JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Identifikasi Penyakit Diabetes Mellitus Menggunakan Algoritma Support Vector Machine dan Random Forest

Anggi Renata Agusti*, Ahmad Fauzi, Kiki Ahmad Baihaqi, Tatang Rohana

Fakultas Ilmu Komputer, Teknik Informatika, Universitas Buana Perjuangan Karawang, Karawang, Indonesia
Email: [1,*]if21.anggiagusti@mhs.ubpkarawang.ac.id , [2]afauzi@ubpkarawang.ac.id, [3]kikiahmad@ubpkarawang.ac.id, [4]tatang.rohana@ubpkarawang.ac.id
Email Penulis Korespondensi: if21.anggiagusti@mhs.ubpkarawang.ac.id
Submitted **30-05-2025;** Accepted **14-08-2025;** Published **30-08-2025**

Abstrak

Diabetes mellitus merupakan penyakit metabolisme kronis yang semakin umum di Indonesia, diperkirakan akan memengaruhi pada tahun 2020, tercatat lebih dari 10,8 juta jiwa. Penyakit ini perlu dikenali sejak dini untuk mencegah komplikasi berat yang dapat meningkatkan angka kesakitan dan kematian. Dengan membandingkan kedua metode, Penelitian ini dilakukan untuk mengetahui apakah salah satu pendekatan menunjukkan tingkat akurasi yang lebih baik serta untuk mengembangkan model klasifikasi berdasarkan data pasien. Data penelitian ini disediakan oleh Puskesmas Anggadita yang meliputi data demografi, gaya hidup, dan hasil penilaian kesehatan dari 1001 pasien. Salah satu langkah penelitian adalah pra-pemrosesan data sampai dengan evaluasi. Pemodelan SVM dan RF, dapat mengevaluasi model yang menggunakan metrik akurasi, presisi, *recall*, dan *F1-score*. Berdasarkan hasil pengujian, algoritma *Random Forest* menunjukkan performa terbaik dengan akurasi *sebesar* 99%, presisi 99%, *recall* 100%, dan *F1-score* 99%, sedangkan SVM mendapatkan akurasi 91%, presisi 0,93%, *recall* 0,91%, dan *F1-score* 0,92%. Hal ini menunjukkan seberapa baik *Random Forest* memisahkan pasien dengan dan tanpa diabetes. Penelitian ini diharapkan mampu menjadi salah satu rujukan dalam memperoleh informasi pembuatan sistem pendukung keputusan medis sehingga para tenaga kesehatan dapat lebih cepat dan akurat dalam mendiagnosis penyakit diabetes mellitus.

Kata Kunci: Machine Learning; Random Forest; Support Vector Machine; Klasifikasi; Diagnosis

Abstract

Diabetes mellitus is a chronic metabolic disease that is increasingly common in Indonesia, estimated to affect more than 10.8 million people in 2020. This disease needs to be recognized early to prevent serious complications that can increase morbidity and mortality. By comparing the two methods, this study was conducted to determine whether one approach shows a better level of accuracy and to develop a classification model based on patient data. The research data was provided by the Anggadita Health Center which includes demographic data, lifestyle, and health assessment results from 1001 patients. One of the research steps is data pre-processing to evaluation. SVM and RF modeling can evaluate models using accuracy, precision, recall, and F1-score metrics. Based on the test results, the Random Forest algorithm showed the best performance with an accuracy of 99%, precision of 99%, recall of 100%, and F1-score of 99%, while SVM got an accuracy of 91%, precision of 0.93%, recall of 0.91%, and F1-score of 0.92%. This shows how well Random Forest separates patients with and without diabetes. This study is expected to be one of the references in obtaining information for making medical decision support systems so that health workers can be faster and more accurate in diagnosing diabetes mellitus.

Keywords: Machine Learning; Random Forest; Support Vector Machine; Classification; Diagnosis

1. PENDAHULUAN

Penyakit atau gangguan metabolisme kronis yang dikenal sebagai diabetes mellitus (DM) disebabkan oleh sejumlah keadaan dan ditandai dengan peningkatan tekanan darah, gula darah, dan masalah metabolisme karbohidrat, lemak, dan protein akibat tidak memadainya aksi insulin [1]. Penyebab utama diabetes tipe 2 adalah pola konsumsi makanan yang tidak baik dan perilaku hidup yang tidak mendukung kesehatan, termasuk makan terlalu banyak makanan manis, bertambahnya usia, dan bahkan memiliki riwayat penyakit dalam keluarga [2]. Gejala kekurangan insulin atau masalah pada penderita diabetes meliputi gangguan penglihatan, haus, lesu, dan sering buang air kecil. Hiperglisemia kronis merupakan ciri lain dari diabetes, suatu kondisi metabolisme yang dapat membahayakan organ seperti ginjal, jantung, saraf, pembuluh darah, mata, dan stroke. Selain itu, diabetes mellitus adalah salah satu masalah kesehatan global yang berkembang paling cepat dan terus meningkat, berkontribusi pada tinggi nya angka kematian [3].

Jumlah data penderita diabetes di Indonesia telah melampaui 10,8 juta pada tahun 2020, dan angka ini diprediksi akan terus bertambah setiap tahun. Banyak pasien tidak menyadari kondisi diabetes, sehingga penyakit ini berkembang menjadi komplikasi serius, seperti penyakit kardiovaskular dan kerusakan saraf, yang akhirnya meningkatkan angka morbiditas dan mortalitas [4]. Penelitian ini bertujuan untuk mengklasifikasikan diabetes mellitus (DM) menggunakan model *Support Vector Machine* (SVM) dan *Random Forest.* Proses klasifikasi ini dirancang untuk mengidentifikasi pola yang dapat digunakan dalam membangun model prediksi berdasarkan variabel tertentu, guna menentukan apakah seseorang menderita diabetes atau tidak [5]. Penelitian tentang klasifikasi penyakit DM sangat penting, terutama di Indonesia, di mana prevalensi diabetes terus meningkat. Meningkatnya jumlah penderita diabetes menyoroti pentingnya interensi dini melalui klasifikasi berbasis teknologi, seperti penggunaan algoritma *Machine Learning* yang efektif untuk klasifikasi diabetes karena, mampu mengelola berbagai variabel dan data kompleks secara efisien [4].

Berbagai studi sebelumnya telah menunjukan potensi algoritma *Support Vector Machine* dan *Random Forest* dalam klasifikasi penyakit diabetes. Sebagai contoh, Maulana [6] Mengoptimalkan parameter *Support Vector Machine*

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

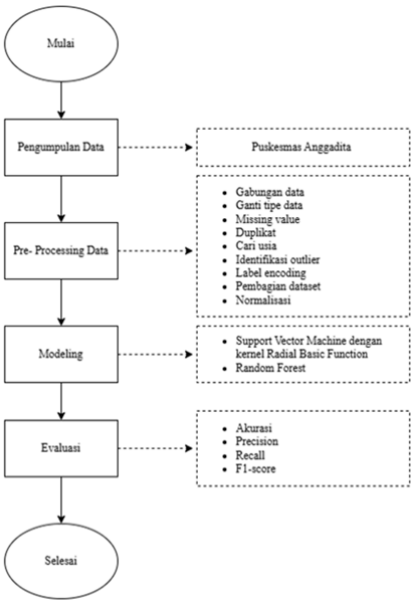
dengan menerapkan *Particle Swarm Optimization* untuk meningkatkan klasifikasi diabetes, di dapatkan hasil analisis bahwa model SVM yang dioptimasi PSO dengan hasil akurasi 74,57%, presisi 74,99%, *recall* 74,57%, dan *F1-score* 74,54%. Penelitian lainnya oleh Desiani, membandingkan algoritma SVM dan *Naïve Bayes* pada kasus klasifikasi diabetes menggunakan dua teknik evaluasi, seperti *persentase split* dan *k-fold cross-validation*. Pada pengujian melalui persentase split, pemodelan SVM mencapai akurasi 77,27%, sedangkan *Naïve Bayes* mendapatkan akurasi 79%. Dalam pengujian menggunakan *k-fold cross-validation*, SVM memperoleh akurasi 71%, sementara *Naïve Bayes* mencapai 75% [7]. Selanjutnya, penelitian ini membandingkan beberapa algoritma, termasuk *Naïve Bayes, Regresi Logistik, Random Forest,* SVM, dan KNN untuk prediksi diabetes. Akurasi *Regresi Logistik* adalah 78%, dengan presisi 77%, *recall* 73%, dan *F1-score* 74%. Untuk *Random Forest*, akurasi mencapai 83%, presisi 82%, *recall* 80%, dan *F1-score* 81%. KNN menunjukkan akurasi 80%, presisi 79%, *recall* 75%, dan *F1-score* 77%, sementara SVM memiliki akurasi 81%, presisi 81%, *recall* 76%, dan *F1-score* 77%. *Naïve Bayes* mencapai tingkat akurasi 77%, presisi 76%, *recall* 74%, dan *F1-score* 74% [8]. Selain itu, penelitian juga dilakukan untuk mengkategorikan diabetes mellitus tipe 2 menerapkan algoritma *Iterative Dichotomiser Tree* dan *Support Vector Machine*. Algoritma SVM diterapkan dengan kernel linear dan RBF, di mana akurasi SVM dengan *kernel linear* mencapai 78,5% dan dengan *kernel* RBF 78%. Hasil klasifikasi terendah dari ID3 adalah 74% [9]. Penelitian selanjutnya menerapkan metode klasifikasi K-Nearest Neigbor pada dataset penderita penyakit diabetes, berdasarkan hasil penelitian nilai akurasi tertinggi sebesar 39% diperoleh pada K=3, dengan presisi maksimum 65% yang dicapai pada K=3 dan K=5, recall tertinggi sebesar 36% pada K=3, serta nilai F-Measure tertinggi sebesar 46% juga pada K=3 [10]. Penelitian oleh Siregar, membandingkan tiga algoritma klasifikasi dalam prediksi cuaca, yaitu *Naïve Bayes, Decision Tree,* dan *Random Forest*. Dari hasil yang diperoleh, akurasi *Naïve Bayes* sebesar 72,22%, *Decision Tree* sebesar 54,01%, dan *Random Forest* mencatat akurasi tertinggi sebesar 82,38% [11]. Penelitian berikutnya oleh Rohana, membandingkan tiga metode yaitu *Multilayer Perceptron, Support Vector Machine, dan Decision Tree*, hasil penelitian menunjukkan bahwa *Multilayer Perceptron* memiliki akurasi sebesar 97,1%, *Support Vector Machine* 98,9%, dan *Decision Tree* memiliki akurasi 100% [12].

