

**PENERAPAN METODE *SUPPORT VECTOR MACHINE* UNTUK
DETEKSI PENYAKIT GAGAL GINJAL**

SKRIPSI

Oleh :
SUCI WULANDARI
NIM. 210605110154



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

**PENERAPAN METODE *SUPPORT VECTOR MACHINE* UNTUK
DETEKSI PENYAKIT GAGAL GINJAL**

SKRIPSI

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

Oleh :
SUCI WULANDARI
NIM. 210605110154

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

HALAMAN PERSETUJUAN

**PENERAPAN METODE *SUPPORT VECTOR MACHINE* UNTUK
DETEKSI PENYAKIT GAGAL GINJAL**

SKRIPSI

Oleh :

SUCI WULANDARI
NIM. 210605110154

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

Pembimbing I,



Prof. Dr. Suhartono, M.Kom
NIP. 19680519 200312 1 001

Pembimbing II,



Okta Qomaruddin Aziz, M.Kom
NIP. 19911019 201903 1 013

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



Prof. Dr. Faehri Kurniawan, M.MT., IPU
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

PENERAPAN METODE *SUPPORT VECTOR MACHINE* UNTUK DETEKSI PENYAKIT GAGAL GINJAL

SKRIPSI

Oleh :
SUCI WULANDARI
NIM. 210605110154

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 06 Mei 2025

Susunan Dewan Penguji

Ketua Penguji	: <u>Roro Inda Melani, MT., M.Sc</u> NIP. 19780925 200501 2 008	()
Anggota Penguji I	: <u>Tri Mukti Lestari, M.Kom</u> NIP. 19911108 202012 2 005	()
Anggota Penguji II	: <u>Prof. Dr. Suhartono, M.Kom</u> NIP. 19680519 200312 1 001	()
Anggota Penguji III	: <u>Okta Oomaruddin Aziz, M.Kom</u> NIP. 19911019 201903 1 013	()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang




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

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Suci Wulandari
NIM : 210605110154
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Penerapan Metode *Support Vector Machine* untuk
Deteksi Penyakit Gagal Ginjal

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, 1 Mei 2025

Yang membuat pernyataan,



Suci Wulandari

NIM.210605110154

MOTTO

”Boleh jadi kamu membenci sesuatu, padahal ia amat baik bagimu, dan boleh jadi (pula) kamu menyukai sesuatu, padahal ia amat buruk bagimu; Allah mengetahui, sedang kamu tidak mengetahui.”

(QS. Al-Baqarah: 216)

HALAMAN PERSEMBAHAN

Alhamdulillah, Segala puji bagi Allah SWT yang telah memberikan limpahan rahmat dan hidayah kepada hamba-Nya, sehingga saya bisa menyelesaikan karya ini. Shalawat dan salam selalu tercurahkan kepada Nabi Muhammad SAW.

Skripsi ini saya persembahkan kepada:

Keluarga kecil penulis

Ayah dan ibu tercinta, Bapak Sul Khan dan ibu Khusnul Khotimah. Malaikat penulis, terimakasih atas doa, dukungan dan kasih sayang yang tak pernah henti kepada penulis. Serta adik tersayang, Adzkie Nadiya Maulida Khusna yang selalu memberikan doa kepada penulis.

Diri sendiri

Terima kasih karena sudah berjuang dan bertahan hingga saat ini, terima kasih karena tidak menyerah. Namun, perjalanan masih panjang, jadi tolong tetap kuat.

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya, yang telah memberikan kemudahan sehingga skripsi berjudul "Penerapan Metode *Support Vector Machine* untuk Deteksi Penyakit Gagal Ginjal" dapat terselsaikan dengan baik.

Dalam penulisan skripsi ini, tak lepas dari peran dan dukungan kepada banyak pihak yang tak ternilai. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada :

1. Prof. Dr. H. M. Zainuddin, M.A., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi.
3. Dr. Fachrul Kurniawan, M.MT, IPU selaku Ketua Program Studi Teknik Informatika.
4. Prof Suhartono, M.Kom selaku Dosen Pembimbing I atas bimbingan dan arahnya hingga skripsi ini dapat diselesaikan dengan baik.
5. Bapak Okta Qomaruddin Aziz, M.Kom selaku Dosen Pembimbing II atas masukan dan dukungan selama penulisan skripsi.
6. Ibu Roro Inda Melani, M.Sc selaku Penguji I dan Ibu Tri Mukti Lestari, M.Kom selaku Penguji II atas saran dan kritik yang membangun dalam proses ujian skripsi.
7. Ibu Puspa Miladin Nuraida Safitri, M.Kom dan Bapak Dr. Cahyo Crys dian M.Cs selaku Dosen Wali atas arahan dan bimbingannya selama masa studi.

8. Ibu Nia Faricha S, Si., selaku Admin Program Studi Teknik Informatika atas bantuan dan informasi yang selalu diberikan dengan sabar selama perkuliahan dan penyusunan skripsi.
9. Seluruh dosen, laboran, dan staf Prodi Teknik Informatika atas ilmu yang diberikan hingga penulis menyelesaikan studi ini.
10. Keluarga kecil penulis, Ayah Sulkhan, Ibu Khusnul Khotimah, dan Adik Adzkie Nadiya Maulida Khusna atas doa, dukungan, dan cinta yang tak pernah putus. Terima kasih telah menjadi tempat ternyaman dalam setiap langkah penulis. Semoga segala pengorbanan dibalas berlipat oleh Allah SWT dan kita dipertemukan kembali di surga-Nya.
11. Teman dekat penulis "Ceciwi", Mala, Fira, dan Afifah yang selalu menemani sejak awal kuliah hingga penyusunan skripsi, serta atas dukungan dan semangat yang tak henti diberikan.
12. Teman dekat penulis Apis dan Nafisah, yang selalu memberikan bantuan, perhatian, dan dukungan kepada penulis.
13. Teman seperjuangan, Adila, Nurul, Nova, Zidan, Qimal, Hamidah, Vivin, terima kasih atas segala bantuan yang diberikan.
14. Manusia-manusia religius '21, Niya, Shafira, dan Hafidz, yang selalu siap kapan saja dimana saja. Terima kasih sudah kebersamaan penulis selama ini.
15. Teman-teman Himatif Encoder, Divisi Relligius '22 dan '23. atas pengalaman yang sangat berharga bagi penulis.

16. Seluruh warga besar Teknik Informatika UIN Malang, khususnya angkatan 2021 "ASTER" yang telah memberikan dukungan dan bantuan kepada penulis.
 17. Saudara dekat penulis, Siti Nila Masyifi'ah yang selalu memberikan nasehat kehidupan yang sangat berharga bagi penulis.
 18. Seluruh pihak yang telah terlibat, baik secara langsung maupun tidak langsung dari awal perkuliahan hingga akhir penulisan skripsi ini.
 19. Diri sendiri, Suci Wulandari, atas perjuangan dan usaha yang telah dilakukan sejauh ini di jurusan yang tak pernah terbayangkan sebelumnya. Semoga setiap langkah dan ilmu yang didapat selama perkuliahan membawa manfaat bagi banyak orang serta menjadi amal jariyah yang terus mengalir pahalanya.
- Akhir kata, penulis menyadari bahwa skripsi ini masih memiliki kekurangan, baik dalam aspek keilmuan maupun penulisan. Oleh karena itu, penulis dengan senang hati menerima kritik dan saran yang membangun. Harapannya, skripsi ini dapat memberikan manfaat bagi berbagai pihak.

Malang, 01 Mei 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT	xvi
مستخلص البحث.....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB II STUDI PUSTAKA	5
2.1 Penelitian Terkait.....	5
2.2 Gagal Ginjal Kronis	10
2.3 <i>Support Vector Machine</i> (SVM).....	11
2.4 Fungsi Kernel.....	14
2.5 <i>Synthetic Minority Oversampling Technique</i> (SMOTE)	14
2.6 <i>Principal Component Analysis</i> (PCA).....	15
BAB III METODE PENELITIAN	18
3.1 Desain Penelitian	18
3.2 Pengumpulan Data	19
3.3 Desain Sistem	22
3.4 <i>Preprocessing</i> Data.....	23
3.4.1 Pengecekan <i>Missing Value</i>	23
3.4.2 Penanganan <i>Missing Value</i>	23
3.4.3 <i>Label Encoding</i>	24
3.4.4 Standarisasi Fitur	25
3.4.5 <i>Principal Component Analysis</i> (PCA).....	27
3.5 Split Data	28
3.6 <i>Synthetic Minority Oversampling Technique</i> (SMOTE)	29
3.7 Implementasi <i>Support Vector Machine</i>	29
3.8 Evaluasi Performa.....	31
3.9 Skenario Uji Coba.....	33
BAB IV HASIL DAN PEMBAHASAN	35
4.1 Hasil <i>Preprocessing</i> Data	35
4.1.1 Pengecekan <i>Missing Value</i>	35
4.1.2 Penanganan <i>Missing Value</i>	36

4.1.3 <i>Label Encoding</i>	36
4.1.4 Standarisasi Fitur	38
4.1.5 PCA	39
4.2 Hasil Split Data.....	40
4.3 Hasil Uji Coba Model	41
4.3.1 Hasil Uji Coba Skenario 1	41
4.3.2 Hasil Uji Coba Skenario 2	43
4.3.3 Hasil Uji Coba Skenario 3	45
4.4 Pembahasan	47
4.5 Integrasi Islam.....	59
BAB V KESIMPULAN DAN SARAN	61
5.1 Kesimpulan	61
5.2 Saran	62
DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar 2. 1 Konsep SVM.....	11
Gambar 3. 1 Desain Penelitian.....	18
Gambar 3. 2 Desain Sistem.....	22
Gambar 4. 1 <i>Confusion Matrix</i> Skenario 1	42
Gambar 4. 2 <i>Confusion Matrix</i> Skenario 2	44
Gambar 4. 3 <i>Confusion Matrix</i> Skenario 3	46
Gambar 4. 4 Grafik Akurasi Setiap Skenario	48
Gambar 4. 5 Grafik <i>F1-Score</i> Setiap Skenario	51
Gambar 4. 6 Distribusi Data Chronic Kidney Disease Pada PC 1.....	54
Gambar 4. 7 Distribusi Data Baru pada PC 1	58

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	8
Tabel 3. 1 Deskripsi Atribut Dataset Chronic Kidney Disease.....	19
Tabel 3. 2 Contoh Sampel Dataset Chronic Kidney Disease.....	21
Tabel 3. 3 Contoh Data Sebelum <i>Label Encoding</i>	24
Tabel 3. 4 Contoh Data Setelah <i>Label Encoding</i>	24
Tabel 3. 5 Contoh Data Sebelum di Standarisasi.....	25
Tabel 3. 6 Contoh Data Setelah di Standarisasi	26
Tabel 3. 7 Contoh Data Perhitungan Manual SVM.....	30
Tabel 3. 8 <i>Confusion Matrix</i>	31
Tabel 3. 9 Skenario Pengujian Pembagian Data	33
Tabel 3. 10 Sub-skenario	34
Tabel 4. 1 Nilai NaN Pada Setiap Kolom	35
Tabel 4. 2 Data Sebelum <i>Encoding</i>	37
Tabel 4. 3 Data Setelah <i>Encoding</i>	37
Tabel 4. 4 Data Sebelum Standarisasi.....	38
Tabel 4. 5 Data Setelah Standarisasi	38
Tabel 4. 6 Nilai <i>Eigenvalue</i> dan <i>Variance Ratio</i>	39
Tabel 4. 7 Hasil Split Data	40
Tabel 4. 8 Hasil Split Data Setelah SMOTE.....	40
Tabel 4. 9 Matrix Evaluasi Skenario 1	41
Tabel 4. 10 Matrix Evaluasi Skenario 2.....	43
Tabel 4. 11 Matrix Evaluasi Skenario 3.....	45
Tabel 4. 12 Matrix Evaluasi Setiap Skenario.....	48
Tabel 4. 13 Matrix Evaluasi Data Baru.....	56

ABSTRAK

Wulandari, Suci. 2025. **Penerapan Metode *Support Vector Machine* untuk Deteksi Penyakit Gagal Ginjal**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Suhartono, M.Kom (II) Okta Qomaruddin Aziz, M.Kom.

Kata Kunci: Gagal Ginjal Kronis, Klasifikasi, *Machine Learning*, *Support Vector Machine* (SVM), *Principal Component Analisis* (PCA)

Gagal ginjal kronis merupakan salah satu penyakit yang terjadi akibat penurunan fungsi ginjal secara bertahap dan dapat menyebabkan kematian apabila tidak ditangani dengan tepat. Pencegahan dan deteksi dini menjadi penting untuk meminimalkan risiko yang ditimbulkan, hal tersebut bisa dilakukan dengan menggunakan algoritma machine learning. Penelitian ini bertujuan untuk mengevaluasi performa algoritma *Support Vector Machine* (SVM) dalam klasifikasi penyakit gagal ginjal kronis menggunakan dataset Chronic Kidney Disease dari UCI Machine Learning Repository. Data diproses melalui beberapa tahap yaitu penghapusan nilai yang hilang, *label encoding*, normalisasi, dan penyeimbangan data menggunakan metode SMOTE. Penelitian ini menggunakan tiga skenario pembagian data dan menggunakan dua jenis kernel yaitu linear dan polynomial. Model diuji dalam dua kondisi, yaitu tanpa reduksi dimensi dan dengan reduksi dimensi menggunakan *Principal Component Analisis* (PCA). Hasil penelitian menunjukkan bahwa model SVM dengan kernel linear, baik tanpa PCA maupun dengan PCA, mampu mencapai akurasi hingga 100% pada beberapa skenario. Sementara itu, kernel polynomial menunjukkan peningkatan performa saat menggunakan PCA, dengan akurasi tertinggi sebesar 92%.

ABSTRACT

Wulandari, Suci. 2025. **Application of Support Vector Machine Method for Kidney Failure Disease Detection**. Undergraduate Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Supervisor: (I) Prof. Dr. Suhartono, M.Kom (II) Okta Qomaruddin Aziz, M.Kom.

Keywords: Chronic Kidney Failure, Classification, Machine Learning, Support Vector Machine (SVM), Principal Component Analysis (PCA)

Chronic kidney disease is a condition that occurs due to a gradual decline in kidney function and can be fatal if not treated properly. Prevention and early detection are important to minimize the risks involved, and this can be achieved using machine learning algorithms. This study aims to evaluate the performance of the Support Vector Machine (SVM) algorithm in classifying chronic kidney disease using the Chronic Kidney Disease dataset from the UCI Machine Learning Repository. The data was processed through several stages, including missing value imputation, label encoding, normalization, and data balancing using the SMOTE method. This study employed three data splitting scenarios and utilized two types of kernels: linear and polynomial. The model was tested under two conditions: without dimension reduction and with dimension reduction using Principal Component Analysis (PCA). The results showed that the SVM model with a linear kernel, both without PCA and with PCA, achieved accuracy of up to 100% in some scenarios. Meanwhile, the polynomial kernel demonstrated improved performance when using PCA, with the highest accuracy reaching 92%.

مستخلص البحث

ولانداري، سوجي 2025. تطبيق طريقة آلة دعم ناقلات الدعم للكشف عن أمراض الفشل الكلوي. البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: أ. د. سهارتونو، ماجستير. المشرف الثاني: أوقتا قوما رودين عزيز، ماجستير.

الكلمات الرئيسية: الفشل الكلوي المزمن، التصنيف، تعلم الآلة، آلة المتجهات الداعمة (SVM)، تحليل المكونات الرئيسية (PCA)

الفشل الكلوي المزمن هو مرض يحدث بسبب التدهور التدريجي في وظائف الكلى ويمكن أن يسبب الوفاة إذا لم يتم علاجه بشكل صحيح. من المهم الوقاية والكشف المبكر لتقليل المخاطر، ويمكن القيام بذلك باستخدام خوارزميات التعلم الآلي. تهدف هذه الدراسة إلى تقييم أداء خوارزمية آلة دعم المتجهات (SVM) في تصنيف الفشل الكلوي المزمن باستخدام مجموعة بيانات مرض الكلى المزمن من مستودع التعلم الآلي لجامعة كاليفورنيا في كاليفورنيا. تتم معالجة البيانات من خلال عدة مراحل، وهي إزالة القيم المفقودة، وترميز التسمية، والتطبيع، وموازنة البيانات باستخدام طريقة SMOTE. وتستخدم هذه الدراسة ثلاثة سيناريوهات لمشاركة البيانات وتستخدم نوعين من النوى، وهما النواة الخطية ومتعددة الحدود. تم اختبار النموذج في حالتين، وهما دون تقليل الأبعاد ومع تقليل الأبعاد باستخدام تحليل المكونات الرئيسية (PCA). أظهرت النتائج أن نماذج SVM ذات النواة الخطية، سواءً بدون تحليل المكونات الرئيسية أو باستخدام تحليل المكونات الرئيسية، كانت قادرة على تحقيق دقة تصل إلى 100٪ في بعض السيناريوهات. وفي الوقت نفسه، أظهرت النواة متعددة الحدود أداءً محسناً عند استخدام تحليل المكونات الرئيسية، حيث بلغت أعلى دقة 92٪.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Ginjal merupakan organ vital yang berperan penting dalam kesehatan. Organ ini berperan dalam mengatur keseimbangan cairan tubuh melalui dengan cara membuang sisa metabolisme dan mempertahankan zat-zat yang masih dibutuhkan oleh tubuh. Namun, fungsi ginjal bisa mengalami penurunan ketika mengalami kerusakan atau terkena penyakit, seperti gagal ginjal kronis (Anwar and Manuharawati 2021).

