalam hasil prediksi.

D. Pelatihan Model dengan SVM

Model SVM dilatih menggunakan kernel RBF, yang dikenal efektif dalam menangani data non-linear dengan kompleksitas tinggi. Model ini dilatih menggunakan data pelatihan yang telah melalui tahap reduksi dimensi menggunakan PCA dan penanganan ketidakseimbangan kelas dengan *undersampling*. Dengan pendekatan ini, SVM diharapkan mampu membangun *hyperplane* yang lebih akurat dalam memisahkan kelas positif dan negatif, bahkan pada data yang lebih rumit atau tidak terpisahkan secara linier.

E. Optimasi *Hyperparameter* dengan *GridSearchCV*

Dalam rangka mengoptimalkan kinerja model, dilakukan pencarian *hyperparameter* menggunakan teknik *GridSearchCV*. Proses ini bertujuan untuk menemukan kombinasi nilai terbaik dari parameter C dan γ pada model SVM, yang sangat memengaruhi hasil klasifikasi. Dengan menguji berbagai kombinasi nilai parameter melalui *cross-validation*, diharapkan dapat diperoleh konfigurasi model yang memberikan hasil terbaik pada data yang tersedia.

F. Evaluasi Model

Setelah pelatihan dan optimasi selesai, model dievaluasi untuk mengukur sejauh mana kemampuannya dalam memprediksi data yang belum pernah dilihat sebelumnya. Evaluasi dilakukan dengan menggunakan *data testing*, dan metrik yang digunakan mencakup akurasi, *precision*, *recall*, dan *F1-score*. Metrik-metrik ini memberikan gambaran komprehensif tentang seberapa baik model bekerja dalam mendeteksi kelas positif, serta keseimbangan antara keandalan prediksi dan kemampuan deteksi.

3.9.2 Hasil Eksperimen

Tabel 3.6 menyajikan hasil eksperimen yang mengevaluasi kinerja model berdasarkan tiga skenario pembagian *data training* dan *testing* yang berbeda. Dalam tabel ini, metrik evaluasi yang digunakan mencakup akurasi, *precision*, *recall*, dan *F1-Score*, yang menggambarkan seberapa baik model dapat

memprediksi data berdasarkan berbagai proporsi data yang digunakan untuk pelatihan dan pengujian.

Tabel 3.3 Simulasi Hasil Eksperimen

No.	Skenario	*Data Training*	*Data Testing*	Akurasi	*Precision*	*Recall*	*F1-Score*
1.	1	90%	10%	89.56%	88.34%	87.89%	88.12%
2.	2	80%	20%	88.34%	87.12%	86.78%	86.95%
3.	3	70%	30%	87.23%	85.78%	84.45%	85.10%

Berdasarkan Tabel tersebut, dapat dilihat bahwa masing-masing skenario mempengaruhi kinerja model. Meskipun ada penurunan dalam nilai akurasi, *precision*, *recall*, dan *F1-Score* ketika proporsi *data testing* meningkat, model tetap menunjukkan performa yang cukup baik. Hasil ini memberikan wawasan lebih lanjut tentang pengaruh pembagian data terhadap efisiensi model dalam memprediksi kasus positif dan negatif, serta potensi untuk meningkatkan kinerja dengan mengoptimalkan parameter model atau teknik pra-pemrosesan data.

3.9.3 Analisis Hasil

Pada bagian ini, hasil eksperimen yang telah dilakukan akan dianalisis untuk melihat pengaruh pembagian data terhadap kinerja model. Setiap skenario memiliki kelebihan dan kekurangan terkait dengan jumlah *data training* dan *testing* yang digunakan, yang dapat mempengaruhi hasil evaluasi model secara keseluruhan. Berikut adalah analisis mendalam dari masing-masing skenario:

a) Skenario 1 (90:10) memberikan akurasi tertinggi (89.56%) karena jumlah *data training* yang besar. Namun, *data testing* yang kecil membuat evaluasi generalisasi model kurang komprehensif.

b) Skenario 2 (80:20) memberikan keseimbangan terbaik antara jumlah *data training* dan *testing*, dengan akurasi yang masih tinggi (88.34%).
c) Skenario 3 (70:30) menunjukkan penurunan akurasi (87.23%) karena jumlah *data training* yang lebih kecil. Namun, *data testing* yang lebih banyak memberikan evaluasi generalisasi yang lebih kuat.

3.9.4 Kesimpulan

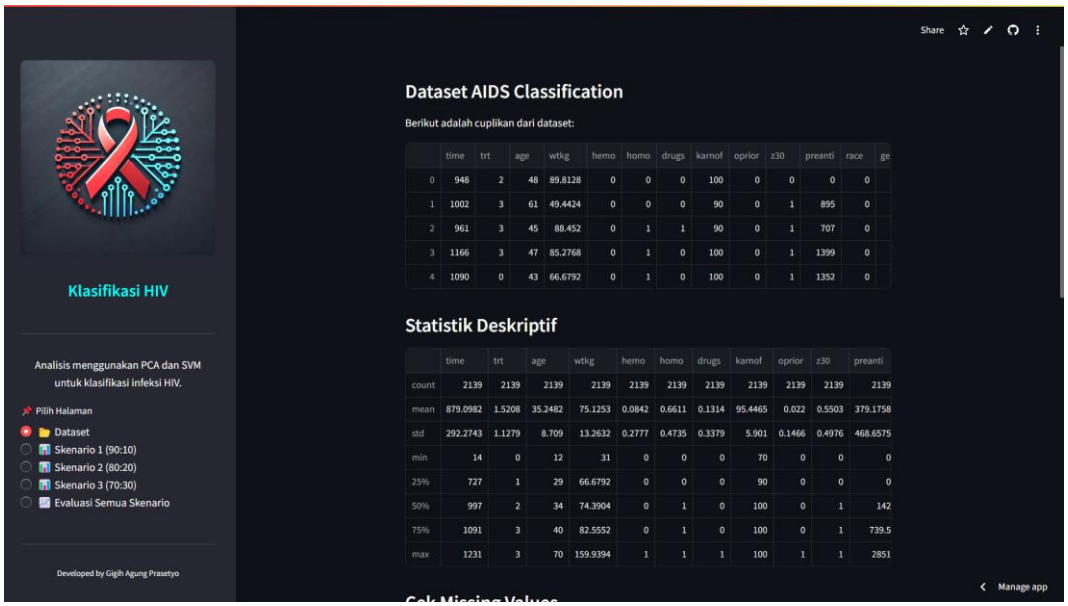
Eksperimen skenario menunjukkan bahwa rasio pembagian data memengaruhi kinerja model secara signifikan, dengan variasi dalam jumlah *data training* dan *testing* memberikan dampak langsung terhadap hasil evaluasi. Skenario 2 (80:20) memberikan hasil yang paling seimbang, dengan akurasi yang tinggi dan evaluasi generalisasi yang cukup kuat, sehingga skenario ini dianggap sebagai kombinasi optimal antara jumlah data yang digunakan untuk pelatihan dan pengujian model.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pengumpulan Data

Pada Gambar 4.1 menunjukkan tampilan antarmuka aplikasi yang digunakan untuk menampilkan dataset *AIDS Classification*. Aplikasi ini menyajikan informasi mengenai data klinis pasien yang meliputi berbagai fitur seperti waktu, usia, berat badan, kadar *hemoglobin*, dan beberapa variabel medis lainnya. Dalam gambar ini, selain dataset mentah, juga ditampilkan statistik deskriptif yang memberikan gambaran umum tentang distribusi data, seperti nilai rata-rata, standar deviasi, dan nilai minimum serta maksimum.



Gambar 4.1 Tampilan Antarmuka Aplikasi: Dataset *AIDS Classification*

Gambar ini memberikan tampilan yang jelas mengenai struktur dataset yang digunakan dalam penelitian, di mana setiap fitur diorganisasikan dalam tabel yang

rapi. Informasi statistik deskriptif yang disajikan juga memungkinkan peneliti untuk memperoleh pemahaman awal tentang data, sehingga memudahkan dalam tahap-tahap *preprocessing* selanjutnya, seperti pembersihan data, normalisasi, dan reduksi dimensi. Antarmuka aplikasi ini memfasilitasi eksplorasi data secara interaktif, yang penting dalam memastikan kualitas data sebelum masuk ke tahap pelatihan model klasifikasi.

4.2 Hasil Normalisasi

Tabel 4.1 menyajikan data sebelum dilakukan normalisasi pada beberapa fitur yang digunakan dalam penelitian ini, seperti waktu (*time*), tipe pengobatan (*txt*), usia (*age*), berat badan (*wtkg*), kadar *hemoglobin* (*hemo*), dan status *homo* (*homo*). Setiap kolom memiliki rentang nilai yang berbeda, yang bisa memengaruhi model jika tidak dilakukan penyesuaian. Variabel berat badan (*wtkg*) memiliki nilai yang lebih besar dibandingkan dengan variabel lainnya, yang dapat menyebabkan ketidakseimbangan pengaruh antar fitur dalam proses pelatihan model. Untuk mengatasi hal ini, dilakukan normalisasi menggunakan *StandardScaler*, yang menghitung *Z-Score* untuk setiap fitur, sehingga nilai-nilai fitur disesuaikan dengan rata-rata nol dan deviasi standar satu, menghasilkan skala yang seragam di seluruh variabel.

Tabel 4.1 Data Sebelum Normalisasi

No.	***time***	***trt***	***age***	***wtkg***	***hemo***	***homo***	**…**
1.	948	2	48	89.8128	0	0	
2.	1002	3	61	49.4424	0	0	
3.	961	3	45	88.4520	0	1	
4.	1166	3	47	85.2768	0	1	
5.	1090	0	43	66.6792	0	1	

Pada Tabel 4.2 menunjukkan data setelah dilakukan normalisasi menggunakan metode *Z-Score* dengan *StandardScaler*. Dalam tabel ini, setiap fitur yang sebelumnya memiliki skala dan rentang nilai yang berbeda, sekarang telah disesuaikan sehingga rata-rata setiap fitur mendekati nol dan deviasi standar mendekati satu. Proses normalisasi ini memastikan bahwa semua variabel dalam dataset memiliki pengaruh yang setara dalam proses klasifikasi, yang sangat penting untuk meningkatkan kinerja model, terutama dalam algoritma seperti SVM.

Tabel 4.2 Data Setelah Dinormalisasi

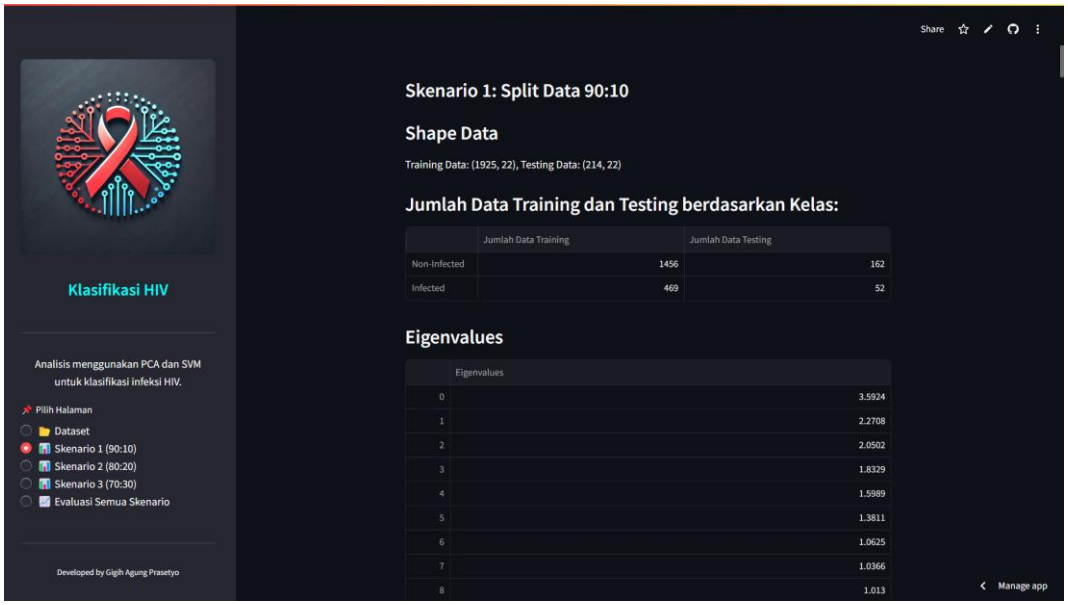
No.	***time***	***trt***	***age***	***wtkg***	***hemo***	***homo***	**…**
1.	0.23579	0.42496	1.46454	1.10764	-0.30312	-1.39654	
2.	0.42060	1.31177	2.95759	-1.93686	-0.30312	-1.39654	
3.	0.28028	1.31177	1.11999	1.00502	-0.30312	0.71605	
4.	0.98184	1.31177	1.34969	0.76556	-0.30312	0.71605	
5.	0.72175	-1.34867	0.89029	-0.63695	-0.30312	0.71605	

Tabel ini memperlihatkan hasil setelah normalisasi, di mana nilai-nilai fitur seperti *age*, *wtkg*, dan *homo* kini berada pada skala yang seragam. Sebagai contoh, fitur *age* yang sebelumnya memiliki rentang nilai antara 43 hingga 61, kini memiliki rentang antara -0.85 hingga 1.11. Demikian pula, fitur *wtkg* yang sebelumnya memiliki skala besar kini dinormalisasi dengan rentang antara -0.98 hingga 0.95. Normalisasi ini sangat penting untuk memastikan bahwa model SVM dapat bekerja secara optimal, meningkatkan akurasi prediksi, dan menghindari bias akibat perbedaan skala antar fitur.

4.3 Hasil Eksperimen Skenario 1

Gambar 4.2 menampilkan tampilan dari eksperimen pada skenario 1 dengan rasio pembagian data 90:10. Gambar ini menunjukkan informasi mengenai bentuk data, termasuk jumlah *data training* dan *testing* berdasarkan kelas, serta hasil

reduksi dimensi melalui PCA. Pembagian data yang jelas antara kelas "*Non-infected*" dan "*Infected*" juga terlihat di sini, yang memungkinkan pemahaman lebih lanjut mengenai distribusi data yang digunakan dalam eksperimen ini. Tampilan ini memberikan gambaran awal tentang bagaimana data diproses sebelum diterapkan pada model klasifikasi.



Gambar 4.2 Tampilan Pada Skenario 1

Gambar ini memberikan visualisasi yang komprehensif mengenai proses eksperimen pada skenario 1, yang mencakup informasi terkait dengan pembagian data serta hasil reduksi dimensi dan evaluasi model. Selain itu, gambar ini juga menunjukkan *eigenvalues* yang diperoleh dari proses PCA dan distribusi data untuk setiap kelas, memberikan gambaran yang lebih jelas tentang bagaimana data diproses dan disiapkan untuk pelatihan model. Visualisasi ini memudahkan analisis tentang pengaruh pembagian data terhadap kinerja model dan pentingnya pemilihan rasio pembagian yang tepat.

4.3.1 *Split Data*

Pada Tabel 4.3 menunjukkan hasil pembagian data dalam eksperimen skenario 1 dengan rasio 90:10, yang memisahkan data menjadi dua bagian utama, yaitu *data training* dan *data testing*. Pembagian ini bertujuan untuk memastikan bahwa model memiliki cukup data untuk belajar dari pola infeksi HIV, sementara juga memungkinkan evaluasi yang akurat dengan menggunakan data yang tidak terlibat dalam pelatihan. Rasio pembagian ini memengaruhi jumlah data yang dialokasikan untuk setiap bagian, yang selanjutnya berdampak pada kinerja model dalam memprediksi infeksi.

Tabel 4.3 *Split Data* Pada Skenario 1

Kelas	**Jumlah *Data Training***	**Jumlah *Data Testing***
Infected	469	52
Non-Infected	1456	162

Tabel ini memperlihatkan distribusi jumlah *data training* dan *data testing* untuk masing-masing kelas, yaitu "*Infected"* dan "*Non-Infected"*. Dalam skenario 1, sebanyak 469 *data training* dan 52 *data testing* digunakan untuk kelas "*Infected"*, sementara kelas "*Non-Infected"* memiliki 1456 *data training* dan 162 *data testing*. Pembagian ini memastikan bahwa model memiliki representasi yang cukup dari kedua kelas untuk mempelajari pola infeksi HIV dan dievaluasi secara objektif dengan *data testing* yang tidak terlihat selama pelatihan.

4.3.2 PCA

Tabel 4.4 menunjukkan nilai *eigenvalue* dan *variance ratio* untuk masing-masing komponen utama (PC) yang dihasilkan dari analisis PCA pada skenario 1. PCA digunakan dalam penelitian ini untuk mereduksi dimensi dataset, yang

memungkinkan penyederhanaan data yang kompleks tanpa kehilangan informasi yang signifikan. Setiap komponen utama (PC) menggambarkan sebagian variasi data, dengan nilai *eigenvalue* yang lebih besar menunjukkan komponen yang lebih dominan dalam menjelaskan variasi tersebut. Dalam hasil analisis ini, komponen pertama (PC1) memiliki *eigenvalue* terbesar, yang berarti bahwa PC1 adalah komponen yang paling dominan dalam menggambarkan struktur data.

Tabel 4.4 Nilai *Eigenvalue* dan *Variance Ratio* Pada Skenario 1

No.	PC	*Eigenvalue*	*Variance Ratio*
1.	PC1	3.59242355	16.34%
2.	PC2	2.27083252	10.33%
3.	PC3	2.0501587	9.33%
4.	PC4	1.83293561	8.34%
5.	PC5	1.5988731	7.27%
6.	PC6	1.38105029	6.28%
7.	PC7	1.06252316	4.83%
8.	PC8	1.03662209	4.72%
9.	PC9	1.01299304	4.61%
10.	PC10	0.95655299	4.35%
11.	PC11	0.86265348	3.92%
12.	PC12	0.82050176	3.73%
13.	PC13	0.77254735	3.51%
14.	PC14	0.65719312	2.99%
15.	PC15	0.53820601	2.45%
16.	PC16	0.43897367	1.99%
17.	PC17	0.39387511	1.79%
18.	PC18	0.22225027	1.01%
19.	PC19	0.19345758	0.88%
20.	PC20	0.14594409	0.66%
21.	PC21	0.09963407	0.45%
22.	PC22	0.0452054	0.21%

Tabel ini menunjukkan bagaimana setiap komponen utama (PC) berkontribusi terhadap variasi total data yang ada. Berdasarkan nilai *eigenvalue* dan *variance ratio*, PC1 menjelaskan sekitar 16.34% dari variasi total, diikuti oleh PC2 dengan 10.33%, dan seterusnya. Dari 22 komponen yang ada, hanya komponen yang menjelaskan sebagian besar variasi yang dipertahankan, dengan pendekatan 95% variasi yang dipilih untuk reduksi dimensi. Dalam hal ini, 17 komponen utama

dipertahankan, sehingga proses reduksi dimensi ini menyederhanakan data sambil mempertahankan sebagian besar informasi yang relevan untuk analisis lebih lanjut.

Pada Tabel 4.5 menyajikan hasil analisis PCA yang menghasilkan komponen utama dari data yang telah melalui proses reduksi dimensi. Dalam penelitian ini, sebanyak 17 komponen utama (PC1 hingga PC17) dipilih untuk mewakili variansi data yang signifikan. Setiap komponen utama tersebut mencerminkan kontribusinya terhadap variasi data secara keseluruhan. Proses PCA ini bertujuan untuk menyederhanakan data yang kompleks tanpa kehilangan informasi penting, yang akan digunakan sebagai input untuk pemodelan dengan algoritma SVM. Dengan mengurangi jumlah dimensi, PCA memungkinkan pengolahan data yang lebih efisien dan meningkatkan kinerja model.