Tinjauan penelitian sebelumnya telah mengungkapkan bahwa tidak ada penelitian yang secara langsung membandingkan efektivitas algoritma *Random Forest* (RF) dan *Support Vector Machine* (SVM) pada klasifikasi diabetes mellitus. Dengan demikian, studi ini ditujukan untuk mengembangkan sebuah model klasifikasi berdasarkan data pasien dan algoritma mana yang tingkat akurasi nya lebih optimal melalui perbandingan kedua algoritma tersebut. Evaluasi kinerja dilakukan menggunakan empat metrik utama, khususnya akurasi, presisi, *recall, F1-score*. Terdapat harapan bahwa studi ini akan berkontribusi pada kemajuan deteksi diabetes, serta menjadi acuan dalam pengambilan keputusan medis dini guna mengurangi risiko komplikasi dan angka kematian akibat diabetes [13].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dirancang berdasarkan prosedur yang terstruktur dan sistematis untuk memastikan setiap langkah dilakukan secara terorganisasi. Prosedur tersebut mencakup beberapa tahap, dimulai dari pengumpulan data, pre-processing data, modeling dan evaluasi. Setiap tahap saling berhubungan dan dirancang untuk menghasilkan kesimpulan yang valid serta selaras dengan tujuan penelitian. Gambar 1 menjelaskan langkah-langkah utama dalam pelaksanaan penelitian ini, yang terdiri dari empat tahap sebagai berikut:



Gambar 1. Tahap Penelitian

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

2.2 Klasifikasi

Proses kategorisasi melibatkan pengidentifikasian model dengan fitur yang memperjelas atau membedakan konsep kelas data. Metode *Random Forest* dan *Support Vector Machine* adalah dua contoh algoritma kategorisasi [14]. Dalam penelitian ini, klasifikasi juga melibatkan proses penghitungan data yang tersedia, disebut juga data pelatihan, dengan data baru atau data pengujian. Dalam klasifikasi, kumpulan data yang digunakan harus mempunyai label atau atribut, untuk mengetahui objek kelas setiap permasalahan pada data, ini disebut sebagai tujuan dari klasifikasi [15].

2.3 Pengumpulan Data

Sumber data penelitian berasal dari hasil pemeriksaan di Puskesmas Anggadita, dengan total 1001 data pasien. Data ini terbagi ke dalam dua kelas, yaitu "Diabetes" untuk pasien yang terdiagnosis diabetes dan "Tidak" untuk pasien yang tidak terdiagnosis diabetes. Identitas pasien, riwayat penyakit tidak menular dalam keluarga atau diri sendiri, dan faktor risiko (merokok, tidak aktif, asupan gula, garam, dan lemak tinggi, kurang buah dan sayur, serta penggunaan alkohol) merupakan beberapa variabel terkait yang termasuk dalam kumpulan data tersebut.

Data juga mencakup pengukuran tekanan darah (sistolik dan diastolik), tinggi badan, berat badan, lingkar perut, Pemeriksaan gula. Seluruh informasi ini disusun dalam format tabular seperti pada Excel. Hal ini bertujuan untuk mempercepat tahap selanjutnya dari pemrosesan dan analisis data. Kita dapat melihat kumpulan data objek penelitian pada Tabel 1.

Tabel 1. Dataset Diabetes

No	Nama	Sistol	Diastol	Berat Badan (Kg)	Pemeriksaan Gula	Diagnosis
1	Ruswanti	145	85	62	356	Diabetes
2	Ikarsem	192	113	51	337	Diabetes
3	Pipin	154	81	96	96	Tidak
4	Saturi	127	116	66	112	Tidak
5	Darwil	141	54	41	113	Tidak
…	…	…	…	…	…	…
1001	Sarwono	160	90	66	207	Diabetes

2.4 *Pre-processing*

Data yang tersedia adalah data per tahun, sehingga perlu digabungkan data pertahun tersebut menjadi dataset utuh agar lebih terstruktur. Setelah data berhasil digabungkan, langkah selanjutnya adalah melakukan pemeriksaan dan penyesuaian tipe data pada masing-masing atribut. Misalnya pada atribut tinggi badan dan berat badan, yang awalnya bertipe *object,* diubah menjadi tipe numerik *float* supaya dapat dipakai dalam perhitungan matematis dan pemodelan *machine learning.* Langkah berikutnya adalah memeriksa nilai yang hilang (*missing values*) pada dataset. Untuk atribut numerik yang memiliki data kosong, dilakukan proses imputasi dengan menggantinya menggunakan nilai rata-rata (*mean*) dari kolom terkait. Sementara itu, untuk atribut kategorikal, nilai yang kosong diisi menggunakan nilai modus (*mode*), yaitu nilai yang paling sering muncul dalam kolom tersebut.

Tahap selanjutnya pengecekan data duplikat, dilakukan pemeriksaan seluruh entri data untuk memastikan tidak ditemukan data yang berulang melalui teknik pendeteksian duplikasi. Data yang terindikasi duplikat kemudian dihapus untuk memastikan bahwa setiap baris mewakili individu yang berbeda, sekaligus menghindari bias dalam hasil analisis model. Tahapan *pre-processing* dilanjutkan dengan penambahan atribut baru berupa usia pasien. Usia dihitung dengan mengurangi tanggal lahir dan tanggal pemeriksaan, lalu hasilnya dikonversi ke dalam satuan tahun. Setelah atribut usia berhasil ditambahkan, proses dilanjutkan dengan mendeteksi keberadaan data *outlier*. Teknik *Z-Score* digunakan untuk mengidentifikasi nilai-nilai ekstrem, di mana data dengan *Z-Score* lebih dari 3 atau kurang dari -3 dikategorikan sebagai *outlier. Outlier* yang ditemukan kemudian dihapus dari dataset untuk menjaga distribusi data tetap normal dan meningkatkan performa model klasifikasi.

Pada tahap selanjutnya, atribut kategorikal diubah ke dalam format numerik menggunakan metode *Label Encoding*. Variabel seperti jenis kelamin, riwayat penyakit, faktor risiko, rujukan, dan diagnosis dikonversi menjadi nilai angka agar dapat diolah oleh algoritma *machine learning*. Setelah semua atribut berhasil dikonversi, dataset kemudian dipisahkan dengan dua bagian 80% untuk data *training* dan 20% untuk data *testing*. Proses pemisahan ini dilakukan agar model dapat dilatih menggunakan sebagian besar data dan diuji akurasi nya, sehingga hasil evaluasi model menjadi lebih objektif dan akurat. Tahapan akhir yaitu, melakukan normalisasi pada atribut numerik dengan menggunakan metode *Standard Scaler*. Normalisasi ini mengubah data sehingga memiliki nilai dengan mean sebesar 0 dan deviasi standar bernilai 1.