Gagal ginjal kronis adalah keadaan ketika kemampuan ginjal menurun secara beransur karena kerusakan pada ginjal. Gagal ginjal kronis mengakibatkan tubuh tak mampu lagi untuk menjaga metabolisme serta keseimbangan cairan tubuh. Apabila masalah pada ginjal ini tidak segera ditangani, resiko kerusakan gagal ginjal lanjutan akan meningkat, yang pada akhirnya dapat berujung pada kematian(Putri et al. 2023).

Data Riset Kesehatan Dasar (Riskesdas) Kementerian Kesehatan RI pada tahun 2018, menunjukkan bahwa lebih dari 700 ribu penduduk indonesia, atau sekitar 3,8% dari populasi, menderita penyakit gagal ginjal kronis. Data dari Institute for Health Metrics and Evaluation (IHME) *Global Burden of Disease* (GBD) 2019, penyakit gagal ginjal kronis ini masuk ke dalam sepuluh besar penyebab kematian tertinggi di Indonesia. Direktur Pencegahan dan Pengendalian Penyakit Tidak Menular Kemenkes RI, Dr. Eva Susanti, menyatakan bahwa jumlah

kematian akibat penyakit ginjal kronis di Indonesia mencapai lebih dari 42 ribu jiwa (Salsabilla 2024).

Penyakit pada ginjal disebabkan oleh beberapa faktor seperti kerusakan pembuluh darah, diabetes, hipertensi, infeksi, dan gaya hidup yang tidak sehat. Penyebab terbanyak gagal ginjal kronik di Indonesia adalah diabetes dan hipertensi, dimana ketika gula darah atau tekanan darah tidak terkontrol maka akan semakin mempercepat perkembangan diabetes dan hipertensi, yang dapat menyebabkan komplikasi, salah satunya adalah gagal ginjal.

Dalam kehidupan sehari-hari, menjaga kesehatan merupakan salah satu bentuk ibadah yang diperintahkan oleh Allah SWT. Kesehatan adalah anugerah yang harus dijaga, dan upaya untuk mencegah penyakit merupakan tanggung jawab kita sebagai hamba-Nya. Allah SWT mengingatkan kita akan pentingnya menjaga diri dalam Surah Al-Baqarah (2:195):

وَأَنْفِقُوا فِي سَبِيلِ اللَّهِ وَلَا تُلْقُوا بِأَيْدِيكُمْ إِلَى التَّهْلُكَةِ ۚ وَأَحْسِنُوا ۚ إِنَّ اللَّهَ يُحِبُّ الْمُحْسِنِينَ

"Dan belanjakanlah (hartamu) di jalan Allah, dan janganlah kamu menjatuhkan dirimu sendiri ke dalam kebinasaan, dan berbuat baiklah; sesungguhnya Allah menyukai orang-orang yang berbuat baik." (QS. Al-Baqarah:195).

Ayat tersebut mengingatkan pentingnya menjaga kesehatan dan menjauhkan diri dari hal-hal yang membahayakan tubuh, seperti menghindari faktor-faktor risiko yang dapat menyebabkan penyakit gagal ginjal kronik. Beberapa upaya untuk menjaga kesehatan, seperti memantau gula darah dan tekanan darah, serta memanfaatkan teknologi untuk mendeteksi dini penyakit, merupakan bagian cara kita menghindari 'kebinasaan' yang disebutkan dalam ayat ini.

Selain itu, dalam sabda Rasulullah SAW juga ditegaskan bahwa setiap penyakit yang ada pasti memiliki obat atau penanganan, sebagaimana hadits yang diriwayatkan oleh Imam Bukhari:

مَا أَنْزَلَ اللَّهُ دَاءً إِلَّا أَنْزَلَ لَهُ شِفَاءً

"Tidaklah Allah menurunkan suatu penyakit, melainkan Dia juga menurunkan obatnya." (HR. Bukhari)

Hadis ini memberikan motivasi bahwa dengan usaha dan pengetahuan, manusia dapat terus mencari solusi terbaik dalam menangani penyakit. Kemajuan ilmu pengetahuan dan teknologi merupakan bagian dari anugerah Allah yang dapat dimanfaatkan untuk meningkatkan kualitas kesehatan manusia. Kemajuan teknologi kini banyak diterapkan dalam bidang medis, termasuk penggunaan algoritma *machine learning* untuk memprediksi penyakit gagal ginjal kronis secara cepat dan efisien. Salah satu algoritma *machine learning* yaitu *support vector machine* (SVM). Algoritma SVM bekerja dengan menemukan hyperplane terbaik yang berfungsi sebagai pembatas untuk memisahkan data ke dalam kelas-kelas yang berbeda. (Imaniar Ikko Mulya Rizky, Suhendro Yusuf Irianto, and Sriyanto 2023).

Dalam penelitian ini, metode *Support Vector Machine* akan digunakan untuk mengklasifikasikan pasien berdasarkan data kesehatan guna mendeteksi penderita gagal ginjal kronis. Diharapkan hasil penelitian ini bermanfaat bagi dunia kesehatan dan memberikan kontribusi pada upaya pencegahan penyakit gagal ginjal kronis.

1.2 Rumusan Masalah

Bagaimana performa metode *Support Vector Machine* dalam mengklasifikasikan penyakit gagal ginjal kronis?

1.3 Batasan Masalah

Adapun batasan penelitian ini adalah:

- a. Data yang digunakan pada penelitian ini yaitu Dataset Chronic Kidney Disease yang didapat dari website UCI.
- b. Menggunakan 2 jenis kernel yaitu kernel linear dan kernel polynomial.
- c. SVM tanpa PCA menggunakan 25 fitur.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengevaluasi performa metode *Support Vector Machine* dalam mengklasifikasikan penyakit gagal ginjal kronis berdasarkan dataset Chronic Kidney Disease.

1.5 Manfaat Penelitian

Berikut beberapa manfaat yang diharapkan:

1. Memberikan wawasan kepada pembaca tentang penerapan metode *Support Vector Machine* dalam mengklasifikasikan penyakit gagal ginjal kronis.
2. Hasil penelitian ini bisa digunakan sebagai landasan pada penelitian selanjutnya.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian oleh (Swain et al. 2023) berjudul “*A Robust Chronic Kidney Disease Classifier Using Machine Learning*”. Penelitian ini membahas tentang klasifikasi penyakit ginjal kronis dengan membandingkan dua metode yaitu *Support Vector Machine* dan *Random Forest* dalam klasifikasi penyakit gagal ginjal kronis. Data yang digunakan berasal dari UCI Machine Learning dengan total 400 data dan 25 fitur. Pada penelitian tersebut dilakukan *preprocessing* data yaitu proses imputasi nilai yang hilang, penyeimbangan data, dan penskalaan fitur. Dalam pemilihan fitur (Swain et al. 2023) menggunakan Teknik *chi-squared*. Dimana dengan teknik tersebut dipilih 9 fitur untuk proses klasifikasi. 9 fitur tersebut meliputi: sg, al, su, pcc, dm, cad, appet, htn dan ane. Hasil penelitian ini menunjukkan bahwa model SVM lebih unggul dari *Random Forest*. Model SVM berhasil mendapatkan akurasi sebesar 97,75% sementara *Random Forest* mendapatkan akurasi 97%.

Penelitian oleh (Madani et al. 2024) berjudul “*Perbandingan Kinerja Klasifikasi Penyakit Ginjal Menggunakan Algoritma Support Vector Machine (SVM) dan Decision Tree*”. Penelitian ini membahas tentang perbandingan algoritma klasifikasi pada penyakit gagal ginjal kronis yaitu menggunakan algoritma *Support Vector Machine* dan *Decision Tree*. Pada penelitian tersebut beberapa tahapan pada *preprocessing* data yang dilakukan meliputi penghapusan

nilai yang hilang, menghapus outlier, dan menormalisasi data. Penelitian ini menggunakan keseluruhan fitur yang tersedia untuk klasifikasi penyakit gagal ginjal kronis. Data yang digunakan pada penelitian tersebut berasal dari kaggle. Hasil penelitian ini menunjukkan bahwa algoritma *Support Vector Machine* lebih unggul dari pada *Decision Tree*, dengan hasil akurasi dari algoritma *Support Vector Machine* mencapai 97,5%, dengan presisi sebesar 0,98, *recall* sebesar 0,96, dan *f1-score* sebesar 0,97. Sementara itu, algoritma *Decision Tree* menghasilkan akurasi sebesar 92,5%, presisi 0,92, *recall* 0,90, dan *f1-score* 0,91.

Penelitian lainnya oleh (Yunianto 2024) berjudul “*Diagnostik Penyakit Ginjal Kronis Menggunakan Model Klasifikasi Support Vector Machine*”. Penelitian ini mengkaji tentang penerapan model *Support Vector Machine* dalam pengklasifikasian penyakit ginjal kronis. Data yang dipakai bersumber dari UCI Machine Learning dengan total 400 data dan 25 atribut. Pada penelitian tersebut dari 25 atribut yang ada peneliti memilih 5 atribut yang digunakan dan nilai yang hilang dihapus dari dataset. Pemilihan 5 atribut tersebut berdasarkan statistik dan wawancara. 5 atribut yang dipilih yaitu *sc*, *pcu*, *bp*, *ane*, dan *htn*. Penelitian ini berhasil menghasilkan suatu sistem yang dapat mengklasifikasikan kondisi penyakit ginjal kronis dengan akurasi yang didapatkan sebesar 96,34%.

Penelitian lain oleh (Akbar, Yudistira, and Ridok 2023) berjudul “*Identifikasi Gagal Ginjal Kronis dengan Mengimplemtasikan Metode Support Vector Machine beserta K-Nearest Neighbour(SVM-KNN)*”. Penelitian ini mengkaji identifikasi gagal ginjal kronis dengan mengimplementasikan metode *Support Vector Machine(SVM)* dan *K-Nearest Neighbour(KNN)*, serta

penggabungan kedua metode tersebut yang dikenal sebagai SVM-KNN. Penelitian ini menggunakan data yang berasal dari UCI Machine Learning. Pada penelitian ini dari 400 data dan 25 atribut yang ada pada dataset, (Akbar et al. 2023) hanya menggunakan 302 data dan 6 atribut. 6 atribut yang digunakan yaitu sg, bp, al, su, rbcc dan wbcc. Hasil penelitian menunjukkan SVM-KNN menghasilkan rata-rata akurasi terbaik sebesar 94,25, sedangkan SVM sebesar 94,09%, dan KNN sebesar 91,73%.

Penelitian oleh(Saputra 2023) dengan judul "*Perbandingan Kernel pada Support Vector Machine untuk Klasifikasi Penyakit Padi*". Penelitian ini bertujuan untuk mengevaluasi performa berbagai kernel pada SVM, yaitu linear, polynomial, RBF, dan sigmoid, dalam mengklasifikasikan jenis penyakit tanaman padi. Hasil penelitian ini menunjukkan bahwa kernel linear secara konsisten menghasilkan akurasi tertinggi dibandingkan kernel lainnya. Pada komposisi data latih dan uji 70:30, kernel linear mencapai akurasi sebesar 89%, yang signifikan lebih tinggi dibandingkan kernel RBF yang hanya mencapai akurasi terendah sebesar 38%.

Penelitian lain oleh(Rahayu and Yamasari 2024) dengan judul "*Klasifikasi Penyakit Stroke dengan Metode Support Vector Machine(SVM)*". Penelitian ini bertujuan untuk mengevaluasi performa beberapa kernel pada SVM, yaitu linear, polynomial, RBF, dan sigmoid, dalam mengklasifikasikan data pasien stroke. Hasil penelitian menunjukkan bahwa kernel polynomial secara konsisten menghasilkan akurasi tertinggi. Pada rasio data latih dan uji 80:20, kernel polynomial mencapai akurasi sebesar 78.86%, dengan presisi 73.98% dan *recall* 56.75%. Hasil ini lebih tinggi dibandingkan dengan kernel linear, yang mencapai akurasi 75.73%, serta

kernel RBF dan sigmoid, yang menunjukkan performa lebih rendah. Penelitian ini menyimpulkan bahwa kernel polynomial adalah pilihan yang paling efektif untuk klasifikasi penyakit stroke pada SVM, memberikan hasil yang lebih akurat dibandingkan kernel lainnya.

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti	Judul Penelitian	Metode	Hasil	Perbedaan Penelitian
1	Swain et al., 2023	<i>A Robust Chronic Kidney Disease Classifier Using Machine Learning</i>	<i>Support Vector Machine</i> <i>Random Forest</i>	SVM memberikan hasil akurasi sebesar 97,75% dan RF yaitu sebesar 97%.	Menggunakan 9 fitur (<i>sg, al, su, pcc, dm, cad, appet, htn dan ane</i>), kernel RBF Rasio data 70:30.
2	Madani et al., 2024	<i>Perbandingan Kinerja Klasifikasi Penyakit Ginjal Menggunakan Algoritma Support Vector Machine (SVM) dan Decision Tree</i>	<i>Support Vector Machine</i> <i>Decision Tree</i>	Algoritma Support Vector Machine lebih unggul dibandingkan Decision Tree, dengan akurasi 97,5%, Sementara itu, Decision Tree menghasilkan akurasi 92,5%.	Menggunakan semua fitur, kernel RBF, dan Rasio pembagian data 90:10, 80:20, 70:30.
3	Yunianto, 2024	<i>Diagnostik Penyakit Ginjal Kronis Menggunakan Model Klasifikasi Support Vector Machine</i>	<i>Support Vector Machine</i>	Algoritma SVM menghasilkan akurasi yang baik sebesar 96,34%	Menggunakan 5 fitur (<i>sc, pcu, bp, ane, dan htn</i>), Menggunakan Kernel linear, polynomial, RBF, dan Sigmoid. Rasio pembagian data 90:10, 80:20, 70:30.
4	Akbar et al., 2023	<i>Identifikasi Gagal Ginjal Kronis dengan Mengimplemtasikan Metode Support Vector Machine beserta K-Nearest Neighbour (SVM-KNN)</i>	<i>Support Vector Machine beserta K-Nearest Neighbour (SVM-KNN)</i>	Hasil penelitian menunjukkan SVM-KNN menghasilkan rata-rata akurasi terbaik sebesar 94,25%, sedangkan SVM sebesar 94,09%, dan	Menggunakan 6 fitur(<i>sg, bp, al, su, rbcc dan wbcc</i>), kernel RBF, dan Rasio pembagian data 60:40

No	Peneliti	Judul Penelitian	Metode	Hasil	Perbedaan Penelitian
				KNN sebesar 91,73%.	
5	Saputra 2023	<i>Perbandingan Kernel pada Support Vector Machine untuk Klasifikasi Penyakit Padi</i>	<i>Support Vector Machine</i>	Kernel Linear menghasilkan akurasi yang lebih tinggi dibanding kernel-kernel lainnya.	Dataset yang digunakan berbeda
6	Rahayu and Yamasari 2024	<i>Klasifikasi Penyakit Stroke dengan Metode Support Vector Machine (SVM)</i>	<i>Support Vector Machine</i>	Kernel Polinomial menghasilkan Akurasi yang lebih baik dibanding kernel-kernel lainnya.	Dataset yang digunakan berbeda

Perbedaan utama penelitian ini dibandingkan penelitian terdahulu terletak pada penggunaan metode *Principal Component Analysis* (PCA) sebagai teknik reduksi dimensi sebelum pengklasifikasian menggunakan *Support Vector Machine* (SVM). Pendekatan ini dilakukan untuk mengidentifikasi pengaruh reduksi fitur terhadap kinerja model, yang belum banyak dibahas dalam penelitian sebelumnya pada konteks penyakit ginjal kronis. Selain itu, penelitian ini juga melakukan tahap *preprocessing* berupa penghapusan nilai yang hilang dalam dataset untuk memastikan data bersih dan siap digunakan dalam proses pelatihan model, sehingga data yang digunakan dalam pelatihan berjumlah 158 data. Untuk mengatasi ketidakseimbangan kelas, peneliti juga menerapkan SMOTE pada data latih. Kombinasi penghapusan nilai yang hilang, reduksi fitur menggunakan PCA, dan SMOTE diharapkan dapat meningkatkan efisiensi serta akurasi model SVM dalam mengklasifikasi pasien yang terkena penyakit ginjal kronis.

2.2 Gagal Ginjal Kronis

Gagal ginjal kronis adalah kerusakan pada organ ginjal yang menyebabkan ginjal tidak dapat menyaring ataupun membuang cairan elektrolit didalam tubuh. Penyakit ini mengakibatkan ginjal tidak dapat menjaga keseimbangan cairan dan zat kimia di dalam tubuh(Suryani, Indra, and Saputra 2022). Penyakit Ginjal Kronis (PGK) atau *Chronic Kidney Disease* (CKD) adalah kondisi penurunan fungsi ginjal yang berlangsung secara kronis dan progresif selama lebih dari 3 bulan, ditandai dengan kerusakan struktur atau fungsi ginjal. Kerusakan ini bisa terjadi dengan atau tanpa penurunan Laju Filtrasi Glomerulus (LFG), yang umumnya ditandai dengan LFG di bawah 60 ml/menit/1,73 m². PGK sering dikaitkan dengan kondisi seperti diabetes mellitus, hipertensi, penyakit kardiovaskular, dan penuaan(Susanto 2020).