Tabel 4.5 Sampel Data Komponen PCA Pada Skenario 1

PC	Sampel Data 1	Sampel Data 2	Sampel Data 3	Sampel Data 4	Sampel Data 5
PC1	-2.407723	1.584889	1.759147	-1.995672	3.033830
PC2	0.605532	0.349586	-0.405273	-0.351292	1.431987
PC3	-0.988089	1.007580	0.384584	1.782287	-2.188331
PC4	1.417869	-1.062565	-1.698490	0.161388	0.207258
PC5	-0.065956	-2.110960	3.084650	0.748434	0.452249
PC6	-0.061050	-0.850360	0.965863	0.854486	2.167190
PC7	0.914423	0.651330	0.271201	-0.697557	0.108225
PC8	-1.111903	-0.769246	-0.333033	0.246559	1.349159
PC9	-0.411975	0.455696	0.462475	0.183545	-0.790445
PC10	-0.642109	0.168775	-1.257351	-0.506061	0.739348
PC11	-0.059325	-0.154110	-0.913058	-0.857530	-1.439391
PC12	-0.660062	0.582627	-0.107861	-0.078782	1.630954
PC13	0.673198	0.347445	-0.476346	-0.000462	-0.580576
PC14	0.739544	-0.106707	0.673352	-0.878818	1.622589
PC15	0.675139	0.560232	0.163867	0.253677	-0.537132
PC16	-0.221536	-0.182983	-0.137771	-0.414460	0.039528
PC17	0.141136	-0.671783	-0.384942	0.181828	1.593847

Tabel 4.6 menunjukkan penamaan komponen utama (PC) yang dihasilkan dari analisis PCA pada skenario 1, beserta fitur-fitur yang terkait dengan setiap komponen utama tersebut. Setiap komponen utama dalam PCA adalah hasil dari

kombinasi linier berbagai fitur asli dalam dataset, yang menggambarkan kontribusi varians terbesar dalam data. Tabel ini memberikan gambaran tentang fitur-fitur yang memiliki kontribusi signifikan terhadap setiap komponen, yang sangat penting untuk memahami bagaimana data direduksi dan bagaimana informasi yang relevan digunakan dalam proses pemodelan.

Tabel 4.6 Penamaan Komponen PCA Pada Skenario 1

No.	Komponen Utama	Nama Fitur
1.	PC1	*Time*
2.	PC2	*Trt*
3.	PC3	*Age*
4.	PC4	*Wtkg*
5.	PC5	*Hemo*
6.	PC6	*Homo*
7.	PC7	*Drugs*
8.	PC8	*Karnof*
9.	PC9	*Oprior*
10.	PC10	*z30*
11.	PC11	*Gender*
12.	PC12	*Str*
13.	PC13	*Strat*
14.	PC14	*Symptom*
15.	PC15	*Treat*
16.	PC16	*Offtrt*

Tabel ini memperlihatkan hubungan antara komponen utama (PC) dan nama fitur yang berkontribusi pada komponen-komponen tersebut. Sebagai contoh, PC1 berhubungan dengan fitur "*Time*", sedangkan PC2 berkaitan dengan "*trt*", dan seterusnya. Informasi ini membantu untuk lebih memahami bagaimana setiap komponen utama merepresentasikan variansi data berdasarkan kombinasi dari berbagai fitur. Hal ini sangat berguna dalam tahap pemodelan selanjutnya, di mana komponen utama yang relevan digunakan untuk meningkatkan akurasi dan efisiensi model SVM dalam mengidentifikasi pola dan melakukan klasifikasi.

Pada Tabel 4.7 menyajikan sampel data berdasarkan komponen utama yang diperoleh melalui proses PCA pada skenario 1. Setiap komponen utama adalah kombinasi linier dari fitur-fitur asli yang mewakili varians terbesar dalam dataset. Proses PCA ini membantu menyederhanakan data yang kompleks tanpa mengorbankan informasi penting, sehingga memungkinkan analisis lebih lanjut menggunakan model SVM menjadi lebih efisien dan akurat. Tabel ini memberikan ilustrasi tentang bagaimana data diproses dan ditransformasi ke dalam bentuk yang lebih mudah dianalisis.

Tabel 4.7 Sampel Data Dengan Nama Fitur Pada Skenario 1

PC	Sampel Data 1	Sampel Data 2	Sampel Data 3	Sampel Data 4	Sampel Data 5
Time	-2.407723	1.584889	1.759147	-1.995672	3.033830
Trt	0.605532	0.349586	-0.405273	-0.351292	1.431987
Age	-0.988089	1.007580	0.384584	1.782287	-2.188331
Wtkg	1.417869	-1.062565	-1.698490	0.161388	0.207258
Hemo	-0.065956	-2.110960	3.084650	0.748434	0.452249
Homo	-0.061050	-0.850360	0.965863	0.854486	2.167190
Drugs	0.914423	0.651330	0.271201	-0.697557	0.108225
Karnof	-1.111903	-0.769246	-0.333033	0.246559	1.349159
Oprior	-0.411975	0.455696	0.462475	0.183545	-0.790445
z30	-0.642109	0.168775	-1.257351	-0.506061	0.739348
Gender	-0.059325	-0.154110	-0.913058	-0.857530	-1.439391
Str	-0.660062	0.582627	-0.107861	-0.078782	1.630954
Strat	0.673198	0.347445	-0.476346	-0.000462	-0.580576
Symptom	0.739544	-0.106707	0.673352	-0.878818	1.622589
Treat	0.675139	0.560232	0.163867	0.253677	-0.537132
Offtrt	-0.221536	-0.182983	-0.137771	-0.414460	0.039528
cd40	0.141136	-0.671783	-0.384942	0.181828	1.593847

Tabel ini menunjukkan hasil transformasi data berdasarkan komponen utama yang telah diidentifikasi, dengan nilai-nilai untuk setiap sampel data yang dihasilkan dari analisis PCA. Setiap komponen utama dalam tabel mencerminkan kontribusi linier dari fitur-fitur yang ada, dan nilai-nilai tersebut menggambarkan bagaimana data diproyeksikan ke dalam ruang dimensi yang lebih rendah. Hal ini memberikan gambaran yang lebih jelas tentang bagaimana PCA mengurangi

kompleksitas data sambil mempertahankan informasi penting yang relevan untuk pemodelan dan analisis lebih lanjut.

4.3.3 *Undersampling*

Tabel 4.8 menunjukkan hasil dari proses *undersampling* yang diterapkan untuk menyeimbangkan distribusi data antara kelas mayoritas dan kelas minoritas. Dalam eksperimen ini, kelas mayoritas, yaitu "*Non-Infected"*, memiliki jumlah data yang lebih besar dibandingkan kelas "*Infected*", yang dapat menyebabkan ketidakseimbangan yang memengaruhi kinerja model. Oleh karena itu, dilakukan teknik *random undersampling* dengan mereduksi jumlah data pada kelas mayoritas sehingga keduanya memiliki jumlah yang seimbang, yang penting untuk menghindari bias model terhadap kelas mayoritas.

Tabel 4.8 Jumlah Data Sebelum dan Sesudah *Undersampling* Pada Skenario 1

Kelas	**Jumlah Data Sebelum *Undersampling***	**Jumlah Data Setelah *Undersampling***
Infected	469	469
Non-Infected	1456	469

Tabel ini memperlihatkan perbandingan jumlah data sebelum dan sesudah proses *undersampling*. Sebelum *undersampling*, kelas "*Infected"* memiliki 469 data, sementara kelas "*Non-Infected"* memiliki 1456 data, yang menunjukkan ketidakseimbangan antara kedua kelas. Setelah melalui proses *undersampling*, jumlah data pada kelas "*Non-Infected"* dikurangi menjadi 469, yang sekarang setara dengan jumlah data pada kelas "*Infected"*. Dengan penyamaan jumlah data antar kelas ini, model SVM dapat belajar dari kedua kelas secara lebih seimbang, meningkatkan performa model dan mengurangi bias terhadap kelas mayoritas.

4.3.4 Pengujian Model SVM

Pada Tabel 4.9 menunjukkan konfigurasi kernel dan *hyperparameter* yang digunakan dalam eksperimen SVM pada skenario 1. Dalam eksperimen ini, kernel yang dipilih adalah RBF, yang dikenal memiliki kemampuan baik dalam menangani data *non-linear* dengan memetakan data ke dalam ruang dimensi yang lebih tinggi. Proses *tuning hyperparameter* juga dilakukan dengan menyesuaikan nilai parameter C dan γ untuk memperoleh konfigurasi terbaik yang dapat meningkatkan performa model. Tabel ini merinci berbagai percobaan yang dilakukan untuk menemukan kombinasi *hyperparameter* yang optimal.

Tabel 4.9 Kernel dan *Hyperparameter* Pada Skenario 1

No.	Percobaan	Kernel	Hyperparameter	
			C	γ
1.	1	RBF	0.1	0.1
2.	2			1
3.	3			10
4.	4		1	0.1
5.	5			1
6.	6			10
7.	7		10	0.1
8.	8			1
9.	9			10
10.	10		100	0.1
11.	11			1
12.	12			10

Tabel ini memberikan gambaran tentang berbagai konfigurasi yang diuji selama eksperimen SVM, termasuk parameter C dan γ yang diuji dengan nilai yang bervariasi. Setiap kombinasi pengaturan ini bertujuan untuk meningkatkan kemampuan model dalam melakukan klasifikasi yang akurat. Dengan adanya rincian *hyperparameter* dalam tabel, proses *tuning* dapat dianalisis untuk memahami bagaimana perubahan parameter mempengaruhi kinerja model dan efektivitasnya dalam klasifikasi data.

Tabel 4.10 menyajikan hasil evaluasi eksperimen pada skenario 1, yang dilakukan dengan menggunakan kernel RBF dan berbagai kombinasi *hyperparameter*. Setiap percobaan menguji nilai parameter C dan γ yang berbeda untuk mengukur pengaruhnya terhadap kinerja model dalam melakukan klasifikasi. Evaluasi ini dilakukan dengan menggunakan 214 *data testing* dan menghitung metrik-metrik evaluasi utama, seperti akurasi, *precision*, *recall*, dan *F1-Score*, yang diambil berdasarkan hasil prediksi model. Proses ini memberikan gambaran tentang seberapa baik model dapat memprediksi data yang belum pernah dilihat sebelumnya selama pelatihan.

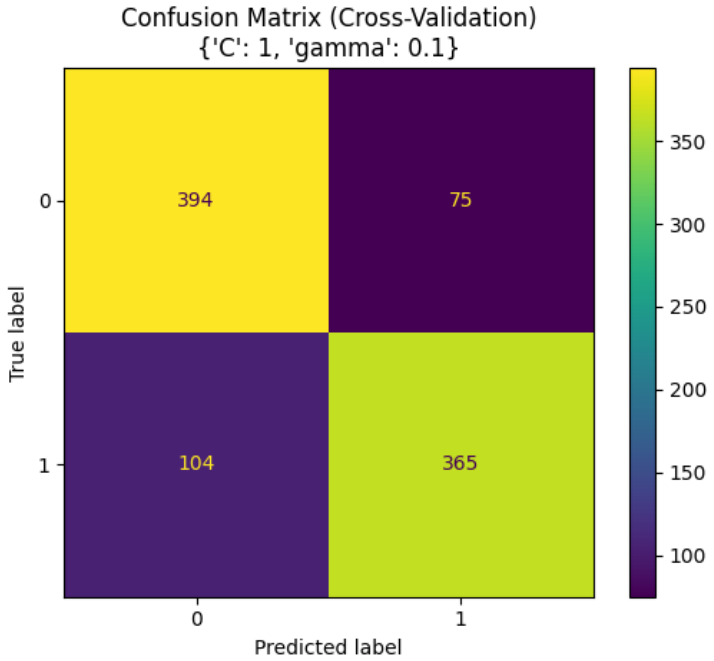
Tabel 4.10 Hasil Evaluasi Pada Skenario 1

No	**Per cobaan**	**Kernel**	***Hyperparameter***		**Evaluasi**			
			C	γ	**Akurasi**	***Precision***	***Recall***	***F1-Score***
1.	1	RBF	0.1	0.1	76.65%	73.32%	84.23%	78.31%
2.	2			1	51.91%	41.03%	79.79%	54.18%
3.	3			10	50.11%	40.05%	80.00%	53.38%
4.	4		1	0.1	80.92%	83.24%	77.82%	80.32%
5.	5			1	60.13%	56.13%	94.03%	70.27%
6.	6			10	50.64%	40.33%	80.00%	53.62%
7.	7		10	0.1	79.11%	79.82%	78.05%	78.90%
8.	8			1	60.02%	56.25%	91.90%	69.74%
9.	9			10	51.28%	40.66%	79.79%	53.86%
10.	10		100	0.1	78.68%	79.13%	78.04%	78.57%
11.	11			1	60.02%	56.25%	91.90%	69.74%
12.	12			10	51.28%	40.66%	79.79%	53.86%

Tabel ini memberikan gambaran komprehensif mengenai hasil evaluasi dari 12 percobaan yang dilakukan dengan konfigurasi *hyperparameter* yang berbeda. Berdasarkan hasil yang diperoleh, nilai akurasi tertinggi dicapai dengan pengaturan C = 1 dan γ = 0.1, yang menghasilkan akurasi sebesar 80.92%. Selain itu, metrik lainnya, seperti *precision* (83.24%), *recall* (77.82%), dan *F1-Score* (80.32%), menunjukkan performa yang cukup seimbang dan baik. Hasil ini mengindikasikan

bahwa model SVM dengan kernel RBF berhasil mengklasifikasikan data dengan akurat dan konsisten pada percobaan ini.

Pada Gambar 4.3 menampilkan *confusion matrix* dari hasil evaluasi model SVM dengan kernel RBF menggunakan konfigurasi *hyperparameter* $C = 1$ dan $\gamma = 0.1$. *Confusion matrix* ini menunjukkan jumlah prediksi yang benar dan salah dibandingkan dengan data aktual, yang memberikan wawasan mengenai bagaimana model mengklasifikasikan data dalam dua kelas: *"infected"* dan "*non-infected"*. Dalam analisis ini, kita dapat melihat bagaimana model menangani kesalahan klasifikasi, yaitu FP dan FN, yang dapat mempengaruhi kinerja model secara keseluruhan.



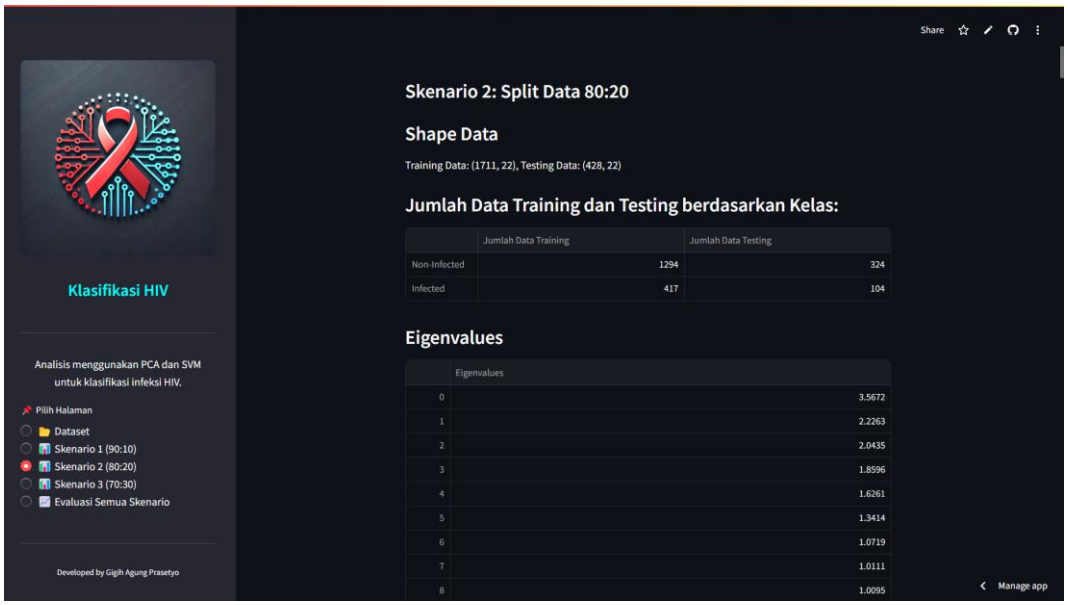
Gambar 4.3 *Hyperparameter* C = 1 dan γ= 0.1

Dari Gambar tersebut, terlihat bahwa model berhasil memprediksi 365 data terinfeksi dengan benar (TP), dan 394 data non-terinfeksi dengan benar (TN). Namun, model juga membuat beberapa kesalahan, dengan 75 data non-terinfeksi

yang salah diprediksi sebagai terinfeksi (FP) dan 104 data terinfeksi yang salah diprediksi sebagai non-terinfeksi (FN). Meskipun ada kesalahan klasifikasi, model ini masih menunjukkan performa yang baik dengan akurasi, *precision*, *recall*, dan *F1-Score* yang cukup tinggi, yang mengindikasikan bahwa model SVM dengan kernel RBF bekerja dengan baik dalam tugas klasifikasi ini.

4.4 Hasil Eksperimen Skenario 2

Gambar 4.4 menunjukkan tampilan hasil eksperimen pada skenario 2 dengan rasio pembagian data 80:20. Dalam gambar ini, ditampilkan informasi terkait dengan bentuk data dan distribusi antara *data training* dan *testing*. Pembagian data ini penting untuk memastikan model diuji dengan data yang belum pernah dilihat sebelumnya, yang memungkinkan evaluasi yang objektif terhadap performa model. Gambar ini juga menunjukkan hasil reduksi dimensi melalui PCA dan langkah-langkah pengolahan data lainnya sebelum model SVM diterapkan.



Gambar 4.4 Tampilan Pada Skenario 2

Gambar ini memberikan gambaran yang jelas mengenai seluruh proses yang terjadi pada skenario 2, mulai dari pembagian data hingga evaluasi model. Tampilan ini mencakup metrik-metrik evaluasi model seperti akurasi, *precision*, *recall*, dan *F1-score*, yang digunakan untuk menilai performa model dalam klasifikasi. Selain itu, gambar ini juga memperlihatkan hasil evaluasi model yang dilakukan melalui tuning parameter dan visualisasi *confusion matrix*, memberikan wawasan lebih dalam tentang bagaimana rasio pembagian data mempengaruhi hasil akhir eksperimen.

4.4.1 *Split Data*

Pada Tabel 4.11 menunjukkan hasil pembagian data pada skenario 2 dengan rasio 80:20, yang membagi data menjadi dua bagian utama, yaitu *data training* dan *data testing*. Pembagian ini penting untuk memastikan bahwa model dapat belajar dengan cukup data melalui *training*, sementara juga diuji secara objektif dengan data yang belum pernah dilihat sebelumnya. Dalam eksperimen ini, 80% data digunakan untuk *training*, dan 20% data sisanya digunakan untuk *testing*. Rasio pembagian ini membantu menjaga keseimbangan antara jumlah data yang digunakan untuk melatih model dan data yang digunakan untuk evaluasi model.

Tabel 4.11 *Split Data* Pada Skenario 2

Kelas	**Jumlah *Data Training***	**Jumlah *Data Testing***
Infected	417	104
Non-Infected	1294	324

Tabel ini mengilustrasikan distribusi data antara kelas "*Infected*" dan "*Non-Infected*" untuk proses *training* dan *testing*. Dari total 2139 data, 1711 data digunakan untuk *training* dan 428 data untuk *testing*. Pada kelas "*Infected*", terdapat

417 data yang digunakan untuk *training* dan 104 data untuk *testing*, sementara pada kelas "*Non-Infected*", sebanyak 1294 data digunakan untuk *training* dan 324 data untuk *testing*. Pembagian ini memastikan bahwa model memiliki cukup data untuk dilatih dan diuji, dengan data testing yang cukup untuk memberikan evaluasi yang akurat dan representatif mengenai kinerja model.

4.4.2 PCA

Tabel 4.12 menunjukkan hasil analisis PCA yang menghasilkan 22 komponen utama (PC) beserta nilai *eigenvalue* dan *variance ratio* untuk masing-masing komponen. Setiap *eigenvalue* menggambarkan kontribusi komponen utama terhadap variasi total dalam dataset. Dalam penelitian ini, PCA diterapkan untuk mereduksi dimensi data yang kompleks, sehingga memudahkan analisis lebih lanjut tanpa kehilangan informasi yang signifikan. Komponen pertama (PC1) memiliki *eigenvalue* terbesar, yang menunjukkan bahwa PC1 menjelaskan bagian terbesar dari variasi data, menjadikannya komponen paling dominan dalam menggambarkan struktur data yang ada.

Tabel 4.12 Nilai *Eigenvalue* dan *Variance Ratio Pada Skenario 2*

No.	PC	*Eigenvalue*	*Variance Ratio*
1.	PC1	3.56722073	16.30%
2.	PC2	2.22634987	10.17%
3.	PC3	2.04350521	9.34%
4.	PC4	1.85956366	8.50%
5.	PC5	1.62609057	7.43%
6.	PC6	1.34144082	6.13%
7.	PC7	1.07189168	4.90%
8.	PC8	1.01111461	4.62%
9.	PC9	1.00953054	4.61%
10.	PC10	0.9449031	4.32%
11.	PC11	0.86854418	3.97%
12.	PC12	0.81793373	3.74%
13.	PC13	0.75272027	3.44%
14.	PC14	0.6606249	3.02%
15.	PC15	0.54952473	2.51%

No.	PC	*Eigenvalue*	*Variance Ratio*
16.	PC16	0.44334211	2.03%
17.	PC17	0.39175055	1.79%
18.	PC18	0.22123912	1.01%
19.	PC19	0.19003616	0.87%
20.	PC20	0.14642481	0.67%
21.	PC21	0.09957587	0.45%
22.	PC22	0.04450824	0.20%

Tabel ini juga menunjukkan *variance ratio* untuk setiap komponen utama, yang mengindikasikan seberapa besar kontribusi setiap komponen terhadap total variasi data. Berdasarkan hasil analisis, komponen yang memiliki *eigenvalue* lebih tinggi seperti PC1 dan PC2 memberikan kontribusi yang lebih signifikan terhadap variasi data, sementara komponen lainnya memiliki kontribusi yang lebih kecil. Proses reduksi dimensi dilakukan dengan mempertahankan komponen-komponen yang menjelaskan sebagian besar variasi dalam data, sehingga meskipun jumlah komponen berkurang, informasi yang relevan tetap dipertahankan. Hasil ini mendukung upaya untuk menyederhanakan model dan meningkatkan efisiensi analisis dengan tetap menjaga informasi penting.