2.5 *Modeling*

Algoritma *Random Forest* dan *Support Vector Machine* (SVM) digunakan dalam penelitian ini karena keduanya dikenal memiliki kinerja yang baik dalam menyelesaikan masalah klasifikasi, khususnya dalam mengkategorikan kasus diabetes. Model SVM diimplementasikan menggunakan kernel *Radial Basis Function* (RBF) dengan nilai C=1 dan gamma=*'scale'*. Data pelatihan yang dinormalisasi digunakan untuk melatih model, sedangkan data pengujian digunakan untuk

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

mengevaluasi kinerjanya [16]. Rangkaian proses dalam membangun model SVM dijelaskan secara sistematis pada Tabel 2.

Tabel 2. Membuat Model Klasifikasi dengan SVM

Klasifikasi dengan algoritma SVM
Input : Dataset Training
Output : Hasil prediksi model SVM pada data uji 1. Baca dataset 2. Tentukan Kernel yang akan digunakan (RBF) 3. Tentukan nilai *hyperparameter* (C, Gamma) 4. Temukan kombinasi *hyperparameter* terbaik untuk model *training* 5. Melatih model SVM menggunakan data pelatihan untuk mempelajari pola dan membuat *hyperplane* pemisah 6. data Prediksi label diabetes pada data uji dilakukan menggunakan model yang telah melalui proses pelatihan

Berdasarkan Tabel 2, langkah awal dalam pengembangan model *Support Vector Machine* (SVM) dimulai dengan proses membaca dataset serta pemilihan kernel yang digunakan, yaitu *Radial Basis Function* (RBF). Setelah itu, dilakukan penentuan nilai hyperparameter, berupa nilai C dan gamma. Pada penelitian ini, digunakan konfigurasi parameter C=1 dan gamma=*'scale'* yang telah disesuaikan dengan karakteristik data yang sebelumnya telah melalui proses normalisasi.

Tahapan berikutnya adalah menentukan kombinasi hyperparameter yang paling optimal guna meningkatkan kinerja model pada data pelatihan. Setelah kombinasi terbaik ditemukan, model SVM dilatih menggunakan data training agar mampu mengenali pola dan membentuk hyperplane yang memisahkan antar kelas. Pada tahap akhir, dilakukan prediksi terhadap data pengujian untuk mengidentifikasi apakah seseorang termasuk dalam kategori diabetes atau tidak. Output dari proses ini berupa label hasil prediksi, yang kemudian digunakan dalam tahap evaluasi untuk menilai tingkat akurasi serta efektivitas model yang telah dibangun.

Selain itu, penelitian ini juga menggunakan algoritma *Random Forest* untuk melakukan klasifikasi pada data diabetes. Pada *Random Forest* juga menggunakan *estimato*r=100 dan *random_state*=42 yang ditentukan secara langsung tanpa melalui proses tuning. Di samping itu, data pelatihan yang telah dinormalisasi dimanfaatkan untuk membangun model, sedangkan data pengujian dipakai untuk mengevaluasi performanya [17]. Rangkaian proses pembentukan model *Random Forest* dalam penelitian ini dapat dilihat secara rinci pada Tabel 3.

Tabel 3. Membuat Model Klasifikasi dengan *Random Forest*

Klasifikasi dengan algoritma *Random Forest*
Input : Dataset Testing
Output : Hasil prediksi model RF pada data uji 1. Baca dataset 2. Membuat objek menggunakan *classifier* dengan *hyperparameter* 3. Melatih model menggunakan data latih 4. Memprediksi label pada data uji 5. Hasil perbandingkan prediksi model dengan label 6. Hasil prediksi label pada data uji dengan model yang telah dilatih

Pada Tabel 3, tahapan awal dimulai dengan membaca dataset, kemudian dilanjutkan dengan membentuk objek menggunakan *classifier* yang dikonfigurasi dengan *hyperparameter* tertentu. Dalam studi ini, digunakan parameter *estimato*r=100 dan *random_state*=42. Nilai tersebut ditentukan secara langsung tanpa proses tuning. Setelah model terbentuk, proses berikutnya adalah melatih model menggunakan data training yang telah melalui proses normalisasi. Selanjutnya, dilakukan prediksi terhadap data testing guna memperoleh hasil klasifikasi. Hasil prediksi tersebut kemudian dibandingkan dengan label aktual untuk menilai performa model.
Langkah ini bertujuan untuk mengukur seberapa akurat dan andalnya algoritma *Random Forest* dalam mengklasifikasikan data pasien ke dalam kategori diabetes atau tidak.

2.6 Evaluasi

Studi ini menggunakan *confussion matrix* dan mengevaluasi kualitas model klasifikasi menggunakan sejumlah matriks yang berbeda Fraksi prediksi akurat relatif terhadap semua data yang tersedia dikenal sebagai akurasi. Metrik Presisi merupakan rasio antara jumlah prediksi positif yang tepat dengan total prediksi positif yang dilakukan oleh model [2]. pada tahap evaluasi memiliki tujuan untuk mengukur kualitas kemampuan suatu model klasifikasi. *Confusion Matrix* digunakan pada proses ini guna mencari nilai akurasi, persisi, *recall, dan F1-Score*, hal ini merupakan tahapan penting agar dapat menjawab perbandingan model algoritma mana yang memiliki performa tinggi dalam klasifikasi [18]. Evaluasi model dilakukan menggunakan *confusion matrix* yang tampilkan pada Tabel 4. Melalui matriks ini, hasil klasifikasi terhadap data uji dibandingkan dengan data pelatihan, serta hasil prediksi pada data pelatihan dibandingkan dengan label sebenarnya. Perbandingan ini digunakan untuk menilai kinerja model secara menyeluruh [19].

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Tabel 4. *Confussion Matrix*

Kelas Asli	Prediksi Positif	Prediksi Negatif
Positif	*True positive* (TP)	*False Negative* (FN)
Negatif	*False Positive* (FP)	*True Negative* (TN)

a. Kuantitas data yang benar-benar diklasifikasikan sebagai positif dan yang secara akurat diidentifikasi oleh model klasifikasi sebagai positif dikenal sebagai *True positive* [20].
b. Istilah *False positive* mengacu pada kuantitas data yang diprediksi model sebagai positif tetapi sebenarnya diklasifikasikan sebagai negatif [20].
c. Kuantitas data yang idealnya masuk dalam kategori positif, justru salah diklasifikasikan sebagai negatif, hal ini disebut sebagai *False Negative* [21].
d. Proporsi data dengan label negatif yang secara akurat dikenali oleh sistem klasifikasi sebagai *True negative* [21].

Empat teknik berikut memberikan pandangan yang lebih menyeluruh tentang kinerja klasifikasi model:
Akurasi merupakan tingkat kesesuaian antara hasil prediksi dengan nilai sebenarnya.

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP+FP+TN+FN}} \text{x} 100\% \quad (6)$$

Presisi menunjukkan proporsi prediksi positif yang akurat terhadap semua prediksi positif model [22].

$$\text{Presisi} = \frac{\text{TP}}{\text{TP+FP}} \text{x} 100\% \quad (7)$$

Recall menunjukkan proporsi data positif yang benar-benar dideteksi oleh model dalam kaitannya dengan jumlah total data positif[22].

$$\textit{Recall} = \frac{\text{TP}}{\text{TP+FN}} \text{x} 100\% \quad (8)$$

F1-score menggabungkan presisi dan *recall* menjadi satu metrik harmonis, sering kali disebut juga sebagai *F1-measure* [22].

$$F-score = 2\text{x}\frac{\text{Presisi x } Recall}{\text{Presisi}+Recall} \quad (9)$$

3. HASIL DAN PEMBAHASAN

3.1 Hasil *Pre-processing* Data

Berdasarkan proses ini, dilakukan proses *pra-pemrosesan* untuk meningkatkan kualitas model. Langkah pertama yang dilakukan adalah memeriksa informasi mengenai data, termasuk tipe data setiap kolom, dengan menggunakan fungsi *info()*. Namun telah diketahui bahwa atribut tinggi badan (cm) dan berat badan (kg) masih bertipe data *object*. Oleh karena itu, perlu dilakukan konversi tipe data menjadi *float64* dengan menggunakan fungsi *astype()*. Tabel 5 memperlihatkan hasil dari proses konversi.