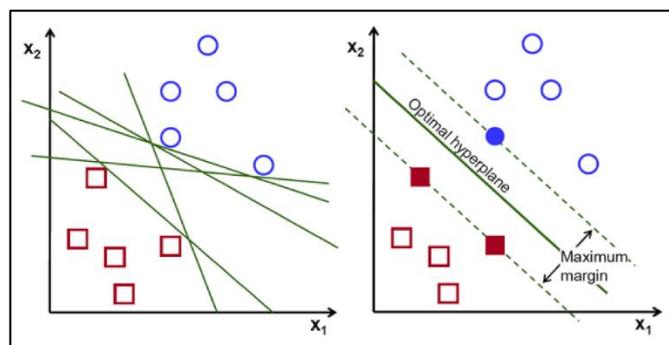
Tanda dan gejala awal Penyakit Ginjal Kronis sering kali tidak spesifik, namun dapat dikenali melalui beberapa indikasi berikut: perubahan pola buang air kecil seperti sering terbangun di malam hari, air seni berbuih, bercampur darah, atau perubahan frekuensi dan warna, pembengkakan pada tubuh, mudah lelah, gatal dan ruam pada kulit, mual dan muntah, sesak napas, pusing dan sulit berkonsentrasi, nyeri di pinggang bagian bawah, demam, serta bau mulut tidak sedap(Penyusun 2019).

Gejala sering kali tidak muncul hingga fungsi ginjal turun di bawah 15%. Pada tahap awal, PGK dapat menyebabkan komplikasi seperti anemia dan penyakit tulang, yang meningkatkan risiko kesakitan dan kematian, termasuk akibat penyakit jantung. Penatalaksanaan PGK bertujuan memperlambat progresivitas penyakit, mencegah gagal ginjal, serta meningkatkan kualitas hidup pasien. Faktor penyebab

PGK meliputi diabetes mellitus, hipertensi, obesitas, merokok, penyakit autoimun, batu saluran kemih, penyumbatan saluran kemih, keracunan obat, infeksi sistemik, usia di atas 50 tahun, serta riwayat keluarga dengan diabetes, hipertensi (Penyusun 2019).

2.3 *Support Vector Machine (SVM)*

Support Vector Machine (SVM), yang diperkenalkan oleh Vapnik pada tahun 1995, merupakan metode klasifikasi dan regresi. Metode ini unggul dalam mengolah data linear maupun data non-linear. Pengklasifikasian dengan algoritma SVM ini dilakukan dengan cara menemukan *hyperplane* optimal yang berfungsi sebagai pemisah data antar kelas. Jarak antara satu kelas dengan kelas lainnya direpresentasikan oleh nilai yang disebut *margin*. Nilai margin maksimum inilah yang menentukan seberapa optimal *hyperplane* tersebut. *Support vector* sendiri ialah titik-titik pada setiap kelas yang memiliki jarak paling dekat dengan *hyperplane*.



Gambar 2. 1 Konsep SVM

Pada Gambar 2.1, bagan kiri terdapat beberapa garis pemisah (*hyperplane*) yang beragam, namun garis pemisah tersebut kurang optimal dalam memisahkan dua kelas data yang ada, yaitu kotak merah dan lingkaran biru. *Support Vector Machine* bertujuan untuk menemukan *hyperplane* terbaik, seperti yang terlihat di

gambar sebelah kanan. Dimana *hyperplane* yang ditemukan memaksimalkan jarak antar kedua kelas. *Hyperplane* yang optimal adalah garis pemisah yang memiliki jarak terjauh dari data terdekat dari kedua kelas. Untuk menemukan *hyperplane* optimal yang memisahkan dua kelas, *Support Vector Machine*(SVM) mengoptimalkan parameter-model yang terkait dengan w (vektor normal) dan b (bias). Proses ini melibatkan optimasi kuadrat yang bertujuan untuk meminimalkan fungsi objektif yang terkait dengan w , dengan kendala dimana data memenuhi syarat margin yang terpisah.

Salah satu metode optimasi yang digunakan untuk memecahkan masalah ini adalah *Sequential Minimal Optimization*(SMO). SMO adalah teknik yang lebih efisien dalam melatih SVM, terutama ketika jumlah data sangat besar. SMO bekerja dengan memecah masalah optimasi yang besar menjadi sub-masalah yang lebih kecil, di mana setiap sub-masalah hanya melibatkan dua parameter α yang diperbarui pada setiap iterasi. Setelah proses optimasi, model SVM akan memiliki parameter α yang optimal, yang digunakan untuk menentukan fungsi keputusan.

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

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

$$\vec{w} \cdot \vec{x} + b = 0 \quad (2.1)$$

Dari persamaan 2.1 diperoleh:

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

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

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

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

Dengan kendala:

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

Keterangan:

y_i : target ke- i

\vec{w} : bidang normal

\vec{x}_i : data input

b : posisi bidang relatif terhadap pusat koordinat

4. Membuat fungsi keputusan untuk persamaan 2.6 dan 2.7:

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

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

Keterangan:

n : jumlah support vector

\vec{x}_i : support vector ke- i

\vec{x}_d : data uji

α : bobot dari setiap vector, yang dihitung saat optimasi

y_i : label ke- i

b : bias, yang dihitung saat optimasi

Karena metode *Support Vector Machine* memiliki kesederhanaan dan fleksibilitas dalam menangani berbagai masalah klasifikasi, metode ini memberikan kinerja prediksi yang konsisten, bahkan ketika ukuran sampel terbatas. Oleh sebab itu, peneliti memilih menggunakan metode *Support Vector Machine*.

2.4 Fungsi Kernel

Kernel dalam SVM memiliki peran penting karena memungkinkan model untuk menangani data yang tidak dapat dipisahkan secara linear di ruang fitur aslinya. Dengan menerapkan fungsi kernel, SVM dapat memetakan data ke ruang dimensi yang lebih tinggi, sehingga pemisahan antara kelas-kelas data menjadi lebih jelas dan mudah. Berbagai fungsi kernel seperti linear dan polinomial memberikan fleksibilitas yang diperlukan untuk menangani berbagai bentuk dan pola data, sehingga dapat menemukan *hyperplane* yang optimal.

2.5 *Synthetic Minority Oversampling Technique (SMOTE)*

Synthetic Minority Oversampling Technique (SMOTE) adalah teknik penyeimbangan distribusi jumlah sampel data pada kelas minoritas. SMOTE bekerja dengan menambah data pada kelas minoritas agar setara dengan data kelas mayoritas dengan membuat data sintetis (Celena, Furqon, and Ridok 2022). SMOTE dapat membantu memperbaiki ketidakseimbangan klasifikasi data dan model klasifikasi. Berikut merupakan tahapan pada SMOTE:

1. Memilih sampel dari kelas minoritas.
2. Menghitung jarak antar data pada kelompok minoritas dengan persamaan jarak euclidean sebagai berikut:

$$\text{Jarak euclidean} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (2.8)$$

Keterangan :

x_i : Nilai dari fitur ke-i pada titik data pertama

y_i : Nilai dari fitur ke-i pada titik data kedua

3. Menentukan presentasi SMOTE yang akan diterapkan untuk menghasilkan data sintesis.
4. Membuat data sintesis, dengan persamaan berikut:

$$X_{syn} = x_i(x_{knn} - x_i) \sigma \quad (2.9)$$

Keterangan:

- x_{syn} : data sintesis yang akan dibuat
- x_i : data yang akan dijadikan referensi untuk pembuatan data sintesis
- x_{knn} : data yang paling dekat jaraknya dari data referensi
- σ : nilai acak antara 0 dan 1

2.6 *Principal Component Analysis (PCA)*

Principal Component Analysis (PCA) adalah metode yang digunakan untuk mereduksi dimensi data yang terdiri dari banyak variabel saling berkaitan, dengan tujuan mempertahankan sebanyak mungkin informasi dari dataset tersebut (Ihsani, Arifin, and Fatoni 2020). PCA bertujuan mengidentifikasi pola dan struktur mendasar dalam data berdimensi tinggi dengan mentransformasikannya ke ruang berdimensi lebih rendah, sambil mempertahankan informasi penting.

Proses PCA bekerja dengan menghilangkan korelasi antara variabel independen melalui transformasi ke variabel baru yang saling bebas, yang disebut sebagai *principal component*. Metode ini merupakan pengembangan dari *Karhunen-Loève Transform (KLT)*, yaitu metode transformasi linier yang digunakan dalam pengolahan citra. Penentuan jumlah *principal component (PC)* biasanya didasarkan pada *cumulative explained ratio*, di mana jumlah komponen yang dipertahankan harus mencakup minimal 95% dari total varians (Jolliffe 2018). Pada penelitian oleh (Almais et al. 2023) penentuan jumlah PC mempertimbangkan nilai eigen dan rasio variansnya. Dalam penelitian yang dilakukan oleh (Purnama

2019), terdapat beberapa metode yang dapat digunakan untuk menentukan komponen utama dalam PCA. Salah satunya adalah dengan memilih komponen yang memiliki nilai eigen lebih dari 1.

Beberapa langkah yang terlibat dalam reduksi dimensi menggunakan PCA menurut (Sirait et al. 2019) adalah sebagai berikut:

1. Data disusun sebagai satu set vektor data X_1 hingga X_n , di mana matriks input untuk PCA adalah $X(i, j)$ untuk proses pelatihan, dengan i menunjukkan baris dan j menunjukkan kolom.
2. Hitung nilai rata-rata (*mean*) dari data menggunakan rumus:

$$\bar{X} = \sum_{i=1}^n \frac{1}{X_i} \quad (2.10)$$

Keterangan:

n : jumlah sampel atau data observasi,
 X_i : data observasi.

3. Hitung matriks kovarian dari data menggunakan rumus:

$$C(x) = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X}) \quad (2.11)$$

Keterangan:

X_i : nilai rata-rata

4. Hitung nilai eigen (v_m) dan vektor eigen (λ_m) menggunakan rumus:

$$C_X v_m = \lambda_m v_m \quad (2.12)$$

5. Urutkan nilai eigen dalam urutan menurun.
6. *Principal Component* (PC) merupakan kumpulan vektor eigen yang telah disusun sesuai dengan nilai eigen yang telah diurutkan pada langkah 5.

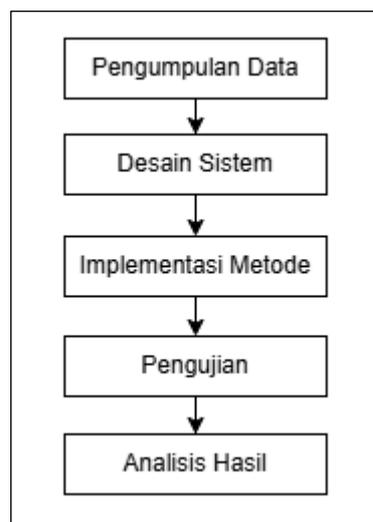
7. Kurangi dimensi *Principal Component* berdasarkan nilai eigen.
8. Pilih beberapa vektor eigen yang memiliki nilai eigen tertinggi.
9. Terakhir, lakukan transformasi data menggunakan vektor eigen yang telah dipilih.

BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian merupakan pedoman dalam perancangan penelitian. Dalam penelitian ini, serangkaian langkah dilakukan secara berurutan untuk memudahkan peneliti dalam menentukan langkah berikutnya, sehingga hasil yang diharapkan dapat tercapai.



Gambar 3. 1 Desain Penelitian

Berdasarkan desain penelitian pada Gambar 3.1. Desain penelitian ini terdiri dari beberapa tahapan. Tahap pertama adalah pengumpulan data, yaitu proses memperoleh data yang relevan dan sesuai kebutuhan penelitian. Selanjutnya desain sistem, yaitu perancangan alur kerja dan pemilihan metode yang akan digunakan untuk mengolah data. Setelah desain sistem ditentukan, dilakukan implementasi metode, yaitu penerapan algoritma yang sudah dipilih. Tahap berikutnya yaitu pengujian, dimana model yang sudah dibangun diuji menggunakan data uji untuk

mengukur kinerja model. Tahap terakhir yaitu analisis hasil, yaitu mengevaluasi performa model berdasarkan matrix evaluasi untuk mengetahui seberapa baik model dalam melakukan klasifikasi.

3.2 Pengumpulan Data

Pada penelitian ini peneliti menggunakan dataset “*Chronic Kidney Disease*” yang diperoleh dari website UCI Machine Learning. Dataset ini diunggah pada tahun 2015 dan berisi 400 data dengan 25 atribut. Data tersebut terdiri dari 250 pasien yang menderita penyakit ginjal kronis, dan 150 pasien yang tidak menderita penyakit ginjal kronis. Selain data dari UCI Machine Learning, penulis juga mencoba melatih model dengan menambahkan data sejenis dari dataset dengan judul “*Kidney Disease Dataset*” yang diperoleh dari platform Kaggle. Penjelasan lebih lanjut mengenai dataset “*Chronic Kidney Disease*” disajikan pada Tabel 3.1.

Tabel 3. 1 Deskripsi Atribut Dataset *Chronic Kidney Disease*

No	Atribut	Deskripsi	Nilai
1	Age	Usia pasien saat data diambil	2 – 90, tahun
2	Bp (Blood pressure)	Tekanan darah pasien	50 – 180, mm/Hg
3	Sg (Specific gravity)	Konsentrasi partikel urine	1.005 - 1.025
4	Al (Albumin)	Kadar albumin dalam urine	0 - 5
5	Su (sugar)	Kadar gula dalam urine	0 - 5
6	Rbc (Red blood cells)	Status sel darah dalam urine	normal/abnormal
7	Pc (Pus cells)	Status sel merah dalam urine	normal/abnormal
8	Pcc (Pus cell clumps)	Indikator adanya gumpalan sel nanah dalam urine	present/not present
9	Ba (Bacteria)	Indikator keberadaan bakteri dalam urine	present/not present
10	Bgr (Blood glucose random)	Kadar glukosa darah acak	22 – 490, mgs/dl
11	Bu (Blood urea)	Kadar urea dalam darah	1.05 – 391, mgs/dl
12	Sc (Serum creatinine)	Kadar kreatinin dalam darah	0.04 – 76, mgs/dl
13	Sod (Sodium)	Kadar natrium dalam darah	4.05 – 163, mEq/L
14	Pot (Potassium)	Kadar kalium dalam darah	2.05 – 47, mEq/L
15	Hemo (Hemoglobin)	Kadar hemoglobin dalam darah	3.01 - 17.08, gms
16	Pcv (Packed cell volume)	Presentase volume darah yang terdiri dari sel darah merah	9 – 54
17	Wbcc (White blood cell count)	Kadar sel darah putih dalam darah	2200 – 26400, cells/cumcn

18	Rbcc (Red blood cell count)	Kadar sel darah merah dalam darah	2.01 – 8.00, millions/cmm
19	Htn (Hypertensi)	Indikator adanya hipertensi	yes/no
20	Dm (Diabetes mellitus)	Indicator adanya diabetes mellitus	yes/no
21	Cad (Coronary artery disease)	Indicator adanya penyakit arteri koroner	yes/no
22	Appet (Appetite)	Status nafsu makan pasien	good, poor
23	Pe (Pedal adema)	Indikator adanya pembengkakan pada kaki	yes/no
24	Ane (Anemia)	Indicator adanya enemia	yes/no
25	Class ()	klasifikasi apakah pasien menderita penyakit gagal ginjal kronis	ckd/not ckd

Pada Tabel 3.1 dapat diketahui bahwa dataset “*Chronic Kidney Disease*” memiliki 25 atribut, dengan 24 atribut sebagai fitur dan 1 atribut sebagai class. Secara keseluruhan, dari 25 atribut tersebut, terdapat 13 atribut bertipe kategorik dan 12 atribut bertipe numerik. Contoh sampel dataset yang digunakan disajikan pada Tabel 3.2.

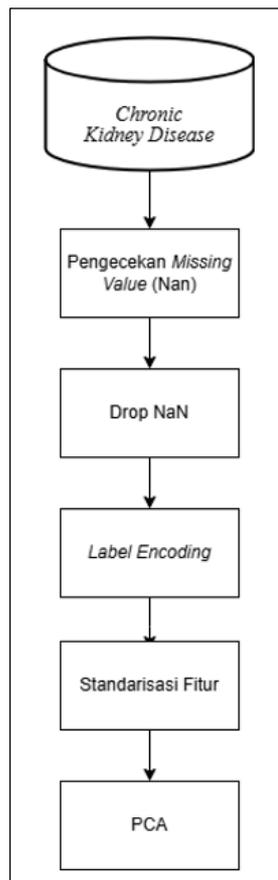
Peneliti juga melakukan percobaan dengan dataset yang ditambah menjadi 300 data, dengan tetap menggunakan struktur fitur yang sama seperti dataset awal dari UCI Machine Learning. Dengan demikian, meskipun jumlah datanya bertambah, jenis dan jumlah fitur yang digunakan dalam pelatihan model tetap konsisten.