Pada Tabel 4.13 menyajikan hasil analisis PCA yang menghasilkan 16 komponen utama (PC1 hingga PC16), yang dipilih untuk mewakili varians data yang signifikan setelah melalui proses reduksi dimensi. Dalam penelitian ini, PCA digunakan untuk menyederhanakan data yang kompleks, dengan mempertahankan informasi penting dari data asli. Komponen-komponen utama ini akan digunakan sebagai input untuk pemodelan SVM, dengan jumlah total data hasil PCA yang mencakup proses *training* dan *testing*. Dengan mengurangi jumlah dimensi data, PCA tidak hanya mempermudah pemahaman pola dan hubungan tersembunyi

dalam dataset, tetapi juga meningkatkan efisiensi pemrosesan data pada tahap selanjutnya.

Tabel 4.13 Sampel Data Komponen PCA Pada Skenario 2

PC	Sampel Data 1	Sampel Data 2	Sampel Data 3	Sampel Data 4	Sampel Data 5
PC1	2.020339	-1.916606	-2.23520	0.433861	2.403991
PC2	-2.385532	0.390932	0.180370	1.245208	0.771574
PC3	-1.749919	1.475792	-0.101052	-1.631683	1.308187
PC4	-2.131623	0.366513	2.043200	1.008756	-0.531510
PC5	-0.119323	0.624106	-0.942733	-0.404869	-2.919630
PC6	-2.462811	-0.059843	0.383347	0.216659	-1.605738
PC7	-0.978018	0.939717	0.064719	-0.652262	-0.113995
PC8	0.748966	-0.545801	-0.432922	-0.662951	0.439792
PC9	1.193643	-1.944649	0.166253	-0.600690	1.390832
PC10	-0.505140	-0.924236	0.140198	1.839566	-0.444421
PC11	0.125865	1.012794	0.136850	0.308240	-0.489601
PC12	-0.914025	1.054119	-0.109613	-0.959601	-0.241401
PC13	0.307536	0.360510	0.319585	0.010108	-1.371018
PC14	0.155851	-0.092953	-0.308795	1.685539	0.194833
PC15	0.815807	0.504305	-0.949255	-0.802073	0.000450
PC16	0.448655	-0.905840	0.218245	-0.236631	0.068795

Tabel ini menunjukkan nilai setiap komponen utama (PC) untuk lima sampel data, yang mewakili kontribusi terhadap variasi data. Setiap nilai pada kolom sampel data menggambarkan proyeksi data ke ruang dimensi yang lebih rendah setelah proses PCA. Proses ini mengurangi kompleksitas data sambil mempertahankan informasi relevan, yang meningkatkan efisiensi dalam pemodelan dan analisis lebih lanjut. Hasil PCA ini memberikan dasar yang kuat untuk eksperimen selanjutnya dengan model SVM.

Tabel 4.14 menunjukkan penamaan setiap komponen utama (PC) yang dihasilkan dari analisis PCA pada skenario 2, serta fitur-fitur yang berkontribusi terhadap komponen-komponen tersebut. Setiap komponen utama adalah kombinasi linier dari fitur-fitur asli yang mencerminkan kontribusi varians terbesar dalam dataset. Proses PCA membantu menyederhanakan data yang kompleks tanpa

mengorbankan informasi penting, sehingga memungkinkan analisis lebih lanjut dengan model SVM menjadi lebih efisien dan akurat. Tabel ini memberikan wawasan tentang bagaimana setiap komponen utama merepresentasikan variasi dalam data dan bagaimana fitur-fitur yang relevan berkontribusi terhadap komponen tersebut.

Tabel 4.14 Penamaan Komponen PCA Pada Skenario 2

No.	Komponen Utama	Nama Fitur
1.	PC1	*Time*
2.	PC2	*Trt*
3.	PC3	*Age*
4.	PC4	*Wtkg*
5.	PC5	*Hemo*
6.	PC6	*Homo*
7.	PC7	*Drugs*
8.	PC8	*Karnof*
9.	PC9	*Oprior*
10.	PC10	*z30*
11.	PC11	*Gender*
12.	PC12	*Str*
13.	PC13	*Strat*
14.	PC14	*Symptom*
15.	PC15	*Treat*
16.	PC16	*Offtrt*

Pada Tabel 4.15 menunjukkan nilai-nilai dari setiap komponen utama (PC) untuk lima sampel data yang berbeda. Setiap komponen utama merupakan hasil dari proses PCA yang menggabungkan berbagai fitur untuk menciptakan dimensi baru yang lebih sederhana, namun tetap mempertahankan informasi penting dari data asli. Tabel ini memberikan gambaran tentang bagaimana setiap komponen utama mempengaruhi setiap sampel data dan menunjukkan kontribusi setiap fitur dalam menjelaskan variasi yang ada. Nilai-nilai yang ditampilkan menggambarkan proyeksi data ke dalam ruang dimensi yang lebih rendah setelah dilakukan reduksi dimensi menggunakan PCA.

Tabel 4.15 Sampel Data Dengan Nama Fitur Pada Skenario 2

PC	Sampel Data 1	Sampel Data 2	Sampel Data 3	Sampel Data 4	Sampel Data 5
Time	2.020339	-1.916606	-2.23520	0.433861	2.403991
Trt	-2.385532	0.390932	0.180370	1.245208	0.771574
Age	-1.749919	1.475792	-0.101052	-1.631683	1.308187
Wtkg	-2.131623	0.366513	2.043200	1.008756	-0.531510
Hemo	-0.119323	0.624106	-0.942733	-0.404869	-2.919630
Homo	-2.462811	-0.059843	0.383347	0.216659	-1.605738
Drugs	-0.978018	0.939717	0.064719	-0.652262	-0.113995
Karnof	0.748966	-0.545801	-0.432922	-0.662951	0.439792
Oprior	1.193643	-1.944649	0.166253	-0.600690	1.390832
z30	-0.505140	-0.924236	0.140198	1.839566	-0.444421
Gender	0.125865	1.012794	0.136850	0.308240	-0.489601
Str	-0.914025	1.054119	-0.109613	-0.959601	-0.241401
Strat	0.307536	0.360510	0.319585	0.010108	-1.371018
Symptom	0.155851	-0.092953	-0.308795	1.685539	0.194833
Treat	0.815807	0.504305	-0.949255	-0.802073	0.000450
Offtrt	0.448655	-0.905840	0.218245	-0.236631	0.068795

Tabel ini memperlihatkan hasil transformasi data setelah dilakukan proses PCA, di mana setiap komponen utama (PC) menggambarkan kontribusi linier dari berbagai fitur yang ada. Dengan mengurangi jumlah dimensi data, proses PCA memungkinkan untuk lebih mudah memahami pola yang tersembunyi dalam data dan meningkatkan efisiensi pemodelan. Setiap nilai pada tabel ini mencerminkan bagaimana data sampel diproyeksikan ke ruang dimensi yang lebih rendah, yang akan digunakan dalam eksperimen selanjutnya untuk pemodelan SVM atau analisis lebih lanjut.

4.4.3 *Undersampling*

Tabel 4.16 menunjukkan jumlah data sebelum dan setelah dilakukan *undersampling* pada skenario 2. Proses *undersampling* digunakan untuk menangani ketidakseimbangan data antar kelas, yang dapat mempengaruhi kinerja model dalam mengklasifikasikan data. Dalam eksperimen ini, kelas mayoritas (*Non-Infected*) memiliki jumlah data yang jauh lebih besar dibandingkan dengan kelas

minoritas (*Infected*), yang dapat membuat model cenderung lebih sensitif terhadap kelas mayoritas. Oleh karena itu, dilakukan pengurangan jumlah data pada kelas *Non-Infected* untuk menyamakan jumlahnya dengan kelas *Infected*, guna mencapai distribusi data yang lebih proporsional.

Tabel 4.16 Jumlah Data Sebelum dan Sesudah *Undersampling* Pada Skenario 2

Kelas	**Jumlah Data Sebelum *Undersampling***	**Jumlah Data Setelah *Undersampling***
Infected	417	417
Non-Infected	1294	417

Tabel ini menunjukkan hasil pembagian data sebelum dan setelah dilakukan *undersampling*. Sebelum *undersampling*, kelas *Infected* memiliki 417 data dan kelas *Non-Infected* memiliki 1294 data, yang menciptakan ketidakseimbangan antara kedua kelas. Setelah melalui proses *undersampling*, jumlah data kelas *Non-Infected* dikurangi menjadi 417, sehingga kedua kelas memiliki jumlah data yang seimbang. Pembagian yang seimbang ini bertujuan untuk menghindari bias pada model dan memungkinkan pelatihan yang lebih adil serta evaluasi yang lebih objektif.

4.4.4 Pengujian Model SVM

Pada Tabel 4.17 menunjukkan konfigurasi kernel dan *hyperparameter* yang digunakan dalam eksperimen SVM pada skenario 2. Pada eksperimen ini, kernel yang dipilih adalah RBF, yang dikenal efektif untuk menangani data non-linear dengan memetakan data ke dalam ruang dimensi yang lebih tinggi, sehingga memudahkan model dalam menemukan pemisah yang optimal. Selain itu, proses *tuning hyperparameter* juga dilakukan untuk mencari kombinasi terbaik antara

parameter C dan γ, yang dapat mempengaruhi kinerja model dalam mengklasifikasikan data.

Tabel 4.17 Kernel dan *Hyperparameter* Pada Skenario 2

No.	Percobaan	Kernel	Hyperparameter	
			C	γ
1.	1	RBF	0.1	0.1
2.	2			1
3.	3			10
4.	4		1	0.1
5.	5			1
6.	6			10
7.	7		10	0.1
8.	8			1
9.	9			10
10.	10		100	0.1
11.	11			1
12.	12			10

Tabel ini memberikan rincian eksperimen yang dilakukan dengan berbagai kombinasi nilai *hyperparameter* untuk parameter C dan γ. Setiap percobaan menguji berbagai nilai dari C dan γ untuk melihat bagaimana perubahan nilai tersebut mempengaruhi performa model SVM. Kombinasi pengaturan *hyperparameter* yang berbeda bertujuan untuk meningkatkan kemampuan model dalam melakukan klasifikasi dengan lebih baik. Dengan melihat tabel ini, kita dapat mengevaluasi bagaimana perubahan *hyperparameter* mempengaruhi efektivitas model dan membantu dalam memilih pengaturan yang optimal untuk eksperimen lebih lanjut.

Tabel 4.18 menunjukkan hasil evaluasi eksperimen pada skenario 2 dengan menggunakan kernel RBF dan berbagai kombinasi *hyperparameter*. Dalam eksperimen ini, evaluasi model dilakukan menggunakan 428 data testing untuk mengukur sejauh mana model dapat memprediksi dengan akurat. Evaluasi ini melibatkan perhitungan metrik-metrik utama seperti akurasi, *precision*, *recall*, dan

F1-Score, yang dihitung berdasarkan nilai TP, FP, TN, dan FN. Hasil dari eksperimen ini memberikan gambaran yang jelas tentang performa model dalam melakukan klasifikasi pada data yang belum pernah dilihat sebelumnya.

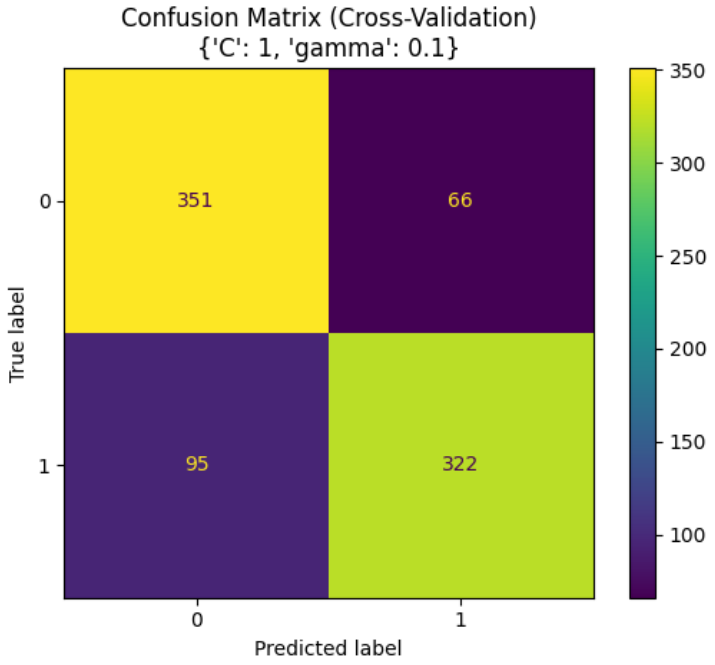
Tabel 4.18 Hasil Evaluasi Pada Skenario 2

No	Per cobaan	Kernel	*Hyperparameter*		Evaluasi			
			C	γ	Akurasi	*Precision*	*Recall*	*F1-Score*
1.	1	RBF	0.1	0.1	77.10%	74.76%	82.02%	78.17%
2.	2			1	50.48%	30.26%	59.76%	40.17%
3.	3			10	49.76%	29.88%	60.00%	39.89%
4.	4		1	0.1	80.70%	83.13%	77.21%	79.97%
5.	5			1	57.20%	54.27%	91.38%	68.08%
6.	6			10	50.12%	50.00%	60.24%	40.47%
7.	7		10	0.1	80.10%	80.68%	79.37%	79.93%
8.	8			1	58.04%	54.89%	89.93%	68.15%
9.	9			10	50.12%	50.06%	60.24%	40.53%
10.	10		100	0.1	80.46%	81.36%	79.36%	80.20%
11.	11			1	58.04%	54.89%	89.93%	68.15%
12.	12			10	50.12%	50.06%	60.24%	40.53%

Tabel ini menunjukkan hasil dari 12 percobaan yang dilakukan dengan berbagai konfigurasi *hyperparameter* C dan γ. Berdasarkan hasil evaluasi, nilai akurasi tertinggi diperoleh pada percobaan dengan konfigurasi C = 1 dan γ= 0.1, yang menghasilkan akurasi sebesar 80.70%. Selain akurasi, metrik lainnya seperti *precision* (83.13%), *recall* (77.21%), dan *F1-Score* (79.97%) juga menunjukkan performa yang seimbang dan baik. Hasil ini menunjukkan bahwa model SVM dengan kernel RBF mampu mengklasifikasikan data secara efisien dan konsisten, dengan keseimbangan yang baik antara precision dan recall.

Pada Gambar 4.5 menunjukkan *confusion matrix* dari hasil evaluasi model SVM yang menggunakan kernel RBF dengan konfigurasi *hyperparameter* C = 1 dan γ= 0.1. *Confusion matrix* ini memungkinkan analisis performa model dengan menunjukkan jumlah prediksi yang benar dan salah terhadap data aktual. Dari hasil ini, kita bisa melihat bagaimana model mengklasifikasikan data kelas positif

(terinfeksi HIV) dan kelas negatif (tidak terinfeksi HIV), serta mengidentifikasi kesalahan klasifikasi seperti FP dan FN. Dengan informasi ini, metrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1-Score* dapat dihitung untuk menilai kinerja model secara keseluruhan.

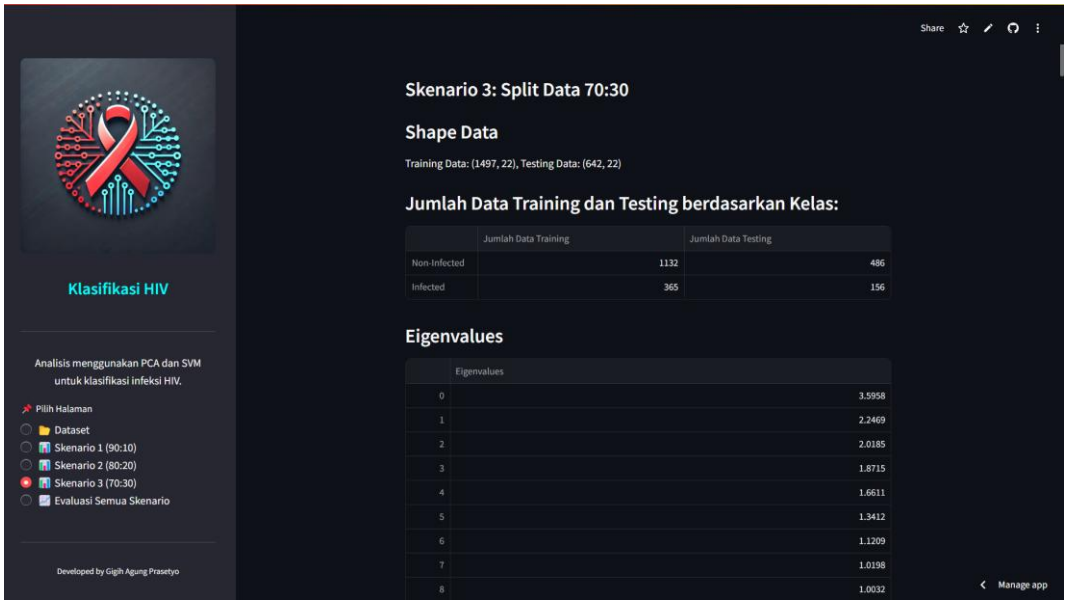


Gambar 4.5 *Hyperparameter* C = 1 dan γ= 0.1

Gambar ini memberikan rincian tentang performa model SVM dengan konfigurasi C = 1 dan γ= 0.1. Berdasarkan *confusion matrix*, model berhasil memprediksi 322 data terinfeksi dengan benar (TP) dan 351 data non-terinfeksi dengan benar (TN). Namun, model juga membuat kesalahan dengan memprediksi 66 data non-terinfeksi sebagai terinfeksi (FP) dan 95 data terinfeksi sebagai non-terinfeksi (FN). Meskipun terdapat kesalahan klasifikasi, hasil evaluasi menunjukkan nilai akurasi yang cukup tinggi, dengan *precision*, *recall*, dan *F1-Score* yang seimbang, mengindikasikan bahwa model SVM dengan kernel RBF berfungsi dengan baik dalam mengklasifikasikan data.

4.5 Hasil Eksperimen Skenario 3

Gambar 4.6 menunjukkan tampilan hasil eksperimen pada skenario 3 dengan rasio pembagian data 70:30. Gambar ini memperlihatkan informasi terkait bentuk data dan distribusi antara *data training* dan *data testing*, yang merupakan langkah penting dalam memastikan kualitas pelatihan dan evaluasi model. Proses *split data* ini memastikan model diuji dengan data yang belum pernah dilihat selama pelatihan, memberikan gambaran objektif tentang kinerja model. Gambar ini juga memberikan insight tentang bagaimana rasio pembagian data mempengaruhi eksperimen dan performa model.



Gambar 4.6 Tampilan Pada Skenario 3

Gambar ini memberikan gambaran visual yang komprehensif mengenai langkah-langkah yang dilakukan pada skenario 3. Di dalamnya, terdapat informasi mengenai distribusi data training dan testing, proses reduksi dimensi melalui PCA, dan evaluasi model menggunakan metrik-metrik seperti akurasi, *precision*, *recall*, dan *F1-score*. Gambar ini membantu untuk memahami bagaimana pemrosesan data

dan evaluasi model berjalan secara keseluruhan, memberikan pemahaman yang lebih mendalam mengenai eksperimen dan hasil yang diperoleh pada skenario ini.

4.5.1 *Split Data*

Pada Tabel 4.19 menunjukkan pembagian data pada skenario 3 dengan rasio 70:30, di mana 70% data digunakan untuk *training* dan 30% untuk testing. Pembagian ini bertujuan untuk memastikan model dilatih dengan cukup data sambil tetap diuji dengan data yang tidak terlibat dalam proses pelatihan, sehingga evaluasi model dapat dilakukan secara objektif. Rasio ini memberikan porsi *data testing* yang lebih besar dibandingkan dengan skenario sebelumnya, yang memungkinkan untuk pengujian yang lebih menyeluruh mengenai kemampuan model dalam melakukan klasifikasi data yang belum pernah dilihat sebelumnya.