Tabel 5. Hasil Perubahan Tipe Data

Column	Non-Null Count	Dtype
Tanggal pemeriksaan	1000 non-null	datetime64[ns]
Nama pasien	1000 non-null	Object
Tanggal lahir	1000 non-null	datetime64[ns]
…	…	…
Tinggi badan (cm)	1000 non-null	float64
Berat badan (kg)	999 non-null	float64
Lingkar perut (cm)	996 non-null	float64
Pemeriksaan gula	1001 non-null	int64
Rujuk	999 non-null	Object
Diagnosis	1001 non-null	Object

Seperti yang ditujukkan pada Tabel 5, tipe data atribut tinggi badan(cm) dan berat badan (kg) telah berhasil diubah menjadi *'float64'* sesuai dengan kebutuhan analisis lebih lanjut. Selain itu, kolom seperti lingkar perut, rujukan, dan diagnosis turut ditampilkan untuk menggambarkan distribusi nilai serta kesiapan data dalam tahap *pre-processing* yang akan dilakukan berikutnya.

Selanjutnya, dilakukan pemeriksaan terhadap nilai yang hilang (*missing value*) pada dataset menggunakan fungsi *isnull(),* yang digunakan untuk mendeteksi nilai yang kosong, serta *sum()* untuk menghitung jumlah nilai kosong yang ada pada setiap kolom, telah ditemukan adanya *missing value* pada beberapa atribut, mencakup tinggi badan (cm), berat badan (kg), lingkar pinggang (cm), rujuk, konsumsi alkohol, kurang konsumsi buah dan sayur, serta gula berlebihan. Untuk mengatasi hal ini, dilakukan penanganan dengan mengisi nilai yang hilang menggunakan fungsi *fillna()* yang

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

digunakan untuk menggantikan *missing value* dengan nilai tersebut, memastikan kelengkapan data yang digunakan dalam analisis lebih lanjut. Tahapan selanjutnya adalah melakukan pengecekan data duplikat untuk memastikan tidak adanya entri yang tercatat lebih dari satu kali yang artinya nilai '0' menandakan tidak adanya *missing value*. Hasil setelah dilakukan penanganan *missing value* dapat dilihat pada Tabel 6.

Tabel 6. Penanganan *Missing Value*

Tanggal pemeriksaan		0
…	…	…
Gula berlebih		0
Garam berlebih		0
Lemak berlebih		0
Kurang makan buah dan sayur		0
Konsumsi alkohol		0
Sistol		0
Diastol		0
Tinggi badan(cm)		0
Berat badan (kg)		0
Lingkar perut(cm)		0
Pemeriksaan gula		0
Rujuk		0
Diagnosis		0

Berdasarkan Tabel 6, seluruh atribut dalam dataset tidak mengandung nilai yang hilang, yang ditunjukkan oleh angka nol (0) pada masing-masing kolom. Oleh karena itu, data telah siap untuk digunakan pada tahap analisis selanjutya.

Langkah berikutnya pengecekan data duplikat, dilakukan menggunakan fungsi *duplicated()* dari *library pandas*, yang akan mengembalikan nilai *True* untuk setiap baris yang memiliki duplikasi identik dengan baris sebelumnya. Baris-baris yang terdeteksi sebagai duplikat kemudian difilter menggunakan *indexing*. Hasil pengecekan menunjukkan bahwa terdapat 36 data duplikat berdasarkan keseluruhan baris dalam dataset. Data duplikat ini dapat memengaruhi hasil analisis dan performa model apabila tidak ditangani secara tepat. Maka dari itu, dilakukan penghapusan data duplikat menggunakan fungsi *drop_duplicates()*. Setelah proses penghapusan, jumlah baris data yang tersisa adalah 965 dari 1001 dataset, seperti ditunjukkan pada Gambar 2.

```
Jumlah baris setelah menghapus duplikat: 965
```

Gambar 2. Hasil Penanganan data duplikat

Gambar 2 memperlihatkan jumlah data setelah duplikasi berhasil dihapus, yaitu sebanyak 965 baris. Kondisi ini menunjukkan bahwa dataset telah bebas dari data ganda dan siap digunakan dalam tahap pelatihan model secara optimal.

Selanjutnya dilakukan proses penambahan atribut baru yaitu usia, dengan tujuan untuk mengetahui distribusi umur pasien pada saat pemeriksaan, hal ini dapat berfungsi sebagai pendukung analisis. Dari segi usia, sebagian besar penderita diabetes berusia antara 55-74 tahun. Namun, kaum muda berusia antara 20-40 tahun juga menderita penyakit ini. Atribut usia dihitung berdasarkan selisih antara tanggal pemeriksaan dan tanggal lahir, kemudian dikonversi ke dalam satuan tahun dengan membagi total hari menggunakan (// 365). Kolom usia ini kemudian ditambahkan ke dataset menggunakan *fungsi insert()*. Proses tersebut menghasilkan dataset yang kini mencakup kolom usia, seperti yang ditampilkan pada Tabel 7.

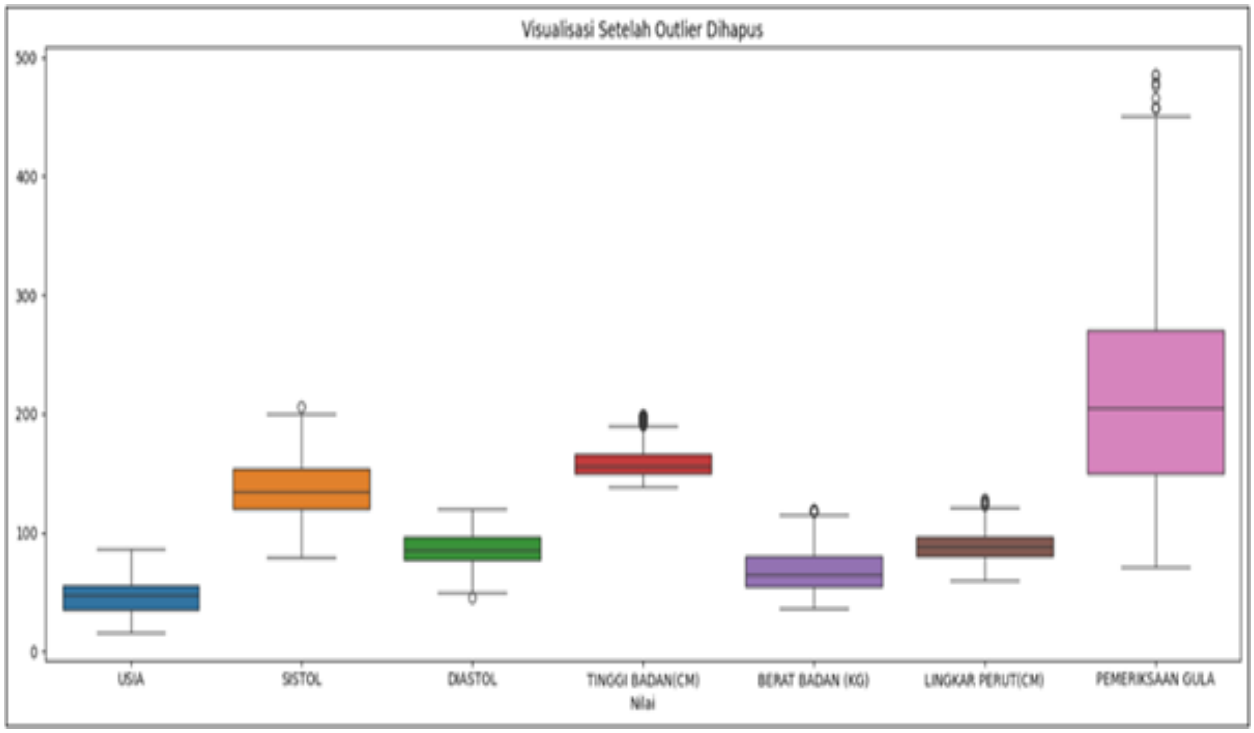
Tabel 7. Data Hasil Penambahkan Atribut Usia

Tanggal Pemeriksaan	**Nama Pasien**	**Tanggal Lahir**	**Usia**	**…**	**Diagnosis**
2024-01-12	Ruswanti	1973-01-29	50		Diabetes
2024-02-20	Entiin S	1962-08-13	61		Diabetes
2023-10-11	Asep	1995-08-26	28		Diabetes

Berdasarkan Tabel 7, kolom usia telah berhasil ditambahkan ke dalam dataset dan menampilkan hasil perhitungan usia setiap pasien saat pemeriksaan dilakukan. Atribut ini selanjutnya akan digunakan dalam proses pelatihan model guna untuk mengetahui apakah variabel usia berkontribusi dalam proses klasifikasi kondisi diabetes.