Tabel 3. 2 Contoh Sampel Dataset *Chronic Kidney Disease*

No	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wbcc	rbcc	htn	dm	cad	appet	pe	ane	class
1	48	80	1.020	1	0	?	normal	notpresent	notpresent	121	36	01.02	?	?	15.04	44	7800	05.02	yes	yes	no	good	no	no	ckd
2	7	50	1.020	4	0	?	normal	notpresent	notpresent	?	18	00.08	?	?	11.03	38	6000	?	no	no	no	good	no	no	ckd
3	62	80	1.010	2	3	normal	normal	notpresent	notpresent	423	53	01.08	?	?	09.06	31	7500	?	no	yes	no	poor	no	yes	ckd
4	48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56	03.08	111	02.05	11.02	32	6700	03.09	yes	no	no	poor	yes	yes	ckd
5	51	80	1.010	2	0	normal	normal	notpresent	notpresent	106	26	01.04	?	?	11.06	35	7300	04.06	no	no	no	good	no	no	ckd
6	60	90	1.015	3	0	?	?	notpresent	notpresent	74	25	01.01	142	03.02	12.02	39	7800	04.04	yes	yes	no	good	yes	no	ckd
7	68	70	1.010	0	0	?	normal	notpresent	notpresent	100	54	24.00.00	104	04.00	12.04	36	?	?	no	no	no	good	no	no	ckd
8	24	?	1.015	2	4	normal	abnormal	notpresent	notpresent	410	31	01.01	?	?	12.04	44	6900	5	no	yes	no	good	yes	no	ckd
9	52	100	1.015	3	0	normal	abnormal	present	notpresent	138	60	01.09	?	?	10.08	33	9600	4	yes	yes	no	good	no	yes	ckd
10	53	90	01.02	2	0	abnormal	abnormal	present	notpresent	70	107	07.02	114	03.07	09.05	29	12100	03.07	yes	yes	no	poor	no	yes	ckd

3.3 Desain Sistem

Desain sistem merupakan tahapan yang digunakan oleh peneliti dalam melakukan penelitian. Gambar 3.2 menggambarkan desain sistem yang digunakan dalam penelitian ini.



Gambar 3. 2 Desain Sistem

Gambar diatas menggambarkan alur sistem, tahapan diawali dengan penggunaan dataset "*Chronic Kidney Disease*". Selanjutnya data melalui tahapan *preprocessing* dan reduksi dimensi dengan PCA. Setelah itu data dibagi menjadi data latih dan data uji, dimana data uji ini diseimbangkan dengan SMOTE sebelum dilatih dengan model dengan algoritma SVM. Terakhir, model menghasilkan hasil

klasifikasi yang menunjukkan apakah seseorang terindikasi penyakit gagal ginjal kronis atau tidak.

3.4 *Preprocessing Data*

Preprocessing data adalah Langkah-langkah yang dilakukan untuk mengolah data mentah menjadi data berkualitas yang sesuai dengan kebutuhan (Syahrani, Suhartono, and Zaman 2023). Tujuan dari tahapan *preprocessing* data ini adalah agar data yang akan diproses berada dalam keadaan bersih dan siap dianalisis.

Tahapan *preprocessing* ini meliputi pengecekan *missing value*, penanganan NaN, *label encoding*, standarisasi fitur, dan PCA untuk mereduksi dimensi data. Dengan *preprocessing* ini, data menjadi lebih bersih dan efisien untuk digunakan dalam proses klasifikasi.

3.4.1 Pengecekan *Missing Value*

Pengecekan *missing value* adalah proses identifikasi keberadaan nilai yang hilang (NaN atau null) dalam dataset. Proses ini dilakukan untuk mengetahui kolom atau fitur mana yang memiliki nilai yang hilang dan seberapa banyak nilai yang hilang dalam dataset. *Missing value* ini nantinya bisa mempengaruhi hasil analisis dan performa model jika tidak ditangani.

3.4.2 Penanganan *Missing Value*

Setelah dilakukan pengecekan, ditemukan adanya sejumlah nilai yang hilang dalam dataset. Dalam penelitian ini, penanganan dilakukan dengan cara menghapus baris data yang mengandung nilai NaN (*drop NaN*). Pendekatan ini dipilih untuk menjaga keaslian data.

3.4.3 Label Encoding

Label encoding adalah mengubah data kategori menjadi angka sehingga dapat dengan mudah dipahami dan diproses oleh model. Setiap kategori unik dalam kolom-kolom ini diberikan angka yang berbeda untuk memudahkan interpretasi oleh model. Data kategori adalah data yang terdiri dari nilai-nilai yang berbentuk label. Sementara itu, data numerik adalah data yang berisi nilai-nilai dalam bentuk angka.

Untuk menunjukkan perbedaan antara data sebelum dan sesudah *label encoding*, peneliti akan menampilkan 5 sampel data dari dataset yang digunakan dalam penelitian ini. Data ini terdiri dari 5 atribut, yaitu age, bp, sg, al, su, rbc, dan pc. Tabel 3.3 contoh menunjukkan data sebelum *label encoding*.

Tabel 3. 3 Contoh Data Sebelum *Label Encoding*

age	bp	sg	al	su	rbc	pc
48	80	1.020	1	0	?	normal
7	50	1.020	4	0	?	normal
62	80	1.010	2	3	normal	normal
48	70	1.005	4	0	normal	abnormal
51	80	1.010	2	0	normal	abnormal

Setelah melakukan *label encoding* pada atribut tersebut, peneliti akan menampilkan data yang telah diubah dalam bentuk tabel. Data ini berisi 5 sampel data dari dataset yang digunakan dalam penelitian ini. Tabel 3.4 menunjukkan contoh data yang telah diubah.

Tabel 3. 4 Contoh Data Setelah *Label Encoding*

age	bp	sg	al	su	rbc	pc
48	80	1.020	1	0	?	0
7	50	1.020	4	0	?	0
62	80	1.010	2	3	0	0
48	70	1.005	4	0	0	1
51	80	1.010	2	0	0	1

Setelah dilakukan *label encoding*, data kategorikal pada atribut *rbc* dan *pc* yang semula berisi nilai ‘normal’ dan ‘abnormal’ diubah menjadi format numerik. Nilai ‘normal’ dikonversi menjadi 0, sedangkan nilai ‘abnormal’ menjadi 1. Perubahan ini ditampilkan pada Tabel 3.4, yang berisi 5 sampel data dari dataset penelitian. Proses *label encoding* ini bertujuan agar data dapat diproses lebih mudah oleh algoritma klasifikasi.

3.4.4 Standarisasi Fitur

Standarisasi Fitur adalah proses perubahan skala data, dimana data akan diubah agar mempunyai rata-rata = 0 dan standar deviasi = 1. Tujuan dari standarisasi adalah untuk memastikan bahwa setiap fitur dalam dataset memiliki skala yang seragam, dengan rata-rata nol dan simpangan baku satu. Dengan standarisasi fitur, algoritma *machine learning* yang sensitif terhadap skala data dapat bekerja lebih optimal. Metode ini tidak hanya meningkatkan kinerja model, tetapi juga mencegah bias dalam analisis, memastikan semua fitur memiliki pengaruh yang seimbang dalam proses pembelajaran mesin.

Proses standarisasi menggunakan rumus *Z-score* yang dihitung dengan rumus:

$$Z = \frac{X - \mu}{\sigma} \quad (3.1)$$

Keterangan:

X : nilai data individu yang akan distandarisasi,

μ : rata-rata (mean) dari data

σ : standar deviasi dari data

Tabel 3. 5 Contoh Data Sebelum di Standarisasi

age	bp	sg	al	su	rbc	pc
62	80	1.010	2	3	0	0

48	70	1.005	4	0	0	1
51	80	1.010	2	0	0	0
52	100	1.015	3	0	0	1
53	90	1.020	2	0	1	1

Contoh perhitungan manual standarisasi data pada fitur age pada Tabel 3.5.

1. Menentukan nilai kolom age, yaitu 62, 48, 51, 52, 53.
2. Menghitung rata-rata.

$$\mu = \frac{62 + 48 + 51 + 52 + 53}{5} = \frac{266}{5} = 53,2$$

3. Menghitung standar deviasi.

$$\sigma = \sqrt{\frac{(62 - 53,2)^2 + (48 - 53,2)^2 + (51 - 53,2)^2 + (52 - 53,2)^2 + (53 - 53,2)^2}{5}}$$

$$\sigma = \sqrt{\frac{110,8}{5}} = \sqrt{22,6} \approx 4,71$$

4. Menhitung *Z-Score* untuk setiap nilai dengan rumus 3.1.

Misal, *Z-Score* untuk nilai 62:

$$Z_{(62)} = \frac{62 - 53,2}{4,71} = \frac{8,8}{4,71} \approx 1,87$$

Berikut adalah tabel setelah standarisasi fitur.

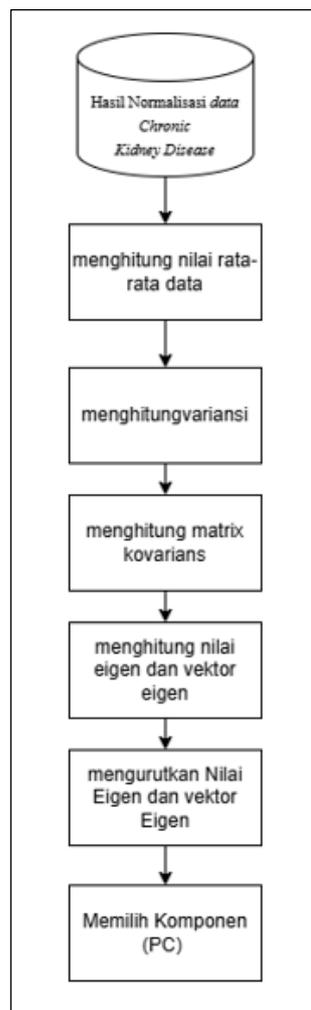
Tabel 3. 6 Contoh Data Setelah di Standarisasi

age	bp	sg	al	su	rbc	pc
1,87	-0,39	-0,39	0,75	2,00	-0,50	-1,22
-1,10	-1,37	-1,37	1,75	-0,50	-0,50	0,82
-0,47	-0,39	-0,39	-0,75	-0,50	-0,50	-1,22
-0,25	1,57	0,59	0,50	-0,50	-0,50	0,82
-0,04	0,59	1,57	-0,75	-0,50	2,00	0,82

Dengan adanya standarisasi ini, kita dapat memastikan bahwa setiap fitur memberikan pengaruh yang setara dalam proses pelatihan model, tanpa ada fitur yang lebih dominan.

3.4.5 *Principal Component Analysis (PCA)*

Principal Component Analysis (PCA) digunakan untuk mengurangi dimensi data tanpa menghilangkan karakteristik penting secara signifikan. Metode ini mengubah variabel-variabel asli yang saling berkorelasi menjadi kumpulan variabel baru yang lebih sedikit dan independen satu sama lain.



Gambar 3. 3 Alur PCA

Gambar 3.3 merupakan alur proses PCA. Proses dimulai dengan hasil dari normalisasi data yang akan digunakan. Kemudian menghitung nilai rata-rata pada setiap fitur dari data hasil normalisasi. Kemudian variansi dan matriks kovarian

dihitung, ini bertujuan untuk memahami hubungan yang ada antar fitur. Dari matriks kovarians ini, dilakukan penghitungan nilai eigen dan vektor eigen. Nilai eigen menunjukkan kontribusi masing-masing komponen terhadap variasi data, sedangkan vektor eigen menunjukkan arah komponen tersebut. Kemudian nilai eigen diurutkan dari terbesar ke terkecil. Dan terakhir komponen utama (*Principal Componen*) dipilih dengan nilai eigen tertinggi. Dalam penelitian ini, komponen dengan nilai eigen ≥ 1 akan dipilih untuk dilatih dalam proses pelatihan model SVM. Komponen dengan nilai eigen ≥ 1 dipilih karena dianggap memberikan kontribusi yang signifikan pada variasi data. Setelah itu, data asli ditransformasi ke bentuk baru berdasarkan komponen utama yang dipilih. Data hasil PCA inilah yang kemudian digunakan untuk melatih model SVM.

3.5 Split Data

Setelah dilakukan *preprocessing* data, langkah selanjutnya adalah pembagian data atau sering disebut dengan “*split data*”. Langkah ini berisi pembagian *dataset* menjadi dua bagian, yaitu data latih dan data uji. Split data merupakan langkah untuk membagi dataset menjadi dua bagian, yaitu data latih dan data uji. Data latih berfungsi sebagai data yang memiliki atribut atau kelas yang digunakan untuk mengenali pola atau karakteristik tertentu dalam proses pelatihan model *machine learning*. Sementara itu, data uji digunakan untuk mengevaluasi akurasi model dengan memanfaatkan data yang sudah memiliki label atau kelas (Kurniawan et al. 2023). Pada tahapan ini peneliti membagi data dengan perbandingan 80% : 20%, 70% : 30%, dan 60% : 40%.

3.6 *Synthetic Minority Oversampling Technique (SMOTE)*

SMOTE (*Synthetic Minority Over-sampling Technique*) adalah proses penanganan masalah ketidakseimbangan kelas pada dataset. Ketika terdapat ketidak seimbangan kelas pada suatu dataset maka hal tersebut bisa menyebabkan model lebih fokus pada kelas mayoritas. SMOTE membantu mengatasi masalah ini dengan cara menghasilkan data sintetis untuk kelas minoritas. pada penelitian ini SMOTE hanya diterapkan pada data training untuk memastikan bahwa model dapat mempelajari pola-pola yang relevan dari kelas minoritas tanpa mempengaruhi distribusi data uji.

3.7 *Implementasi Support Vector Machine*

Langkah pertama dalam penerapan *Support Vector Machine* adalah melatih model dengan data pelatihan yang sudah dibagi sebelumnya. Untuk melatih model *Support Vector Machine*, digunakan metode optimasi seperti *Quadratic Programming (QP)*, yang sebelumnya umum digunakan namun cenderung lambat dan kompleks untuk dataset besar. *Sequential Minimal Optimization (SMO)* adalah peningkatan dari QP yang lebih efisien, karena memecah masalah besar menjadi bagian-bagian kecil yang lebih mudah dioptimalkan, sehingga mempercepat proses pelatihan *Support Vector Machine*.

Secara umum, proses pelatihan model *Support Vector Machine* ini mencakup:

1. Representasi data dalam bentuk vektor dan label kelas.
2. Pembentukan hyperplane pemisah fungsi keputusan, yang sudah dirumuskan pada persamaan 2.1.

3. Optimasi parameter model untuk memperoleh nilai w , b , dan α yang optimal.
4. Penentuan fungsi keputusan akhir untuk mengklasifikasikan data uji.

Berikut adalah contoh perhitungan fungsi keputusan SVM:

Tabel 3. 7 Contoh Data Perhitungan Manual SVM

age	bp	sg	class
48	80	1020	1
70	50	1020	1
62	80	1010	1
48	70	1005	1
40	80	1025	-1
51	80	1010	1
23	80	1025	-1

1. Kita asumsikan setelah optimasi data dari Tabel 3.7 didapatkan support vector dan α nya:

$$x_1 = (48, 80, 1020) \text{ dengan } y_1 = 1 \text{ dan } \alpha_1 = 0.5$$

$$x_5 = (40, 80, 1024) \text{ dengan } y_5 = -1 \text{ dan } \alpha_5 = 0.6$$

$$\text{Bias} = 1$$

2. Fungsi keputusan:

$$f(x_i) = \sum_{i=1}^n \alpha_i y_i (\vec{x}_i \vec{x}_{test}) + b \quad (3.2)$$

Dot product antara vektor-vektor fitur:

Dot Product $X_1 \cdot X_{test}$

$$\begin{aligned} X_1 \cdot X_{test} &= (48 \times 45) + (80 \times 80) + (1020 \times 1022) \\ &= 1049000 \end{aligned}$$

Dot product $X_5 \cdot X_{test}$

$$\begin{aligned} X_5 \cdot X_{test} &= (40 \times 45) + (80 \times 80) + (1024 \times 1022) \\ &= 1050904 \end{aligned}$$

3. Substitusi ke persamaan:

$$f(x_{test}) = (0.5 \times 1 \times 1049000) + (0.6 \times 1 \times 1050904) + 1$$

$$\begin{aligned} f(x_{test}) &= 524500 + (-630542.4) + 1 \\ &= -106041.4 \end{aligned}$$

4. Menggunakan fungsi *sign* sehingga:

$$f(x_{test}) = \text{sign}(-106041.4)$$

Karena $-106041.4 < 0$, maka masuk ke kelas -1, yang berarti data uji masuk ke dalam kategori tidak terkena penyakit gagal ginjal kronis.

3.8 Evaluasi Performa

Tahapan terakhir dalam penelitian ini adalah evaluasi performa model. Proses evaluasi performa merupakan langkah penting untuk menilai seberapa efektif model atau algoritma yang telah dibuat dalam melakukan prediksi terhadap data tertentu. Evaluasi performa ini dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* merupakan metode untuk meninjau hasil implementasi algoritma klasifikasi dalam bentuk tabel (Grandis, Arumsari, and Indriati 2021). Tabel 3.8 merupakan *confusion matrix*.

Tabel 3. 8 *Confusion Matrix*

Kelas	Prediksi	
	Positif	Negatif
Positif	TP	FN
Negatif	FP	TN

True Positif (TP) merupakan metrik yang menunjukkan jumlah data positif yang berhasil diklasifikasikan dengan benar oleh model. Sebaliknya, *False Positif* (FP) adalah metrik yang menghitung jumlah data negatif yang salah diklasifikasikan sebagai positif oleh model. *False Negatif* (FN) adalah metrik yang

menghitung jumlah data positif yang salah diklasifikasikan sebagai negatif, sementara *True Negative* (TN) mengukur jumlah data negatif yang berhasil diklasifikasikan dengan benar oleh model. Evaluasi kinerja model meliputi perhitungan *accuracy*, *precision*, *recall*, dan *f1-score*, yang semuanya didasarkan pada nilai-nilai yang dihasilkan oleh *confusion matrix*.