Tabel 4.19 *Split Data* Pada Skenario 3

Kelas	**Jumlah *Data Training***	**Jumlah *Data Testing***
Infected	365	156
Non-Infected	1132	486

Tabel ini memperlihatkan distribusi data untuk setiap kelas, dengan jumlah *data training* dan *testing* untuk kelas "*Infected*" dan "*Non-Infected*". Pada skenario ini, *data training* untuk kelas "*Infected*" sebanyak 365 dan *data testing* sebanyak 156, sementara untuk kelas "*Non-Infected*", *data training* berjumlah 1132 dan *data testing* sebanyak 486. Meskipun jumlah *data training* berkurang, peningkatan jumlah *data testing* memungkinkan evaluasi yang lebih akurat terhadap model, memberikan gambaran yang lebih luas mengenai bagaimana model mengklasifikasikan data yang tidak seimbang antara kedua kelas tersebut.

4.5.2 PCA

Tabel 4.20 menunjukkan hasil analisis PCA yang menghasilkan nilai *eigenvalue* dan *variance ratio* untuk setiap komponen utama (PC) pada skenario 3. Setiap *eigenvalue* menggambarkan kontribusi dari setiap komponen utama terhadap variasi data, sementara *variance ratio* menunjukkan persentase kontribusi dari masing-masing komponen utama terhadap total variasi dalam dataset. Dalam penelitian ini, PCA diterapkan untuk mereduksi dimensi dataset yang kompleks, dengan tujuan untuk menyederhanakan data tanpa menghilangkan informasi penting. Komponen pertama (PC1) memiliki *eigenvalue* terbesar, yang menunjukkan bahwa PC1 menjelaskan sebagian besar variasi dalam data, menjadikannya komponen yang paling dominan.

Tabel 4.20 Nilai *Eigenvalue* dan *Variance Ratio* Pada Skenario 3

No.	**PC**	***Eigenvalue***	***Variance Ratio***
1.	PC1	3.59580759	16.28%
2.	PC2	2.24691582	10.17%
3.	PC3	2.0185469	9.14%
4.	PC4	1.87154236	8.47%
5.	PC5	1.66109022	7.52%
6.	PC6	1.34121123	6.07%
7.	PC7	1.12092759	5.07%
8.	PC8	1.01982149	4.62%
9.	PC9	1.00316386	4.54%
10.	PC10	0.96587811	4.37%
11.	PC11	0.88120099	3.99%
12.	PC12	0.84161684	3.81%
13.	PC13	0.74855091	3.39%
14.	PC14	0.66489556	3.01%
15.	PC15	0.55929107	2.53%
16.	PC16	0.44865725	2.03%
17.	PC17	0.39698783	1.80%
18.	PC18	0.21711181	0.98%
19.	PC19	0.19081849	0.86%
20.	PC20	0.14789889	0.67%
21.	PC21	0.10336106	0.47%
22.	PC22	0.0451002	0.20%

Tabel ini memberikan informasi rinci tentang *eigenvalue* dan *variance ratio* dari 22 komponen utama, dengan PC1 menunjukkan kontribusi terbesar terhadap variasi data. Berdasarkan pendekatan *variance* 95%, komponen-komponen yang menjelaskan sebagian besar variasi dalam data dipertahankan, sementara komponen yang kontribusinya kecil dapat diabaikan. Dalam hal ini, meskipun ada 22 komponen, hanya 16 komponen yang dipertahankan karena mereka dapat menjelaskan lebih dari 95% variasi data. Proses reduksi dimensi ini membantu menyederhanakan model dan meningkatkan efisiensi analisis dengan tetap mempertahankan informasi yang relevan dari data asli.

Pada Tabel 4.21 menunjukkan hasil analisis PCA yang menghasilkan komponen utama dari data yang telah melalui proses reduksi dimensi. Dalam penelitian ini, sebanyak 16 komponen utama (PC1 hingga PC16) dipilih sebagai representasi dari varians data yang signifikan. Setiap komponen utama merupakan kombinasi linier dari fitur-fitur yang ada dalam dataset, yang mewakili sebagian besar variasi dalam data. Hasil PCA ini akan digunakan sebagai input untuk pemodelan SVM, dengan total data hasil PCA sebanyak 2.139 yang mencakup data untuk proses *training* dan *testing*. Penggunaan PCA memungkinkan penyederhanaan data yang kompleks sambil tetap mempertahankan informasi penting dari data asli.

Tabel 4.21 Sampel Data Komponen PCA Pada Skenario 3

PC	Sampel Data 1	Sampel Data 2	Sampel Data 3	Sampel Data 4	Sampel Data 5
PC1	-2.008795	-1.819306	- 2.257306	-1.945967	2.103730
PC2	-1.374471	-1.231499	-0.072864	-1.139868	-0.323666
PC3	-1.255959	0.576377	0.271511	-1.219789	-0.666245
PC4	1.588414	0.783569	1.375709	0.916152	0.712752
PC5	-0.162329	1.451587	-1.746477	0.774616	1.272913
PC6	-0.011029	-1.041123	-0.201105	-0.373572	1.092982

PC	Sampel Data 1	Sampel Data 2	Sampel Data 3	Sampel Data 4	Sampel Data 5
PC7	0.533714	-0.241787	-0.107878	-0.580846	-0.366987
PC8	-1.122244	0.666400	-0.089039	1.824852	0.349316
PC9	2.060990	0.107568	0.735093	0.650865	-0.491197
PC10	0.531312	-1.549649	0.272828	0.760487	-0.312200
PC11	0.611522	-1.975786	0.260275	0.581556	-0.812371
PC12	-0.994823	1.086240	0.096865	1.490824	-0.402504
PC13	-0.471526	-0.525750	0.005281	1.600374	0.338392
PC14	0.252394	2.378718	-0.379503	0.511299	-0.933262
PC15	-0.098327	0.627832	-0.087753	-0.309448	0.162717
PC16	-0.112851	0.163981	-0.328754	1.108854	0.912442

Tabel ini memberikan gambaran tentang bagaimana komponen utama (PC1 hingga PC16) mempengaruhi data sampel yang berbeda. Setiap nilai dalam tabel menunjukkan proyeksi sampel data ke dalam ruang dimensi yang lebih rendah setelah dilakukan reduksi dimensi dengan PCA. Dengan mengurangi jumlah dimensi, data yang lebih sederhana dapat dianalisis dengan lebih efisien tanpa kehilangan informasi penting yang relevan untuk pemodelan lebih lanjut. Hasil transformasi ini memberikan dasar yang kuat untuk eksperimen selanjutnya, khususnya dalam penerapan model SVM untuk klasifikasi yang lebih akurat dan efisien.

Tabel 4.22 menunjukkan penamaan komponen utama yang diperoleh dari hasil analisis PCA pada skenario 3. Setiap komponen utama merupakan kombinasi linier dari berbagai fitur asli dalam dataset yang berkontribusi terhadap variasi terbesar dalam data. Dengan menggunakan PCA, kita dapat mengurangi kompleksitas data yang tinggi, tetapi tetap mempertahankan informasi yang penting. Komponen-komponen utama ini memainkan peran penting dalam analisis data lebih lanjut dan memungkinkan model untuk melakukan identifikasi pola yang lebih efisien, terutama dalam klasifikasi menggunakan SVM.

Tabel 4.22 Penamaan Komponen PCA Pada Skenario 3

No.	Komponen Utama	Nama Fitur
1.	PC1	*Time*
2.	PC2	*Trt*
3.	PC3	*Age*
4.	PC4	*Wtkg*
5.	PC5	*Hemo*
6.	PC6	*Homo*
7.	PC7	*Drugs*
8.	PC8	*Karnof*
9.	PC9	*Oprior*
10.	PC10	*z30*
11.	PC11	*Gender*
12.	PC12	*Str*
13.	PC13	*Strat*
14.	PC14	*Symptom*
15.	PC15	*Treat*
16.	PC16	*Offtrt*

Tabel ini memberikan daftar nama fitur yang terkait dengan setiap komponen utama (PC), yang dihasilkan melalui proses PCA. Proses ini memastikan bahwa setiap komponen utama menggambarkan varians terbesar dalam data yang ada. Nama-nama fitur ini mencerminkan kontribusi linier dari fitur asli dalam dataset, yang memungkinkan model untuk menangkap pola-pola penting. Dengan menggunakan komponen-komponen utama ini, analisis lebih lanjut dapat dilakukan dengan lebih efisien, meningkatkan kualitas dan akurasi pemodelan SVM dalam mengklasifikasikan data sesuai dengan hasil yang diharapkan.

Pada Tabel 4.23 menyajikan hasil transformasi data sampel dengan menggunakan komponen utama (PC) dari hasil analisis PCA pada skenario 3. Setiap komponen utama dihasilkan dari kombinasi linier berbagai fitur dalam dataset yang mewakili sebagian besar varians data. Proses PCA ini memungkinkan data yang awalnya kompleks untuk disederhanakan ke dalam dimensi yang lebih rendah tanpa kehilangan informasi penting. Tabel ini menunjukkan bagaimana data

sampel diproyeksikan ke dalam ruang dimensi yang lebih rendah setelah dilakukan reduksi dimensi, yang berfungsi untuk memperjelas pola data yang ada.

Tabel 4.23 Sampel Data Dengan Nama Fitur Pada Skenario 3

PC	Sampel Data 1	Sampel Data 2	Sampel Data 3	Sampel Data 4	Sampel Data 5
Time	-2.008795	-1.819306	- 2.257306	-1.945967	2.103730
Trt	-1.374471	-1.231499	-0.072864	-1.139868	-0.323666
Age	-1.255959	0.576377	0.271511	-1.219789	-0.666245
Wtkg	1.588414	0.783569	1.375709	0.916152	0.712752
Hemo	-0.162329	1.451587	-1.746477	0.774616	1.272913
Homo	-0.011029	-1.041123	-0.201105	-0.373572	1.092982
Drugs	0.533714	-0.241787	-0.107878	-0.580846	-0.366987
Karnof	-1.122244	0.666400	-0.089039	1.824852	0.349316
Oprior	2.060990	0.107568	0.735093	0.650865	-0.491197
z30	0.531312	-1.549649	0.272828	0.760487	-0.312200
Gender	0.611522	-1.975786	0.260275	0.581556	-0.812371
Str	-0.994823	1.086240	0.096865	1.490824	-0.402504
Strat	-0.471526	-0.525750	0.005281	1.600374	0.338392
Symptom	0.252394	2.378718	-0.379503	0.511299	-0.933262
Treat	-0.098327	0.627832	-0.087753	-0.309448	0.162717
Offtrt	-0.112851	0.163981	-0.328754	1.108854	0.912442

Tabel ini memperlihatkan nilai-nilai proyeksi data untuk setiap komponen utama (PC) dari lima sampel data yang berbeda. Nilai yang ditampilkan pada setiap kolom menggambarkan bagaimana setiap sampel data diproyeksikan ke dalam ruang komponen utama setelah dilakukan PCA. Hasil ini memberikan gambaran mengenai pengaruh setiap komponen utama terhadap data yang lebih sederhana dan memfasilitasi pemahaman yang lebih baik terhadap hubungan antar fitur yang ada. Dengan menggunakan hasil transformasi ini, analisis lanjutan, seperti pemodelan SVM, dapat dilakukan dengan lebih efisien.

4.5.3 *Undersampling*

Tabel 4.24 menyajikan hasil pembagian data sebelum dan setelah dilakukan proses *undersampling* pada skenario 3. Pada eksperimen ini, metode *random undersampling* diterapkan untuk menangani ketidakseimbangan kelas antara data

"*Infected*" dan "*Non-Infected*". Penyeimbangan ini bertujuan untuk memastikan bahwa model yang dibangun tidak cenderung lebih sensitif terhadap kelas mayoritas, yang bisa mengurangi efektivitas dalam mengklasifikasikan data dari kelas minoritas. Dengan melakukan *undersampling*, data dari kelas mayoritas (*Non-Infected*) direduksi sehingga jumlahnya sama dengan data dari kelas minoritas (*Infected*).

Tabel 4.24 Jumlah Data Sebelum dan Sesudah *Undersampling* Pada Skenario 3

Kelas	**Jumlah Data Sebelum *Undersampling***	**Jumlah Data Setelah *Undersampling***
Infected	365	365
Non-Infected	1132	365

Tabel ini menunjukkan pembagian data sebelum dan setelah proses *undersampling*. Sebelum dilakukan *undersampling*, kelas "*Infected*" memiliki 365 data, sedangkan kelas "*Non-Infected*" memiliki 1132 data. Ketidakseimbangan ini dapat mempengaruhi kinerja model, karena model SVM akan lebih terpengaruh oleh kelas mayoritas. Setelah proses *undersampling*, jumlah data pada kelas "*Non-Infected*" dikurangi hingga setara dengan kelas "*Infected*", yaitu sebanyak 365 data untuk masing-masing kelas. Proses ini memastikan bahwa model dilatih dengan data yang seimbang dan dapat mengklasifikasikan kedua kelas secara lebih adil.

4.5.4 Pengujian Model SVM

Pada Tabel 4.25 menunjukkan rincian konfigurasi kernel dan *hyperparameter* yang digunakan dalam eksperimen skenario 3. Dalam eksperimen ini, model SVM menggunakan kernel RBF, yang dipilih karena kemampuannya dalam menangani data yang tidak linear dengan memetakan data ke ruang dimensi yang lebih tinggi. Selain pemilihan kernel, proses *tuning hyperparameter* juga

diterapkan untuk mengoptimalkan performa model, dengan parameter C dan γ yang diatur dengan berbagai nilai. Tuning *hyperparameter* ini bertujuan untuk menemukan kombinasi terbaik yang menghasilkan model dengan akurasi tertinggi.

Tabel 4.25 Kernel dan *Hyperparameter* Pada Skenario 3

No.	Percobaan	Kernel	Hyperparameter	
			C	γ
1.	1	RBF	0.1	0.1
2.	2			1
3.	3			10
4.	4		1	0.1
5.	5			1
6.	6			10
7.	7		10	0.1
8.	8			1
9.	9			10
10.	10		100	0.1
11.	11			1
12.	12			10

Tabel ini merinci hasil percobaan yang dilakukan dengan menguji beberapa konfigurasi *hyperparameter*, termasuk nilai C dan γ. Setiap kombinasi parameter diuji untuk mengevaluasi pengaruhnya terhadap kinerja model dalam mengklasifikasikan data. Dengan mencoba berbagai konfigurasi, diharapkan diperoleh set parameter yang optimal, yang akan meningkatkan akurasi dan keseimbangan antara *precision*, *recall*, dan *F1-score* pada model SVM. Analisis hasil dari percobaan ini memberikan gambaran yang jelas mengenai pengaruh tuning parameter terhadap performa model klasifikasi.

Tabel 4.26 menyajikan hasil evaluasi dari eksperimen yang dilakukan menggunakan model SVM dengan kernel RBF dan berbagai konfigurasi *hyperparameter*, termasuk nilai C dan γ. Evaluasi dilakukan untuk mengukur performa model berdasarkan empat metrik utama: Akurasi, *Precision*, *Recall*, dan *F1-Score*, yang dihitung berdasarkan *confusion matrix*. Hasil evaluasi ini

menggunakan 642 *data testing* untuk menguji kemampuan model dalam mengklasifikasikan data dengan benar. Evaluasi yang dilakukan mencakup berbagai percobaan untuk menemukan konfigurasi *hyperparameter* terbaik yang menghasilkan performa optimal.

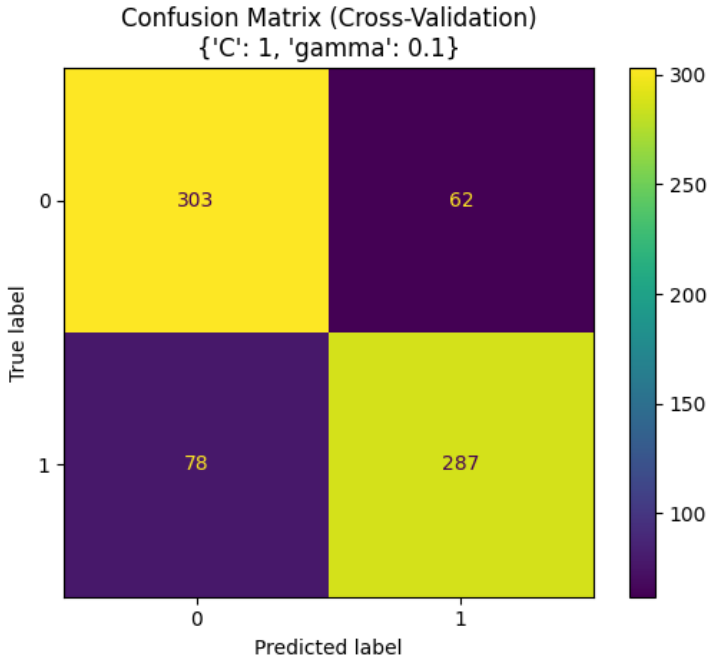
Tabel 4.26 Hasil Evaluasi Pada Skenario 3

No	Percobaan	Kernel	*Hyperparameter*		Evaluasi			
			C	γ	Akurasi	*Precision*	*Recall*	*F1-Score*
1.	1	RBF	0.1	0.1	70.55%	65.71%	86.85%	74.74%
2.	2			1	53.15%	51.66%	99.73%	68.05%
3.	3			10	50.00%	30.00%	60.00%	40.00%
4.	4		1	0.1	80.82%	82.54%	78.63%	80.39%
5.	5			1	60.68%	56.47%	95.07%	70.81%
6.	6			10	50.14%	50.00%	60.00%	40.48%
7.	7		10	0.1	79.32%	80.28%	78.36%	79.14%
8.	8			1	61.23%	57.05%	91.78%	70.33%
9.	9			10	50.27%	50.07%	60.00%	40.54%
10.	10		100	0.1	77.40%	78.38%	76.16%	77.13%
11.	11			1	61.23%	57.05%	91.78%	70.33%
12.	12			10	50.27%	50.07%	60.00%	40.54%

Tabel ini menunjukkan hasil dari 12 percobaan yang dilakukan pada skenario 3. Dalam eksperimen ini, konfigurasi *hyperparameter* C = 1 dan γ= 0.1 menghasilkan akurasi tertinggi sebesar 80.82%. Selain itu, metrik evaluasi lainnya juga menunjukkan keseimbangan yang baik, dengan *precision* 82.54%, *recall* 78.63%, dan *F1-Score* 80.39%. Hasil ini membuktikan bahwa model SVM dengan kernel RBF mampu memberikan kinerja yang baik dalam klasifikasi data. Rincian lebih lanjut mengenai performa model ini, khususnya pada konfigurasi terbaik, dapat dilihat pada gambar selanjutnya yang menunjukkan *confusion matrix* hasil dari kombinasi *hyperparameter* tersebut.

Pada Gambar 4.7 menampilkan *confusion matrix* dari hasil evaluasi model SVM dengan kernel RBF menggunakan konfigurasi *hyperparameter* C = 1 dan γ= 0.1. *Confusion matrix* ini memberikan gambaran mengenai performa model dalam

mengklasifikasikan *data testing*, dengan memisahkan jumlah prediksi yang benar dan salah untuk setiap kelas. Dalam hal ini, data yang diprediksi benar sebagai kelas positif dan negatif dihitung sebagai TP dan TN, sementara data yang diprediksi salah sebagai kelas positif atau negatif dihitung sebagai FP dan FN. Analisis ini memberikan wawasan mendalam tentang bagaimana model bekerja dalam mendeteksi pola-pola penting dalam data yang terinfeksi HIV dan tidak terinfeksi.