Tahap selanjutnya adalah mendeteksi adanya *outlier* atau nilai ekstrem pada data numerik. Metode *Z-score*, yang menghitung deviasi suatu nilai dari rata-rata dalam satuan deviasi standar, berfungsi dalam mengenali nilai-nilai *outlier*. *Outlier* adalah nilai dengan *Z-score* kurang dari -3 atau lebih besar dari 3. Setelah dilakukan deteksi *outlier* menggunakan metode *Z-Score*, sebanyak 36 data diketahui berada di luar rentang normal. Data tersebut kemudian dihapus dari dataset untuk menjaga kualitas data dan keakuratan model. Setelah penghapusan, jumlah data yang tersisa adalah 926 data. Hasil penghapusan *outlier* ditampilkan pada Gambar 4.

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom



Gambar 4. Visualisasi Setelah Hapus *Outlier*

Hasil Gambar 4, memperlihatkan bahwa distribusi data untuk seluruh atribut numerik seperti usia, tekanan darah (sistol dan diastol), tinggi badan, berat badan, lingkar perut, serta pemeriksaan gula telah berada dalam kisaran normal. Tidak ditemukan lagi nilai-nilai ekstrem di luar batas atas maupun bawah pada boxplot, yang mengindikasikan bahwa proses pembersihan data telah dilakukan secara efektif.

Selanjutnya, berdasarkan data pada kolom jenis kelamin, riwayat penyakit tidak menular pada keluarga maupun individu, kebiasaan merokok, serta kurangnya aktivitas fisik, gaya hidup konsumsi (gula, garam, lemak, buah dan sayur), serta status rujukan dan diagnosis masih terdapat data kategorikal, sehingga perlu diubah menjadi representasi numerik dengan menggunakan *Label Encoding* menggunakan fungsi *LabelEncoder* dari pustaka *sklearn.preprocessing.* Misalnya, kolom Diagnosis diubah menjadi angka dengan '0' untuk diabetes dan '1' untuk tidak, begitupun pada kolom yang lain diubah menjadi angka berdasarkan urutan abjad yang ada. Dengan demikian, data yang awalnya berbentuk kategorikal kini dapat digunakan dalam proses analisis lebih lanjut yang membutuhkan data numerik. Tabel 8 menyajikan hasil dari label encoding.

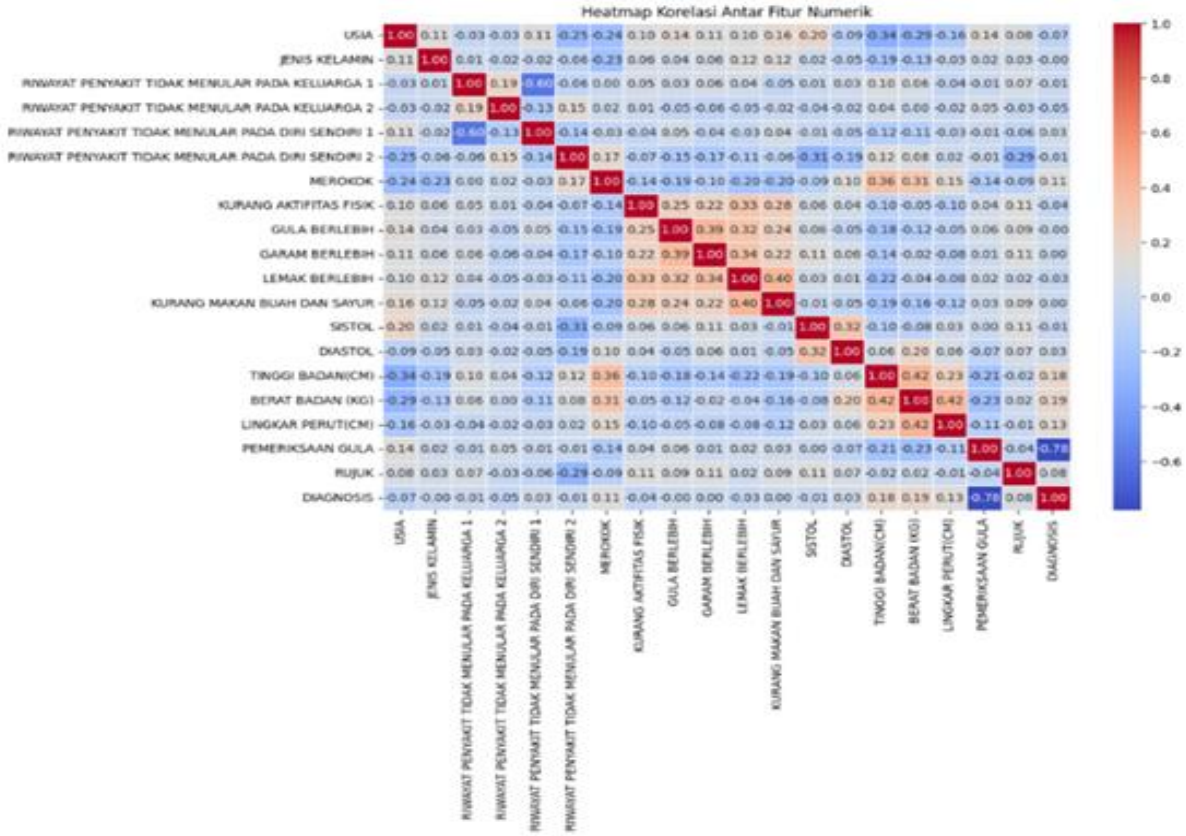
Tabel 8. Hasil *Label Encoding*

Tanggal Pemeriksaan	Jenis Kelamin	…	Rujuk	Diagnosis
2024-01-12	1		1	0
2024-01-12	1		1	1
2024-01-12	1		1	1

Tabel 8 memperlihatkan bahwa semua nilai dalam atribut kategorikal telah berhasil diubah menjadi format numerik. Oleh karena itu, dataset sudah siap untuk digunakan dalam proses pemodelan klasifikasi yang memerlukan input data dalam bentuk numerik secara konsisten.

Setelah seluruh data dikonversi ke bentuk numerik seperti laki-laki '0', Perempuan '1', langkah selanjutnya adalah melakukan analisis korelasi antar fitur untuk mengetahui seberapa kuat hubungan antar variabel numerik dalam dataset. Beberapa fitur yang menunjukkan korelasi cukup tinggi terhadap diagnosis antara lain adalah pemeriksaan gula, merokok, dan riwayat penyakit tidak menular pada diri sendiri. Sementara itu, fitur seperti tinggi badan (cm) dan garam berlebihan memiliki korelasi yang lebih rendah terhadap label 'Diagnosis', sehingga mungkin memiliki pengaruh yang lebih kecil dalam proses klasifikasi. Hasil hubungan korelatif antar fitur ditampilkan pada Gambar 5.

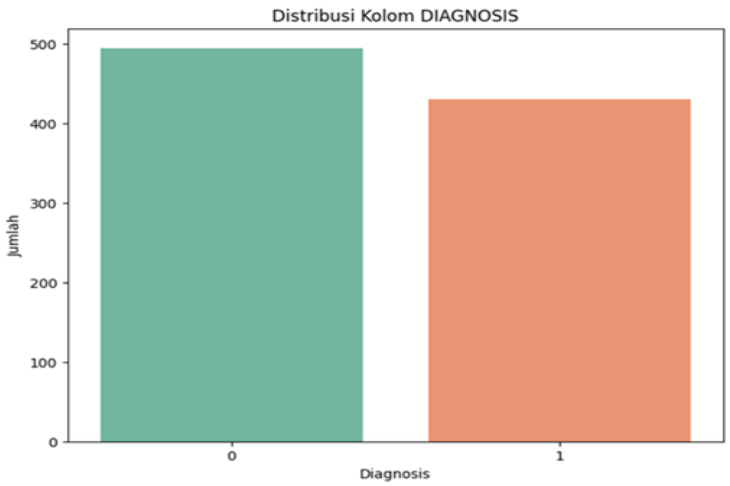
JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom



Gambar 5. Hasil Korelasi Fitur

Berdasarkan Gambar 5, dapat diamati bahwa analisis ini memberikan gambaran kekuatan hubungan antar variabel numerik. Warna merah pada heatmap menunjukkan korelasi positif yang kuat, sedangkan warna biru menunjukkan korelasi negatif. Nilai korelasi yang mendekati 1 atau -1 menandakan hubungan yang kuat antar variabel.