Akurasi mengukur seberapa sering model bisa memprediksi dengan tepat dari keseluruhan data yang ada. Ini berarti seberapa sering prediksi model cocok dengan kategori data sesungguhnya. Nilai akurasi dihitung dengan rumus 3.3.

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \quad (3.3)$$

Presisi mengukur tingkat keakuratan klasifikasi pada kelas positif, yaitu sejauh mana model berhasil mengklasifikasikan data sebagai positif dan memang benar-benar positif. Presisi dihitung dengan membandingkan jumlah klasifikasi positif yang tepat dengan jumlah total hasil klasifikasi positif. Presisi membantu untuk mengetahui seberapa banyak data yang diklasifikasikan sebagai positif dan memang benar-benar positif pada kenyataannya. Nilai presisi dihitung dengan rumus 3.4.

$$Presisi = \frac{TP}{TP + FP} \quad (3.4)$$

Recall adalah metrik yang mengukur sejauh mana model mampu mengidentifikasi seluruh data sesungguhnya positif dari keseluruhan data positif yang ada. *Recall* dihitung dengan membandingkan jumlah identifikasi positif yang akurat dengan keseluruhan data yang sebenarnya positif. Nilai *recall* dihitung dengan rumus 3.5.

$$Recall = \frac{TP}{TP + FN} \quad (3.5)$$

F1-score adalah metrik yang mencerminkan keseimbangan data antara presisi dan *recall*. Nilai *f1-score* dihitung dengan rumus 3.6.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3.6)$$

Evaluasi ini tidak hanya bertujuan untuk mengukur tingkat akurasi prediksi, tetapi juga memberikan wawasan yang lebih mendalam mengenai pola kesalahan dalam klasifikasi. Hal ini memungkinkan pengoptimalan model untuk meningkatkan performa dalam pengklasifikasian data di masa mendatang (Mellina, Suhartono, and Yaqin 2024).

3.9 Skenario Uji Coba

Skenario uji coba adalah serangkaian langkah uji yang dilakukan untuk mengevaluasi sistem yang sudah dikembangkan. Pada penelitian ini, dataset yang digunakan dibagi menjadi tiga skenario berbeda untuk data latih dan data uji. Pembagian ini bertujuan untuk mengidentifikasi skenario mana yang menghasilkan nilai akurasi terbaik. Jumlah data yang digunakan pada setiap skenario pembagian dataset disajikan pada Tabel 3.9.

Tabel 3. 9 Skenario Pengujian pembagian data

Skenario	Presentase Data Latih	Presentase Data Uji
1	80%	20%
2	70%	30%
3	60%	40%

Selain skenario pembagian dataset, terdapat juga sub-skenario pengujian untuk mengevaluasi performa model *Support Vector Machine* (SVM) dengan dan tanpa *Principal Component Analysis* (PCA).

Tabel 3. 10 Sub-skenario

Sub-skenario	Model	Kernel
1	SVM tanpa PCA	Kernel Linear
		Kernel Polynomial
	SVM dengan PCA	Kernel Linear
		Kernel Polynomial
2	SVM tanpa PCA	Kernel Linear
		Kernel Polynomial
	SVM dengan PCA	Kernel Linear
		Kernel Polynomial
3	SVM tanpa PCA	Kernel Linear
		Kernel Polynomial
	SVM dengan PCA	Kernel Linear
		Kernel Polynomial

Dengan adanya sub-skenario ini, penelitian dapat mengevaluasi bagaimana PCA memengaruhi performa SVM dalam berbagai kondisi kernel dan skenario pembagian data.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil *Preprocessing* Data

Data terlebih dahulu dilakukan *preprocessing* agar data siap digunakan dalam proses pelatihan model.

4.1.1 Pengecekan *Missing Value*

Pengecekan *missing value* dilakukan untuk mengetahui kolom atau fitur mana yang memiliki nilai yang hilang. Setelah dilakukan pengecekan, terdapat 242 nilai yang hilang pada dataset. Pengecekan *missing value* menggunakan *library* 'pandas', dengan fungsi 'isnull()'. Jumlah masing-masing data yang hilang pada setiap kolom disajikan pada Tabel 4.1.

Tabel 4. 1 Nilai Nan pada setiap kolom

No	Nama Kolom	Jumlah <i>Missing Value</i>
1	Age	9
2	Bp	12
3	Sg	47
4	Al	46
5	Su	49
6	Rbc	152
7	Pc	65
8	Pcc	4
9	Ba	4
10	bgr	44
11	Bu	19
12	Sc	17
13	Sod	87
14	Pot	88
15	Hemo	52
16	Pcv	71
17	Wbcc	106
18	Rbcc	131
19	Htn	2
20	Dm	2
21	Cad	2
22	Appet	1

23	Pe	1
24	Anc	1
25	Class	0

Pada Tabel 4.1 diatas kolom Rbc memiliki jumlah Nan terbanyak, yaitu 152 data, sementara kolom lainnya memiliki jumlah yang bervariasi. Pengecekan ini membantu peneliti untuk memahami sejauh mana data memiliki nilai yang hilang.

4.1.2 Penanganan *Missing Value*

Pada tahap sebelumnya telah diidentifikasi jumlah nilai yang hilang pada dataset. peneliti memutuskan untuk menghapus seluruh baris data yang memiliki setidaknya satu nilai NaN (*drop NaN*). Langkah ini diambil guna menjaga keaslian data, serta untuk menghindari potensi bias atau distorsi yang dapat memengaruhi hasil analisis dan performa model. Setelah proses penghapusan dilakukan, hanya data yang lengkap dan bersih yang digunakan dalam proses analisis dan pemodelan selanjutnya.

Pada awalnya data berjumlah 400, setelah dilakukan penghapusan NaN jumlah data yang tersisa yaitu 158 data dengan fitur tetap seperti awal yaitu 25 fitur. Penghapusan nilai yang hilang menggunakan *library* `pandas` dengan fungsi `dropna(inplace=True)`.

4.1.3 *Label Encoding*

Label encoding dilakukan dengan mengubah tipe data kategorik menjadi numerik, sehingga bisa dipahami oleh mesin. Proses ini menggunakan *library* `scikit-learn` dengan fungsi `LabelEncoder()`. Tabel 4.2 merupakan data sebelum dilakukan *labeling*.

Tabel 4. 2 Data Sebelum *Encoding*

no	age	bp	sg	al	su	rbc	pc	...	class
0	48	70	1.005	4	0	normal	abnormal	...	ckd
1	53	90	1.020	2	0	abnormal	abnormal	...	ckd
2	63	70	1.010	3	0	abnormal	abnormal	...	ckd
3	68	80	1.010	3	2	normal	abnormal	...	ckd
4	61	80	1.015	2	0	abnormal	abnormal	...	ckd

Pada tabel tersebut, terdapat beberapa nilai yang bertipe kategorik, seperti pada kolom rbc dan class. Dimana tipe data harus diubah menjadi numerik agar bisa diproses oleh model.

Tabel 4. 3 Data Setelah *Encoding*

No	age	bp	sg	al	su	rbc	pc	...	class
0	48	70	1.005	4	0	1	0	...	0
1	53	90	01.02	2	0	0	0	...	0
2	63	70	01.01	3	0	0	0	...	0
3	68	80	01.01	3	2	1	0	...	0
4	61	80	1.015	2	0	0	0	...	0

Tabel 4.3 merupakan hasil setelah dilakukan *labeling*, di mana fitur rbc, pc, dan class, serta beberapa fitur kategorikal lainnya yang sebelumnya berisi data dalam bentuk teks telah diubah menjadi data numerik menggunakan *'LabelEncoder'*. Dalam proses ini, setiap kategori diberikan nilai numerik yang telah ditentukan untuk memastikan data dapat diproses oleh model *Support Vector Machine* (SVM).

Sebagai contoh, pada fitur rbc, nilai 'normal' diubah menjadi 1, dan 'abnormal' diubah menjadi 0. Sementara itu, pada fitur class, kategori 'ckd' dikonversi menjadi 0, dan 'notckd' menjadi 1. Teknik ini memungkinkan model untuk memahami data dengan lebih baik tanpa kehilangan informasi dari fitur kategorikal.

4.1.4 Standarisasi Fitur

Standarisasi fitur digunakan untuk memastikan fitur yang ada pada dataset memiliki skala yang seragam. Dengan standarisasi fitur, algoritma *machine learning* yang sensitif terhadap skala data dapat bekerja lebih optimal. Pada penelitian ini peneliti menggunakan metode *Z-Score*. Proses ini menggunakan library `scikit-learn` dengan fungsi `StandardScaler()`.

Tabel 4. 4 Data Sebelum Standarisasi

No	age	bp	sg	al	su	rbc	pc	...	class
0	48	70	1.005	4	0	1	0	...	0
1	53	90	01.02	2	0	0	0	...	0
2	63	70	01.01	3	0	0	0	...	0
3	68	80	01.01	3	2	1	0	...	0
4	61	80	1.015	2	0	0	0	...	0

Tabel 4.4 menunjukkan data sebelum normalisasi. Pada tabel tersebut, dapat diketahui bahwa setiap fitur mempunyai rentang nilai yang berbeda-beda. Sebagai contoh, pada fitur 'age' memiliki rentang nilai puluhan, sementara fitur 'sg' berada dalam skala satuan yang kecil. Perbedaan skala antar fitur ini bisa mempengaruhi kinerja model *machine learning* Oleh karena itu diperlukan standarisasi untuk menyamakan skala antar fitur.

Tabel 4. 5 Data Setelah Standarisasi

no	age	bp	sg	al	su	rbc	...	class
0	-0.101098	-0.363613	-2.713365	22.734.74	-0.312233	0.358569	...	0
1	0.222253	14.317.25	0.023092	0.853676	-0.312233	-2.788867	...	0
2	0.868954	-0.363613	-18.012.12	1.563575	-0.312233	-2.788867	...	0
3	11.923.05	0.534056	-18.012.12	1.563575	2.154410	0.358569	...	0
4	0.739614	0.534056	-0.889060	0.853676	-0.312233	2.788867	...	0

Tabel 4.5 merupakan data setelah distandarisasi menggunakan metode *Z-Score*. Pada tabel tersebut dapat diketahui bahwa nilai pada setiap fitur telah mengalami perubahan dalam skala sehingga memiliki rata-rata 0 dan standar deviasi 1.

4.1.5 PCA

Peneliti menggunakan PCA untuk mereduksi dimensi dataset yang digunakan. Proses PCA menggunakan *library* `scikit-learn` dengan fungsi `PCA()`. Hasil dari analisis PCA terdapat 24 *eigenvalue*, dimana masing-masing *eigenvalue* ini mewakili variasi yang dijelaskan oleh komponen utama.

Tabel 4. 6 Nilai *Eigenvalue* dan *Variance Ratio*

No	PC	<i>Eigenvalue</i>	<i>Variance ratio</i>
1	PC 1	11.07615855	45.86%
2	PC 2	1.91821643	7.94%
3	PC 3	1.47246383	6.10%
4	PC 4	1.18402932	4.90%
5	PC 5	1.06444593	4.41%
6	PC 6	0.97284877	4.03%
7	PC 7	0.871354	3.61%
8	PC 8	0.73806078	3.06%
9	PC 9	0.69782403	2.89%
10	PC 10	0.60291467	2.50%
11	PC 11	0.51350508	2.13%
12	PC 12	0.43158056	1.79%
13	PC 13	0.40556408	1.68%
14	PC 14	0.36513337	1.51%
15	PC 15	0.329445	1.36%
16	PC 16	0.28884274	1.20%
17	PC 17	0.24902995	1.03%
18	PC 18	0.20704306	0.86%
19	PC 19	0.16900436	0.70%
20	PC 20	0.1643774	0.68%
21	PC 21	0.13146337	0.54%
22	PC 22	0.12140592	0.50%
23	PC 23	0.09642392	0.40%
24	PC 24	0.08173111	0.34%

Tabel 4.6 menunjukkan hasil nilai *eigenvalue* dan *variance ratio* dari setiap komponen. Pada penelitian ini hanya komponen dengan nilai eigen >1 yang dipilih untuk melatih model. Sehingga terdapat 5 komponen yang akan dipilih yaitu PC1, PC2, PC3, PC4, dan PC5. Lima komponen inilah yang akan digunakan untuk melatih model SVM.

4.2 Hasil Split Data

Proses pembagian data menghasilkan dua bagian, yaitu data latih dan data test. Berikut adalah hasil pembagian data dengan berbagai rasio yang berbeda. Pembagian data menggunakan fungsi `train_test_split()` dari *library* `sckit-learn`.

Tabel 4. 7 Hasil Split Data

Rasio	Jumlah Data Latih	Jumlah Data Test
80:20	126	32
70:30	110	48
60:40	94	64

Tabel 4.7 menunjukkan hasil pembagian data menjadi data latih dan data uji berdasarkan tiga skenario rasio yang berbeda. ada skenario 80:20, terdapat 126 data latih dan 32 data uji, skenario 70:30 menghasilkan 110 data latih dan 48 data uji, sedangkan skenario 60:40 membagi dataset menjadi 94 data latih dan 64 data uji. Pada data latih, diterapkan SMOTE untuk menyeimbangkan kelas sebelum model dilatih, sementara data uji tetap dalam distribusi aslinya.

Tabel 4. 8 Hasil Split Data Setelah SMOTE

Rasio	Jumlah Data Latih Setelah SMOTE	Jumlah Data Test
80:20	184	32
70:30	160	48
60:40	140	64

Tabel 4.8 menunjukkan jumlah data latih dan data test setelah penerapan SMOTE pada data latih untuk menyeimbangkan kelas. Pada skenario 80:20, data latih meningkat menjadi 184, skenario 70:30 menghasilkan 160 data latih, dan skenario 60:40 menghasilkan 140 data latih, sementara jumlah data uji tetap sama sesuai dengan pembagian awal.

4.3 Hasil Uji Coba Model

Pada sub bab ini, penulis akan membahas hasil uji coba sistem deteksi penyakit ginjal menggunakan metode *Support Vector Machine* (SVM) dengan PCA, dan SVM tanpa PCA.

4.3.1 Hasil Uji Coba Skenario 1

Pada skenario 1, pengujian rasio splitting data yang digunakan adalah 80:20, dengan 184 data latih, dan 32 data uji. Berikut ini merupakan tabel hasil evaluasi pengujian skenario 1:

Tabel 4. 9 Matrix Evaluasi Skenario 1

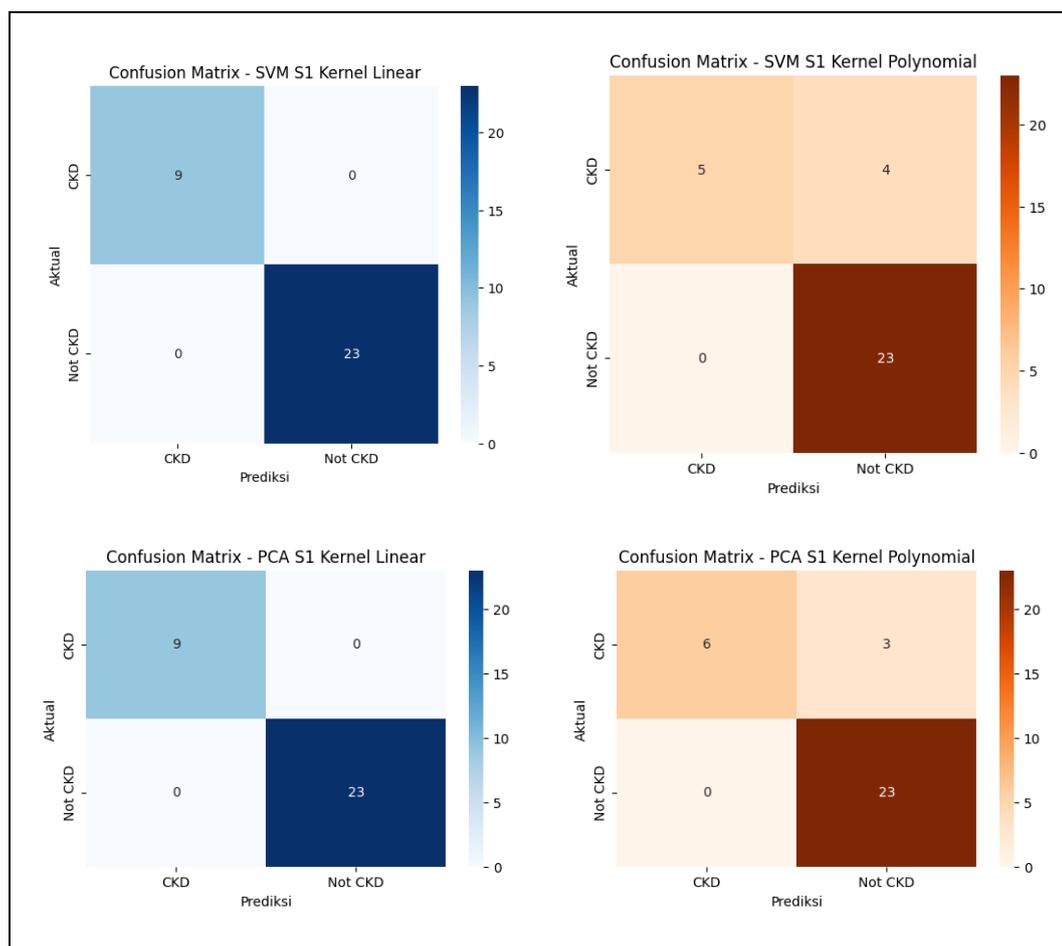
No	Skenario	Model	Kernel	Akurasi	Presisi	Recall	F1-Score
1	1	SVM tanpa PCA	Linear	100%	100%	100%	100%
2			Polynomial	88%	93%	78%	82%
3		SVM PCA	Linear	100%	100%	100%	100%
4			Polynomial	91%	94%	83%	87%

Tabel 4.9 menunjukkan hasil performa pengujian pada Skenario 1. Pada model SVM tanpa PCA dengan kernel linear, diperoleh performa terbaik dengan akurasi 100%, presisi 100%, *recall* 100%, dan *F1-score* 100%. Hal ini menunjukkan bahwa model mampu mengenali seluruh data uji dengan benar. Namun, ketika menggunakan kernel polynomial, performa model mengalami penurunan dengan akurasi 88%, presisi 93%, *recall* 78%, dan *F1-score* 82%. Penurunan ini menunjukkan bahwa kernel polynomial kurang optimal dalam menangani pola data dibandingkan dengan kernel linear.