Gambar 4.7 *Hyperparameter* C = 1 dan γ= 0.1

Gambar tersebut menunjukkan bahwa model SVM dengan kernel RBF menghasilkan hasil klasifikasi dengan baik. Total TP yang terdeteksi berjumlah 287, di mana model berhasil mengidentifikasi data positif (terinfeksi HIV) dengan tepat. Sedangkan TN sebanyak 303, yang menunjukkan bahwa model dengan benar mengklasifikasikan data negatif (tidak terinfeksi HIV) sebagai negatif. Meskipun terdapat beberapa kesalahan klasifikasi, dengan FP sebanyak 62 dan FN sebanyak 78, hasil evaluasi ini menunjukkan bahwa model memiliki kinerja yang baik dalam

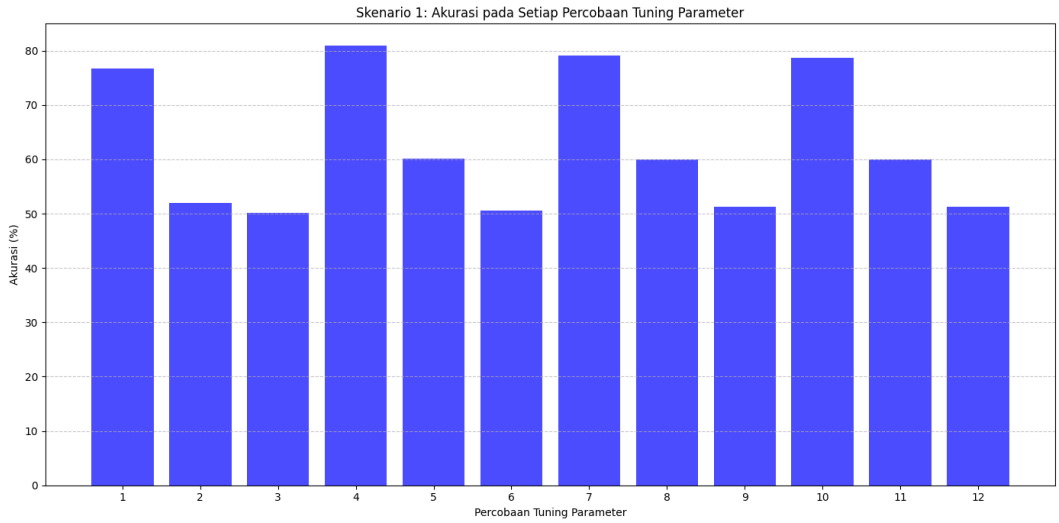
hal akurasi, *precision*, *recall*, dan *F1-Score*. Dengan demikian, meskipun model memiliki kekurangan dalam beberapa kesalahan klasifikasi, secara keseluruhan, model SVM dengan kernel RBF menunjukkan performa yang cukup memadai.

4.6 Analisis dan Pembahasan Hasil Eksperimen

Penelitian ini dilakukan dengan tiga skenario utama yang menggunakan rasio pembagian data berbeda untuk menentukan skenario dengan performa optimal dalam mengklasifikasikan infeksi HIV menggunakan metode SVM. Pada setiap skenario, peneliti menerapkan kernel RBF sebagai fungsi kernel, disertai dengan variasi parameter C dan γ. Tujuannya adalah untuk menemukan kombinasi parameter terbaik yang dapat meningkatkan akurasi dan performa model.

4.6.1 Analisis Eksperimen Skenario 1

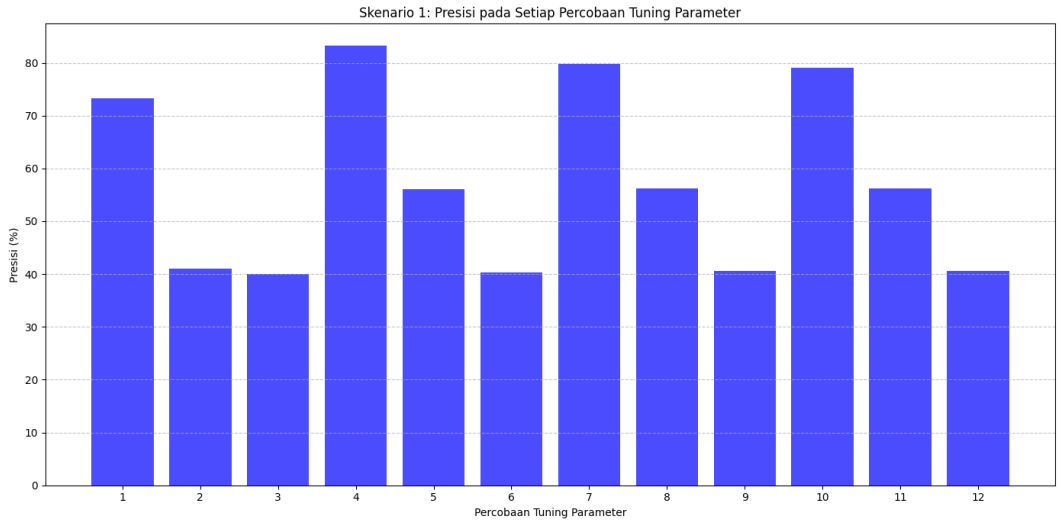
Gambar 4.8 menunjukkan diagram yang menggambarkan hasil akurasi dari setiap percobaan *tuning* parameter pada eksperimen skenario 1. Dalam eksperimen ini, model SVM dengan kernel RBF diuji dengan berbagai kombinasi nilai parameter C dan γ. Grafik ini memberikan gambaran tentang bagaimana akurasi model berubah sesuai dengan variasi parameter yang diterapkan dalam setiap percobaan. Hasil ini mencerminkan seberapa baik model dapat mengklasifikasikan data dengan akurat, tergantung pada konfigurasi *hyperparameter* yang diuji.



Gambar 4.8 Diagram Akurasi Pada Skenario 1

Berdasarkan Gambar tersebut, dapat dilihat bahwa nilai akurasi model SVM mengalami variasi pada setiap percobaan yang dilakukan dalam eksperimen skenario 1. Dari grafik tersebut, dapat dilihat bahwa akurasi model tertinggi dicapai pada percobaan dengan nilai C = 1 dan γ = 0.1, yang menghasilkan akurasi sebesar 80.92%. Ini menunjukkan bahwa kombinasi parameter tersebut memberikan kinerja terbaik dalam mengklasifikasikan data, sementara kombinasi lainnya menghasilkan akurasi yang lebih rendah. Dengan menggunakan grafik ini, kita dapat mengevaluasi secara visual pengaruh *tuning hyperparameter* terhadap performa model.

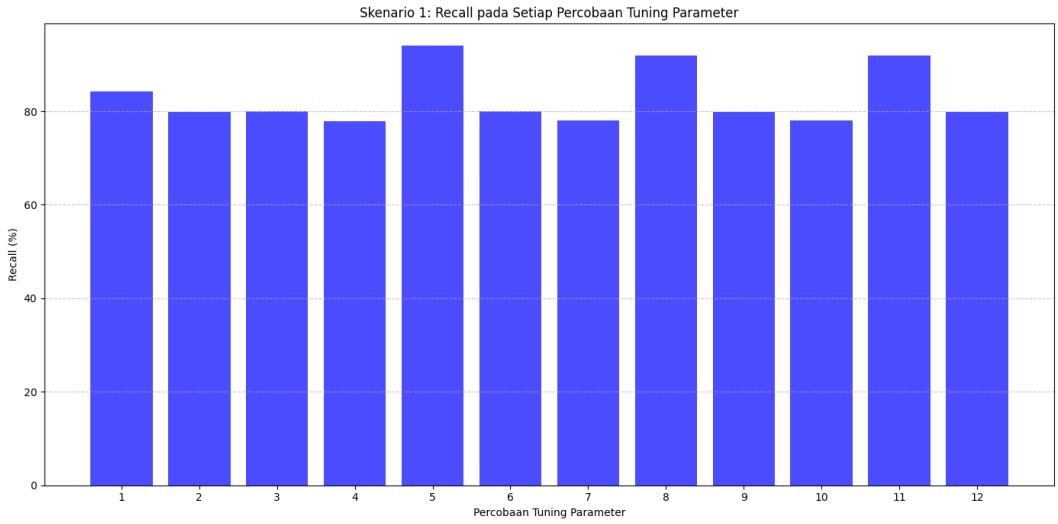
Pada Gambar 4.9, ditampilkan hasil pengujian model berdasarkan nilai *precision* yang diukur pada setiap percobaan *tuning* parameter. Diagram ini menggambarkan perbandingan *precision* yang dicapai oleh model pada berbagai percobaan dengan variasi parameter yang digunakan. Hasil ini memberikan gambaran penting tentang seberapa andal model dalam mengklasifikasikan data dengan benar, yang pada gilirannya berkontribusi pada evaluasi kinerja model.



Gambar 4.9 Diagram *Precision* Pada Skenario 1

Gambar tersebut menunjukkan bahwa percobaan ke-4 menghasilkan nilai *precision* tertinggi, mencapai 83.24%. Hal ini menunjukkan bahwa kombinasi parameter pada percobaan tersebut berhasil memberikan kinerja terbaik dalam klasifikasi. Meskipun demikian, ada fluktuasi nilai precision pada percobaan lainnya, yang menunjukkan bahwa model memiliki sensitivitas terhadap variasi parameter, yang perlu diperhatikan dalam pengoptimalan model untuk hasil yang lebih konsisten.

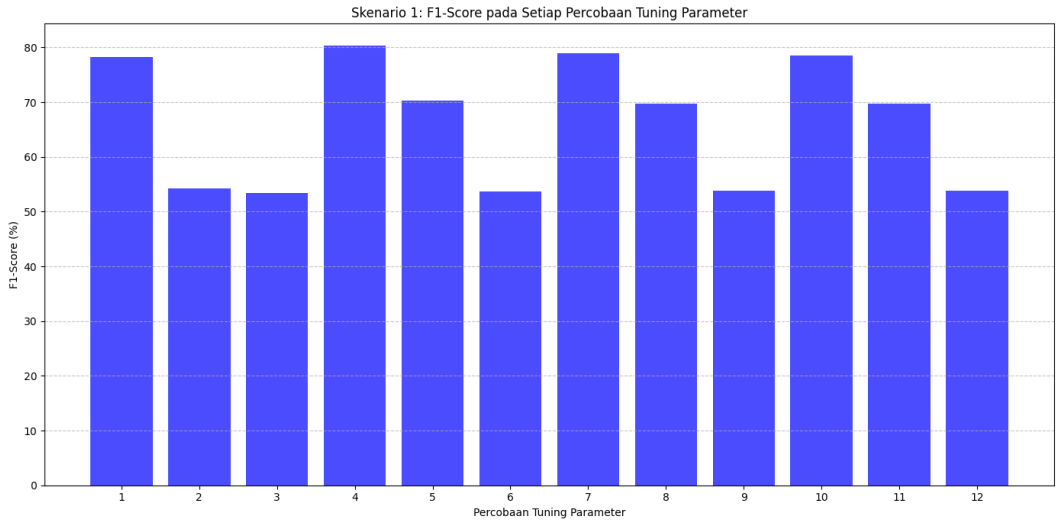
Gambar 4.10, dapat dilihat diagram yang menggambarkan nilai *recall* pada setiap percobaan *tuning* parameter. Recall adalah metrik yang mengukur kemampuan model dalam mendeteksi semua kasus positif dengan benar. Diagram ini memberikan gambaran mengenai kinerja model dalam mengenali data yang relevan, yang sangat penting dalam konteks aplikasi yang mengutamakan deteksi kasus positif sebanyak mungkin.



Gambar 4.10 Diagram *Recall* Pada Skenario 1

Dari Gambar tersebut, terlihat bahwa nilai *recall* tertinggi tercatat pada percobaan ke-5 dengan angka mencapai 94.03%. Angka ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mendeteksi seluruh kasus positif. Namun, perlu dicatat bahwa meskipun nilai *recall* tinggi, hal ini perlu diwaspadai, karena dapat mengarah pada penurunan *precision*, yang menunjukkan potensi *overfitting* terhadap kelas positif.

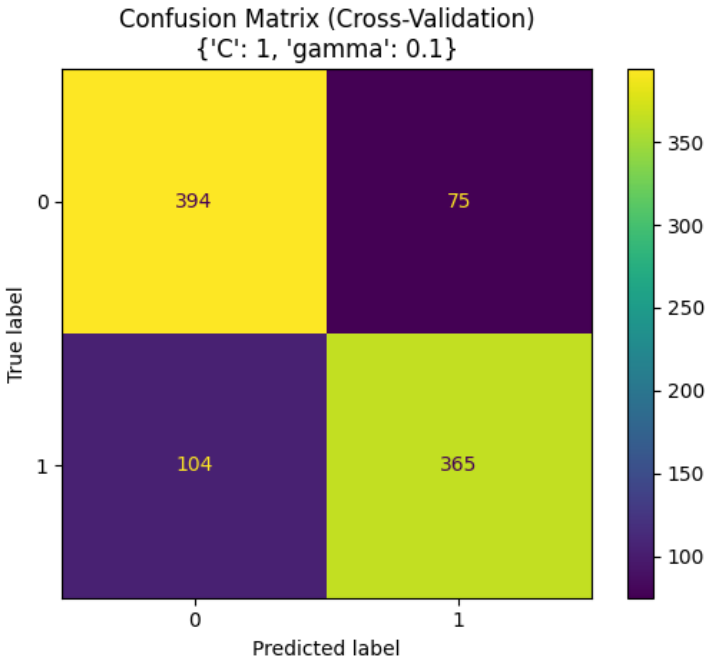
Pada Gambar 4.11, ditampilkan nilai *F1-Score* pada setiap percobaan *tuning* parameter, yang merupakan gabungan antara *precision* dan *recall*. *F1-Score* memberikan gambaran yang lebih seimbang tentang kemampuan model dalam mengklasifikasikan data dengan mempertimbangkan baik kemampuan model dalam mendeteksi kasus positif (*recall*) maupun ketepatannya (*precision*). Visualisasi ini penting untuk menilai kinerja model secara keseluruhan.



Gambar 4.11 Diagram *F1-Score* Pada Skenario 1

Berdasarkan Gambar tersebut, dapat dilihat bahwa percobaan ke-4 menghasilkan F1-Score tertinggi, yaitu 80.32%, yang menunjukkan bahwa model dapat mencapai keseimbangan optimal antara *precision* dan *recall*. Dengan demikian, percobaan ke-4 menunjukkan performa terbaik dalam menjaga keseimbangan deteksi positif yang tepat dan mengurangi kesalahan klasifikasi. Secara keseluruhan, *F1-Score* memberikan gambaran yang lebih komprehensif terkait kualitas model yang dikembangkan.

Gambar 4.12, ditampilkan *confusion matrix* yang menggambarkan distribusi prediksi model berdasarkan konfigurasi terbaik menggunakan kernel RBF dengan *hyperparameter* C = 1 dan γ = 0.1. *Confusion matrix* ini memberikan gambaran yang jelas mengenai jumlah prediksi yang benar dan salah pada setiap kelas, serta membantu dalam menganalisis kinerja model berdasarkan kesalahan klasifikasi yang terjadi.



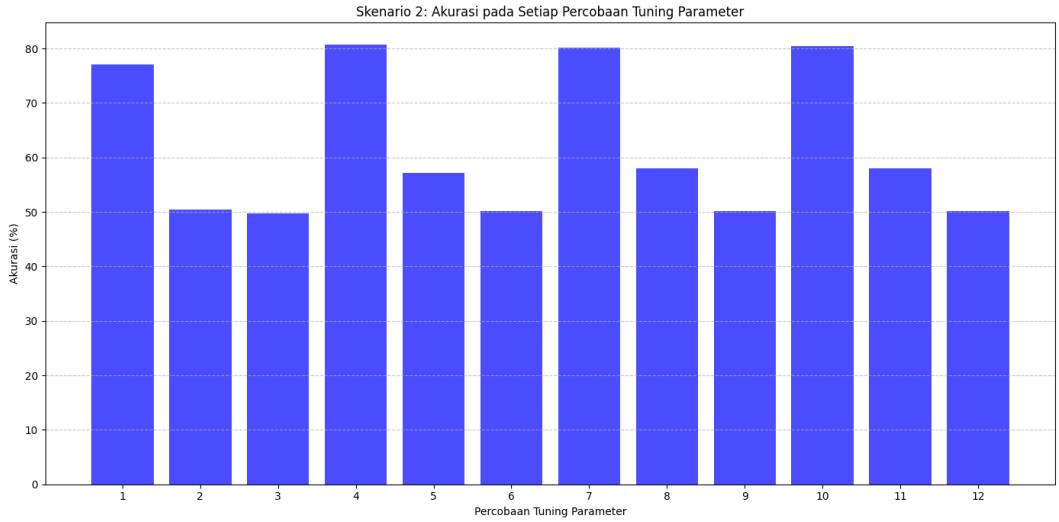
Gambar 4.12 *Confusion Matrix* Kernel RBF C = 1 dan γ= 0.1

Gambar tersebut menunjukkan bahwa model dengan konfigurasi C = 1 dan γ = 0.1 berhasil memprediksi dengan benar 394 data pada kelas 0 dan 365 data pada kelas 1. Namun, terdapat kesalahan klasifikasi, yaitu 75 data kelas 0 yang diprediksi sebagai kelas 1 (FP) dan 104 data kelas 1 yang diprediksi sebagai kelas 0 (FN). Secara keseluruhan, *confusion matrix* ini menunjukkan bahwa model memiliki akurasi yang baik, meskipun masih ada ruang untuk perbaikan dalam hal mengurangi kesalahan prediksi.

4.6.2 Analisis Eksperimen Skenario 2

Pada Gambar 4.13, ditampilkan grafik yang menggambarkan hasil akurasi model pada eksperimen skenario 2 untuk setiap percobaan *tuning* parameter. Eksperimen ini bertujuan untuk mengevaluasi performa model dengan variasi konfigurasi *hyperparameter* C dan γ. Grafik ini memberikan informasi terkait

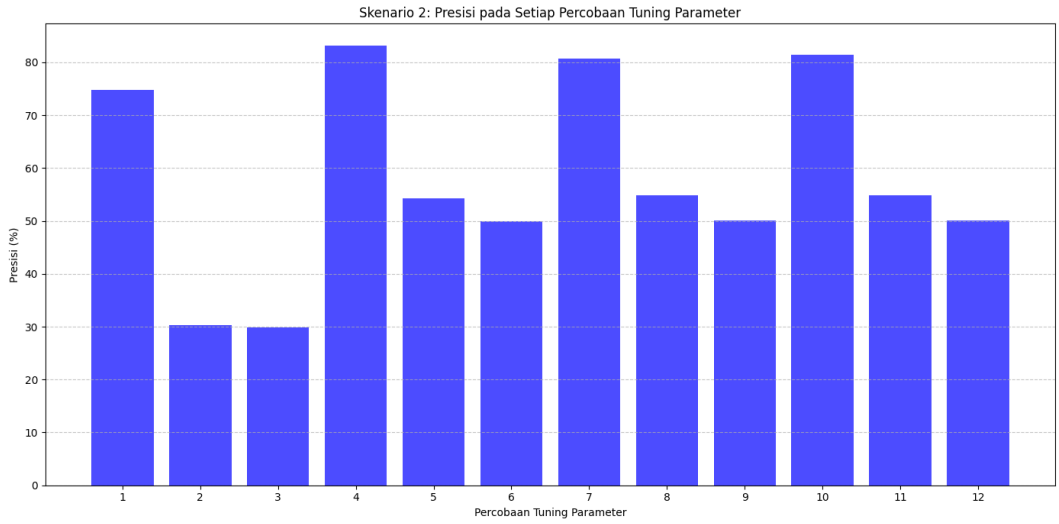
bagaimana perubahan parameter mempengaruhi akurasi model, yang membantu dalam pemilihan konfigurasi terbaik untuk model SVM dengan kernel RBF.



Gambar 4.13 Diagram Akurasi Pada Skenario 2

Dari gambar tersebut, terlihat bahwa akurasi model mengalami variasi pada setiap percobaan. Nilai tertinggi yang dicapai adalah 80.70% pada konfigurasi *hyperparameter* C = 1 dan γ = 0.1. Hal ini menunjukkan bahwa kombinasi nilai parameter tersebut memberikan performa terbaik di antara konfigurasi lainnya. Grafik ini membantu dalam menentukan parameter yang paling efektif untuk meningkatkan akurasi model pada skenario 2.

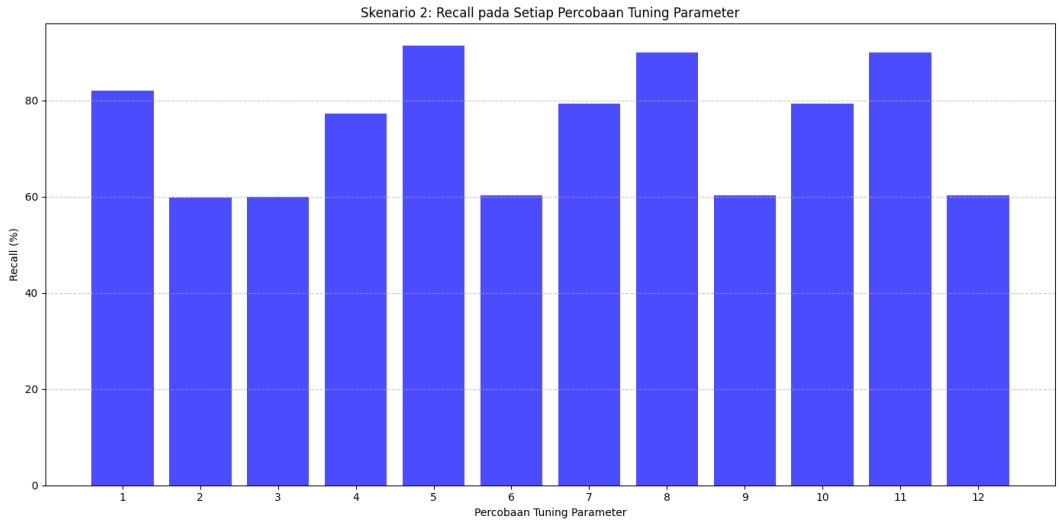
Gambar 4.14, ditampilkan grafik yang menggambarkan nilai *precision* pada setiap percobaan *tuning* parameter dalam eksperimen skenario 2. *Precision* adalah metrik yang menunjukkan seberapa akurat model dalam mengklasifikasikan data positif, atau dengan kata lain, seberapa banyak prediksi positif yang benar. Grafik ini memberikan gambaran mengenai seberapa andal model dalam mendeteksi kasus positif di setiap percobaan dengan variasi parameter yang diuji.