Sesudah analisis korelasi dilakukan, langkah berikutnya adalah melakukan visualisasi distribusi label 'Diagnosis' untuk memahami proporsi data antara pasien yang terdiagnosis diabetes dan yang tidak. Visualisasi ini menggunakan diagram batang (*barchart*) yang menunjukkan jumlah masing-masing kelas diagnosis setelah proses *encoding*. Distribusi data tersebut divisualisasikan dan ditampilkan pada Gambar 6.



Gambar 6. Distribusi Kolom Diagnosis

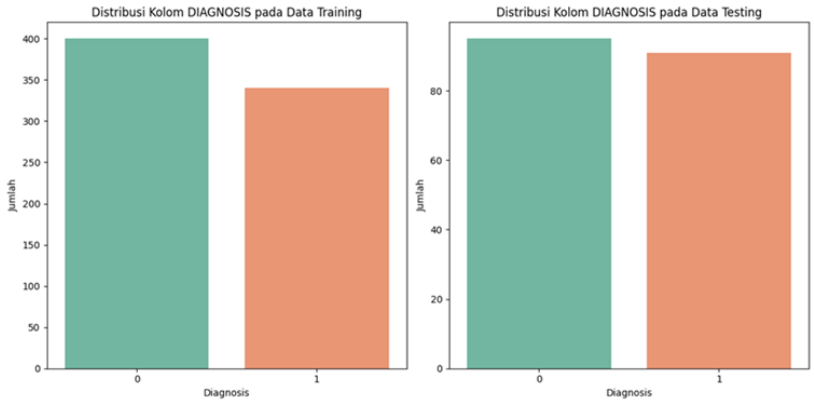
Berdasarkan Gambar 6, terlihat bahwa jumlah pasien yang terdiagnosis diabetes (label 0) sedikit lebih banyak dibandingkan pasien yang tidak terdiagnosis diabetes (label 1). Hal ini penting untuk diketahui karena distribusi label yang tidak seimbang dapat mempengaruhi performa model klasifikasi. Dengan distribusi yang relatif seimbang seperti ini, model memiliki peluang yang lebih baik untuk mempelajari pola dari kedua kelas secara optimal.

Setelah memastikan distribusi label, langkah berikutnya adalah membagi dataset menjadi data pelatihan dan data pengujian serta memilih atribut mana yang digunakan selama tahap pemodelan. Pembagian data penelitian ini dilakukan menggunakan rasio 80% terhadap data *training* dan 20% terhadap data *testing* menggunakan fungsi *train_test_split* dari

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

pustaka *sklearn.model_selection*, fitur yang digunakan selama proses *training* meliputi semua atribut numerik yang relevan kecuali kolom target 'Diagnosis'. Dengan demikian, proses pemisahan fitur dan target dilakukan untuk mempersiapkan data dalam pelatihan model. Gambar 7 menampilkan visualisasi distribusi data setelah dilakukan proses pembagian dataset.

Gambar 7 menunjukkan visualisasi ulang distribusi label pada data data training dan testing mengikuti distribusi data.



Gambar 7. Distribusi Diagnosis pada Data *Training* dan *Testing*

Gambar 7, memperlihatkan distribusi label 'Diagnosi' pada data *training* dan *testing*. Terlihat bahwa distribusi antara pasien dengan diagnosis diabetes dan tidak diabetes tetap seimbang di kedua subset data, sehingga model dapat dilatih dan diuji dengan kondisi data yang proporsional.

Langkah selanjutnya adalah melakukan normalisasi data agar seluruh fitur memiliki skala nilai yang seragam. Normalisasi dilakukan dengan menggunakan metode *StandardScaler* dari pustaka *sklearn.preprocessing*, *StandardScaler* mengubah distribusi data agar data memiliki mean 0 dan standar deviasi 1. Pada proses normalisasi melibatkan dua tahap dengan melakukan *fitting* dan transformasi pada data pelatihan dan menggunakan *scaler* yang sama untuk mentransformasi data pengujian. Dengan demikian, data yang digunakan untuk pelatihan dan pengujian berada pada skala yang sama, sehingga dapat meningkatkan performa dan konvergensi model.

Tahapan *pre-processing* dalam penelitian ini memiliki kesamaan dengan studi Maulana [6], terutama pada proses pembagian data menjadi data latih dan data uji, serta penerapan normalisasi dalam implementasi algoritma SVM untuk klasifikasi diabetes. Meski demikian, penelitian Maulana tidak mencakup penghapusan *outlier* maupun penambahan atribut usia, yang justru dalam studi ini terbukti berkontribusi terhadap peningkatan akurasi model. Selain itu, berbeda dengan studi Desiani [7] yang tidak menyebutkan proses imputasi *missing value* secara *eksplisit*, penelitian ini telah menerapkan teknik imputasi menggunakan nilai rata-rata *(mean)* dan modus *(mode)* untuk mempertahankan kualitas data. Oleh karena itu, tahapan *pre-processing* dalam studi ini dapat dikatakan lebih komprehensif dan berpotensi memberikan kontribusi positif terhadap kinerja model klasifikasi.

3.2 Hasil *Modeling*

3.2.1 *Modeling Support Vector Machine*

Pada tahap pemodelan, algoritma SVM digunakan dengan menggunakan kernel RBF *(Radial Basis Function)*. Model SVM diinisialisasi dengan *parameter* C=1*, gamma='scale',* dan *random_state*=42 untuk menjaga konsistensi hasil saat pelatihan ulang. *Parameter C* berfungsi untuk mengatur keseimbangan antara akurasi pelatihan dan margin keputusan, sedangkan *gamma='scale'* secara otomatis menyesuaikan nilai gamma berdasarkan jumlah fitur pada data. Setelah proses inisialisasi, data pelatihan digunakan untuk melatih model *(X_train, y_train)* untuk mempelajari pola klasifikasi yang dapat membedakan antar kelas dalam dataset. Tahap berikutnya adalah mengimplementasikan model yang sudah dilatih guna memprediksi label pada data uji *(X_test)* melalui fungsi *predict()*. Seluruh proses pemodelan ini ditunjukkan secara rinci pada Gambar 8.

```
# Inisialisasi model SVM dengan kernel linear
svm_model = SVC(kernel='rbf', C=1, gamma='scale', random_state=42)

# Latih model
svm_model.fit(X_train, y_train)

# Prediksi dengan model
y_pred_svm = svm_model.predict(X_test)
```

Gambar 8. *Modeling Support Vector Machine*

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

3.2.2 Modeling Random Forest

Pada algoritma *Random Forest*, model diinisialisasi dengan parameter n_estimators dengan membangun 100 pohon keputusan untuk meningkatkan akurasi prediksi, serta *random_state*=42 untuk menjaga konsistensi hasil pelatihan. Setelah tahap inisialisasi selesai Untuk mengidentifikasi pola kategorisasi dalam dataset, model dilakukan pelatihan menerapkan data latih *(X_train, y_train).* Metode *predict()* kemudian digunakan untuk menggunakan model yang dikembangkan guna memperkirakan label pada data uji *(X_test)*. Proses pemodelan *Random Forest* dapat dilihat pada Gambar 9.

```
# Model Random Forest
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
```

Gambar 9. Modeling *Random Forest*

3.3. Evaluasi

3.3.1 Evaluasi Model *Support Vector Machine*

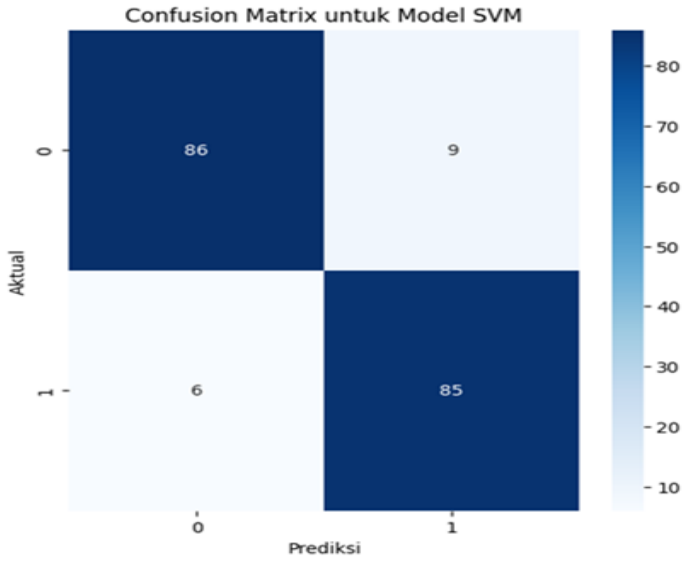
Proses evaluasi dilakukan untuk mengukur performa model dengan memanfaatkan *confusion matrix* dan *classification report*, yang mencakup metrik akurasi, presisi, *recall,* serta *F1-score.* Hasil evaluasi model SVM disajikan dalam Tabel 9.