Sementara itu, pada model SVM dengan PCA, hasil yang diperoleh menunjukkan pola yang sama. Dengan kernel linear, model kembali mencapai

performa terbaik, yaitu akurasi 100%, presisi 100%, *recall* 100%, dan *F1-score* 100%. Hal ini menunjukkan bahwa proses reduksi dimensi melalui PCA tidak menurunkan performa model. Sedangkan saat menggunakan kernel polynomial, performa model mengalami peningkatan sebesar 3% dalam akurasi (dari 88% menjadi 91%), 1% pada presisi (dari 93% menjadi 94%), 5% pada *recall* (dari 78% menjadi 83%), dan 5% pada *F1-score* (dari 82% menjadi 87%) dibandingkan dengan SVM tanpa PCA. Peningkatan ini berarti PCA mampu membantu kinerja kernel polynomial, meskipun performanya tetap lebih rendah dibandingkan kernel linear dalam hal akurasi.

Berikut *confusion matrix* kedua model pada skenario 1:



Gambar 4. 1 *Confusion Matrix* Skenario 1

Gambar 4.1 menunjukkan confusion matrix untuk skenario 1. Pada model SVM tanpa PCA, kernel linear mengklasifikasikan seluruh data dengan benar, menghasilkan akurasi 100%. Sebaliknya, kernel polynomial salah mengklasifikasikan 4 dari 9 data CKD sebagai Not CKD, menurunkan akurasi menjadi 88%. Ini menunjukkan performa yang kurang optimal. Dengan penerapan PCA, akurasi kernel linear tetap 100%, menandakan bahwa PCA tidak mengurangi kinerja model. Sementara itu, kernel polynomial salah mengklasifikasikan 3 data CKD, sehingga akurasi meningkat menjadi 91%. Hal ini mengindikasikan bahwa PCA membantu menyederhanakan fitur sehingga model lebih efektif mengenali pola.

4.3.2 Hasil Uji Coba Skenario 2

Pada skenario 2, pengujian rasio splitting data yang digunakan adalah 70:30, dengan 160 data latih, dan 48 data uji. Berikut ini merupakan tabel hasil evaluasi pengujian skenario 2:

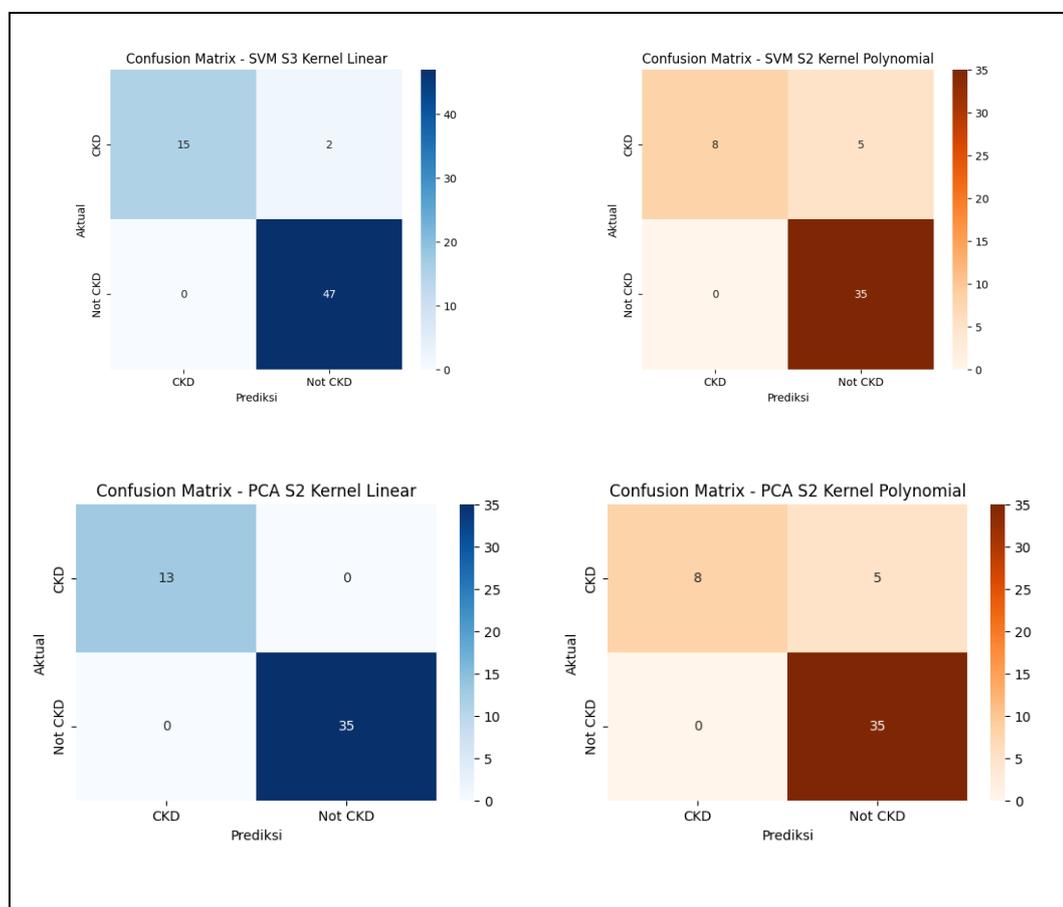
Tabel 4. 10 Matrix Evaluasi Skenario 2

No	Skenario	Model	Kernel	Akurasi	Presisi	Recall	F1-Score
1	2	SVM tanpa PCA	Linear	96%	97%	92%	94%
2			Polynomial	90%	94%	81%	85%
3		SVM PCA	Linear	100%	100%	100%	100%
4			Polynomial	90%	94%	81%	85%

Tabel 4.10 menunjukkan hasil evaluasi performa pengujian pada skenario 2. Pada model SVM tanpa PCA, hasil pengujian menggunakan kernel linear memperoleh akurasi sebesar 96%, presisi 97%, recall 92%, dan F1-score 94%. Sementara pada kernel polynomial, diperoleh akurasi 90%, presisi 94%, recall

81%, dan $F1$ -score 85%. Nilai akurasi dan $F1$ -score pada kernel linear lebih unggul daripada kernel polynomial. Selanjutnya, pada model SVM dengan PCA, hasil pengujian menggunakan kernel linear menunjukkan akurasi 100%, presisi 100%, $recall$ 100%, dan $F1$ -score 100%. Sedangkan pada kernel polynomial, diperoleh akurasi 90%, presisi 94%, $recall$ 81%, dan $F1$ -score 85%. Sama seperti pada skenario sebelumnya, kernel linear memberikan hasil tertinggi.

Berikut *confusion matrix* kedua model pada skenario 2.



Gambar 4. 2 *Confusion Matrix* Skenario 2

Gambar 4.2 menunjukkan *confusion matrix* pada skenario 2. Pada model SVM tanpa PCA, kernel linear mengalami penurunan performa dibandingkan skenario sebelumnya, dengan 2 data CKD salah diklasifikasi sebagai Not CKD dari

13 data CKD, sementara semua data Not CKD tetap diklasifikasikan dengan benar. Pada kernel polynomial, penurunan lebih jelas terlihat, dengan 5 data CKD salah diklasifikasikan sebagai Not CKD dari 13 data CKD, meskipun semua data Not CKD tetap diklasifikasikan dengan benar.

Pada SVM dengan PCA, kernel linear menunjukkan perbaikan performa dari pada SVM tanpa PCA, dengan semua data berhasil diklasifikasikan dengan benar. Hal ini menunjukkan bahwa PCA mereduksi data dengan tetap mempertahankan pola antar kelas, sehingga model mampu mengklasifikasikan data dengan baik. Namun, pada kernel polynomial, SVM dengan PCA tidak memberikan perubahan performa dibandingkan SVM tanpa PCA, terdapat 5 data CKD yang salah diklasifikasi sebagai Not CKD dari 13 data CKD. Ini menunjukkan bahwa efektivitas PCA bergantung pada karakteristik data dan jenis kernel yang digunakan.

4.3.3 Hasil Uji Coba Skenario 3

Pada skenario 3, pengujian rasio splitting data yang digunakan adalah 60:40, dengan 140 data latih, dan 64 data uji. Berikut ini merupakan tabel hasil evaluasi pengujian skenario 3:

Tabel 4. 11 Matrix Evaluasi Skenario 3

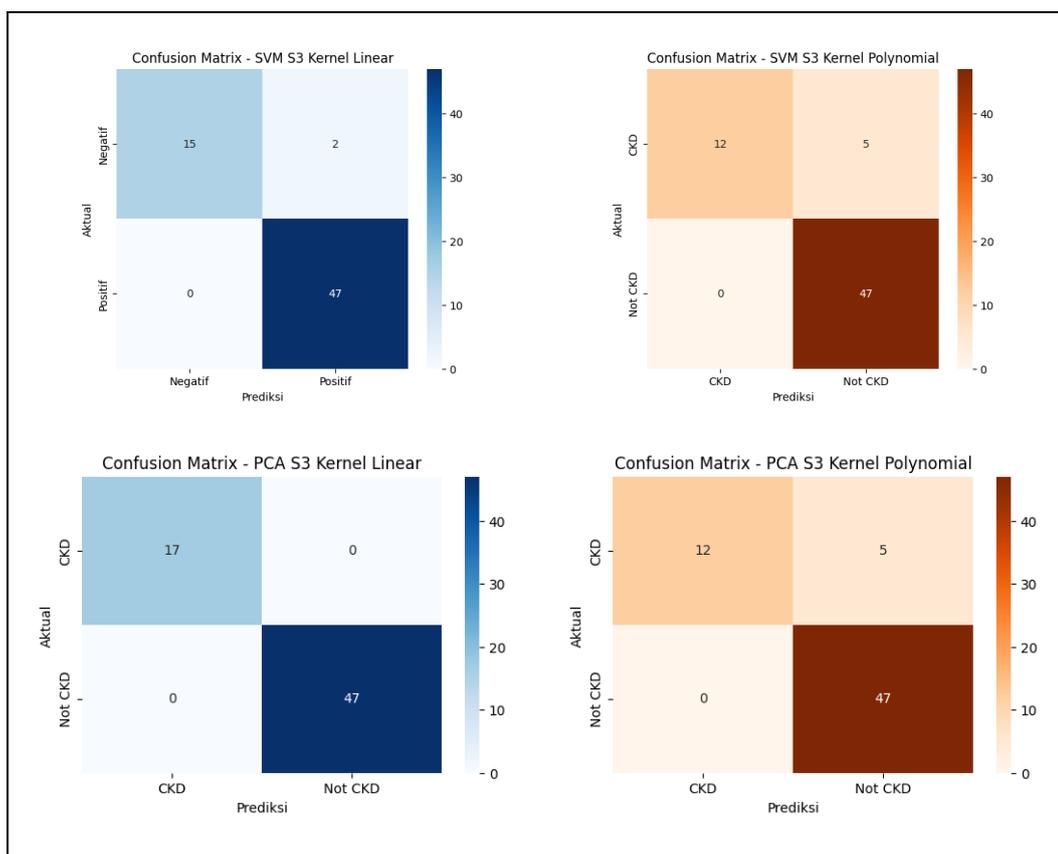
No	Skenario	Model	Kernel	Akurasi	Presisi	Recall	F1-Score
1	3	SVM tanpa PCA	Linear	97%	98%	94%	96%
2			Polynomial	92%	95%	85%	89%
3		SVM PCA	Linear	100%	100%	100%	100%
4			Polynomial	92%	95%	85%	89%

Tabel 4.11 menyajikan hasil evaluasi performa pengujian pada skenario 3. Pada model SVM tanpa PCA, kernel linear menghasilkan akurasi sebesar 97%,

dengan presisi 98%, *recall* 94%, dan *F1-score* 96%. Sementara itu, pada kernel polynomial, akurasi yang diperoleh sebesar 92%, dengan presisi 95%, *recall* 85%, dan *F1-score* 89%. Pada model ini, kernel linear kembali menunjukkan performa yang lebih baik dibandingkan kernel polynomial.

Kemudian, pada model SVM dengan PCA, pengujian menggunakan kernel linear berhasil tetap memperoleh akurasi 100%, presisi 100%, *recall* 100%, dan *F1-score* 100%. Sedangkan untuk kernel polynomial, akurasi yang dicapai sebesar 94%, dengan presisi 96%, *recall* 88%, dan *F1-score* 91%. Sama seperti model tanpa PCA, kernel linear tetap memberikan hasil tertinggi pada skenario ini.

Berikut *confusion matrix* kedua model pada skenario 3.



Gambar 4.3 *Confusion Matrix* Skenario 3

Gambar 4.3 menunjukkan *confusion matrix* pada skenario 3. Pada model SVM tanpa PCA, kernel linear mengalami 2 kesalahan klasifikasi pada data CKD dari total 17 data CKD, sementara semua data Not CKD tetap diklasifikasikan dengan benar, ini menunjukkan sedikit penurunan performa dibandingkan skenario sebelumnya. Pada kernel polynomial, kesalahan klasifikasi pada data CKD tetap yaitu sebanyak 5 dari 17 data CKD, sementara semua data Not CKD tetap diklasifikasikan dengan benar, hal ini menguatkan pola bahwa kernel polynomial kurang efektif dalam memisahkan data yang dipakai.

Pada SVM dengan PCA, sama seperti skenario sebelumnya, kernel linear menunjukkan peningkatan performa dengan semua data berhasil diklasifikasikan dengan benar. Ini menunjukkan bahwa PCA membantu menyederhanakan data sehingga pola antar kelas lebih jelas dan mudah dipisahkan. Namun, pada kernel polynomial, tidak ada perubahan performa dibandingkan SVM tanpa PCA, dengan 5 data CKD masih salah diklasifikasi sebagai Not CKD dari 17 data CKD. Hal ini kembali menegaskan bahwa efektivitas PCA bergantung pada karakteristik data dan jenis kernel yang digunakan.

4.4 Pembahasan

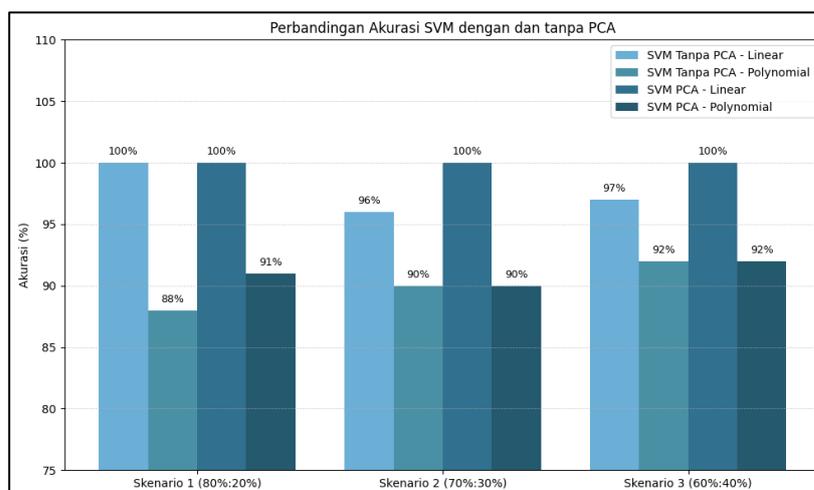
Dalam penelitian ini, penulis menggunakan dataset “Chronic Kidney Disease” yang berasal dari UCI Machine Learning. Data tersebut perlu melalui tahap *preprocessing* sebelum diolah oleh *machine learning*. Tahap *preprocessing* meliputi Pengecekan dan Penghapusan NaN, *label encoding*, dan PCA. Setelah tahapan *preprocessing* peneliti membagi data menjadi data latih dan data uji dengan proporsi sesuai dengan skenario yang sudah ditetapkan. Penelitian ini juga

menerapkan SMOTE untuk menyeimbangkan kelas pada data train. Penulis kemudian melatih model *Support Vector Machine* (SVM) dengan data latih untuk mengklasifikasikan pasien berisiko terkena gagal ginjal kronis dan tidak.

Berdasarkan hasil dari pengujian model, performa SVM akan dievaluasi menggunakan matrix akurasi, presisi, *recall*, dan *F1-Score* dengan berbagai rasio perbandingan data latih dan data uji yang sudah disebutkan sebelumnya, berikut Tabel hasil akurasi, presisi, *recall*, dan *F1-score* untuk setiap skenario.