Gambar 4.14 Diagram *Precision* Pada Skenario 2

Berdasarkan Gambar tersebut, dapat dilihat bahwa percobaan ke-4 menghasilkan nilai *precision* tertinggi, yaitu 83.13%, yang menunjukkan bahwa model cukup andal dalam mengklasifikasikan pasien yang benar-benar terinfeksi HIV. Meskipun demikian, terdapat fluktuasi signifikan pada beberapa percobaan lainnya, yang mencerminkan bahwa *precision* model cukup sensitif terhadap kombinasi parameter yang digunakan. Hal ini mengindikasikan perlunya pengoptimalan lebih lanjut pada pemilihan parameter untuk mendapatkan performa yang lebih konsisten.

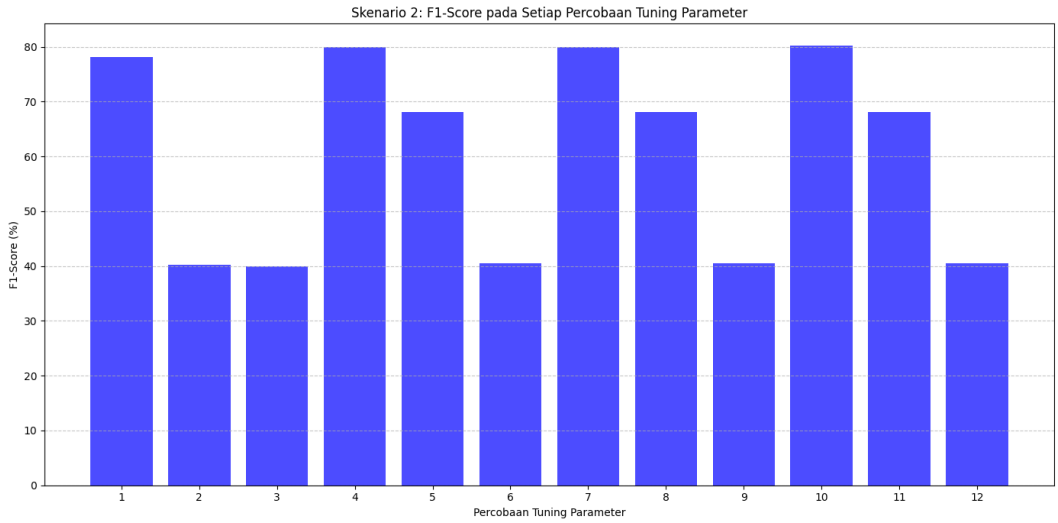
Pada Gambar 4.15, ditampilkan grafik yang menggambarkan nilai *recall* pada setiap percobaan *tuning* parameter dalam eksperimen skenario 2. *Recall* adalah metrik yang mengukur kemampuan model dalam mendeteksi semua kasus positif dengan benar. Grafik ini memberikan gambaran mengenai bagaimana model bekerja dalam mengidentifikasi kasus positif di setiap percobaan, dengan variasi parameter yang diuji.



Gambar 4.15 Diagram *Recall* Pada Skenario 2

Gambar tersebut menunjukkan bahwa percobaan ke-5 menghasilkan nilai *recall* tertinggi, yaitu 91.38%. Hal ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mendeteksi seluruh kasus positif. Namun, meskipun nilai *recall* tinggi, perlu diperhatikan bahwa dalam beberapa kasus, nilai *recall* yang sangat tinggi ini dapat berpotensi mengorbankan *precision*, yang mengindikasikan adanya risiko *overfitting* terhadap kelas positif. Oleh karena itu, perlu dilakukan penyesuaian lebih lanjut untuk mencapai keseimbangan yang optimal antara *precision* dan *recall*.

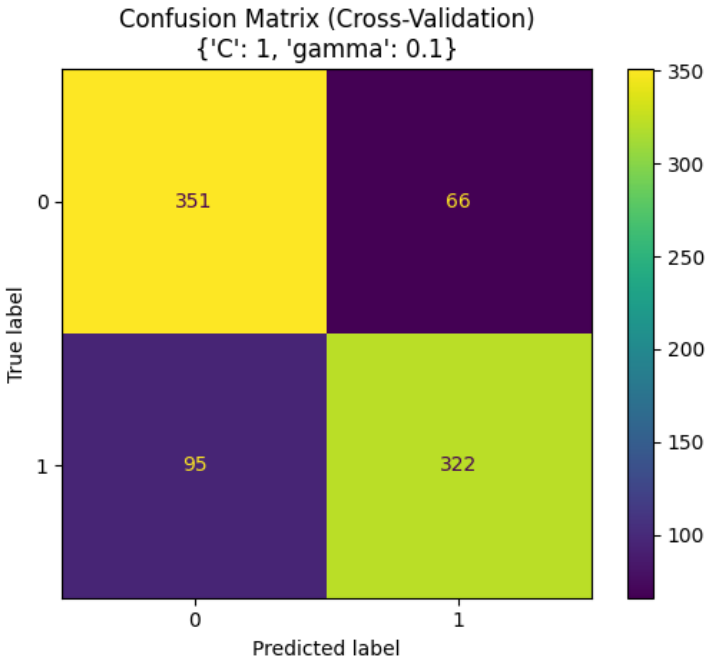
Gambar 4.16, ditampilkan grafik yang menggambarkan nilai *F1-Score* pada setiap percobaan *tuning* parameter dalam eksperimen skenario 2. *F1-Score* adalah metrik yang menggabungkan *precision* dan *recall* untuk memberikan gambaran yang lebih seimbang mengenai kinerja model. Grafik ini menunjukkan bagaimana variasi konfigurasi *hyperparameter* C dan γ mempengaruhi keseimbangan antara mendeteksi kasus positif dan menghindari kesalahan klasifikasi pada setiap percobaan.



Gambar 4.16 Diagram *F1-Score* Pada Skenario 2

Dari gambar tersebut, terlihat bahwa percobaan ke-4 menghasilkan nilai *F1-Score* tertinggi, yaitu 79.97%. Hal ini menunjukkan bahwa model dapat menjaga keseimbangan yang baik antara *precision* dan *recall*, sehingga meminimalkan kesalahan klasifikasi. Dengan *F1-Score* yang relatif tinggi pada percobaan ini, dapat disimpulkan bahwa konfigurasi parameter yang digunakan pada percobaan ke-4 memberikan performa yang paling optimal dan stabil di antara seluruh kombinasi yang diuji.

Pada Gambar 4.17, ditampilkan *confusion matrix* yang menggambarkan distribusi prediksi model berdasarkan konfigurasi terbaik menggunakan kernel RBF dengan *hyperparameter* C = 1 dan γ = 0.1. *Confusion matrix* ini memberikan gambaran yang jelas mengenai jumlah prediksi yang benar dan salah pada setiap kelas. Ini sangat berguna untuk memahami bagaimana model mengklasifikasikan data dalam kedua kelas yang ada, yaitu kelas 0 dan kelas 1.



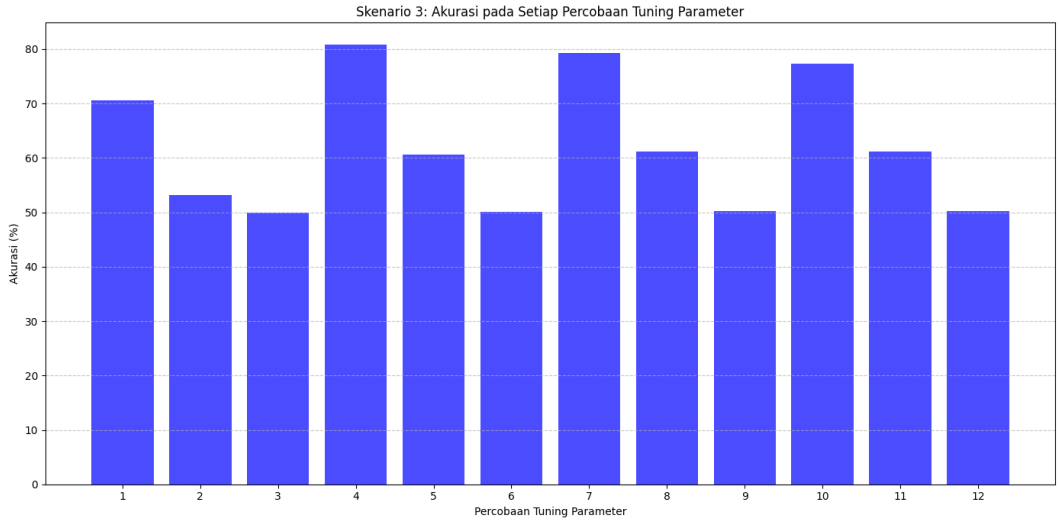
Gambar 4.17 *Confusion Matrix* Kernel RBF C = 1 dan γ= 0.1

Berdasarkan Gambar tersebut, dapat dilihat bahwa model dengan kernel RBF dan konfigurasi $C = 1$ dan $\gamma = 0.1$ berhasil memprediksi dengan benar 351 data pada kelas 0 dan 322 data pada kelas 1. Namun, terdapat kesalahan klasifikasi, yaitu 66 data kelas 0 yang diprediksi sebagai kelas 1 (FP) dan 95 data kelas 1 yang diprediksi sebagai kelas 0 (FN). Secara keseluruhan, *confusion matrix* ini menunjukkan bahwa model memiliki akurasi yang baik, meskipun ada kesalahan prediksi yang perlu diperhatikan untuk meningkatkan kinerja model di masa depan.

4.6.3 Analisis Eksperimen Skenario 3

Gambar 4.18, ditampilkan grafik yang menggambarkan nilai akurasi pada eksperimen skenario 3 untuk setiap percobaan *tuning* parameter. Eksperimen ini bertujuan untuk mengevaluasi kinerja model dengan berbagai konfigurasi *hyperparameter* C dan γ. Grafik ini menunjukkan bagaimana variasi parameter

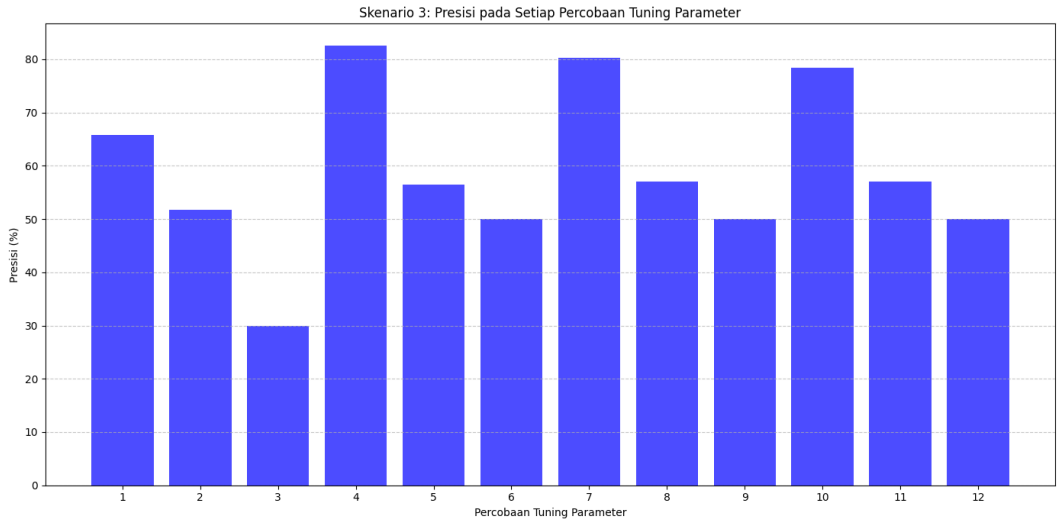
mempengaruhi akurasi model SVM dengan kernel RBF, memberikan gambaran terkait performa model di setiap percobaan



Gambar 4.18 Diagram Akurasi Pada Skenario 3

Gambar tersebut menunjukkan bahwa akurasi model mengalami variasi pada setiap percobaan. Nilai tertinggi yang tercatat adalah 80.82% pada percobaan dengan konfigurasi hyperparameter C = 1 dan γ = 0.1. Hal ini menunjukkan bahwa kombinasi parameter tersebut memberikan performa terbaik di antara seluruh percobaan, dengan akurasi yang cukup tinggi untuk klasifikasi data pada skenario 3.

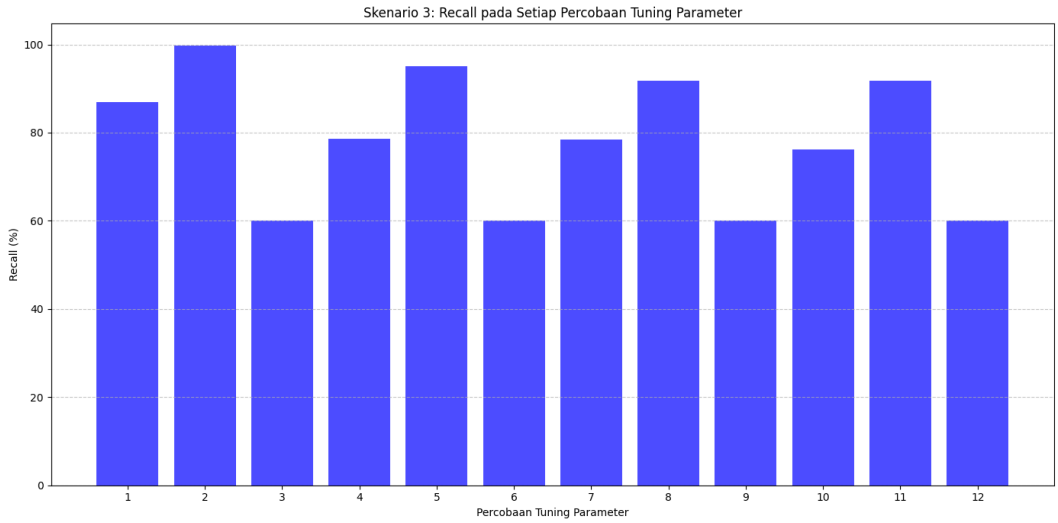
Pada Gambar 4.19, ditampilkan grafik yang menggambarkan nilai *precision* pada setiap percobaan *tuning* parameter dalam eksperimen skenario 3. *Precision* adalah metrik yang menunjukkan seberapa banyak prediksi positif yang benar dibandingkan dengan semua prediksi positif yang dilakukan oleh model. Grafik ini memberikan informasi terkait seberapa andal model dalam mengklasifikasikan data positif dengan tepat pada berbagai konfigurasi parameter yang diuji.



Gambar 4.19 Diagram *Precision* Pada Skenario 3

Dari Gambar tersebut, terlihat bahwa percobaan ke-4 menghasilkan nilai *precision* tertinggi, yaitu 82.54%. Hal ini menunjukkan bahwa model cukup andal dalam mengklasifikasikan pasien yang benar-benar terinfeksi HIV. Namun, terdapat fluktuasi signifikan pada beberapa percobaan lainnya, yang mencerminkan bahwa *precision* model cukup sensitif terhadap variasi kombinasi parameter yang digunakan. Ini menunjukkan bahwa model perlu dioptimalkan lebih lanjut untuk memperoleh konsistensi yang lebih baik pada pengklasifikasian data positif.

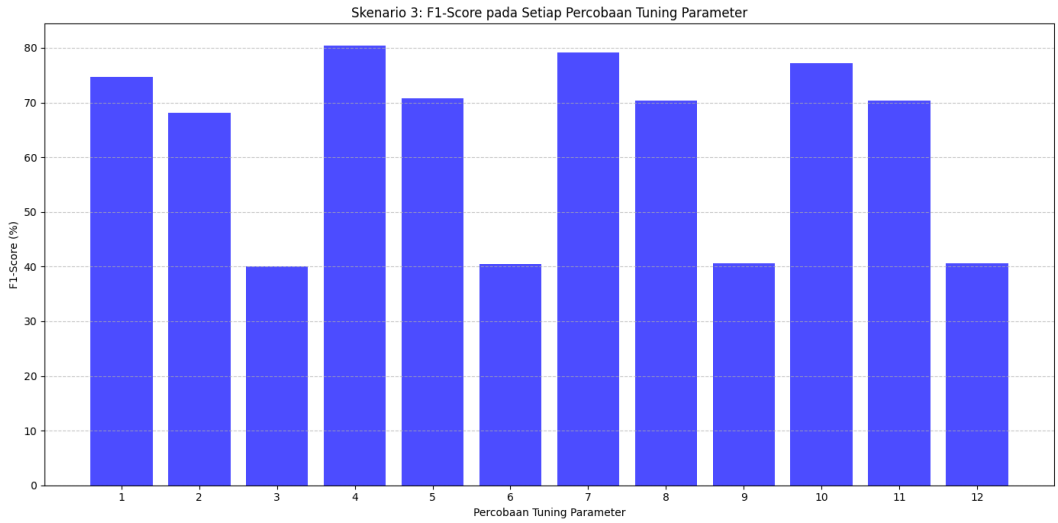
Gambar 4.20, ditampilkan grafik yang menggambarkan nilai *recall* pada setiap percobaan *tuning* parameter dalam eksperimen skenario 3. *Recall* adalah metrik yang mengukur kemampuan model dalam mendeteksi semua kasus positif dengan benar. Grafik ini memberikan gambaran mengenai bagaimana model berhasil mengidentifikasi kasus positif dalam setiap percobaan dengan berbagai konfigurasi *hyperparameter* yang diuji.



Gambar 4.20 Diagram *Recall* Pada Skenario 3

Berdasarkan Gambar tersebut, dapat dilihat bahwa percobaan ke-2 menghasilkan nilai *recall* tertinggi, yaitu 99.73%. Hal ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mendeteksi seluruh kasus positif. Namun, meskipun nilai *recall* sangat tinggi, hal ini perlu diperhatikan lebih lanjut karena dapat mengindikasikan kemungkinan *overfitting* terhadap kelas positif, yang dapat menyebabkan penurunan nilai *precision* pada beberapa percobaan lainnya.

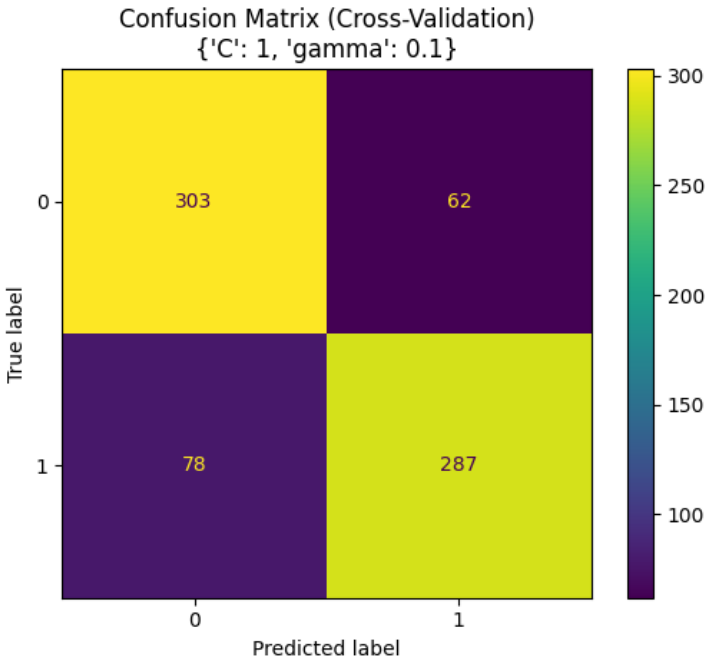
Pada Gambar 4.21, ditampilkan grafik yang menggambarkan nilai *F1-Score* pada setiap percobaan *tuning* parameter dalam eksperimen skenario 3. *F1-Score* adalah metrik yang menggabungkan *precision* dan *recall*, memberikan gambaran yang lebih seimbang mengenai kemampuan model dalam mengklasifikasikan data positif. Grafik ini menunjukkan bagaimana variasi konfigurasi parameter C dan γ mempengaruhi kinerja model secara keseluruhan dalam menjaga keseimbangan antara deteksi yang tepat dan menghindari kesalahan klasifikasi.