Tabel 9. Hasil Evaluasi Algoritma SVM

Kelas	*Precision*	*Recall*	*F1-Score*	Akurasi
0	0.93	0.91	0.92	91%
1	0.90	0.93	0.92	

Tabel 9 memperlihatkan hasil evaluasi model SVM memperlihatkan kinerja yang cukup baik dalam nilai *Precision, Recall, F1-Score*, dan Akurasi secara optimal. Secara keseluruhan, akurasi model mencapai 91%, yang menunjukkan total data berhasil diklasifikasikan dengan benar. Untuk kelas '0' presisi sebesar 93% menujukkan bahwa model menyatakan seseorang mengidap diabetes, f1-Score mencapai 92% untuk kedua kelas menggambarkan keseimbangan yang baik antara presisi dan recall. Adapun niai recall sebesar 91% menujukkan bahwa model dapat mendeteksi kasus diabetes yang sebenarnya ada.

Sementara itu, untuk kelas '1', presisi 90% mengindikasikan bahwa model dapat dengan tepat mengklasifikasikan kasus negatif, sementara itu recall pada kelas ini mencapai 93%, yang berarti model sangat baik dalam mengenali kasus yang seharusnya tidak mengidap diabetes. Dengan tingginya nilai recall pada kedua kelas, model ini memiliki kemampuan yang baik dalam mengidentifikasi kasus diabetes, sehingga berpotensi efektif untuk diterapkan dalam sistem pendukung keputusan medis. Selain itu, *confusion matrix* digunakan untuk menunjukkan jumlah prediksi yang benar dan salah pada masing-masing kelas. Visualisasi ini membantu dalam mengevaluasi kemampuan model dalam membedakan antara pasien yang didiagnosis menderita diabetes dan yang tidak. Hasil *confusion matrix* untuk model SVM ditampilkan pada Gambar 10 berikut.



Gambar 10. *Confusion Matrix* untuk Model SVM

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Pada Gambar 10, ditampilkan hasil *confusion matrix* dengan label 0 untuk 'diabetes' dan label 1 untuk 'tidak diabetes'. Sebanyak 86 data dengan label diabetes berhasil diklasifikasikan secara akurat, sementara 9 data lainnya keliru diprediksi sebagai bukan diabetes. Untuk kategori tidak diabetes, 85 data berhasil dikenali dengan benar, sedangkan 6 data salah diklasifikasikan sebagai diabetes. Hasil ini mengindikasikan bahwa model SVM memiliki kinerja yang cukup baik dalam membedakan antara pasien yang mengidap diabetes dan yang tidak, ditunjukkan oleh jumlah prediksi yang benar yang lebih tinggi dibandingkan dengan jumlah prediksi yang salah.

3.3.2 Evaluasi Model *Random Forest*

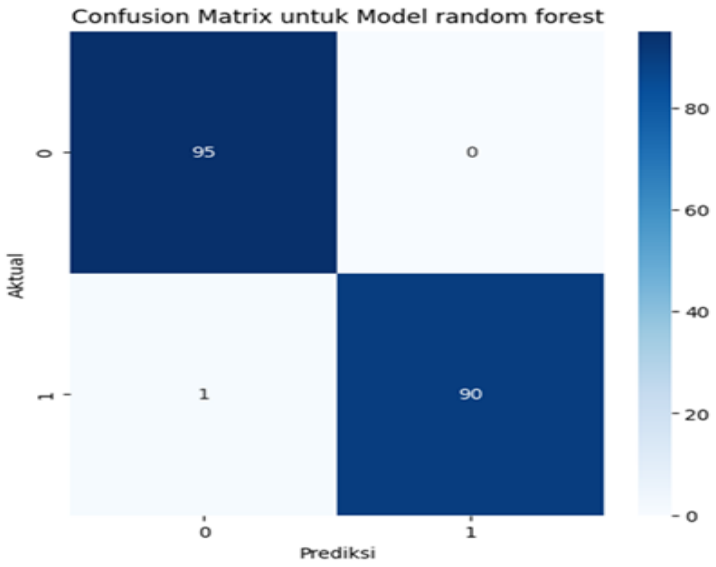
Selanjutnya, model *Random Forest* dinilai menggunakan metrik yang sama dengan SVM dan juga digunakan untuk memperkirakan hasil pengujian. Tabel 10 menampilkannya berdasarkan hasil evaluasi pemodelan *Random Forest.*

Tabel 10. Hasil Evaluasi Algoritma *Random Forest*

Kelas	*Precision*	*Recall*	*F1-Score*	Akurasi
0	0.99	1.00	0.99	99%
1	1.00	0.99	0.99	

Berdasarkan Tabel 10, memperlihatkan hasil evaluasi algoritma Random Forest menunjukkan kinerja yang sangat baik. Secara keseluruhan, akurasi model mencapai 99%, yang mengindikasikan bahwa sebagian besar data telah berhasil diklasifikasikan dengan benar. Pada label 0 yang memperlihatkan kasus 'diabetes', presisi sebesar 99% menandakan bahwa hampir seluruh prediksi yang menyatakan seseorang menderita diabetes adalah benar. *recall* sebesar 100% menunjukkan bahwa semua kasus 'diabetes' dalam data berhasil diklasifikasikan dengan benar tanpa ada yang terlewat. Sementara itu *F1-Score* juga mencapai nilai 99% memperlihatkan bahwa model memiliki keseimbangan yang sangat baik antara presisi dan recall.

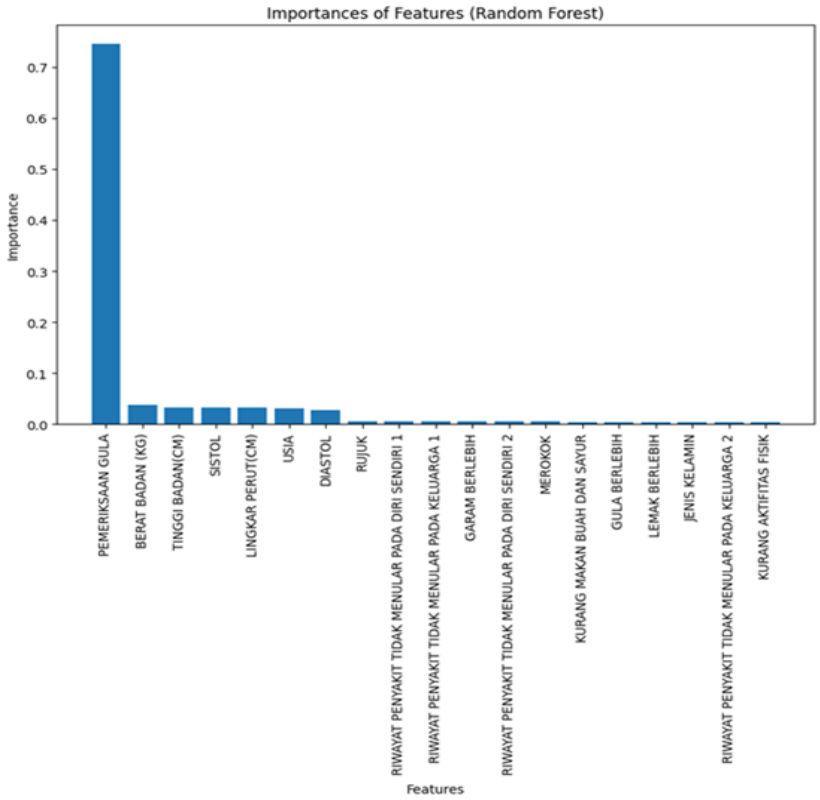
Untuk label 1 menunjukkan 'tidak diabetes', presisi sebesar 100% berarti bahwa seluruh prediksi yang menyatakan seseorang tidak menderita diabetes adalah benar. Pada nilai *recall* sebesar 99% menunjukkan hanya sedikit data 'tidak diabetes' yang salah prediksi, *F1-Score* untuk kedua label bernilai 99%, menandakan keseimbangan yang sangat baik antara presisi dan *recall*. Model *Random Forest* menunjukkan kemampuan yang sangat baik sehingga cocok diterapkan dalam sistem pendukung keputusan medis. *Confusion matrix* juga divisualisasikan untuk model *Random Forest*, seperti ditunjukkan pada Gambar 11.