Tabel 4. 12 Matrix Evaluasi Setiap Skenario

Skenario	Model	Kernel	Akurasi	Presisi	<i>Recall</i>	<i>F1-score</i>
1 (80%:20%)	SVM tanpa PCA	Linear	100%	100%	100%	100%
		Polynomial	88%	93%	78%	82%
	SVM PCA	Linear	100%	100%	100%	100%
		Polynomial	91%	94%	83%	87%
2 (70%:30%)	SVM tanpa PCA	Linear	96%	97%	92%	94%
		Polynomial	90%	94%	81%	85%
	SVM PCA	Linear	100%	100%	100%	100%
		Polynomial	90%	94%	81%	85%
3 (60%:40%)	SVM tanpa PCA	Linear	97%	98%	94%	96%
		Polynomial	92%	95%	85%	89%
	SVM PCA	Linear	100%	100%	100%	100%
		Polynomial	92%	95%	85%	89%



Gambar 4. 4 Grafik Akurasi Setiap Skenario

Pada skenario 1, dataset dibagi menjadi data latih 80% dan data uji 20%. Hasil menunjukkan bahwa pada kedua model (tanpa PCA dan dengan PCA) dengan kernel linear mampu mencapai akurasi 100%. Hal ini berarti data memiliki pemisah yang jelas, sehingga model dengan linear bisa mengklasifikasikan seluruh data dengan sempurna. Namun, pada model tanpa PCA dengan kernel polynomial, akurasi yang diperoleh sebesar 88%. Ini menunjukkan bahwa kernel polynomial kurang baik dalam menangani data pada skenario ini. Sementara itu, model dengan PCA pada kernel polynomial mendapatkan akurasi yang lebih tinggi 3%, yaitu sebesar 91%. Hal ini menunjukkan bahwa reduksi dimensi yang dilakukan oleh PCA dapat membantu model dalam menangkap pola data dengan lebih baik dalam skenario ini.

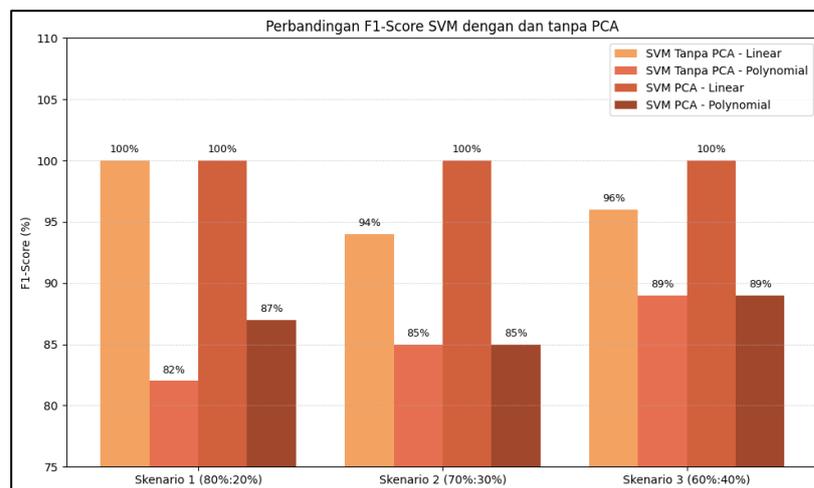
Pada skenario 2, dataset dibagi menjadi data latih 70% dan data uji 30%. Pada skenario ini, jumlah data uji lebih banyak dibandingkan skenario sebelumnya. Dengan data uji yang lebih besar, model memiliki peluang yang lebih baik untuk menguji kemampuannya dalam mengenali pola yang lebih luas. Pada model tanpa PCA, akurasi kernel linear menurun dari 100% menjadi 96%. Penurunan ini terjadi karena dengan jumlah data uji yang lebih banyak, variasi dalam data meningkat, sehingga model linear yang sebelumnya bisa mengklasifikasikan data dengan sempurna pada proporsi 80:20 menjadi kurang tepat dalam mengklasifikasikan beberapa sampel tertentu. Meskipun begitu, akurasi yang diperoleh oleh kernel linear masih menunjukkan nilai yang tinggi, yang menegaskan bahwa data memiliki pola pemisah yang jelas. Sebaliknya, pada model tanpa PCA dengan kernel polynomial, akurasi meningkat menjadi 90%. Ini berarti semakin banyaknya

data uji, model dengan kernel polynomial dapat lebih baik dalam menerapkan pengetahuan yang diperoleh dari data latih ke data baru yang belum pernah dilihat sebelumnya, terutama ketika pola dalam data bersifat kompleks..

Pada model dengan PCA, akurasi kernel linear tetap sempurna di angka 100%. Ini menunjukkan bahwa setelah reduksi dimensi yang dilakukan oleh PCA, model masih mampu mempertahankan informasi data yang cukup untuk tetap mengklasifikasikan data dengan benar. Namun, pada kernel polynomial terjadi penurunan akurasi dari skenario sebelumnya, yaitu dari 91% menjadi 90%. Penurunan ini berarti meskipun model tetap stabil setelah reduksi dimensi, performanya sedikit terpengaruh oleh variasi data uji yang lebih besar.

Pada skenario 3, dataset dibagi menjadi data latih 60% dan data uji 40%. Pada skenario ini, jumlah data uji lebih banyak dibandingkan dua skenario sebelumnya. Hasil menunjukkan bahwa pada model tanpa PCA, kedua kernel mengalami kenaikan akurasi. Kernel linear mengalami peningkatan dari 96% menjadi 97%. Hal ini menunjukkan bahwa ketika data uji bertambah lebih banyak lagi, model mulai beradaptasi dengan lebih banyak variasi data, sehingga mampu mengklasifikasikan dengan lebih baik. Pada kernel polynomial, akurasi juga meningkat seiring bertambahnya data uji, dengan akurasi sebelumnya 90% sekarang menjadi 92%. Ini menunjukkan bahwa semakin banyaknya data uji, model dengan kernel polynomial menunjukkan performa yang lebih stabil dalam mengklasifikasikan data. Hal ini berarti bahwa pemisah yang dibentuk selama training tetap efektif dalam membedakan kelas pada data baru.

Pada model dengan PCA, kernel linear tetap bertahan pada akurasi sempurna, yaitu 100%. Ini menunjukkan bahwa reduksi dimensi yang dilakukan oleh PCA tidak mengurangi informasi yang dibutuhkan oleh model untuk dapat mengklasifikasikan data dengan benar. Sementara itu, pada kernel polinomial, akurasi meningkat 2% setelah sebelumnya mengalami penurunan. Peningkatan ini menunjukkan bahwa setelah reduksi dimensi, semakin banyak data uji dapat membantu model dengan kernel polinomial dalam mempelajari pola yang lebih kompleks dengan lebih stabil.



Gambar 4. 5 Grafik *F1-Score* Setiap Skenario

Gambar 4.5 menampilkan grafik nilai *F1-Score* setiap skenario. *F1-score* memberikan gambaran keseimbangan antara presisi dan *recall*. Pada kernel linear, SVM dengan PCA selalu memiliki *F1-score* 100%, hal ini berarti keseimbangan antara presisi dan *recall* di semua skenario sempurna. Sementara itu, SVM tanpa PCA mengalami sedikit penurunan *F1-score*, dari 100% pada skenario 1 menjadi 94% di skenario 2, sebelum kembali meningkat ke 96% di skenario 3. Penurunan di skenario 2 terjadi akibat turunnya *recall*, meskipun presisi tetap tinggi. Hal ini

menunjukkan bahwa ketika jumlah data uji bertambah, model tanpa PCA mengalami sedikit kesulitan dalam mengklasifikasikan seluruh sampel CKD dengan sempurna.

Pada kernel polynomial, nilai *F1-score* lebih rendah dibandingkan kernel linear karena *recall* yang lebih rendah. SVM tanpa PCA memiliki *F1-score* terendah di skenario 1 (82%), lalu meningkat menjadi 85% di skenario 2, dan 89% di skenario 3. Peningkatan ini menunjukkan bahwa dengan lebih banyak data uji, model semakin baik dalam menangkap pola CKD.

Model SVM dengan PCA memberikan kestabilan pada *F1-score* di semua skenario, terutama pada kernel polynomial, di mana nilainya lebih stabil meskipun tidak selalu lebih tinggi dibandingkan model SVM tanpa PCA. Pada kernel linear, SVM dengan PCA memberikan keunggulan dengan mempertahankan *F1-score* 100% di semua skenario, sedangkan pada model SVM tanpa PCA nilai *F1-score* mengalami sedikit penurunan pada skenario dengan lebih banyak data uji. Hal ini menunjukkan bahwa reduksi dimensi dengan PCA dapat membantu menjaga performa model tetap konsisten meskipun jumlah data uji bertambah.

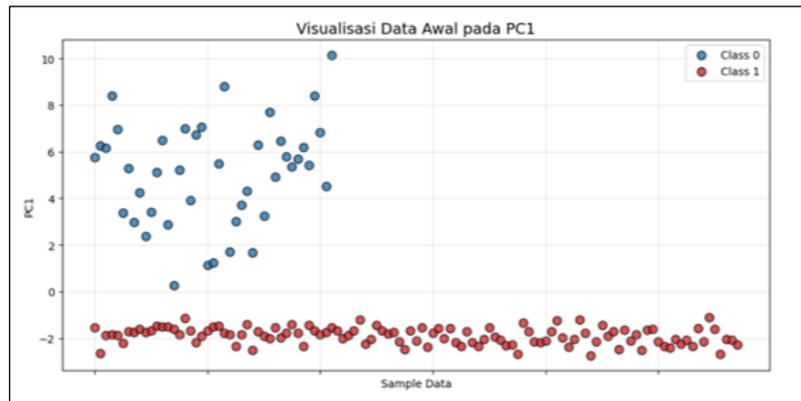
Secara keseluruhan, hasil uji coba menunjukkan bahwa proporsi data latih dan data uji berpengaruh terhadap performa model SVM, baik tanpa maupun dengan PCA. Semakin banyak data uji, semakin tinggi variasi data yang harus diuji, sehingga model menghadapi tantangan lebih besar dalam melakukan generalisasi. Pada SVM tanpa PCA, kernel linear mengalami sedikit penurunan akurasi seiring bertambahnya data uji, sedangkan kernel polynomial justru mengalami peningkatan akurasi, yang mengindikasikan bahwa model dengan kernel polynomial ini

memerlukan lebih banyak data uji untuk mengenali pola yang lebih kompleks secara lebih stabil.

Sementara itu, pada SVM dengan PCA, pola pengaruh data terhadap model polynomial lebih stabil dibandingkan tanpa PCA. Dengan bertambahnya data uji, akurasi yang didapatkan kernel polynomial cenderung lebih stabil dan sedikit meningkat, menunjukkan bahwa reduksi dimensi membantu model mengenali pola lebih baik dengan jumlah data uji yang lebih besar. Namun, pada kernel linear, PCA tidak memberikan perubahan signifikan terhadap akurasi. Hal ini berarti bahwa PCA membantu menjaga konsistensi performa model dengan menghilangkan fitur yang kurang relevan, sehingga model tetap mampu mengklasifikasikan data dengan akurasi tinggi (100%) meskipun proporsi data uji berubah.

Pada penelitian ini kernel linear bekerja sangat baik dalam semua skenario. Model dengan kernel linear mampu mencapai akurasi sempurna dalam beberapa skenario, yang mengidentifikasi bahwa data yang digunakan dapat dipisahkan secara linear dengan baik. Hal ini berarti fitur-fitur pada dataset sudah cukup representatif dalam membedakan kelas CKD dan Not CKD tanpa memerlukan transformasi non-linear yang kompleks. Ketika data divisualisasikan menggunakan PCA, distribusi data memiliki karakteristik *linear separable*, yaitu dapat dipisahkan dengan garis lurus. Kondisi ini sangat menguntungkan bagi kernel linear, karena model dapat dengan mudah menemukan *hyperplane* yang optimal tanpa transformasi non-linear tambahan. Sebaliknya, pada kernel polynomial, pada data yang sudah bisa dipisahkan dengan garis linear, kernel polynomial justru menambah kompleksitas yang ada, hal ini karena dia berkerja dengan memetakan

data ke ruang fitur yang lebih tinggi. Namun Meskipun demikian kernel polynomial tetap menunjukkan performa yang baik, dengan akurasi yang berkisar 88% sampai 92%. Hal ini menunjukkan bahwa kernel polynomial tetap mampu mengenali pola data dengan baik, meskipun tidak sebaik kernel linear.



Gambar 4. 6 Distribusi Data Chronic Kidney Disease Pada PC 1

Untuk memperkuat penjelasan tersebut, pada Gambar 4.6 menunjukkan visualisasi distribusi data hasil transformasi PCA dua Dimensi. Dari gambar tersebut tampak bahwa data CKD dan Not CKD tersebar dengan cukup jelas dan dapat dipisahkan oleh garis lurus, yang menandakan bahwa data memiliki karakteristik *linear separable*. Hal ini juga menunjukkan bahwa terdapat kombinasi fitur dominan yang mampu memisahkan kedua kelas secara jelas, sehingga model linear dapat bekerja secara optimal. Hal ini menjelaskan mengapa model SVM tanpa dan dengan PCA pada kernel linear mampu mencapai akurasi 100% di skenario tertentu.

Penggunaan PCA dalam model SVM memberikan efek yang bervariasi tergantung pada kernel yang digunakan. Pada kernel linear, PCA membantu mempertahankan akurasi yang tinggi yaitu 100% di semua skenario, sementara

tanpa PCA, model sempat mengalami penurunan akurasi pada beberapa skenario. Hal ini menunjukkan bahwa reduksi dimensi yang dilakukan PCA tetap mempertahankan informasi penting bagi model dalam kasus ini. Pada kernel polynomial, PCA memberikan peningkatan performa dalam beberapa skenario. Pada split 80:20, akurasi naik dari 88% (tanpa PCA) menjadi 91% (dengan PCA), menunjukkan bahwa PCA dapat membantu model menangkap pola lebih baik dengan mereduksi dimensi fitur yang kurang relevan. Namun, pada split 70:30 dan 60:40, perbedaan akurasi hanya sekitar 0-3%, yang menunjukkan bahwa manfaat PCA tetap terbatas dalam meningkatkan performa model polynomial.

Sebagai lanjutan dari pengujian pada dataset awal, peneliti melakukan percobaan tambahan dengan menggunakan dataset baru sejumlah 300 data, dengan jumlah kelas seimbang, dengan struktur fitur yang sama seperti dataset awal dari UCI Machine Learning. Dengan demikian, meskipun jumlah datanya bertambah, jenis dan jumlah fitur yang digunakan dalam pelatihan model tetap konsisten. Pengujian tambahan dilakukan karena pada dataset awal, model menunjukkan performa yang sangat tinggi, bahkan mencapai akurasi 100% pada beberapa skenario. Namun, performa sempurna tersebut tidak sepenuhnya mencerminkan kemampuan model secara umum, mengingat adanya kemungkinan dominasi fitur tertentu yang terlalu berpengaruh terhadap hasil klasifikasi. Dataset baru ini memungkinkan penilaian ulang terhadap efektivitas model SVM, khususnya dalam menghadapi data yang lebih kompleks. Selain itu, percobaan ini juga bertujuan untuk mengkaji kembali peran fungsi kernel (linear dan polynomial) serta pengaruh teknik reduksi dimensi PCA dalam meningkatkan performa model.

Hasil pengujian pada dataset baru menunjukkan bahwa meskipun tidak lagi mencapai 100%, model SVM tetap memberikan hasil yang baik dan stabil. Kernel polynomial, terutama jika dikombinasikan dengan PCA, menunjukkan performa paling konsisten di semua skenario, dengan akurasi tertinggi mencapai 88%. Sementara itu, kernel linear menunjukkan sensitivitas yang lebih tinggi terhadap perubahan data, baik dalam hal jumlah maupun distribusi. Performa SVM dengan kernel linear tetap kompetitif, meskipun dalam beberapa skenario PCA justru menurunkan akurasi. Berikut tabel hasil akurasi pada dataset baru.

Tabel 4. 13 Matrix Evaluasi Data Baru

Skenario	Model	Kernel	Akurasi	Presisi	Recall	F1-score
1 (80%:20%)	SVM tanpa PCA	Linear	82	83	82	81
		Polynomial	83	84	83	83
	SVM PCA	Linear	80	82	80	80
		Polynomial	87	89	87	86
2 (70%:30%)	SVM tanpa PCA	Linear	83	84	83	83
		Polynomial	86	86	86	86
	SVM PCA	Linear	83	85	83	83
		Polynomial	88	90	88	88
3 (60%:40%)	SVM tanpa PCA	Linear	86	86	86	86
		Polynomial	85	85	85	85
	SVM PCA	Linear	85	86	85	85
		Polynomial	88	90	88	88

Pada skenario 1, model SVM dengan PCA dan kernel polynomial menunjukkan hasil terbaik dengan akurasi 87%. Model SVM tanpa PCA dengan kernel polynomial juga menunjukkan hasil yang baik yaitu dengan akurasi 83%. Pada kernel linear penggunaan PCA justru menurunkan akurasi menjadi 80% dari yang sebelumnya tanpa PCA 82%. Hal ini berarti pada data ini PCA lebih cocok diterapkan pada kernel polynomial.