Gambar 4.21 Diagram *F1-Score* Pada Skenario 3

Gambar tersebut menunjukkan bahwa percobaan ke-4 menghasilkan nilai *F1-Score* tertinggi, yaitu 80.39%, yang menunjukkan bahwa model dapat menjaga keseimbangan yang baik antara *precision* dan *recall*, sehingga meminimalkan kesalahan klasifikasi. Dengan performa yang stabil pada percobaan ke-4, dapat disimpulkan bahwa konfigurasi parameter $C = 1$ dan $\gamma = 0.1$ memberikan hasil yang optimal di antara seluruh kombinasi yang diuji dalam eksperimen ini.

Gambar 4.22, ditampilkan *confusion matrix* yang menggambarkan distribusi prediksi model berdasarkan konfigurasi terbaik menggunakan kernel RBF dengan *hyperparameter* $C = 1$ dan $\gamma = 0.1$. *Confusion matrix* ini memberikan gambaran yang jelas mengenai jumlah prediksi yang benar dan salah pada setiap kelas. Hal ini sangat penting untuk menganalisis bagaimana model mengklasifikasikan data dalam kedua kelas yang ada dan seberapa baik model mampu membedakan antara kelas 0 dan kelas 1.



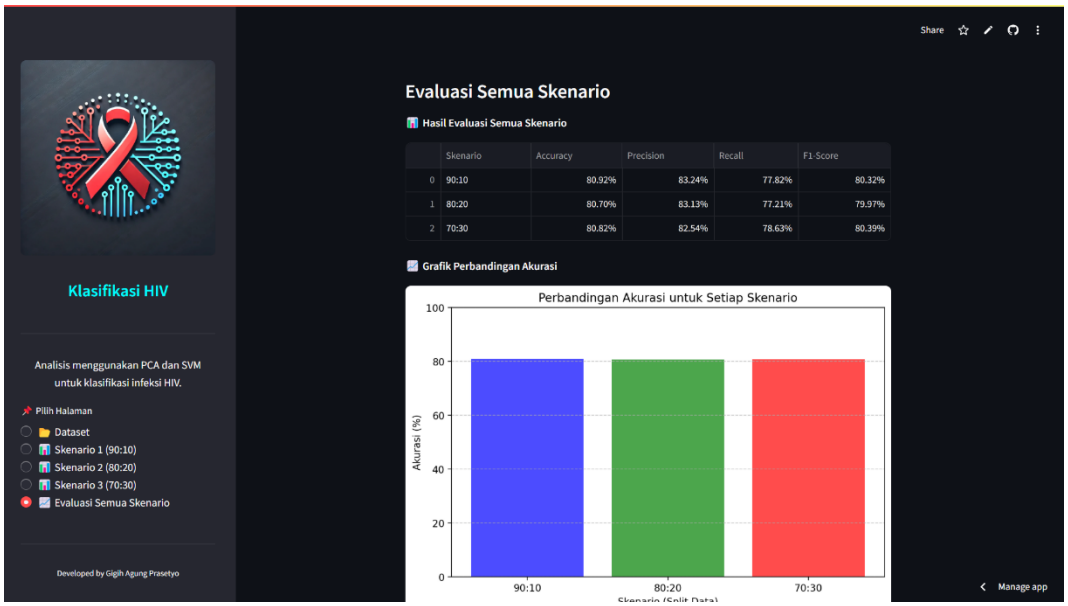
Gambar 4.22 *Confusion Matrix* Kernel RBF C = 1 dan γ= 0.1

Dari Gambar tersebut, terlihat bahwa model dengan konfigurasi kernel RBF $C = 1$ dan $\gamma = 0.1$ berhasil memprediksi dengan benar 303 data pada kelas 0 dan 287 data pada kelas 1. Namun, terdapat kesalahan klasifikasi, yaitu 62 data kelas 0 yang salah diprediksi sebagai kelas 1 (FP) dan 78 data kelas 1 yang salah diprediksi sebagai kelas 0 (FN). Secara keseluruhan, *confusion matrix* ini menunjukkan bahwa model memiliki akurasi yang baik, dengan performa metrik yang cukup stabil meskipun ada kesalahan prediksi yang perlu diperbaiki.

4.6.4 Rata-Rata Evaluasi Skenario

Pada Gambar 4.23 menunjukkan tampilan yang menyajikan hasil evaluasi dari seluruh skenario yang telah diuji. Tampilan ini mencakup metrik akurasi, *precision*, *recall*, dan *F1-score* yang dihitung berdasarkan berbagai rasio *split data*, yaitu 90:10, 80:20, dan 70:30. Grafik yang ditampilkan juga memberikan gambaran

perbandingan akurasi antar skenario untuk memudahkan pemahaman terhadap performa model pada proporsi data yang berbeda.



Gambar 4.23 Tampilan Evaluasi Semua Skenario

Berdasarkan Gambar tersebut, dapat dilihat bahwa evaluasi yang ditampilkan memberikan *insight* yang jelas mengenai bagaimana model bereaksi terhadap berbagai rasio *data training* dan *testing*. Grafik perbandingan akurasi antar skenario menjadi visualisasi yang penting untuk memahami performa model secara keseluruhan. Hasil ini juga berfungsi sebagai referensi dalam memilih skenario terbaik yang menghasilkan performa optimal, dan dapat menjadi dasar dalam implementasi model lebih lanjut.

Tabel 4.27 menyajikan hasil evaluasi rata-rata dari beberapa skenario eksperimen yang telah diuji. Melalui tabel ini, kita dapat melihat metrik akurasi, *precision*, *recall*, dan *F1-Score* pada setiap skenario yang diujikan, memberikan gambaran tentang performa model dengan berbagai rasio *split data*. Data ini sangat berguna untuk menilai seberapa baik model dalam mengklasifikasikan data

berdasarkan pengaturan parameter yang berbeda, serta untuk memahami kekuatan dan kelemahan model dalam setiap percobaan.

Tabel 4.27 Rata-Rata Evaluasi Per Skenario

	Rata-rata evaluasi *confusion matrix*			
	Akurasi	***Precision***	***Recall***	***F1-Score***
Skenario 1	80.92%	83.24%	77.82%	80.32%
Skenario 2	80.70%	83.13%	77.21%	79.97%
Skenario 3	80.82%	82.54%	78.63%	80.39%

Tabel tersebut menunjukkan, bahwa skenario 1 dengan rasio 90:10 menghasilkan akurasi tertinggi sebesar 80,92%. Hal ini menunjukkan bahwa model memiliki kinerja yang optimal pada skenario tersebut, didukung oleh nilai *precision* yang tinggi, yaitu 83,24%, serta *recall* sebesar 77,82% dan *F1-Score* sebesar 80,32%. Hasil ini menandakan bahwa model berhasil menjaga keseimbangan antara mendeteksi kelas positif dan menghindari kesalahan klasifikasi, yang menjadikannya konfigurasi terbaik di antara seluruh skenario yang diuji. Nilai ini dapat dianggap baik berdasarkan beberapa faktor:

A. Standar Akurasi dalam Penelitian *Machine Learning*

Dalam berbagai penelitian terkait klasifikasi dengan SVM, nilai akurasi yang diperoleh umumnya bervariasi tergantung pada kompleksitas data. Sebagai contoh, penelitian terkait klasifikasi status gizi balita menggunakan SVM mencapai akurasi 91,91% pada dataset sederhana (Septyanto, 2023), Sementara itu, penelitian lain mengenai analisis sentimen terhadap kampanye pengurangan plastik di media sosial menggunakan SVM memperoleh akurasi 70,64% (Boro et al., 2025). Dengan demikian, akurasi 80,92% pada penelitian

ini berada dalam rentang yang cukup baik dan dapat diterima untuk klasifikasi berbasis data medis yang kompleks.

B. Konteks dan Kompleksitas Data

Akurasi model sangat bergantung pada kompleksitas data. Dalam klasifikasi infeksi HIV, dataset yang digunakan memiliki banyak variabel dengan tingkat variasi yang tinggi. Oleh karena itu, mendapatkan akurasi di atas 80% menunjukkan bahwa model berhasil menangkap pola dalam data dengan cukup baik. Selain itu, penerapan PCA dalam penelitian ini membantu mereduksi dimensi data tanpa menghilangkan informasi penting, sehingga meningkatkan efisiensi model.

C. Keseimbangan Antara Akurasi dan Metrik Lain

Selain akurasi, evaluasi juga mempertimbangkan metrik *Precision* (83,24%), *Recall* (77,82%), dan *F1-Score* (80,32%). Nilai *Precision* yang tinggi menunjukkan bahwa model jarang memberikan prediksi positif yang salah, sedangkan *Recall* yang cukup baik menunjukkan model masih dapat menangkap sebagian besar kasus positif. Dengan keseimbangan antara metrik-metrik ini, model pada skenario 1 dianggap memiliki performa yang lebih stabil.

Berdasarkan analisis di atas, dapat disimpulkan bahwa akurasi 80,92% pada skenario 1 menunjukkan performa yang baik, terutama jika dibandingkan dengan studi lain dalam klasifikasi data medis. Selain itu, proporsi *data training* yang lebih besar dalam skenario ini membantu model mempelajari pola dengan lebih optimal,

sehingga menghasilkan prediksi yang lebih akurat. Hal ini juga mengindikasikan bahwa model dapat lebih baik menangani variasi dalam data, meningkatkan kemampuan generalisasinya pada data yang belum terlihat sebelumnya.

4.6.5 Evaluasi Nilai *Cost*

Salah satu faktor yang memengaruhi performa model pada ketiga skenario adalah pengaturan parameter *Cost* (C) dalam metode SVM. Parameter ini mengatur keseimbangan antara kesalahan klasifikasi pada *data training* dan kompleksitas model. Ketika nilai C meningkat, margin *hyperplane* menjadi lebih sempit, membuat model lebih ketat dalam memisahkan kelas pada *data training*. Hal ini dapat meningkatkan akurasi pada *data training*, namun berisiko menyebabkan *overfitting*, di mana model kesulitan dalam generalisasi data baru. Sebaliknya, nilai C yang lebih kecil memperlebar margin, memberi fleksibilitas pada model, tetapi mengurangi akurasi pada data training. Pemilihan nilai C yang tepat memungkinkan model mencapai keseimbangan antara akurasi dan generalisasi, yang tercermin pada hasil eksperimen. Dalam eksperimen ini, nilai C yang berbeda diuji pada tiga skenario *split data*: skenario 1 (90:10), skenario 2 (80:20), dan skenario 3 (70:30). Pemilihan nilai C yang optimal bertujuan menyeimbangkan kesalahan pada *data training* dan *testing* serta kemampuan generalisasi model.

A. Skenario 1: *Split Data* 90:10

Pada Tabel 4.28 menyajikan hasil evaluasi performa model dengan kernel RBF pada berbagai nilai C dalam skenario 1, yang menggunakan rasio data 90% untuk *training* dan 10% untuk *testing*. Tabel ini menunjukkan metrik akurasi,

precision, *recall*, dan *F1-score* pada nilai C yang berbeda, yaitu 0.1, 1, 10, dan 100, untuk memberikan gambaran tentang bagaimana variasi nilai C memengaruhi kinerja model dalam mengklasifikasikan data.

Tabel 4.28 Evaluasi Nilai *Cost* Pada Skenario 1

Cost	**Kernel RBF**			
	Akurasi	***Precision***	***Recall***	***F1-Score***
0.1	59.56%	51.47%	81.34%	61.95%
1	63.90%	59.90%	83.95%	68.07%
10	63.47%	58.91%	83.24%	67.50%
100	63.33%	58.68%	83.24%	67.39%

Dari Tabel tersebut, dapat dilihat bahwa nilai C = 0.1 menghasilkan akurasi 59.56%, dengan *precision* 51.47%, *recall* 81.34%, dan *F1-score* 61.95%. Ketika nilai C dinaikkan menjadi 1, performa model meningkat dengan akurasi 63.90%, *precision* 59.90%, *recall* 83.95%, dan *F1-score* 68.07%. Peningkatan ini menunjukkan bahwa model menjadi lebih ketat dalam memisahkan kelas, dengan *recall* yang tetap tinggi dan *precision* yang meningkat. Namun, pada nilai C = 10 dan C = 100, meskipun performa model tetap stabil, tidak ada peningkatan signifikan, yang menunjukkan bahwa model sudah mencapai kestabilan performa pada nilai C yang lebih tinggi.

B. Skenario 2: *Split Data* 80:20

Tabel 4.29 menyajikan hasil evaluasi performa model dengan kernel RBF pada berbagai nilai C dalam skenario 2, yang menggunakan rasio data 80% untuk *training* dan 20% untuk *testing*. Tabel ini menunjukkan metrik akurasi, *precision*, *recall*, dan *F1-score* pada nilai C yang berbeda, yaitu 0.1, 1, 10, dan 100, memberikan gambaran tentang bagaimana variasi nilai C memengaruhi kinerja model dalam mengklasifikasikan data.

Tabel 4.29 Evaluasi Nilai *Cost* Pada Skenario 2

Cost	**Kernel RBF**			
	Akurasi	***Precision***	***Recall***	***F1-Score***
0.1	59.11%	44.97%	67.26%	52.74%
1	62.67%	62.47%	76.28%	62.84%
10	62.75%	61.88%	76.51%	62.87%
100	62.87%	62.10%	76.51%	62.96%

Berdasarkan Tabel tersebut, terlihat bahwa pada nilai C = 0.1, model memiliki akurasi 59.11%, dengan *precision* 44.97%, *recall* 67.26%, dan *F1-score* 52.74%. Meskipun *recall* lebih tinggi, precision yang rendah menunjukkan adanya kesalahan klasifikasi. Ketika nilai C dinaikkan menjadi 1, performa model meningkat dengan akurasi 62.67%, *precision* 62.47%, *recall* 76.28%, dan *F1-score* 62.84%. Pada nilai C = 10, performa model stabil dengan akurasi 62.75%, *precision* 61.88%, *recall* 76.51%, dan *F1-score* 62.87%. Pada nilai C = 100, meskipun terjadi sedikit penurunan pada akurasi 62.87%, *precision* 62.10%, dan *recall* 76.51%, *F1-score* sedikit meningkat menjadi 62.96%, menunjukkan bahwa model tetap stabil pada nilai C yang lebih tinggi.

C. Skenario 3: *Split Data* 70:30

Pada Tabel 4.30 menyajikan hasil evaluasi performa model dengan kernel RBF pada berbagai nilai C dalam skenario 3, yang menggunakan rasio data 70% untuk *training* dan 30% untuk *testing*. Tabel ini menunjukkan metrik akurasi, *precision*, *recall*, dan *F1-score* pada nilai C yang berbeda, yaitu 0.1, 1, 10, dan 100, memberikan gambaran bagaimana variasi nilai C memengaruhi kinerja model dalam mengklasifikasikan data.

Tabel 4.30 Evaluasi Nilai *Cost* Pada Skenario 3

Cost	**Kernel RBF**			
	Akurasi	***Precision***	***Recall***	***F1-Score***
0.1	57.90%	49.12%	82.19%	60.93%
1	63.88%	63.00%	77.90%	63.90%
10	63.61%	62.47%	76.71%	63.34%
100	62.97%	61.83%	75.98%	62.67%

Tabel tersebut menunjukkan bahwa pada nilai C = 0.1, model memiliki akurasi 57.90%, dengan *precision* 49.12%, *recall* 82.19%, dan *F1-score* 60.93%. Meskipun *recall* cukup tinggi, *precision* yang rendah menunjukkan adanya kesalahan klasifikasi pada kelas tertentu. Ketika nilai C dinaikkan menjadi 1, performa model meningkat dengan akurasi 63.88%, *precision* 63.00%, *recall* 77.90%, dan *F1-score* 63.90%, menunjukkan peningkatan pada *precision* dan *recall*. Pada nilai C = 10, performa model stabil dengan akurasi 63.61%, *precision* 62.47%, *recall* 76.71%, dan *F1-score* 63.34%, meskipun terjadi sedikit penurunan pada *recall*. Pada nilai C = 100, meskipun terdapat penurunan kecil pada akurasi 62.97% dan *precision* 61.83%, *F1-score* sedikit meningkat menjadi 62.67%, menunjukkan bahwa model tetap stabil pada nilai C yang lebih tinggi.

Berdasarkan hasil evaluasi pada ketiga skenario, dapat disimpulkan bahwa nilai C yang lebih rendah cenderung meningkatkan *recall*, namun dengan mengorbankan *precision* dan akurasi. Sebaliknya, dengan meningkatkan nilai C, model menjadi lebih ketat dalam memisahkan kelas, yang meningkatkan akurasi dan *precision*, tetapi berisiko menyebabkan penurunan kemampuan generalisasi (*overfitting*). Pada setiap skenario, nilai C = 1 hingga C = 10 memberikan keseimbangan yang relatif baik antara akurasi, *precision*, *recall*, dan *F1-score*, meskipun ada variasi kecil tergantung pada proporsi pembagian data. Model dengan

nilai C yang lebih tinggi (misalnya C = 100) cenderung lebih stabil dalam hal performa, tetapi ada indikasi potensi *overfitting*, terutama pada skenario dengan proporsi data lebih kecil. Oleh karena itu, pemilihan nilai C yang optimal sangat bergantung pada tujuan model, apakah lebih memprioritaskan akurasi atau kemampuan generalisasi.

4.6.6 Evaluasi Nilai *Gamma*

Selain *hyperparameter* (C), *hyperparameter Gamma* (γ) juga memainkan peran penting dalam kernel RBF. *Gamma* menentukan seberapa jauh pengaruh setiap titik data dalam membentuk *hyperplane*. Semakin besar nilai *Gamma*, semakin sempit radius fungsi RBF yang digunakan untuk menghitung kesamaan antara pasangan data, sehingga model menjadi lebih sensitif terhadap *data training*. Sensitivitas yang tinggi ini dapat membuat model lebih kompleks dan berisiko mengalami *overfitting*. Sebaliknya, jika *Gamma* terlalu rendah, radius fungsi RBF menjadi terlalu lebar, menyebabkan model kehilangan kompleksitas dan cenderung mengalami *underfitting*. Oleh karena itu, pemilihan nilai *Gamma* yang tepat sangat bergantung pada karakteristik data dan tujuan dari model. Dalam eksperimen ini, evaluasi performa model dilakukan dengan menggunakan berbagai nilai *Gamma*, Berdasarkan hasil eksperimen, berikut adalah evaluasi nilai *Gamma* pada masing-masing skenario.

A. Skenario 1: *Split Data* 90:10

Tabel 4.31 menyajikan hasil evaluasi performa model dengan kernel RBF pada berbagai nilai γ dalam skenario 1, yang menggunakan rasio data 90% untuk

training dan 10% untuk *testing*. Tabel ini menampilkan metrik akurasi, *precision*, *recall*, dan *F1-score* pada nilai γ yang berbeda, yaitu 0.1, 1, dan 10, memberikan gambaran tentang bagaimana variasi nilai γ mempengaruhi kinerja model dalam mengklasifikasikan data.

Tabel 4.31 Evaluasi Nilai *Gamma* Pada Skenario 1

Gamma	**Kernel RBF**			
	Akurasi	***Precision***	***Recall***	***F1-Score***
0.1	78.84%	78.88%	79.54%	79.02%
1	58.02%	52.41%	89.40%	65.98%
10	50.82%	40.42%	79.89%	53.68%

Dari Tabel tersebut, terlihat bahwa pada nilai γ = 0.1, model menghasilkan akurasi 78.84%, dengan *precision* 78.88%, *recall* 79.54%, dan *F1-score* 79.02%. Hal ini menunjukkan bahwa model dengan γ yang rendah memiliki keseimbangan yang baik antara akurasi, *precision*, *recall*, dan *F1-score*, memberikan performa yang optimal. Namun, pada nilai γ = 1, performa model mengalami penurunan dengan akurasi 58.02%, *precision* 52.41%, *recall* 89.40%, dan *F1-score* 65.98%, menunjukkan bahwa model menjadi lebih sensitif terhadap *data training*. Pada nilai γ = 10, performa model semakin menurun dengan akurasi 50.82%, *precision* 40.42%, *recall* 79.89%, dan *F1-score* 53.68%, yang mengindikasikan *overfitting* pada *data training* dan kesulitan dalam melakukan generalisasi pada *data testing*.