Gambar 11. *Confusion Matrix* untuk Model *Random Forest*

Berdasarkan Gambar 11, sebanyak 95 data dengan label 0 sebagai 'diabetes' berhasil diklasifikasikan dengan benar. Untuk label 1 sebagai 'tidak diabetes', terdapat 90 data yang berhasil diklasifikasikan dengan tepat, sementara satu data mengalami kesalahan prediksi dan teridentifikasi sebagai 'diabetes'. Hasil ini dapat memperlihatkan model *Random Forest* mampu membedakan dengan cukup baik antara pasien yang mengidap diabetes dan yang tidak, dengan jumlah prediksi benar yang lebih banyak dibandingkan jumlah prediksi salah. Selain itu, pada model *Random Forest* juga dilakukan analisis *feature importance* untuk mengidentifikasi fitur yang paling signifikan dalam proses klasifikasi. Visualisasi *feature importance* ditampilkan pada Gambar 12.

JURIKOM (Jurnal Riset Komputer), Vol. 12 No. 4, Agustus 2025
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v12i4.8686
Hal 589-601
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom



Gambar 12. *Visualisasi Feature Importance* pada *Random Forest*

Berdasarkan Gambar 12, 'pemeriksaan gula' merupakan fitur paling berpengaruh dalam klasifikasi diabetes, ditunjukkan oleh nilai *feature importance* tertinggi. Fitur lain seperti 'merokok' dan riwayat penyakit tidak menular juga memberikan kontribusi, meskipun lebih kecil memiliki pengaruh yang rendah terhadap hasil klasifikasi.

4. KESIMPULAN

Berdasarkan temuan yang didapat selama penelitian, dapat dikatakan bahwa tujuan penelitian telah tercapai, yakni berhasil membangun model untuk mengklasifikasikan penyakit diabetes mellitus dengan menerapkan dua algoritma *machine learning*, seperti *Random Forest* (RF) dan *Support Vector Machine* (SVM). Kedua model diperluas berdasarkan data pasien yang telah melalui tahap *pre-processing.* Dari hasil evaluasi, model *Random Forest* memperlihatkan efektivitas yang lebih baik dengan pencapaian akurasi sebesar 99%. sedangkan model SVM memperoleh akurasi sebesar 91%. Penelitian ini mengindikasikan bahwa algoritma *Random Forest* bekerja lebih efektif dalam melakukan klasifikasi diabetes berdasarkan dataset yang digunakan. Oleh karena itu, algoritma *Random Forest* dairekomendasikan sebagai metode yang lebih efektif untuk diterapkan dalam sistem pendukung keputusan medis dalam mengidentifikasi penyakit diabetes secara lebih akurat dan efisien.

REFERENCES

[1] Kemkes, "Penyakit Diabetes Melitus," Kemkes.
[2] R. F. N. Iskandar, D. H. Gutama, D. P. Wijaya, and D. Danianti, "Klasifikasi Menggunakan Metode Random Forest untuk Awal Deteksi Diabetes Melitus Tipe 2," *J. Tek. Ind. Terintegrasi*, vol. 7, no. 3, pp. 1620–1626, 2024, doi: 10.31004/jutin.v7i3.26916.
[3] V. H. Ghaida, "Mengenal Komplikasi Diabetes Melitus," Kemenkes.
[4] R. G. Ginting, E. Girsang, J. B. Ginting, and H. Hartono, "Analisis Determinan Dan Prediksi Penyakit Diabetes Melitus Tipe 2 Menggunakan Metode Machine Learning: Scoping Review," *J. Matern. Kebidanan*, vol. 7, no. 1, pp. 58–72, 2022, doi: 10.34012/jumkep.v7i1.2538.
[5] G. Abdurrahman, "Jurnal Sistem dan Teknologi Informasi Klasifikasi Penyakit Diabetes Melitus Menggunakan Adaboost Classifier," *JUSTINDO (Jurnal Sist. dan Teknol. Informasi)*, vol. 7, no. 1, pp. 59–66, 2022, [Online]. Available: http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO
[6] M. R. A. S. Maulana, "Klasifikasi Diabetes," Pp. 1–23, 2024.
[7] A. Desiani *Et Al.*, "Perbandingan Algoritma Support Vector Machine (Svm) Dan Naïve," Vol. 10, no. 1, pp. 65–74, 2024.
[8] M. U. Khairul Huda, "SENASTIKA Universitas Malikussaleh," pp. 1–10, 2024.
[9] R. Pahlevi, K. Q. Fredlina, and N. W. Utami, "Penerapan Algoritma ID3 dan SVM Pada Klasifikasi Penyakit Diabetes Melitus Tipe 2," *Pros. Semin. Nas. Apl. Sains Teknol.*, vol. 2, pp. A64–A75, 2021, [Online]. Available: https://ejournal.akprind.ac.id/index.php/snast/article/view/3340
[10] A. M. Argina, "Penerapan Metode Klasifikasi K-Nearest Neigbor pada Dataset Penderita Penyakit Diabetes," *Indones. J. Data*

Sci., vol. 1, no. 2, pp. 29–33, 2020, doi: 10.33096/ijodas.v1i2.11.
[11] A. M. Siregar, S. Faisal, Y. Cahyana, and B. Priyatna, "Accounting Information System," pp. 17–30, 2020.
[12] T. Rohana, E. Nurlaelasari, E. E. Awal, and H. Y. Novita, "Kajian Model Jaringan Syaraf Tiruan Untuk Memprediksi Secara Dini Tingkat Kelulusan Mahasiswa," vol. 15, no. 4, pp. 629–640, 2024.
[13] H. Alrasyid, A. Homaidi, M. Kom, Z. Fatah, and M. Kom, "Comparison Support Vector Machine and Random Forest Algorithms in Detect Diabetes," vol. 1, no. 1, pp. 447–453, 2024.
[14] Marlina Haiza, Elmayati, Zulius Antoni, and Wijaya Harma Oktafia Lingga, "Penerapan Algoritma Random Forest Dalam Klasifikasi Penjurusan Di SMA Negeri Tugumulyo," *Penerapan Kecerdasan Buatan*, vol. 4, no. 2, pp. 138–143, 2023.
[15] K. U. K. Fida Maisa Hana, Deka Setia Negara, *Perbandingan Algoritm. Neural Netw. Dengan Linier Discrim. Anal. Pada Klasifikasi Penyakit Diabetes*, vol. 1, pp. 1541–1541, 2020.
[16] S. Rabbani, D. Safitri, and N. Rahmadhani, "Comparative Evaluation of SVM Kernels for Sentiment Classification in Fuel Price Increase Analysis Perbandingan Evaluasi Kernel SVM untuk Klasifikasi Sentimen dalam Analisis Kenaikan Harga BBM," vol. 3, no. October, pp. 153–160, 2023.
[17] B. R. Prasetyo, E. D. Wahyuni, and P. M. Kusumantara, "Komparasi Performa Model Berbasis Algoritma Random Forest Dan Lightgbm Dalam Melakukan Klasifikasi Diabetes Melitus Gestasional," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024, doi: 10.23960/jitet.v12i3.4817.
[18] A. Ridwan, "Penerapan Algoritma Naïve Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus," *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 4, no. 1, pp. 15–21, 2020, doi: 10.47970/siskom-kb.v4i1.169.
[19] Y. N. Paramitha, A. Nuryaman, A. Faisol, E. Setiawan, and D. E. Nurvazly, "Klasifikasi Penyakit Stroke Menggunakan Metode Naïve Bayes," *J. Siger Mat.*, vol. 04, no. 01, pp. 11–16, 2023, [Online]. Available: https://www.kaggle.com/datasets/zzettrkalpakbal/full-filled-
[20] D. Nurnaningsih, D. Alamsyah, A. Herdiansah, and A. A. J. Sinlae, "Identifikasi Citra Tanaman Obat Jenis Rimpang dengan Euclidean Distance Berdasarkan Ciri Bentuk dan Tekstur," *Build. Informatics, Technol. Sci.*, vol. 3, no. 3, pp. 171–178, 2021, doi: 10.47065/bits.v3i3.1019.
[21] P. N. Sabrina and A. Komarudin, "Prediksi Penyakit Diabetes Dengan Metode K-Nearest Neighbor (Knn) Dan Seleksi Fitur Information Gain," vol. 8, no. 6, pp. 11320–11326, 2024.
[22] N. Hidayah and Dodiman, "Implementasi Algoritma Multinomial Naïve Bayes, TF-IDF dan Confusion Matrix dalam Pengklasifikasian Saran Monitoring dan Evaluasi Mahasiswa Terhadap Dosen Teknik Informatika Universitas Dayanu Ikhsanuddin," *J. Akad. Pendidik. Mat.*, vol. 10, no. 1, pp. 8–15, 2024.