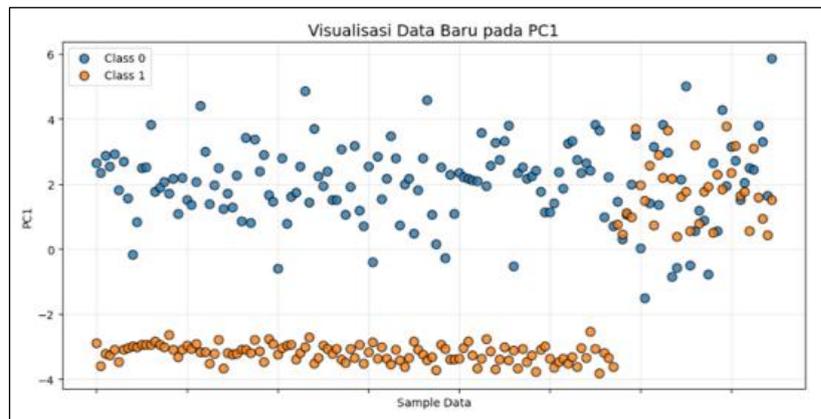
Pada skenario 2, model SVM dengan PCA pada kernel polynomial kembali unggul dengan akurasi 88%. Sedangkan tanpa PCA hasil akurasi yang didapatkan

juga menunjukkan performa tinggi yaitu 86%. Pada skenario kedua ini, model dengan kernel linear dengan dan tanpa PCA memiliki hasil akurasi yang sama yaitu 83%. Namun, model dengan PCA menunjukkan sedikit peningkatan pada presisi (85% dibandingkan 84%) yang mengindikasikan bahwa PCA memberikan kontribusi kecil dalam meningkatkan kemampuan model mengenali kelas positif secara tepat.

Pada skenario 3, model SVM dengan PCA pada kernel polynomial kembali unggul dengan akurasi 88%. Hasil yang didapat pada kernel polynomial sama persis dengan skenario sebelumnya (skenario kedua), hal ini menunjukkan bahwa model stabil dan konsisten meskipun jumlah data latih berkurang. Sebaliknya model SVM tanpa PCA dengan kernel polynomial mengalami penurunan performa dari 86% menjadi 85%. Hal ini menunjukkan model SVM linear tanpa PCA lebih sensitif terhadap pengurangan data latih. Disisi lain model SVM kernel linear tanpa PCA mengalami kenaikan performa dengan akurasi yang didapatkan yaitu 85% (naik 2% dari skenario sebelumnya). Pada skenario ini model SVM linear dengan PCA juga mendapatkan akurasi yang berbeda tipis namun tetap lebih tinggi dibanding tanpa PCA, yaitu 86%.

Jika dibandingkan dengan hasil pada dataset awal (sebelum penambahan data), terlihat adanya perbedaan karakteristik dalam performa model. Pada dataset awal yang jumlahnya lebih sedikit dan tidak seimbang, model SVM kernel linear menunjukkan performa yang terbaik, bahkan mencapai akurasi 100% dalam beberapa skenario. Namun pada dataset baru yang jumlahnya lebih banyak dan

seimbang, model dengan kernel polynomial memberikan perofrma yang lebih tinggi.



Gambar 4. 7 Distribusi Data Baru pada PC 1

Gambar 4.7 menunjukkan persebaran dataset baru. Pada dataset baru terlihat bahwa terdapat beberapa data kelas 1 yang tersebar berdekatan dengan data kelas 0, sehingga batas pemisah antar kelas tidak sepenuhnya linear. Untuk mengatasi hal ini diperlukan transformasi data ke dimensi yang lebih tinggi agar kelas dapat dipisahkan dengan lebih baik. Pada SVM, hal tersebut dilakukan dengan menggunakan fungsi kernel, seperti kernel polynomial. Oleh karena itu pada data baru ini kernel polynomial memberikan performa lebih baik dibandingkan kernel linear.

Perbedaan ini menunjukkan bahwa seiring dengan bertambahnya data, maka kompleksitas pola dalam data juga meningkat, sehingga kernel polynomial (yang bekerja dengan memetakan data ke dimensi yang lebih tinggi) menjadi lebih efektif dibanding kernel linear. Selain itu penggunaan PCA pada dataset baru juga memberikan dampak positif (kenaikan akurasi) yang lebih konsisten, khusus pada kernel polynomial.

Dengan demikian, efektifitas kernel dan teknik reduksi dimensi seperti PCA sangat dipengaruhi oleh karakteristik data. Oleh karena itu penting untuk melakukan evaluasi secara menyeluruh ketika menerapkan model SVM pada data yang berbeda.

4.5 Integrasi Islam

Dalam penelitian ini, klasifikasi didasarkan pada aturan dan perhitungan yang sudah ditentukan. Dalam islam, segala sesuatu yang ada di alam semesta juga telah diatur dengan ketetapan yang pasti. Setiap ciptaan memiliki ukuran, batasan, serta aturan yang tidak terjadi secara kebetulan, melainkan telah ditetapkan oleh Allah dengan penuh hikmah. Hal ini sesuai dengan firman Allah dalam surat Al-Qamar ayat 49:

إِنَّا كُلَّ شَيْءٍ خَلَقْنَاهُ بِقَدَرٍ

“Sesungguhnya Kami menciptakan segala sesuatu menurut ukuran” (Q.S Al-Qamar 49).

Dalam konteks klasifikasi penyakit gagal ginjal, konsep ini dapat diartikan sebagai upaya untuk mengelompokkan pasien gagal ginjal kronis dan non-kronis berdasarkan parameter medis terkait. Proses ini bertujuan untuk mengidentifikasi pola dari data pasien guna memberikan diagnosis yang akurat. Konsep ini juga sejalan dengan firman Allah surat Al-Furqon ayat 2:

الَّذِي لَهُ مُلْكُ السَّمٰوٰتِ وَالْاَرْضِ وَلَمْ يَتَّخِذْ وَلَدًا وَلَمْ يَكُنْ لَهُ شَرِيْكٌ فِى الْمُلْكِ وَخَلَقَ كُلَّ شَيْءٍ
فَقَدَرَهُ تَقْدِيْرًا

“Yang memiliki kerajaan langit dan bumi, tidak mempunyai anak, tidak ada sekutu bagi-Nya dalam kekuasaan (-Nya), dan Dia menciptakan segala sesuatu, lalu menetapkan ukuran-ukurannya dengan tepat” (Q.S Al-Furqon 2).

Dalam konteks klasifikasi, ayat ini menegaskan bahwa segala sesuatu memiliki ukuran dan aturan yang telah ditetapkan dengan jelas. Demikian pula dalam penelitian ini, proses klasifikasi penyakit gagal ginjal kronis menggunakan pendekatan berbasis fitur yang telah ditentukan, sehingga hasil pengelompokan dapat lebih terstruktur.

Hasil penelitian ini diharapkan juga bisa menjadi pengingat masyarakat untuk selalu menjaga kesehatan. Kesehatan berperan penting dalam menjalankan ibadah dengan maksimal. Kondisi kesehatan yang baik tentunya akan memudahkan seseorang dalam menjalankan ibadah seperti shalat, puasa, dan berbagai amal kebajikan lainnya. Rasulullah SAW juga menekankan bahwa kesehatan adalah salah satu nikmat yang sering kali diabaikan oleh manusia, sebagaimana sabdanya:

نِعْمَتَانِ مَغْبُورٌ فِيهِمَا كَثِيرٌ مِنَ النَّاسِ، الصِّحَّةُ وَالْفَرَاغُ

“Dua kenikmatan yang sering dilupakan oleh kebanyakan manusia adalah kesehatan dan waktu luang.” (HR. Al-Bukhari: 6412).

Ibnu Bathal menjelaskan bahwa seseorang tidak dapat dikatakan memiliki waktu luang hingga ia juga memiliki badan yang sehat. Hal ini menunjukkan betapa pentingnya menjaga kesehatan sebagai modal utama dalam menjalani kehidupan sehari-hari. Selain itu, dengan tubuh yang sehat seseorang dapat menjalani kehidupan dengan lebih baik, bekerja dengan optimal, dan beribadah dengan sempurna (Nur 2023).

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, metode *Support Vector Machine* (SVM) terbukti sebagai algoritma klasifikasi yang andal dalam mendeteksi penyakit gagal ginjal kronis. Model ini mampu memberikan performa yang sangat baik, terutama pada data dengan karakteristik yang terdefinisi jelas, seperti pada dataset UCI Machine Learning. Dalam skenario pembagian data 80:20, baik model SVM tanpa PCA maupun dengan PCA berhasil mencapai nilai akurasi, presisi, recall, dan F1-score sebesar 100%, menunjukkan kemampuan klasifikasi yang sangat optimal. Hasil ini mengindikasikan bahwa SVM memiliki kapabilitas yang kuat dalam membedakan antara kelas CKD dan non-CKD.

Seiring peningkatan jumlah data dan kompleksitas pola pada dataset baru yang lebih besar dan seimbang, performa model mengalami penurunan. Namun, SVM tetap mempertahankan kapabilitas yang tinggi dalam klasifikasi, menunjukkan kestabilan meski dihadapkan pada distribusi data yang lebih rumit. Kernel polynomial terbukti lebih andal dan konsisten dibanding kernel linear dalam menangani pola non-linear, terutama ketika dipadukan dengan PCA—kombinasi ini secara konsisten menghasilkan akurasi tertinggi, yakni hingga 88% pada ketiga skenario. Temuan ini menegaskan bahwa dengan pemilihan kernel serta strategi praproses seperti PCA yang tepat, SVM tetap menjadi metode klasifikasi yang tangguh dan adaptif terhadap beragam karakteristik data.

5.2 Saran

1. Menggunakan dataset yang berbeda guna menguji generalisasi dan ketahanan model terhadap data yang lebih bervariasi.
2. Mempertimbangkan penggunaan teknik reduksi dimensi fitur selain PCA dengan 5 komponen. Penelitian selanjutnya diharapkan bisa lebih mengoptimalkan dalam proses reduksi dimensi.
3. Menerapkan metode *preprocessing* yang berbeda, serta lebih berhati-hati pada tahap *preprocessing*, khususnya dalam penghapusan data.

DAFTAR PUSTAKA

- Akbar, Alvin Tarisa, Novanto Yudistira, and Achmad Ridok. 2023. "Identifikasi Gagal Ginjal Kronis Dengan Mengimplementasikan Metode Support Vector Machine Beserta K-Nearest Neighbour (SVM-KNN)." *Jurnal Teknologi Informasi Dan Ilmu Komputer* 10(2):301–8. doi: 10.25126/jtiik.20231026059.
- Almais, Agung Teguh Wibowo, Adi Susilo, Agus Naba, Moehammad Sarosa, Cahyo Crys dian, Imam Tazi, Mokhammad Amin Hariyadi, Muhammad Aziz Muslim, Puspa Miladin Nuraida Safitri Abdul Basid, Yunifa Miftachul Arif, Mohammad Singgih Purwanto, Diyan Parwatiningtyas, Supriyono, and Hendro Wicaksono. 2023. "Principal Component Analysis-Based Data Clustering for Labeling of Level Damage Sector in Post-Natural Disasters." *IEEE Access* 11(July):74590–601. doi: 10.1109/ACCESS.2023.3275852.
- Anwar, Khoirul, and Manuharawati. 2021. "Math Unesa." *Jurnal Ilmiah Matematika* 9(2):437–46.
- Celena, Anindya Kirana Khansa, Muhammad Tanzil Furqon, and Achmad Ridok. 2022. "Klasifikasi Berat Badan Lahir Rendah (BBLR) Menggunakan Metode Support Vector Machine Dengan Teknik SMOTE." *Pengembangan Teknologi Informasi Dan Ilmu Komputer* 6(7):3442–51.
- Grandis, Galuh Fadillah, Yuita Arumsari, and Indriati. 2021. "Seleksi Fitur Gain Ratio Pada Analisis Sentimen Kebijakan Pemerintah Mengenai Pembelajaran Jarak Jauh Dengan K-Nearest Neighbor." *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer* 5(8):3507–14.
- Ihsani, Dian Atia, Achmad Arifin, and Muhammad Hilman Fatoni. 2020. "Klasifikasi DNA Microarray Menggunakan Principal Component Analysis (PCA) Dan Artificial Neural Network (ANN)." *Jurnal Teknik ITS* 9(1). doi: 10.12962/j23373539.v9i1.51637.
- Imaniar Ikko Mulya Rizky, Suhendro Yusuf Irianto, and Sriyanto. 2023. "Perbandingan Kinerja Algoritma Naive Bayes, Support Vector Machine Dan Random Forest Untuk Prediksi Penyakit Ginjal Kronis." *Seminar Nasional Hasil Penelitian Dan Pengabdian Masyarakat* 18:139–51.
- Jolliffe, P. 2018. *Prisons and Forced Labour in Japan*.
- Kurniawan, Puan Maharani, Agung Teguh Wibowo Almais, M. Amin Hariyadi, M. Ainul Yaqin, and Suhartono. 2023. "Prediction of Civil Servant Performance Allowances Using the Neural Network Backpropagation Method." *International Journal on Informatics Visualization* 7(3):673–80. doi: 10.30630/joiv.7.3.1698.
- Madani, Puja Milenia Sriwildan, Tatang Rohana, Kiki Ahmad Baihaqi, and Ahmad Fauzi. 2024. "Perbandingan Kinerja Klasifikasi Penyakit Ginjal Menggunakan

Algoritma Support Vector Machine (SVM) Dan Decision Tree (DT).” *Building of Informatics, Technology and Science (BITS)* 6(1):74–82-74–82. doi: 10.47065/bits.v6i1.5206.

Mellina, Ayu Dian Fitri, Suhartono Suhartono, and M. Ainul Yaqin. 2024. “Prediksi Deteksi Penyakit Kanker Payudara Dengan Menggunakan Algoritma Decision Tree.” *JISKA (Jurnal Informatika Sunan Kalijaga)* 9(1):70–78. doi: 10.14421/jiska.2024.9.1.70-78.

Nur, Nur Shofiatul. 2023. “Kesehatan Jasmani Dan Rohani Menurut Al-Qur’an Dan Hadist.” *Ta’wiluna: Jurnal Ilmu Al-Qur’an, Tafsir Dan Pemikiran Islam* 4(2):265–80. doi: 10.58401/takwiluna.v4i2.977.

Penyusun, T. I. M. 2019. *Mengenal Penyakit Ginjal Kronis Dan Perawatannya*. edited by H. Kusuma. Fakultas Kedokteran Univeritas Diponegoro.

Purnama, Drajat Indra. 2019. “Analisis Komponen Utama Pada Data Potensi Kecamatan Di Kota Palu Sebelum Bencana Gempa Bumi Dan Tsunami 28 September 2018.” *Jurnal Matematika, Statistika Dan Komputasi* 16(1):25. doi: 10.20956/jmsk.v16i1.6329.

Putri, Shofi Inayah, Tri Kesuma Dewi, Akademi Keperawatan, Dharma Wacana, and Kata Kunci. 2023. “Jurnal Cendikia Muda Volume 3 , Nomor 2 , Juni 2023 ISSN : 2807-3469 IMPLEMENTATION OF SLOW DEEP BREATHING ON FATIGUE IN CHRONIC KIDNEY Putri , Penerapan Slow Deep PENDAHULUAN Gagal Ginjal Kronik Adalah Kondisi Saat Fungsi Ginjal Menurun Ginjal , Maka Aka.” 3.

Rahayu, Siskawati, and Yuni Yamasari. 2024. “Klasifikasi Penyakit Stroke Dengan Metode Support Vector Machine (SVM).” *Journal of Informatics and Computer Science (JINACS)* 5(03):440–46. doi: 10.26740/jinacs.v5n03.p440-446.

Salsabilla, Rindi. 2024. “Data Kemenkes: Lebih Dari 700 Ribu Orang RI Menderita Ginjal Kronis.” *16 January*. Retrieved September 26, 2024 (<https://www.cnbcindonesia.com/lifestyle/20240116111340-33-506206/data-kemenkes-lebih-dari-700-ribu-orang-ri-menderita-ginjal-kronis>).

Saputra, Kurniawan. 2023. “Perbandingan Kinerja Fungsi Kernel Algoritma Support Vector Machine Pada Klasifikasi Penyakit Padi.” *Ijccs* 17(1):119–31.

Sirait, D. T. C., A. Adiwijaya, and ... 2019. “Analisis Perbandingan Reduksi Dimensi Principal Component Analysis (PCA) Dan Partial Least Square (PLS) Untuk Deteksi Kanker Menggunakan Data.” *E-Proceeding of Engineering* 6(2):8570–81.

Suryani, Sri, Rani Lisa Indra, and Bayu Saputra. 2022. “Jurnal Keperawatan Hang Tuah (Hang Tuah Nursing Journal) Gambaran Persepsi Penyakit Pada Pasien Dengan Gagal Ginjal Kronik.” *Jurnal Keperawatan Hang Tuah* 2:164–79.

Susanto, F. Haryanto. 2020. *Penyakit Ginjal Kronis (Chronic Kidney Disease) Dan*

Hipertensi. edited by S. R. Wicaksono. CV. Sseribu Bintang.

Swain, Debabrata, Utsav Mehta, Ayush Bhatt, Hardeep Patel, Kevin Patel, Devanshu Mehta, Biswaranjan Acharya, Vassilis C. Gerogiannis, Andreas Kanavos, and Stella Manika. 2023. "Machine Learning." *Electronics* 12(212):1–13.

Syahranita, Rafika, Suhartono, and Syahiduz Zaman. 2023. "Regresi Logistik Multinomial Untuk Prediksi Kategori Kelulusan Mahasiswa." *Jurnal Informatika Sunan Kalijaga* 8(2):102–11.

Yunianto, Era. 2024. "DIAGNOSTIK PENYAKIT GINJAL KRONIS MENGGUNAKAN MODEL."