B. Skenario 2: *Split Data* 80:20

Pada Tabel 4.32 menyajikan hasil evaluasi performa model dengan kernel RBF pada berbagai nilai parameter γ dalam skenario 2, yang menggunakan rasio data 80% untuk *training* dan 20% untuk *testing*. Tabel ini menunjukkan metrik akurasi, *precision*, *recall*, dan *F1-score* pada nilai γ yang berbeda, yaitu 0.1, 1, dan

10, memberikan gambaran bagaimana variasi nilai γ memengaruhi kinerja model dalam mengklasifikasikan data.

Tabel 4.32 Evaluasi Nilai *Gamma* Pada Skenario 2

Gamma	**Kernel RBF**			
	Akurasi	***Precision***	***Recall***	***F1-Score***
0.1	79.59%	79.98%	79.49%	79.57%
1	55.94%	48.57%	82.75%	61.14%
10	50.03%	45.00%	60.18%	40.35%

Berdasarkan Tabel tersebut, dapat dilihat bahwa pada nilai γ = 0.1, model menghasilkan akurasi 79.59%, dengan *precision* 79.98%, *recall* 79.49%, dan *F1-score* 79.57%. Hal ini menunjukkan bahwa model dengan γ yang rendah dapat menjaga keseimbangan yang baik antara akurasi, *precision*, *recall*, dan *F1-score*. Namun, pada nilai γ = 1, performa model mengalami penurunan dengan akurasi 55.94%, *precision* 48.57%, *recall* 82.75%, dan *F1-score* 61.14%, menunjukkan bahwa meskipun *recall* tetap tinggi, *precision* menurun. Pada nilai γ = 10, performa model semakin menurun dengan akurasi 50.03%, *precision* 45.00%, *recall* 60.18%, dan *F1-score* 40.35%, yang mengindikasikan bahwa model menjadi terlalu sensitif terhadap *data training* dan kesulitan dalam melakukan generalisasi pada *data testing.*

C. Skenario 3: *Split Data* 70:30

Tabel 4.33 menyajikan hasil evaluasi performa model dengan kernel RBF pada berbagai nilai parameter γ dalam skenario 3, yang menggunakan rasio data 70% untuk *training* dan 30% untuk *testing*. Tabel ini menyajikan metrik akurasi, *precision*, *recall*, dan *F1-score* pada nilai γ yang berbeda, yaitu 0.1, 1, dan 10,

memberikan gambaran mengenai bagaimana variasi nilai γ memengaruhi kinerja model dalam mengklasifikasikan data.

Tabel 4.33 Evaluasi Nilai *Gamma* Pada Skenario 3

Gamma	**Kernel RBF**			
	Akurasi	***Precision***	***Recall***	***F1-Score***
0.1	77.02%	76.72%	80.00%	77.85%
1	59.08%	55.56%	94.59%	69.88%
10	50.17%	45.03%	60.00%	40.39%

Tabel tersebut menunjukkan bahwa pada nilai γ = 0.1, model menghasilkan akurasi 77.02%, dengan *precision* 76.72%, *recall* 80.00%, dan *F1-score* 77.85%. Hal ini menunjukkan bahwa dengan γ yang rendah, model memiliki keseimbangan yang baik antara akurasi, *precision*, *recall*, dan *F1-score*. Namun, pada nilai γ = 1, performa model mengalami penurunan dengan akurasi 59.08%, *precision* 55.56%, *recall* 94.59%, dan *F1-score* 69.88%, yang menunjukkan bahwa meskipun *recall* meningkat, *precision* dan akurasi menurun. Pada nilai γ = 10, performa model semakin menurun dengan akurasi 50.17%, *precision* 45.03%, *recall* 60.00%, dan *F1-score* 40.39%, yang mengindikasikan bahwa model menjadi terlalu sensitif terhadap *data training* dan kesulitan dalam generalisasi pada *data testing*.

Berdasarkan hasil evaluasi pada ketiga skenario, dapat disimpulkan bahwa nilai *Gamma* yang lebih rendah (*Gamma* = 0.1) memberikan keseimbangan yang terbaik antara akurasi, *precision*, *recall*, dan *F1-score*. Nilai *Gamma* yang terlalu tinggi (*Gamma* = 10) menunjukkan penurunan signifikan dalam performa model, baik dalam hal akurasi, *precision*, maupun *F1-score*. Meskipun *recall* pada Gamma yang lebih tinggi cenderung meningkat, hal ini diimbangi dengan penurunan pada *precision* dan akurasi, yang menunjukkan model cenderung mengalami *overfitting*. Oleh karena itu, untuk memperoleh performa optimal pada model dengan kernel

RBF, disarankan untuk menggunakan nilai *Gamma* yang rendah hingga sedang, bergantung pada proporsi *data training* dan *testing* yang digunakan.

4.7 Integrasi Islam

Penelitian ini tidak hanya bertujuan untuk meningkatkan keakuratan dalam klasifikasi infeksi HIV melalui penerapan metode PCA dan SVM, tetapi juga berupaya untuk mengintegrasikan nilai-nilai Islam dalam setiap tahapannya. Dalam Islam, ilmu pengetahuan dipandang sebagai salah satu karunia Allah Subhanahu wa ta'ala yang harus digunakan untuk kebaikan umat manusia. Seperti halnya dalam Al-Qur'an yang memuat ayat-ayat muhkamat, yang jelas dan menjadi pokok ajaran, serta mutasyabihat, yang memerlukan pemahaman mendalam, ilmu pengetahuan yang kita miliki pun harus disikapi dengan bijaksana. Dalam konteks ini, klasifikasi data HIV menggunakan metode PCA dan SVM dapat dipandang sebagai upaya untuk menyaring dan mengkategorikan informasi yang jelas (muhkamat) dan yang membutuhkan analisis lebih lanjut (mutasyabihat) agar dapat menghasilkan diagnosa yang akurat dan bermanfaat. Allah Subhanahu wa ta'ala berfirman dalam QS. Al-Imran ayat 7:

هُوَ الَّذِي أَنزَلَ عَلَيْكَ الْكِتَابَ مِنْهُ آيَاتٌ مُّحْكَمَاتٌ هُنَّ أُمُّ الْكِتَابِ وَأُخَرُ مُتَشَابِهَاتٌ فَأَمَّا الَّذِينَ فِي قُلُوبِهِمْ زَيْغٌ فَيَتَّبِعُونَ مَا تَشَابَهَ مِنْهُ ابْتِغَاءَ الْفِتْنَةِ وَابْتِغَاءَ تَأْوِيلِهِ وَمَا يَعْلَمُ تَأْوِيلَهُ إِلَّا اللَّهُ وَالرَّاسِخُونَ فِي الْعِلْمِ يَقُولُونَ آمَنَّا بِهِ كُلٌّ مِّنْ عِندِ رَبِّنَا وَمَا يَذَّكَّرُ إِلَّا أُوْلُواْ الْأَلْبَابِ

" Dia-lah yang menurunkan Kitab (Al-Qur'an) kepadamu. Di antara ayat-ayatnya ada yang muhkamāt (jelas), itu adalah pokok-pokok ajaran dalam Kitab itu, dan ada yang mutasyābihāt (samar-samar). Adapun orang-orang yang dalam hatinya condong kepada kesesatan, mereka mengikuti apa yang samar-samar (mutasyābih) daripadanya, untuk menimbulkan fitnah dan untuk mencari-cari takwilnya. Padahal tidak ada yang mengetahui takwilnya selain Allah. Dan orang-orang yang mendalam ilmunya berkata: 'Kami beriman kepadanya, semuanya itu dari sisi

Tuhan kami.' Dan tidak ada yang dapat mengambil pelajaran kecuali orang-orang yang berakal.'" (QS. Al-Imran: 7)

Menurut Tafsir Al-Qurthubi Jilid 4, ayat ini menunjukkan adanya dua jenis ayat dalam Al-Qur'an: muhkamat (yang tegas dan menjadi dasar hukum) dan mutasyabihat (yang samar maknanya dan membutuhkan pendalaman). Orang-orang yang mengikuti hawa nafsunya akan lebih suka mengikuti ayat-ayat yang samar dengan tujuan menimbulkan fitnah dan kekacauan pemahaman. Tafsir ini juga menekankan bahwa hanya Allah Subhanahu wa ta'ala yang mengetahui takwil (makna hakiki) dari ayat-ayat mutasyabihat, sementara ulama yang kokoh ilmunya tunduk kepada ketetapan Allah Subhanahu wa ta'ala dan tidak berani berspekulasi (Qurthubi, 2020).

Dengan demikian, penerapan teknologi modern ini tidak hanya berorientasi pada hasil yang efisien tetapi juga membawa nilai-nilai Islam yang menekankan pentingnya membantu orang lain dan mencari keridaan Allah Subhanahu wa ta'ala. Integrasi ilmu pengetahuan dan nilai-nilai Islam ini menjadi wujud nyata dari harmonisasi antara sains dan agama dalam kehidupan manusia. Selain itu, Islam juga mendorong manusia untuk terus berusaha mencari solusi atas setiap masalah, termasuk dalam bidang kesehatan. Sebagaimana sabda Rasulullah SAW:

مَا أَنْزَلَ اللَّهُ دَاءً إِلَّا أَنْزَلَ لَهُ شِفَاءً

"Tidaklah Allah menurunkan suatu penyakit kecuali Dia juga menurunkan penawarnya." (HR. Bukhari no. 5678).

Hadis ini memberikan motivasi bahwa setiap penyakit pasti memiliki solusinya, dan manusia diperintahkan untuk berusaha mencarinya (Mohamad

Ismail et al., 2021). Dalam penelitian ini, metode PCA dan SVM diterapkan untuk memberikan kontribusi nyata dalam pengelolaan data medis, sehingga hasil klasifikasi yang dihasilkan dapat membantu dokter atau tenaga medis dalam mengambil keputusan yang tepat. Dengan menggunakan metode ini, diharapkan proses diagnosa dapat dilakukan dengan lebih akurat, sehingga langkah penanganan medis dapat segera diambil untuk mencegah dampak yang lebih buruk. Hal ini sejalan dengan ajaran Islam yang mendorong setiap individu untuk menjaga kesehatan dan berikhtiar dalam mencari pengobatan yang baik dan halal. Keseluruhan proses dalam penelitian ini mencerminkan pentingnya usaha manusia dalam memanfaatkan ilmu pengetahuan untuk kemaslahatan umat. Dengan memadukan teknologi modern seperti PCA dan SVM dengan prinsip-prinsip Islam, penelitian ini diharapkan dapat memberikan manfaat yang luas bagi masyarakat, sekaligus menjadi bentuk pengamalan nilai-nilai Islam yang menekankan pentingnya keseimbangan antara usaha, doa, dan pengabdian kepada Allah Subhanahu wa ta’ala.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa model klasifikasi infeksi HIV dengan mengintegrasikan metode PCA dan SVM mampu memberikan performa klasifikasi yang baik dan konsisten pada berbagai rasio pembagian data. Proses PCA terbukti efektif dalam mereduksi dimensi data tanpa menghilangkan informasi penting, sehingga dapat meningkatkan efisiensi algoritma SVM.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa rekomendasi dan saran untuk penelitian selanjutnya dapat dipertimbangkan untuk meningkatkan kualitas dan aplikabilitas model klasifikasi yang dikembangkan. Saran-saran ini bertujuan untuk mengeksplorasi pendekatan yang lebih beragam dan memperluas cakupan penggunaan data serta algoritma yang lebih canggih dalam sistem klasifikasi. Berikut adalah beberapa saran yang dapat dijadikan referensi untuk penelitian lanjutan:

a) Penelitian selanjutnya disarankan untuk menggunakan teknik *balancing data* yang lebih variatif, seperti *SMOTE* atau *ADASYN*, agar performa model dapat dibandingkan dengan pendekatan *undersampling* yang digunakan pada penelitian ini.

b) Model klasifikasi dapat dikembangkan dengan membandingkan beberapa algoritma lain, seperti *Random Forest, XGBoost,* atau *Artificial Neural Networks* (ANN), untuk mengevaluasi apakah performa klasifikasi dapat ditingkatkan lebih lanjut.

c) Penelitian lanjutan juga dapat mempertimbangkan penggunaan data *real-time* atau data yang bersumber langsung dari rumah sakit atau instansi kesehatan, untuk meningkatkan relevansi dan aplikabilitas sistem klasifikasi di dunia nyata.

d) Bagi praktisi kesehatan atau pengembang sistem informasi medis, hasil penelitian ini dapat dijadikan acuan awal dalam merancang sistem pendukung keputusan (*decision support system*) untuk deteksi dini infeksi HIV secara otomatis dan efisien berbasis data numerik.

DAFTAR PUSTAKA

Abdillah, A. A. (2019). Support Vector Machines untuk Menyelesaikan Masalah Klasifikasi pada Pengenalan Pola. *Jurnal Poli-Teknologi*, *18*(2). https://doi.org/10.32722/pt.v18i2.1432

Balzer, L. B., Havlir, D. V., Kamya, M. R., Chamie, G., Charlebois, E. D., Clark, T. D., Koss, C. A., Kwarisiima, D., Ayieko, J., Sang, N., Kabami, J., Atukunda, M., Jain, V., Camlin, C. S., Cohen, C. R., Bukusi, E. A., Van Der Laan, M., & Petersen, M. L. (2020). Machine Learning to Identify Persons at High-Risk of Human Immunodeficiency Virus Acquisition in Rural Kenya and Uganda. *Clinical Infectious Diseases*, *71*(9), 2326–2333. https://doi.org/10.1093/cid/ciz1096

Bekti, R. D. (2019). *Klasifikasi Status Human Immunodeficiency Virus (HIV) Menggunakan Metode Support Vector Machine (SVM) dan Regresi Logistik Biner*.

Boro, C. L. T., Faisol, A., & Rudhistiar, D. (2025). Analisis Sentimen Terhadap Kampanye Pengurangan Plastik pada Media Sosial Menggunakan Metode SVM. *Jurnal Informatika Teknologi Dan Sains (Jinteks)*, *7*(1), 147–157. https://doi.org/10.51401/jinteks.v7i1.5069

Darani, N. P. (2021). Kewajiban Menuntut Ilmu dalam Perspektif Hadis. *Jurnal Riset Agama*, *1*(1), 133–144. https://doi.org/10.15575/jra.v1i1.14345

Diba, F., Lydia, M. S., & Sihombing, P. (2023). Analisis Random Forest Menggunakan Principal Component Analysis pada Data Berdimensi Tinggi. *The Indonesian Journal of Computer Science*, *12*(4). https://doi.org/10.33022/ijcs.v12i4.3329

Eliyanto, J., & Sugiyarto, S. (2020). Meningkatkan Performa Fuzzy Clustering dengan Principal Component Analysis. *Prosiding Seminar Pendidikan Matematika Dan Matematika*, *2*. https://doi.org/10.21831/pspmm.v2i0.103

Faisah, F., Toaha, S., & Kasbawati, K. (2022). Analisis Kestabilan Model Matematika Penyebaran Penyakit HIV Dengan Klasifikasi Gejala Pada Penderita. *Proximal: Jurnal Penelitian Matematika Dan Pendidikan Matematika*, *5*(2), 106–118. https://doi.org/10.30605/proximal.v5i2.1831

Firmansyach, W. A., Hayati, U., & Arie Wijaya, Y. (2023). Analisa Terjadinya Overfitting dan Underfitting pada Algoritma Naive Bayes dan Decision Tree dengan Teknik Cross Validation. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(1), 262–269. https://doi.org/10.36040/jati.v7i1.6329

Hediyati, D., & Suartana, I. M. (2021). Penerapan Principal Component Analysis (PCA) untuk Reduksi Dimensi pada Proses Clustering Data Produksi Pertanian di Kabupaten Bojonegoro. *Journal of Information Engineering and Educational Technology*, *5*(2), 49–54. https://doi.org/10.26740/jieet.v5n2.p49-54

Lela, G., Nay, F. A., & Maure, O. P. (2022). Dinamika Model Penyebaran HIV AIDS Berdasarkan Jumlah Sel CD4. *Leibniz: Jurnal Matematika*, *2*(1), 18–33. https://doi.org/10.59632/leibniz.v2i1.156

Maulana, I., Siregar, A. M., & Fauzi, A. (2024). Optimization of Machine Learning Model Accuracy for Brain Tumor Classification with Principal Component Analysis. *Jurnal Teknik Informatika (JUTIF)*. https://doi.org/10.52436/1.jutif.2024.5.3.2058

Mohamad Ismail, M. F., Mohd Arifin, S. R., Shahadan, S. Z., & Puad, N. A. N. (2021). Knowledge, Attitude, and Practice of Rukhsah in Prayer Among Undergraduate Students at International Islamic University Malaysia Kuantan Campus. *International Journal of Care Scholars*, *4*(2), 3–7. https://doi.org/10.31436/ijcs.v4i2.157

Oktafiani, R. (2024). Breast Cancer Classification with Principal Component Analysis and Smote using Random Forest Method and Support Vector Machine. *International Journal of Computer Applications*, *186*.

Parums, D. V. (2024). Editorial: Forty Years of Waiting for Prevention and Cure of HIV Infection – Ongoing Challenges and Hopes for Vaccine Development and Overcoming Antiretroviral Drug Resistance. *Medical Science Monitor*, *30*. https://doi.org/10.12659/MSM.944600

Qurthubi, I. A. (2020). *Tafsir Qurthubi* (Vol. 4). Pustaka Azzam. https://archive.org/details/tafsir-qurthubi/Tafsir%20Qurthubi%2004/

Rabbani, S., Safitri, D., Rahmadhani, N., Sani, A. A. F., & Anam, M. K. (2023). Perbandingan Evaluasi Kernel SVM untuk Klasifikasi Sentimen dalam Analisis Kenaikan Harga BBM: Comparative Evaluation of SVM Kernels for Sentiment Classification in Fuel Price Increase Analysis. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *3*(2), 153–160. https://doi.org/10.57152/malcom.v3i2.897

Ritonga, A. S., & Muhandhis, I. (2021). Teknik Data Mining untuk Mengklasifikasikan Data Ulasan Destinasi Wisata Menggunakan Reduksi Data Principal Component Analysis (PCA). *Edutic - Scientific Journal of Informatics Education*, *7*(2). https://doi.org/10.21107/edutic.v7i2.9247

Samudra, A. W., Susanto, R. A., Putra, A. R., Kurniadi, F. I., & Juarto, B. (2022). Klasifikasi HIV AIDS dengan Aplikasi Rapid Miner. *Jurnal SISKOM-KB*

(Sistem Komputer dan Kecerdasan Buatan), *6*(1), 15–19. https://doi.org/10.47970/siskom-kb.v6i1.320

Septyanto. (2023). *SVM (Support Vector Machine) untuk Klasifikasi Status Gizi Balita Berdasarkan Indeks Antropometri—LPPM ITK*. https://lppm.itk.ac.id/detail-hasil-penelitan/svm-support-vector-machine-untuk-klasifikasi-status-gizi-balita-berdasarkan-indeks-antropometri

Setiaji, B., & Pramudho, P. A. K. (2022). Pemanfaatan Teknologi Informasi Berbasis Data dan Jurnal untuk Rekomendasi Kebijakan Bidang Kesehatan. *HEALTHY : Jurnal Inovasi Riset Ilmu Kesehatan*, *1*(3), 166–175. https://doi.org/10.51878/healthy.v1i3.1649

Shihab, M. Q. (2021). *Tafsir al-Mishbāh: Pesan, Kesan, dan Keserasian Al-Qur'an* (Cet. 10). Lentera Hati.

Sonawane, R. B., & Barkade, G. D. (2023). A Review: Acquired Immunodeficiency Syndrome (AIDS). *Indian Journal of Pharmacy and Pharmacology*, *10*(3), 142–148. https://doi.org/10.18231/j.ijpp.2023.029

Suryadewiansyah, M. K., & Tju, T. E. E. (2022). Naïve Bayes dan Confusion Matrix untuk Efisiensi Analisa Intrusion Detection System Alert. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *8*(2), 81–88. https://doi.org/10.25077/TEKNOSI.v8i2.2022.81-88

Tommy Rustandi, S. R. K. W., Suhaedi, D., & Pemanasari, Y. (2023). Pemetaan Hyperplane Pada Support Vector Machine. *Bandung Conference Series: Mathematics*, *3*(2), 109–119. https://doi.org/10.29313/bcsm.v3i2.8187

Uçar, M. K., Nour, M., Sindi, H., & Polat, K. (2020). The Effect of Training and Testing Process on Machine Learning in Biomedical Datasets. *Mathematical Problems in Engineering*, *2020*, 1–17. https://doi.org/10.1155/2020/2836236

Yuliastuti, G. E., Prabiantissa, C. N., & Rizki, A. M. (2022). Klasifikasi Penyakit Menular Seksual Menggunakan Naïve Bayes. *INTEGER: Journal of Information Technology*, *7*(1). https://doi.org/10.31284/j.integer.2022.v7i1.2883